



RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE  
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA  
RECHERCHE SCIENTIFIQUE

**UNIVERSITÉ DJILLALI LIABES**  
FACULTÉ DES SCIENCES EXACTES - SIDI BEL ABBES -  
Département d'INFORMATIQUE

# THÈSE

pour obtenir le titre de

## Docteur en Sciences

Présentée par

Afaf MERAZI

Mention : Informatique

## Empaquetage Dynamique à Base de Services Web Sémantiques

Soutenue le :

08 Décembre 2015

Devant le Jury composé de :

<i>Président :</i>	Ahmed LEHIRECHE	(Pr)	UDL - SBA
<i>Examineurs :</i>	Djamel AMAR BENSABER	(MCA)	ESI - SBA
	Zohra SLAMA	(MCA)	UDL - SBA
	Sofiane BOUKLI HACENE	(MCA)	UDL - SBA
	Djelloul BOUCHIHA	(MCA)	CU - Naama
<i>Directeur de thèse :</i>	Mimoun MALKI	(Pr)	ESI - SBA



# Liste des Abbreviations

API	Application Programming Interface
BPEL	Business Process Execution Language
BPM	Business Process Management
BPML	Business Process Modelling Lanugage
Contest S3	Contest Semantic Service Selection
DAML	DARPA Agent Markup Language
ebXML	Electronic Business XML
Golog	alGOI in LOGic
GUI	Graphical User Interface
HTML	Hypertext Markup Language
HTN	Hierarchical Task Network
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IOPE	Inputs,Outputs,Preconditions,Effects
JAR	Java Archive
JENA	Java framework for building Semantic Web applications
LIMES	Link Discovery Framework for metric spaces
LOD	LINKED OPEN DATA
OIL	Ontology Inference Layer
OWL	Web Ontology Language
OWL-QL	OWL Query Language
OWL-S	Web Ontology Language for Services
P2P	Peer-to-Peer
PDDL	Planning Domain Definition Language
PDDL3	Version of PDDL that supports preferences
QoS	Quality of Service
RDF	Resource Description Framework
RDFS	RDF Schema language
SOA	Service-oriented Architecture
SOC	Service-oriented Computing
SWRL	Semantic Web Rule Language
SWSF	Semantic Web Services Framework
UDDI	Universal Description, Discovery and Integration
UML	Unified Modeling Language
URI	Uniform Resource Identifier

---

URL	Uniform Resource Locator
W3C	World Wide Web Consortium
WSA	Web Service Architecture
WSC	Web Service Composition
WSCI	Web Service Choreography Interface
WSD	Web Service Discovery
WSDL	Web Service Description Language
WSDL-S	Web Services Semantics
WSMO	Web Service Modeling Ontology
XML	Extensible Markup Language

# REMERCIEMENTS

*"La connaissance est en fin de compte fondée sur la reconnaissance."*

– Ludwig Wittgenstein.

L'aventure de la thèse s'achève. Au-delà de la satisfaction du travail réalisé, je voudrais ici exprimer ma reconnaissance envers les personnes qui m'ont aidé et soutenu durant cette période.

au Pr. Mimoun MALKI (École Supérieure en Informatique ESI-SBA), mon directeur de thèse depuis 2007, qui m'a apporté le soutien scientifique et les encouragements sans lesquels ces travaux de thèse n'auraient pas pu s'effectuer. Avec patience et pédagogie, il m'a fait découvrir plusieurs facettes de l'activité de chercheur. Je souhaite lui dire que j'ai eu beaucoup de plaisir à travailler avec lui et que cette thèse n'aurait pu être ce qu'elle est sans ses remarques et ses conseils judicieux. Merci de m'avoir encadré pendant ces années et surtout pour m'avoir fait confiance dans le parcours de ma thèse.

à Tous les membres du laboratoire "Evolutionary Engineering and Distributed Information Systems Laboratory " et également à tous les professeurs et professionnels de ce laboratoire que j'ai connus.

à Mes collègues du département d'Informatique avec qui je collabore.

à Chacun des membres du jury de cette thèse de doctorat qui ont accepté la tâche délicate d'examiner cette thèse et qui ont eu la patience de juger ce travail : Pr. Ahmed LEHIRECHE, Professeur à l'Université DJILLALI Liabés(SBA) ; Dr. Djamel AMAR BENSABER, Maître de Conférences (A) à l'École Supérieure en Informatique (ESI-SBA) ; Dr. Zohra SLAMA, Maître de Conférences (A) à l'Université DJILLALI Liabés (SBA) ; Dr. Sofiane BOUKLI HACENE, Maître de Conférences (A) à l'Université DJILLALI Liabés (SBA) et Dr. Djelloul BOUCHIHA, Maître de Conférences(A) au Centre universitaire SALHI Ahmed de Naama.

à Mes AMIS, à qui je tiens énormément, et qui, de près ou de loin, m'ont toujours soutenu. Ils m'ont toujours témoigné d'un attachement et d'une affection sincère et inébranlable. Je pense notamment à ... Non, je ne les citerai pas, ils se reconnaîtront.

à **Ma Mère**, qui est et restera le moteur de ma vie pour son soutien indéfectible et qui a su me faire confiance et vouloir toujours que je vole plus haut. Mes profonds et mes plus grands remerciements s'adressent particulièrement à toi, ma mère pour ton aide et pour tes corrections, ainsi que pour ta patience et pour avoir fait beaucoup de concessions afin que ma thèse se déroule dans de bonnes conditions. Merci pour être une parente formidable et également pour les moments que nous avons partagés. Merci pour être toujours là, à mon côté, comme aujourd'hui. Merci du fond du cœur. "Eu amo vocês" !

à L'université Djillali LIABES - Sidi Bel Abbes et ses services.



# Table des matières

<b>Liste des Abbreviations</b>	<b>i</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Contexte de la thèse . . . . .	1
1.2 Problématique . . . . .	5
1.3 Scénario illustratif . . . . .	9
1.4 Contributions de la thèse . . . . .	11
1.5 Plan de la thèse . . . . .	14
<b>I Background &amp; États de l’art</b>	<b>15</b>
<b>2 Évolution du Web</b>	<b>17</b>
2.1 Introduction . . . . .	17
2.2 Le web sémantique . . . . .	18
2.2.1 Données, Information et Connaissance . . . . .	21
2.2.2 Représentation des connaissances avec RDF(S) et OWL . . . . .	22
2.2.3 Ontologie, sens de données et raisonnement . . . . .	27
2.2.4 Interrogation du Web Sémantique . . . . .	31
2.2.5 Données liées- Le Web sémantique d’aujourd’hui . . . . .	34
2.3 Description de Services Web sémantiques . . . . .	46
2.3.1 Description syntaxique des Services Web . . . . .	49
2.3.2 La nébuleuse des langages sémantiques . . . . .	51
2.4 Conclusion . . . . .	59
<b>3 Coordination de Services Web Sémantiques</b>	<b>61</b>
3.1 Introduction . . . . .	61
3.2 Découverte de Services Web Sémantiques . . . . .	62
3.2.1 Mise en correspondance de Services Web . . . . .	63
3.2.2 Les approches de la découverte sémantique . . . . .	66
3.2.3 Discussion . . . . .	73
3.3 Composition des Services Web Sémantiques . . . . .	74
3.3.1 Approches basées sur les automates et réseaux de pétri . . . . .	76
3.3.2 Approches basées sur les techniques de planification . . . . .	77
3.3.3 D’autres approches . . . . .	85
3.3.4 Discussion . . . . .	85
3.4 Conclusion . . . . .	88
<b>4 E-tourisme &amp; Package dynamique</b>	<b>91</b>
4.1 Introduction : E-tourisme . . . . .	91
4.2 Le package dynamique: la nouvelle tendance de l’e-tourisme . . . . .	93
4.3 Enjeux . . . . .	95
4.4 Approches de Packages dynamiques Sémantiques . . . . .	95

## TABLE DES MATIÈRES

---

4.4.1	Approche touristique orienté web sémantique . . . . .	96
4.4.2	Approche touristique orienté Services Web Sémantiques . . . . .	102
4.5	Discussion et synthèse . . . . .	105
4.6	Conclusion . . . . .	110
<b>II</b>	<b>Contributions</b>	<b>111</b>
<b>5</b>	<b>Contributions Scientifiques</b>	<b>113</b>
5.1	Introduction . . . . .	113
5.2	L'architecture du prototype SQUIREL . . . . .	115
5.2.1	Problématique . . . . .	115
5.2.2	Proposition . . . . .	115
5.3	Couche d'Interrogation Sémantique . . . . .	118
5.3.1	Interrogation des jeux de Données sémantiques touristiques . . . . .	122
5.3.2	Interrogation des Services Web Sémantiques . . . . .	123
5.4	Couche jeu de Données Touristiques . . . . .	124
5.5	Couche Services Web Sémantique Inter-reliés . . . . .	129
5.5.1	Modélisation formelle de OWL-S . . . . .	130
5.5.2	Positionnement de notre approche . . . . .	133
5.6	Conclusion . . . . .	152
<b>6</b>	<b>Implémentation &amp; Expérimentations</b>	<b>155</b>
6.1	Introduction . . . . .	155
6.2	Implémentation & Expérimentations . . . . .	155
6.2.1	Les outils technologiques utilisés . . . . .	155
6.2.2	Les technologies sémantiques . . . . .	156
6.3	Description du prototype . . . . .	156
6.4	Environnement de développement . . . . .	159
6.5	Conclusion . . . . .	162
<b>III</b>	<b>Conclusion Générale</b>	<b>163</b>
<b>7</b>	<b>Bilan et Perspectives</b>	<b>165</b>
7.0.1	Bilan du travail réalisé . . . . .	165
7.0.2	Perspectives . . . . .	167
	<b>Bibliographie</b>	<b>169</b>
<b>8</b>	<b>Généralités sur les graphes</b>	<b>183</b>
8.0.1	Généralités sur les graphes . . . . .	183



# Liste des tableaux

2.1	Déclarations RDF . . . . .	23
2.2	Déclarations RDFS . . . . .	24
2.3	Le vocabulaire RDF Schema. . . . .	25
2.4	Règles d'inférence en RDFS . . . . .	26
2.5	Résultat de la requête Q (Filtrage textuel) . . . . .	33
2.6	Résultat de la requête Q' . . . . .	34
2.7	Evolution des ensembles de données publiées grâce aux Données Liées Ouvertes	36
3.1	Les approches de la découverte sémantique (matchmaking) . . . . .	75
3.2	Les approches de la composition sémantique (matchmaking) . . . . .	87
4.1	Les Ontologies touristiques existantes [Barta <i>et al.</i> 2009, Siricharoen 2010] . .	107
4.2	Les approches sémantiques touristiques . . . . .	110
5.1	les éléments conceptuels de l'ontologie GoodRelations . . . . .	127
6.1	Les mesures de l'expérimentation en millisecondes . . . . .	159



# Table des figures

1.1	L'évolution du Web . . . . .	2
1.2	L'inflation des sites web touristiques en ligne [Phocuswright 2015] . . . . .	3
1.3	Evolution des technologies d'intégration d'applications [Hacid <i>et al.</i> 2009] . . . . .	4
1.4	Explosion des sites de réservation en ligne [Phocuswright 2015] . . . . .	5
1.5	La forte croissance de réservation de séjour en ligne par pays [Phocuswright 2015] . . . . .	6
1.6	Les capacités impliquées dans l'organisation d'un voyage (inspiré de [Hacid <i>et al.</i> 2009]) . . . . .	9
1.7	Les capacités et les instances de services (inspiré de [Hacid <i>et al.</i> 2009]) . . . . .	10
1.8	Les Descriptions de services reposant sur les Données Liées touristiques . . . . .	13
2.1	Les Trois couches du Web Sémantique [Hitzler <i>et al.</i> 2009] . . . . .	19
2.2	Evolution de la recherche Google [Lardinois 2013] . . . . .	20
2.3	Evolution du hummingbird knowledge graph [Lardinois 2013] . . . . .	21
2.4	Semantic Web Layer Cake, repris de Tim Berners-Lee [Antoniou & Harmelen 2008] . . . . .	22
2.5	Représentation graphique d'un exemple de triplets RDF (Graphe RDF) . . . . .	23
2.6	Représentation des sources de données interconnectées grâce à Linking Open Data (en Aout 2014). . . . .	36
2.7	Processus de Liage de données [Heath & Bizer 2011] . . . . .	39
2.8	Techniques de Liage [Jentzsch <i>et al.</i> 2010] . . . . .	40
2.9	Cycle de vie des données liées . . . . .	42
2.10	Les mesures de similarité utilisées par SILK [Jentzsch <i>et al.</i> 2010] . . . . .	45
2.11	Silk Link Specification Language [Jentzsch <i>et al.</i> 2010] . . . . .	46
2.12	La comparaison entre les trois outils . . . . .	47
2.13	Les versions de WSDL . . . . .	50
2.14	Mécanisme d'annotation sémantique des types de données, des opérations, des entrées et des sorties . . . . .	52
2.15	Mécanisme d'annotation sémantique . . . . .	53
2.16	Principaux éléments de description dans USDL . . . . .	54
2.17	L'ontologie supérieure d'OWL-S . . . . .	56
2.18	Service OWL-S de réservation d'hôtel . . . . .	57
2.19	Comparaison entre OWL-S et WSMO . . . . .	58
2.20	Récapitulatif des éléments de description dans les approches étudiées [Jorge 2006] . . . . .	59
3.1	Mise en correspondance de Services Web . . . . .	66
3.2	Classification des approches pour la mise en correspondance [Klusch 2012] . . . . .	68
3.3	Découverte profil vs Process [Stoilos <i>et al.</i> 2007] . . . . .	74
3.4	Composition des Services Web Sémantiques . . . . .	76
3.5	Architecture de [McIlraith & Son 2002] . . . . .	78
3.6	L'architecture de WSPlan, un système de planification de services, extraite de [Wu <i>et al.</i> 2003]. . . . .	79

## TABLE DES FIGURES

---

3.7	Technique d'interprétation a posteriori du plan pour obtenir des compositions complexes, ici d'une séquence de services à un <Split + Join> du langage de description OWL-S . . . . .	83
3.8	Architecture de Porsce [Hatzi <i>et al.</i> 2013] . . . . .	83
3.9	Classification des approches pour la composition dynamique du Process Model [Klusch 2012] . . . . .	88
4.1	Package dynamique vs pré-packaging [Schmeing <i>et al.</i> 2006, Ayazlar 2014] . .	93
4.2	Les trois offres de forfaits disponibles pour l'utilisateur [sia partners 2013] . .	94
4.3	Architecture du système Package dynamique du Projet SEED [Cardoso & Schauer 2006, Cardoso 2006] . . . . .	97
4.4	Structure hiérarchique des médiateurs sémantiques [Cardoso & Schauer 2006, Cardoso 2006] . . . . .	98
4.5	Génération des processus abstraits [Cardoso & Schauer 2006, Cardoso 2006] .	98
4.6	Architecture de CRUZAR [Mànguez <i>et al.</i> 2010] . . . . .	100
4.7	Algorithme détaillé de CRUZAR [Mànguez <i>et al.</i> 2010] . . . . .	101
4.8	Les Cas d'utilisation démontrant les exigences de cette architecture [Corcho <i>et al.</i> 2006] . . . . .	103
4.9	L'architecture S-OGSA [Corcho <i>et al.</i> 2006] . . . . .	104
4.10	Plateforme intermédiaire orienté package dynamique [Bilbao <i>et al.</i> 2010] . . .	106
4.11	Adaptateur Lowering et lifting [Bilbao <i>et al.</i> 2010] . . . . .	106
4.12	Le domaine de l'ontologie touristique . . . . .	108
4.13	Classification des types d'ontologie touristiques [Barta <i>et al.</i> 2009, Siricharoen 2010] . . . . .	109
5.1	SQUIREL: l'architecture touristique proposée . . . . .	117
5.2	Cas d'étude durant cette partie . . . . .	119
5.3	La syntaxe du Motif d'expression régulière PPARQL . . . . .	121
5.4	Le prototype EMMA [Garcàa <i>et al.</i> 2012] . . . . .	125
5.5	Données de Tourpedia . . . . .	128
5.6	Présentation des constructeurs du process model en constructeur de séquence- ment . . . . .	133
5.7	Exemple de matrice d'adjacence du service événement . . . . .	134
5.8	Différentes intra-dépendances entre PM . . . . .	141
5.9	Un multi-graphe représentant toutes les possibilités . . . . .	145
5.10	sélection raffinée par Optimisation locale entre les deux processus de réservation	149
5.11	Sélection finale des processus atomiques . . . . .	150
5.12	La Fermeture Transitive de l'exemple touristique . . . . .	152
6.1	Point d'accès SPARQL: virtuoso . . . . .	157
6.2	Découverte orienté composition . . . . .	160
6.3	Nombre de Processus atomiques . . . . .	161
7.1	Les principes des Services Liés Ouverts . . . . .	168

8.1 Présentation de la matrice d'adjacence et sa liste de successeurs . . . . . 184



# Listings

2.1	Exemple de triplets RDF en format turtle . . . . .	27
2.2	Exemple de requête Q de filtrage textuel SPARQL . . . . .	33
2.3	Exemple de requête Q' de filtrant les ressources cibles SPARQL . . . . .	33
5.1	Exemple de requête $Q_1$ basé sur les expression régulière en CPSPARQL . . .	123
5.2	Exemple de filtre $Q_{ALL}$ en SPARQL . . . . .	123
5.3	Exemple de filtre $Q_{SOME}$ . . . . .	124
5.4	La liste des ontologies formant le jeu de Données liées touristique . . . . .	126
5.5	Exemple d'alignement de schéma retourné par un outil de liage des Données Liées Ouvertes . . . . .	128
5.6	Exemple de requête Q exprimant l'intra-dépendance directe . . . . .	140
5.7	Exemple de requête Q exprimant l'intra-dépendance indirecte . . . . .	140
5.8	Exemple de requête Q exprimant l'intra-dépendance directe étendue . . . . .	140
5.9	Exemple de requête $R_Q$ . . . . .	141
5.10	La réécriture de requête $R_{Q'}$ . . . . .	142
5.11	Exemple de requête Q de filtrage textuel SPARQL . . . . .	142





# 1

## Introduction

### Sommaire

---

<b>1.1</b>	<b>Contexte de la thèse</b> . . . . .	<b>1</b>
<b>1.2</b>	<b>Problématique</b> . . . . .	<b>5</b>
<b>1.3</b>	<b>Scénario illustratif</b> . . . . .	<b>9</b>
<b>1.4</b>	<b>Contributions de la thèse</b> . . . . .	<b>11</b>
<b>1.5</b>	<b>Plan de la thèse</b> . . . . .	<b>14</b>

---

### 1.1 Contexte de la thèse

---

Le World Wide Web (ou tout simplement le web) est devenu la première source de connaissances pour tous les domaines de la vie. On peut le considérer comme un vaste système d'information qui permet d'échanger des ressources. L'évolution d'Internet et la compétitivité entre les entreprises ont été les facteurs de l'explosion des Services Web. Les SERVICES WEB [Haas & Brown 2004] étant des applications modulaires, faiblement couplées et auto-descriptives qui parviennent à fournir un modèle basé sur des normes facile à comprendre et à utiliser dans la programmation et le déploiement d'applications.

Les SERVICES WEB [Haas & Brown 2004] sont aptes à s'exécuter à travers le Web lui permettant de passer du rôle habituel de réseaux d'informations à un Middleware d'applications grâce à l'exploitation croissante des technologies XML, ils peuvent constituer un apport de rapidité et d'efficacité pour l'e-business et plus spécialement l'e-tourisme. Quant au WEB SÉMANTIQUE [Berners-Lee 1998, Berners-Lee *et al.* 2001, Antoniou & Harmelen 2008, Hendler 2008] qui est une extension du web d'aujourd'hui visant à donner une forme bien définie et une sémantique aux ressources du web. Répondre aux requêtes est une fonctionnalité essentielle d'un système d'information, et ainsi Du WEB SÉMANTIQUE [Berners-Lee 1998, Berners-Lee *et al.* 2001, Antoniou & Harmelen 2008, Hendler 2008] .

De nos jours, les domaines des Services web et du Web sémantique se croisent pour donner naissance au concept de SERVICES WEB SÉMANTIQUES [Jorge 2006, McIlraith *et al.* 2001] (voir Figure 1.1). L'application des principes du Web Sémantique aux Services Web ont permis de profiter de la considération du sens et de la signification des données échangées pour améliorer la faisabilité des différentes tâches : description, appariement, découverte, composition, substitution etc. offrant, d'une part, une description sémantique explicite de leurs fonctionnalités aussi compréhensible par les agents logiciels que par les utilisateurs

humains, et d'autre part, une interprétation correcte des informations envoyées et reçues.

L'objectif ultime de l'approche SERVICES WEB SÉMANTIQUES [Jorge 2006, McIlraith *et al.* 2001] est de transformer le Web en un dispositif distribué de calcul où les services peuvent interagir de manière intelligente en étant capables de se découvrir automatiquement, de négocier entre eux et de se composer en des services plus complexes. Ces services seraient intuitivement des programmes dont les effets sur leur environnement seraient connus et dont les données manipulées possèderaient aussi une sémantique. En d'autres termes, l'idée poursuivie avec les Web services, est de mieux exploiter les technologies de l'Internet en substituant, au tant que possible, les humains qui réalisent actuellement un certain nombre de services (ou tâches), par des machines en vue de permettre une découverte et/ou une composition automatique de services sur l'Internet.

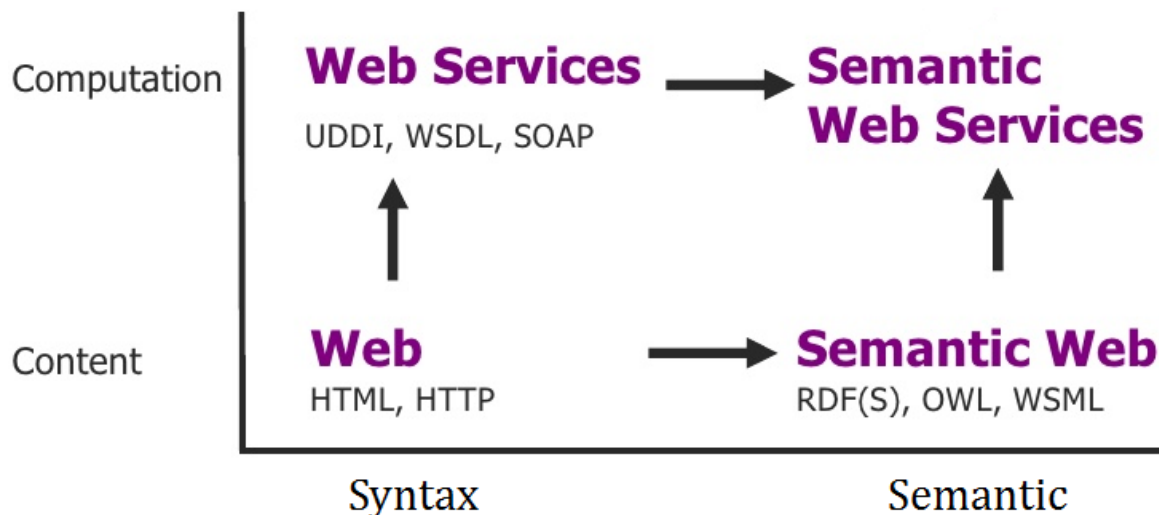


FIGURE 1.1 – L'évolution du Web

L'intégration d'application touristique est l'élément clé de l'évolution actuelle des systèmes d'information comme le montre la Figure 1.2 et des processus en interne à l'entreprise<sup>1</sup> et lors d'entreprise étendue à son réseau de partenaires, l'e-tourisme. En effet, le développement de toute nouvelle application ou d'une procédure commerciale requiert une intégration *manuelle et ad hoc* des sources de données issues d'applications et de bases de données très hétérogènes. La complexité de l'intégration est variable mais elle n'en reste pas moins une des sources de coût des plus lourdes pour les entreprises.

L'objectif des SERVICES WEB [Haas & Brown 2004] est d'aller au-delà des limites de l'intégration telle qu'elle se fait actuellement dans les technologies existantes d'intégration d'applications touristiques. En effet ces dernières induisent en général un grand coût de mise en œuvre, ce qui en restreint l'usage aux grandes entreprises. Or, avec l'explosion et l'ouverture apportées par Internet et l'arrivée massive de petites entreprises<sup>2</sup> (voir Figure 1.4) dans le commerce en ligne, ce n'est clairement plus adapté. Ainsi, le souhait est de passer à des

1. fournisseurs, producteurs et distributeurs d'offres de voyage

2. <http://www.fr.lastminute.com/>, <http://www.tripadvisor.com/>, etc.

## 1.1. CONTEXTE DE LA THÈSE

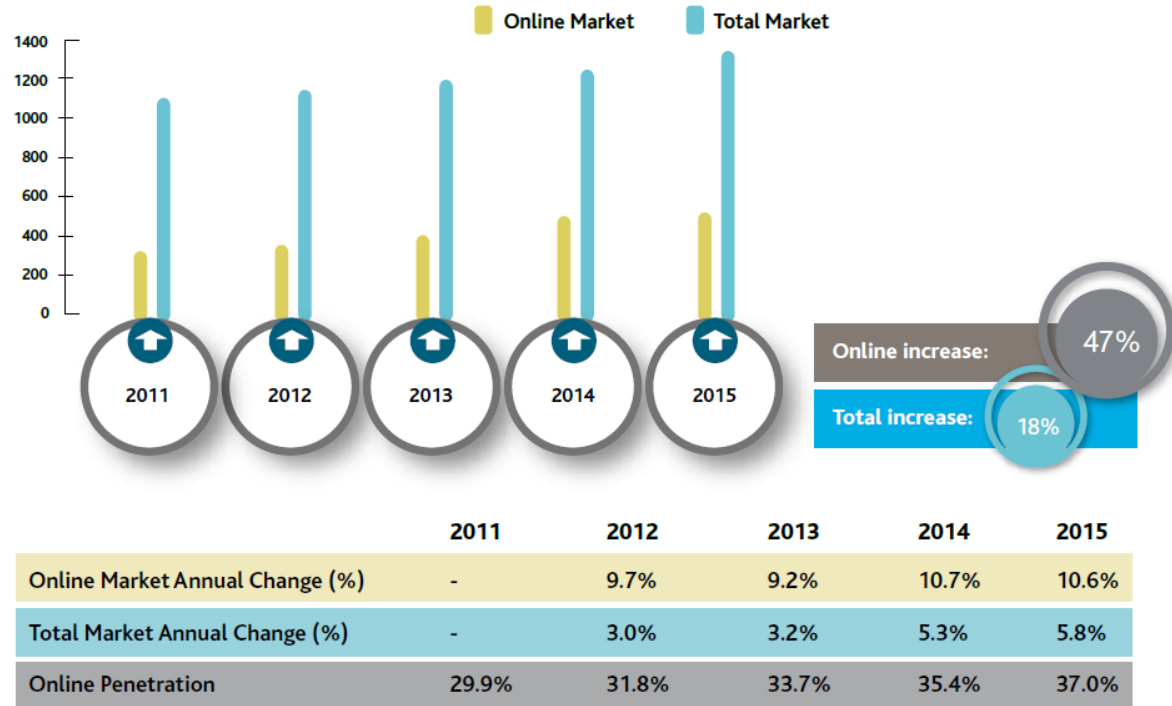


FIGURE 1.2 – L’inflation des sites web touristiques en ligne [Phocuswright 2015]

technologies d’intégration décentralisées où chaque partenaire possède ses propres composants dans un système d’intégration faiblement couplé qui autorise une plus grande flexibilité dans les relations inter-partenaires. Les SERVICES WEB [Haas & Brown 2004] permettent cette évolution où les entreprises commercialisent des prestations de services et des produits de voyages, de séjours, de croisières et de loisirs jouant le rôle d’intermédiaire et/ou d’agrégateur de services entre les clients et les différents prestataires sur le marché du tourisme : compagnies aériennes, hôteliers, loueurs de voiture, compagnies d’assurances, etc.

L’impact potentiel envisagé par ces approches pour des applications touristiques des entreprises, couvre les champs suivants :

- Intégration sémantique en B2B pour le partage des données et services entre entreprises partenaires, mais également pour fiabiliser et sécuriser les processus métiers de partenariats (e.g. remplacement d’un partenaire momentanément défaillant en B2B ; protocole d’échange et de négociation, etc.).
- Interopérabilité sémantique grâce aux métadonnées et aux ontologies qui supportent l’échange flexible et dynamique de données et de services entre systèmes d’information d’entreprises ou unités d’affaires. Elle permet aussi de développer des systèmes d’interrogation de données touristiques, de découverte et d’intégration de services transactionnel automatisée et très fiable.
- La gestion des cycles de vie des processus métiers des entreprises où les métadonnées, les ontologies et les règles deviennent des outils incontournables pour la modélisation et le traitement des données, des processus, des règles commerciales et finalement des

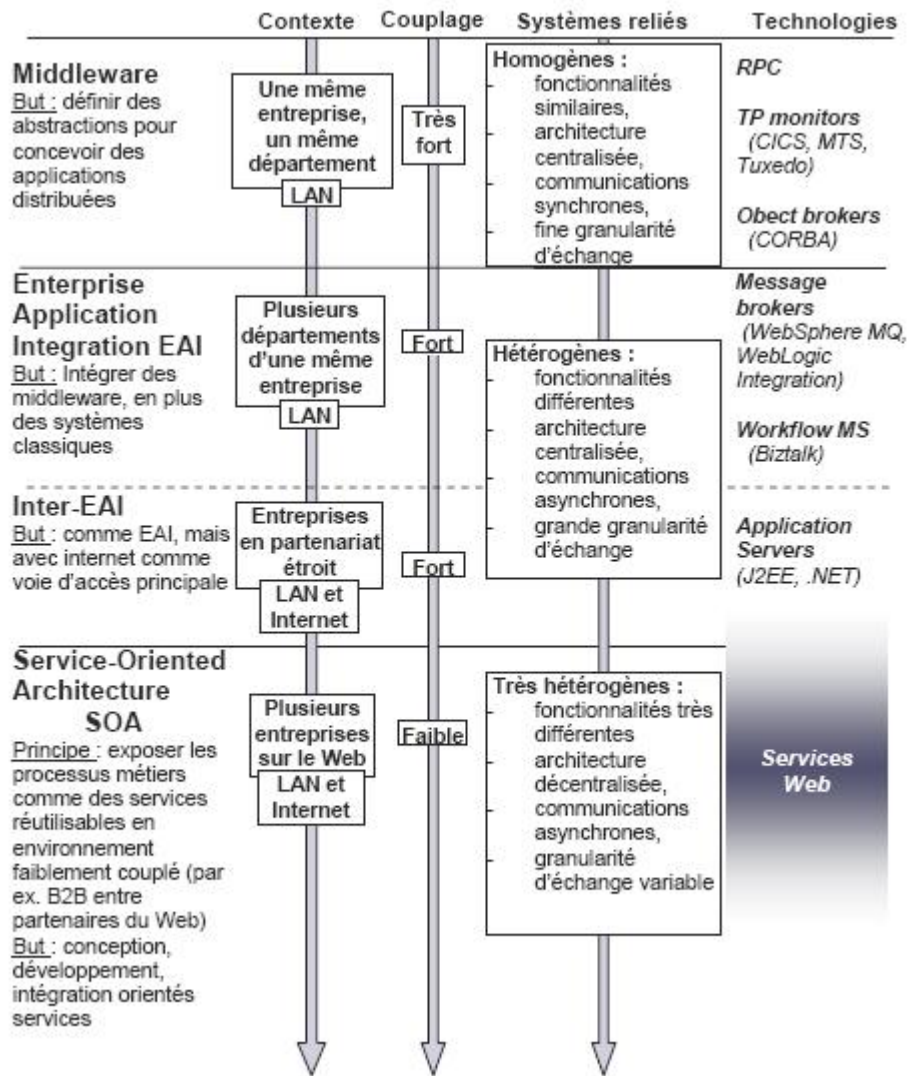


FIGURE 1.3 – Evolution des technologies d'intégration d'applications [Hacid et al. 2009]

savoirs de(s) entreprise(s).

Contrairement aux processus business "traditionnels" qui sont exécutés de manière pré-visible et répétitive dans un environnement statique, les Services web composés s'exécutent dans un environnement versatile où le nombre de services disponibles évolue très rapidement.

De plus, la forte compétition engendrée par la multitude de fournisseurs de services de réservation en ligne<sup>3</sup> et l'apparition du "forfait ou package dynamique" qui permet à un client de composer, sur le site du prestataire, un voyage sur-mesure, à la carte, comme pourrait le proposer un agent de voyage, oblige les entreprises à adapter leurs services pour mieux répondre aux besoins des clients et ce à moindre coût. Ces deux facteurs imposent des contraintes fortes sur les systèmes qui délivrent des services composés. En conséquence, les processus métiers

3. <http://www.fr.lastminute.com/>, <http://www.tripadvisor.com/>, etc.

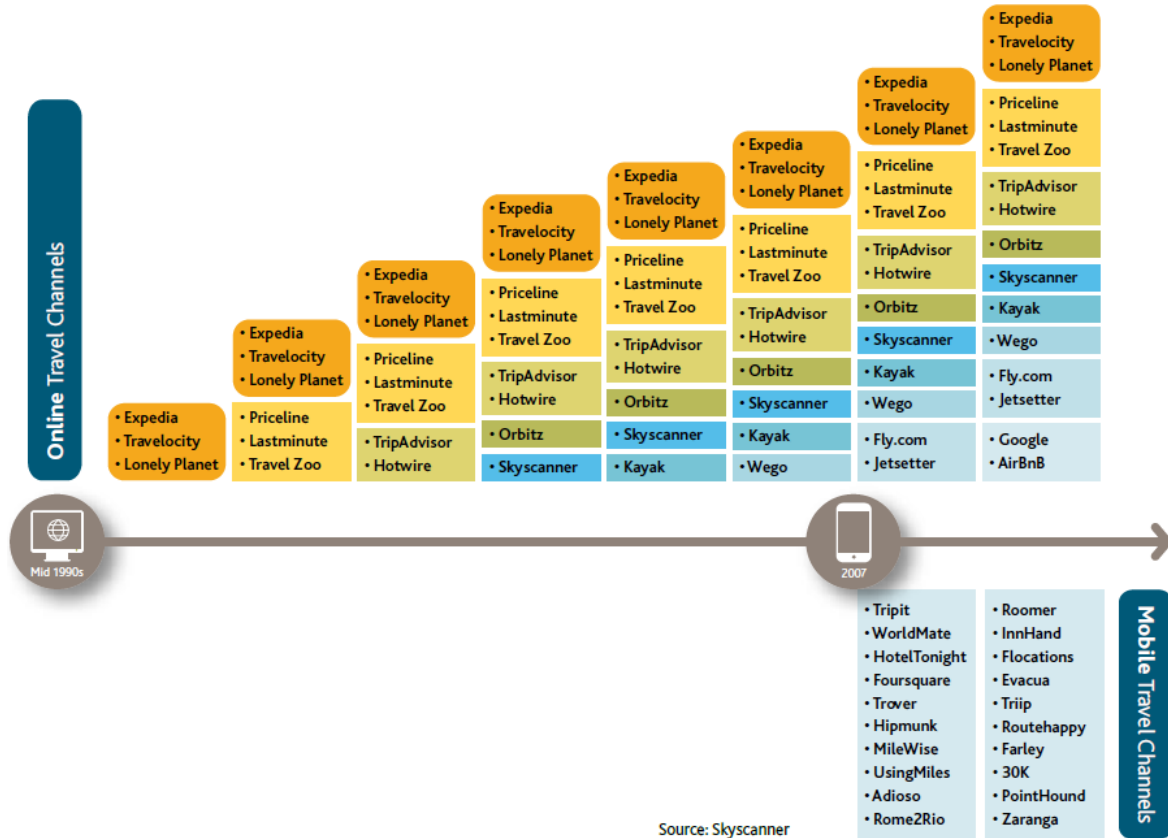


FIGURE 1.4 – Explosion des sites de réservation en ligne [Phocuswright 2015]

qui décrivent des services composés devront intégrer d’emblée ces contraintes en exhibant des possibilités réelles d’adaptabilité à leur environnement.

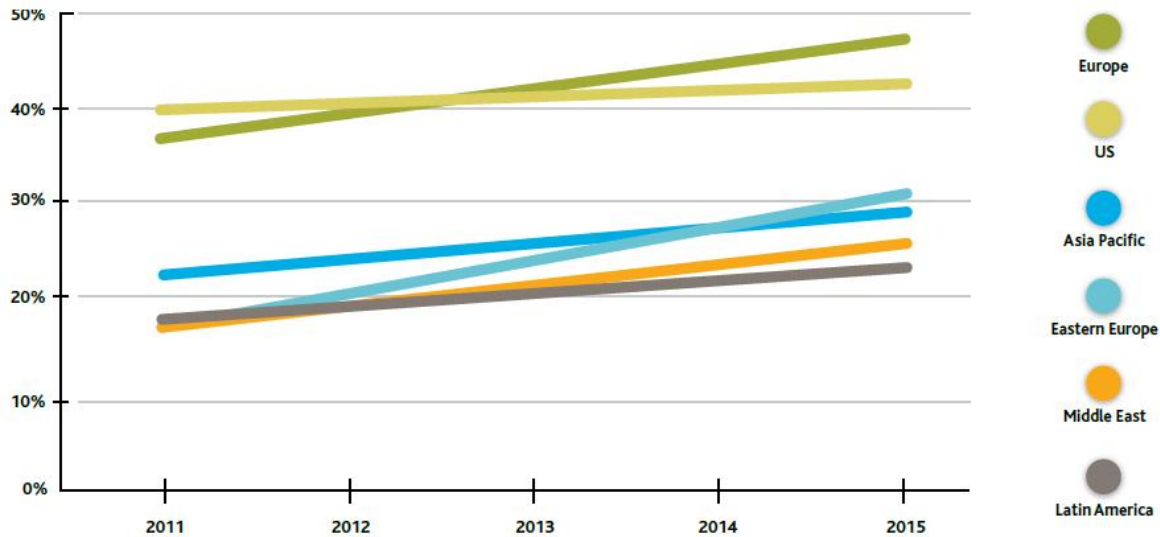
## 1.2 Problématique

L’e-tourisme représente déjà la première activité commerciale sur le Web. La majorité des consommateurs ont utilisé Internet pour préparer leur voyage, au point que la vente en ligne a dépassé en valeur la vente traditionnelle en agence. Côté client, on assiste à la naissance d’un nouveau consommateur, en plus grande recherche de qualité et de sécurité, en demande accrue de personnalisation. Un consommateur plus opportuniste, avec un besoin marqué d’inédit et d’originalité, mais aussi d’authenticité. L’élaboration de produits "sur-mesure", innovants et compétitifs ne sera possible que si les territoires deviennent "intelligents", appelé "Package dynamique".

Pour cela, nous examinons les différents aspects de la problématique liés à notre domaine en termes qualitatif et quantitatif. Les approches de composition sont d’avantage jugées sur des questions de précision, de rappel et de temps de réponse :

1. **Le problème d’accessibilité des données orienté tourisme** : Accéder à l’infor-

Online Travel Penetration by Region, 2011-2015



Note: 2013-2015 projected.

Source: Global Online Travel Overview Third Edition ©2014 PhotoCusWright: Inc. All Rights Reserved.

FIGURE 1.5 – La forte croissance de réservation de séjour en ligne par pays [Phocuswright 2015]

mation pertinente, intègre et authentique n'est pas facile dans de nombreux cas et pour diverses raisons, Le consommateur est appelé à parcourir plusieurs site Web afin de trouver l'offre adéquate à sa requête initiale et de faire plusieurs réservations également sur des sites web disparates<sup>4</sup> : hébergement d'un coté, vol de l'autre etc ce qui devient une tâche fastidieuse, longue et décourageante pour le consommateur.

2. **Le problème lié au besoin d'automatisation** : de la composition dans un environnement dynamique et adaptatif : n'oublions pas qu'Internet est en perpétuelle évolution. Les fournisseurs proposent continuellement de nouveaux services Web qui peuvent être pris en compte, afin d'améliorer la qualité des données, proposer de nouvelles fonctionnalités grâce aux nouvelles compositions possibles de manière complètement automatique sous la forme de services Web, qui est encore plus difficile à mettre en œuvre avec les systèmes classiques. *Ce type de composition est encore très peu utilisé.* Et du fait qu'ils sont généralement établis par des organisations différentes qui utilisent différents modèles conceptuels pour la présentation des caractéristiques des services. Cela exige l'utilisation d'informations pertinentes (les propriétés fonctionnelles et non-fonctionnelles) pour faire correspondre (matching) et composer les services web.

3. **Le problème lié au choix des connaissances essentielles sur les Services**

4. LastMinute, TripAdvisor, Seekda, Expedia, etc.

**Web** : Les méthodes proposées dans la littérature portant sur la composition, utilisent généralement les attributs fonctionnels et des attributs non-fonctionnels définis dans le Service Profile pour proposer à l'utilisateur les services qui semblent les plus appropriés pour répondre à sa requête, cela dit parfois le Service Profile peut être incomplet c'est-à-dire qu'il ne contient pas d'Input ou bien d'Output dans sa description ce qui rend la composition délicate à générer. En plus, Il existe très peu de travaux portant sur la composition du Service Model d'OWL-S , le Process Model qui fournit le fonctionnement du service Web.

4. **Le problème lié aux défaillances des services Web** : Le besoin de prise en charge des défaillances des services Web, lorsqu'un service web tombe en panne, que faire pour la composition ?
5. **Le problème lié à la traduction en un autre langage de planification** : Etant obligé de traduire le SWS en un langage de planification ? et également faire appel à des outils de planification ? ce qui revient à perdre du temps d'exécution et de réponse.

Les avantages attendus de la technologie sémantique de services Web concernent l'utilisation efficace des services Web. La décision d'utiliser un service particulier est faite à l'exécution en temps réel à la place du développement temps du logiciel service demandeur. Il est donc possible d'utiliser les services qui ne sont pas disponibles, ou même existants, au moment du développement mais ne sont découverts lors de l'exécution en temps réel. Cela va considérablement améliorer la convivialité de la technologie des services Web dans des environnements volatiles, tels que l'informatique mobile et ubiquitaire, où le paysage des services disponibles peut changer rapidement, par exemple, quand un dispositif se déplace d'un espace à un autre. Dans la littérature, nous retrouvons différentes approches de composition automatique : les approches sémantiques, les approches basées sur les automates et les approches basées sur les techniques de planification qui nécessitent de traduire les SERVICES WEB SÉMANTIQUES [Jorge 2006,McIlraith *et al.* 2001] en un autre langage et de définir la requête utilisateur sous forme d'un service web. Le second axe concerne la composition dynamique des Process model, un modèle complexe et explicite très peu abordé dans la littérature [Brogi *et al.* 2008,Paulraj *et al.* 2012] , des approches peu évolutives qui reposent sur la composition d'une petite collection des Process Model.

La problématique lié au tourisme sémantique est notamment étudiée par plusieurs approches [Cardoso & Schauer 2006,Mànguez *et al.* 2010,Corcho *et al.* 2006,Bilbao *et al.* 2010] pratiques visant à utiliser des technologies du Web sémantique pour contribuer au développement des packages dynamique. Une spécificité de ce contexte, assez peu abordé dans la littérature existante, est sans doute le besoin d'inciter les découvertes des SERVICES WEB SÉMANTIQUES [Jorge 2006,McIlraith *et al.* 2001] inattendues et fortuites à travers la richesse des Données Liées Ouvertes (LOD), une richesse qui repose notamment sur la diversité de types de liens et de ressources élargissant ainsi le champ de recherche par lesquelles une requête sémantique est initialement décrite, en y incluant des concepts similaires permettant ainsi de solliciter les compétences de d'autres services. Partant de ce point, notre premier objectif est de proposer une approche permettant à l'utilisateur d'accéder aux données pertinentes tout en ayant la possibilité d'exprimer ses préférences / contraintes dans sa requête. Compte tenu du contexte de notre travail et des besoins décrits ci-dessus, nous avons identifié différents points à considérer comme sous-objectifs :

1. Proposer une approche de collecte des données touristiques à partir du nuage des données liées ouvertes existantes en réutilisant les vocabulaires orienté et centré e-commerce déjà disponibles tels que GoodRelations ; afin d'enrichir la spécification d'informations sémantiques.
2. Proposer d'intégrer une couche d'interrogation sémantique qui couvre les principales préférences et contraintes exprimées par notre consommateur.
3. Proposer une approche pertinente et complètement originale qui s'inspire des techniques de planification en se basant sur des spécifications déclaratives du comportement avec le plus possibles d'éléments de chaque service et qui ne dépend pas d'un langage/outil de planification en particulier, *i.e.*, ne faisant aucune traduction au préalable, en se basant uniquement sur les descriptions sémantiques de chaque service. Nous nous intéressons plus spécifiquement à des propriétés d'atomicité. Le modèle permet la prise en compte des propriétés d'atomicité des composants. Tirant partie de l'existence d'un grand nombre de services répondant aux mêmes besoins fonctionnels et afin d'augmenter les chances pour les compositions automatiques de s'exécuter avec succès, l'étude s'appuie sur un principe de sélection de services à base de données liées, à l'exécution. Le concept de "clusters de services web" fournit un cadre à cette sélection. Pour cela, nous faisons appel d'une part aux techniques (1) des théories de graphes appliquées à la matrice de dépendance et d'autres part aux (2) techniques d'alignement d'instances issues des Données Liées Ouvertes : plus on réussit à apparier d'éléments des deux services, requis et offert, plus le résultat de composition est pertinent, et plus on identifie avec précision la similarité entre les éléments des deux services, requis et offert, plus le résultat de découverte est pertinent. L'algorithme doit générer un enchaînement unique portant sur les composants des SERVICES WEB SÉMANTIQUES [Jorge 2006,McIlraith *et al.* 2001] sur la base de cette matrice enrichie et les paramètres fonctionnels de chaque service. Cependant, il existe différentes normes sur les SERVICES WEB SÉMANTIQUES [Jorge 2006,McIlraith *et al.* 2001] (par exemple, OWL-S [Burstein *et al.* 2004] , SAWSDL [Kopecký *et al.* 2007] , WSMO [Bruijn *et al.* 2005] etc..). Nous adoptons le langage OWL-S [Burstein *et al.* 2004] qui s'appuie sur la sémantique opérationnelle d'Erlang [Cesarini & Thompson 2009] et les Concurrent Haskell Programs [Peyton Jones *et al.* 1996] pour décrire la sémantique d'exécution de la composition étudiée et du fait qu'il existe des similarités entre l'approche de description et de composition OWL-S [Burstein *et al.* 2004] et le Planning Domain Definition Language (PDDL) [Mcdermott *et al.* 1998a]. Ce principe a interpellé plusieurs communautés de recherche travaillant dans le domaine de l'Intelligence Artificielle [Russell & Norvig 2003]. Notre algorithme repose essentiellement sur l'alignement de schémas fournis par les données liées ouvertes inter-reliant ainsi les descriptions sémantiques des Process Models mettant en avant l'interdépendance entre eux en se basant évidemment sur la requête initial des consommateurs. De là, Les consommateurs seront en mesure d'énoncer une tâche complexe qu'ils veulent accomplir et le système doit renvoyer un (simple ou composite) WS capable de mener à bien la tâche.



### 1.3 Scénario illustratif

Pour exemples très illustratifs, nous trouvons de manière classique : i) le service web d'agence de voyages qui fournit une liste de vols disponibles et leurs coûts étant donné la ville de départ, la ville d'arrivée, les dates d'aller et retour<sup>5</sup>. ii) le service web de réservation de chambre d'hôtel étant donné les dates d'arrivée et de départ, la ville, le pays et un moyen de paiement<sup>6</sup>. Déjà sur ces exemples le paiement s'effectue généralement via un service bancaire associé ainsi que la livraison qui est assurée par un service tiers de logistique. Il s'agit d'arriver à un Web "intelligent", où les informations ne seraient plus stockées mais "comprises" par les ordinateurs. A titre d'exemple, le Web sémantique doit être capable de donner une réponse précise à une question du type " Je veux partir en vacances l'été prochain en Australie. J'ai un budget de 5 000 €. J'aime les plages et le tennis ". Aujourd'hui, répondre à une telle question va exiger des heures de tri dans des listes distinctes d'hôtels, réservation de vol et la location de voitures

Nous choisissons d'illustrer la problématique introduite ci-dessus par un scénario dans lequel une application est conçue pour l'organisation de voyages (voir Figure 1.6). L'organisation d'un voyage nécessite l'achat d'un billet d'avion aller/retour, la réservation d'une chambre d'hôtel, le cas échéant la location d'un moyen de transport et la consultation de la météo pour le lieu de la destination. Chacune de ces tâches s'appuie sur un ensemble d'opérations (processus atomiques) qui sont regroupés, dans la perspective d'en faire abstraction, sous le terme de *Process Model* de service. La figure 1.6 montre par un graphique les différents services disponibles pour la composition qui découle de cette analyse. Cette étape de la conception fait abstraction du service qui remplira effectivement la capacité identifiée.

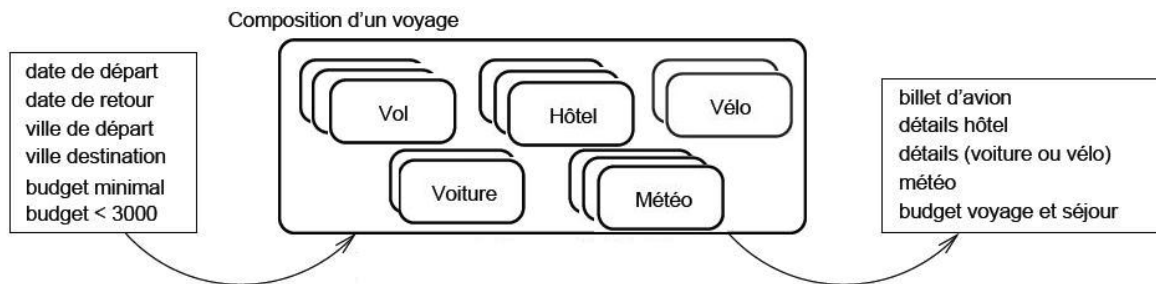


FIGURE 1.6 – Les capacités impliquées dans l'organisation d'un voyage (inspiré de [Hacid et al. 2009])

L'étape suivante consiste à déterminer les capacités parmi celles décrites plus haut, qui sont indispensables et pertinentes à l'élaboration du voyage en fonction de la requête de l'utilisateur en maximisant ces fonctions de préférences. Le fait de disposer de référentiels où sélectionner plusieurs services spécifiques et appropriés à la composition, facilite la construction d'un ensemble d'instances pour chaque configuration possible. Cependant, l'explosion des sites web touristiques met en avant l'existence de plusieurs services couvrant le même

5. <http://www.govoyages.com/>, <http://www.promovols.com/>, <http://www.ebookers.fr/>

6. <http://www.tripadvisor.fr/>, <http://www.trivago.com/>, <http://www.booking.com/>, <http://www.laterooms.com/>, <http://www.agoda.com/>

## CHAPITRE 1. INTRODUCTION

domaine (services d'hébergement, de consultation météo, de location de moyen de transport, etc). Le voyage est possible en configurant ces différents services en fonction de la demande de l'utilisateur, la composition générée doit combiner le meilleur service de chaque domaine et ce en fonction des opérations définies dans chaque Process Model ayant la satisfaction maximale en fonction de ses entrées et sorties sémantiques.

L'étape suivante consiste à associer un service à chaque capacité identifiée. Puisqu'il existe plusieurs services web qui répondent aux mêmes besoins fonctionnels, la question qui se pose alors est : où et comment trouver le service qui répond le mieux aux besoins requis par une capacité ? La notion de *communauté ou cluster* de services nous aide à fournir un cadre pour effectuer la sélection d'un service parmi ceux en concurrence pour une capacité donnée (voir Figure 1.7).

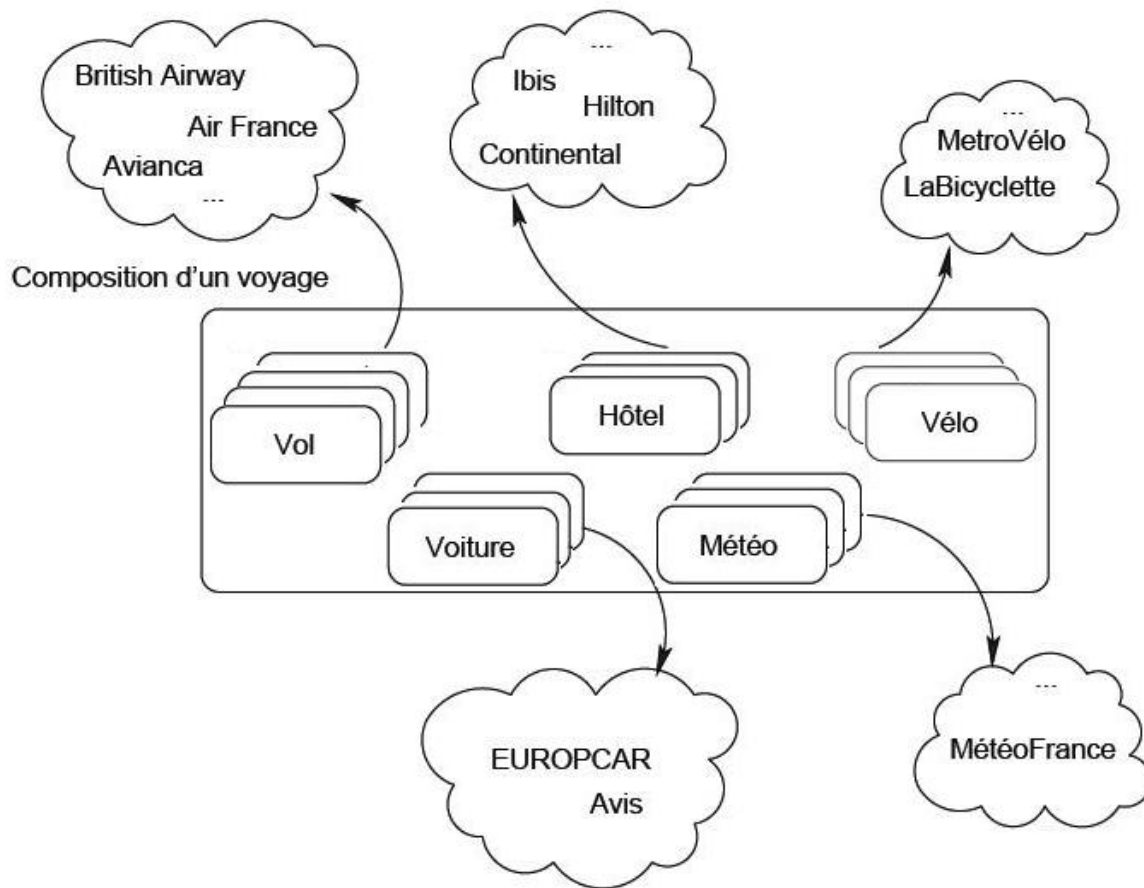


FIGURE 1.7 – Les capacités et les instances de services (inspiré de [Hacid et al. 2009])

La connaissance des propriétés fonctionnelles des services transactionnelles (réservation) à considérer permet de construire ces clusters de services dont les chances de réussite sont maximales. La prise en compte de ces services iso-fonctionnels pour un voyage nécessite d'assouplir la propriété transactionnelle d'atomicité : les capacités impliquées dans le Process Model ne doivent pas toutes nécessairement s'exécuter, la nécessité de raffiner le sous-ensemble de ces services à ces processus atomiques pertinents en éliminant les processus atomiques super-

flus et infructueux à la configuration de la composition en fonction des données en entrée et celles attendues en sortie définies dans la requête. Ainsi, la paramétrisation de la composition est utilisée pour guider la construction et la classification de ces services par rapport aux paramètres spécifiés.

L'exécution de certaines peut échouer, par exemple parce qu'aucun service n'est disponible pour répondre aux besoins de la capacité, ou bien parce que la ressource désirée ne peut pas être acquise. Cet échec peut se produire sans pour autant avoir comme conséquence l'annulation de l'exécution en cours de la composition.

En conclusion, afin d'obtenir la composition optimale et finale, nous préconisons (i) l'assouplissement de la propriété d'atomicité et non la complexité des processus par la restriction du Process Model au minimum requis en terme de processus atomiques utiles et non les inutilisés, (ii) la paramétrisation de la composition pour guider la sélection des services candidats et potentiels en proposant un algorithme d'optimisation locale, (iii) la présélection appropriée de Services Web Sémantiques (Process Model) se fait par le biais d'une requête sémantique exprimé en langage sémantique SPARQL.

### 1.4 Contributions de la thèse

---

La nécessité est de mettre à disposition un canevas intelligent spécialisé dans le e-tourisme et conçu pour développer des Packages dynamiques intelligents en temps réel et répondant ainsi aux caractéristiques, besoins et attentes du consommateur et ce afin de personnaliser l'offre d'information et d'accroître la satisfaction des consommateurs et également de raccourcir le temps qu'ils passent à naviguer sur Internet. Pour cela, Nous pouvons faire appel aux données orienté e-commerce, et plus spécialement les GoodRelations permettant de valoriser les ressources d'un territoire donné en présentant une information riche, précise et pertinente, tout en facilitant l'accessibilité et la compréhension de l'offre du côté des consommateurs. Une telle solution doit permettre aux offreurs / prestataires (territoires, opérateurs ou marchands) de mettre en avant leurs offres en temps réels, et aux consommateurs de disposer d'une application spécialisé permettant la conception de séjours sur mesure à partir d'une information filtrée, organisée, spécialisée et contextualisée leur permettant :

- Soit d'interroger le réseau de Données Liées dédié au tourisme (recherche sémantique orientée préférences et contraintes),
- Soit de passer directement à la phase réservation de son voyage en utilisant un algorithme dynamique et adaptatif de composition purement automatique de services web sémantiques qui consiste à choisir, à l'exécution, une composition répondant le mieux aux besoins du client reposant sur le standard recommandé tel que OWL-S [Burstein *et al.* 2004] et ce en faisant appel aux techniques de théorie de Graphes combinées aux techniques matricielle, tout en garantissant la qualité du résultat.

Les contributions du travail, rassemblées dans le canevas SQUIREL [Merazi & Malki 2015, Merazi 2011] , sont :

- Un modèle dans lequel la composition s'exprime en termes de capacités de services et non pas en termes de services web spécifiques. Une capacité de services est associée à un ensemble de besoins fonctionnels souhaités (par exemple la réservation d'une chambre d'hôtel). Le modèle permet au concepteur d'établir ses propres capacités.

- Un modèle de "découverte orienté composition automatique de services web munis de préférences" pertinent entre une requête du consommateur et les descriptions sémantiques de services publiés qui garantit au consommateur **la qualité du résultat** et également **le temps de réponse**.

L'approche de chorégraphie de services que nous voulons concevoir se veut :

- *automatique*, i.e. n'ayant pas recours à l'utilisateur pour la sélection et la combinaison des services appropriés ;
- et *dynamique*, i.e. les services existants sont disponibles dans un environnement ouvert tel que le web ou l'intelligence ambiante, et sont découverts et composés à la volée, en fonction des besoins de l'utilisateur et de leurs fonctionnalités, par le biais d'un mécanisme de coordination explicite se basant sur la théorie de matrices . La conception d'une telle approche nécessite de disposer de services capables de déterminer s'ils peuvent prendre en charge, en fonction de leurs compétences, tout ou partie des besoins de l'utilisateur à satisfaire. Ceci requiert des services de pouvoir raisonner sur les besoins de l'utilisateur à satisfaire ainsi que sur leurs propres fonctionnalités afin de décomposer adéquatement les besoins de l'utilisateur. L'ensemble des services participant à la chorégraphie doivent alors pouvoir interagir de manière flexible, puisqu'ils ne sont pas connus a priori les uns des autres, et se coordonner, en fonction de ce que chacun a accompli et de ce dont il a besoin, afin découvrir l'intégralité des besoins à satisfaire.
- Un modèle dans lequel il est possible de paramétrer la composition par des contraintes et des préférences fixées par le concepteur et qui permettent de guider la sélection des services, cette sélection étant effectuée au moment de l'exécution de la composition.
- Développer un algorithme optimal et complet pour la Composition dynamique et automatique de service Web sémantiques transactionnels définissant ainsi les packages dynamiques qui permet un développement plus rapide des applications à base de services génère la composition de service adéquate de manière transparente à l'utilisateur.

L'architecture proposée permet une approche de composition d'applications à base de services qui combine à la fois une démarche descendante et une autre ascendante La démarche descendante, du niveau interrogation sémantique jusqu'au niveau réservation, permet de modéliser l'application, et de propager les préférences de l'utilisateur en vue de choisir une composition de services répondant aux contraintes de l'utilisateur. La partie ascendante de la démarche projette le résultat qui est retourné à l'utilisateur. Nous proposons une architecture de composition à trois niveaux, à savoir, le niveau interrogation sémantique, le niveau information sémantique et le niveau réservation/booking. D'abord le niveau interrogation sémantique supporte un langage prenant en considération les préférences de l'utilisateur exprimé en C<sub>SPARQL</sub> [Alkhateeb *et al.* 2008] , un langage étendu du SPARQL [Franconi & Tessaris 2005, Prud'hommeaux & Seaborne 2008, Arenas & Pérez 2011, Pérez *et al.* 2006] ; Une deuxième couche constitue l'ensemble de jeux de données liées ouvertes relatant le domaine de l'e-tourisme permettant de valoriser les offres des prestataires. Ensuite le niveau réservation, constitue la réalisation de la composition qui requiert la description et l'organisation de l'interaction entre les services en se basant des informations fournies par la deuxième couche. Elle nécessite la gestion de plusieurs aspects comme les échanges de données entre les services, les pannes ou erreurs éventuelles, le contexte d'interaction, le degré d'automatisation

des tâches, etc.

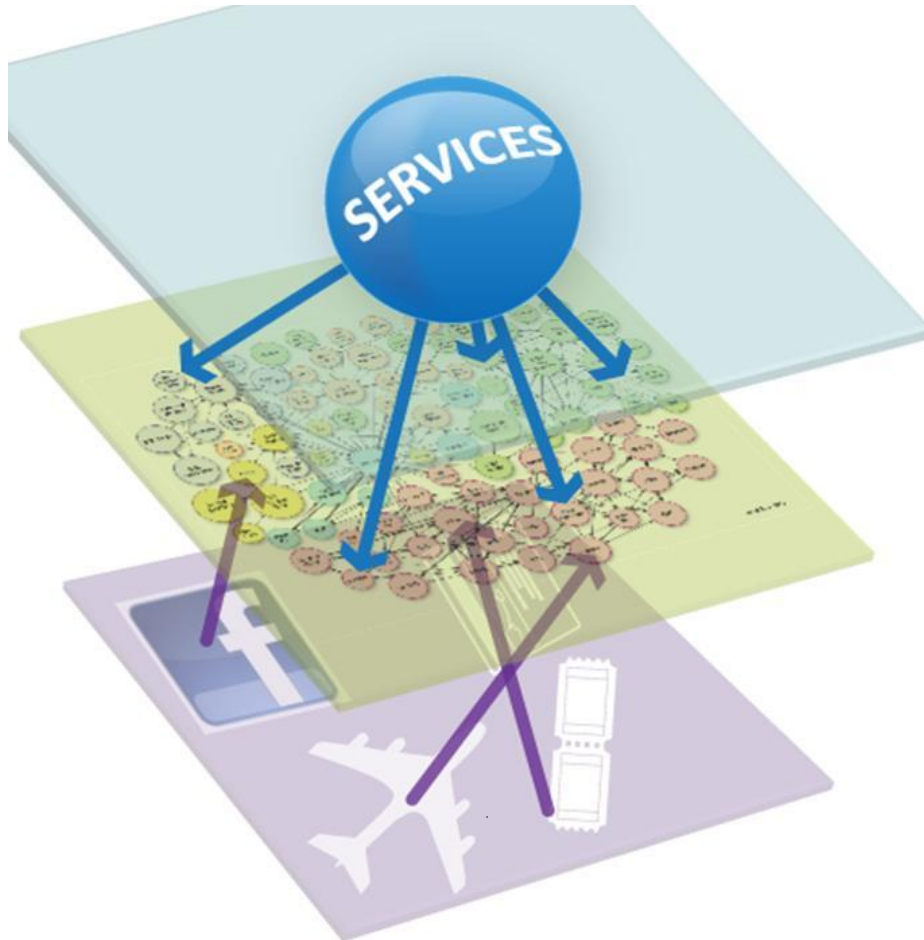


FIGURE 1.8 – Les Descriptions de services reposant sur les Données Liées touristiques

Pour cela, Nous avons développé un prototype en Java qui implémente un outil de planification automatique reposant sur la composition de Service Model et plus spécifiquement le Process Model de chaque services Web qui fournit une description détaillé sur le comportement des Processus atomiques. Le résultat généré représentera la package dynamique dont les descriptions de services reposent sur les données liées touristiques (voir Figure 1.8). L'expérimentation de notre algorithme optimal et complet pour la Composition dynamique et automatique de service Web sémantiques vise à vérifier l'efficacité de la méthodologie qu'on a choisi d'utiliser. En outre, cet ensemble de tests reposent sur un ensemble de plusieurs scénarios qui sont utilisés comme cas pilote pour la composition automatique. Néanmoins, notre intention est de prendre en considération le fait que les services composites risquent de ne pas s'exécuter correctement pour diverses raisons, comme la congestion du réseau, ou de l'indisponibilité d'un service impliqué dans le système.

### 1.5 Plan de la thèse

---

Ce chapitre a présenté une introduction à notre travail de recherche, suivi de la problématique que nous avons traitée, des objectifs que nous poursuivons et des contributions que nous avons apportées.

Notre document a été organisé en deux parties. La première partie est une étude bibliographique, elle-même organisée en trois chapitres : le chapitre 2, présente des éléments clés de l’approche services web sémantiques : web sémantique et données liées, notion de services, description d’un service, les fonctions types d’une architecture orienté service. Ce chapitre introduit aussi les caractéristiques principales de la technologie des services web et sa problématique. Dans cette partie de cette thèse constituant ce panorama le chapitre 3 donne une vision d’ensemble détaillée de la découverte de services web sémantiques dans le contexte spécifique des services ontologiques OWL-S. Puis après une vue d’ensemble, nous terminons par une transition vers la composition de services web. Nous abordons la question de la composition de services web et listons les différentes approches existantes. Nous présentons les avantages et les limites des standards les plus répandus ainsi que les différentes approches spécifiques proposées par la recherche académique et industrielle. Enfin nous concluons par une feuille de route recherche et technologique sur ces technologies. Quant au chapitre 4, il définira la notion de package dynamique, les enjeux liés à cette technologies. Puis nous détaillerons les approches reposant sur les technologies des services Web et web Sémantique pour développer ces packages dynamiques.

La deuxième partie du document présente notre proposition de canevas pour la découverte orienté composition de services web sémantiques munis de propriétés transactionnelles. Cette partie est formée de deux chapitres. Le chapitre 5 présente l’architecture générale de notre prototype nommé SQUIREL, nous détaillerons les trois couches interdépendantes. Nous détaillons et formalisons les mécanismes proposés, initialement Cette approche nous permet aussi de donner un cadre à la sélection de services web en améliorant une approche existante combiné avec les avantages de l’utilisation des données liées touristiques. Nous détaillerons notre algorithme d’optimisation appliqué sur les graphes de cluster de services et expliquons comment générer le plan final. Le chapitre 6 illustre l’utilisation du canevas SQUIREL [Merazi & Malki 2015, Merazi 2011] . Nous présentons les détails de la mise en œuvre du canevas proposé.

Finalement, le dernier chapitre 7 dresse le bilan de notre travail et donne les perspectives que notre travail a ouvert et effectue la liaison avec d’autres travaux de recherche en cours.

Première partie

Background & États de l'art





# 2

## Évolution du Web

### Sommaire

---

<b>2.1 Introduction</b>	<b>17</b>
<b>2.2 Le web sémantique</b>	<b>18</b>
2.2.1 Données, Information et Connaissance	21
2.2.2 Représentation des connaissances avec RDF(S) et OWL	22
2.2.3 Ontologie, sens de données et raisonnement	27
2.2.4 Interrogation du Web Sémantique	31
2.2.5 Données liées- Le Web sémantique d'aujourd'hui	34
<b>2.3 Description de Services Web sémantiques</b>	<b>46</b>
2.3.1 Description syntaxique des Services Web	49
2.3.2 La nébuleuse des langages sémantiques	51
<b>2.4 Conclusion</b>	<b>59</b>

---

### 2.1 Introduction

---

L'évolution du Web ne semble pas converger vers une nomenclature commune selon la catégorisation proposée par Nova Spivack<sup>1</sup>. Cette perspective propose une vision de cette évolution du Web sous forme de " versions distinctes " que nous allons énumérer :

- **Web 1.0** : le web pyramidal, une ressource "read only" conçu pour échanger et partager des documents statiques où les acteurs sont passifs, l'information est diffusée vers les utilisateurs de manière unidirectionnelle à des fins consultatives.
- **Web 2.0** : le web participatif ou social, une ressource " read and write " dont le canal de communication repose sur les échanges sociaux entre les utilisateurs (les consommateurs créés du contenu et le partage).
- **Web 3.0** : le web sémantique [Berners-Lee *et al.* 2001] qui est l'extension du Web 1.0, le rendant ainsi plus intelligent où les données y sont représentées de manière à ce que les machines puissent en comprendre le sens d'une manière concrète indépendamment des supports, donnant ainsi une signification aux balises XML [Maler *et al.* 2004, Bray *et al.* 2006] .

---

1. <http://www.slideshare.net/novaspivack/web-evolution-nova-spivack-twine>

- **Web 4.0** : C'est une réalité augmentée où l'environnement est cliquable : les différents capteurs, GPS, boussoles et centrales à inertie permettent d'obtenir des informations sur les lieux ou objets qui nous entourent.
- **Web 5.0** : l'ère de l'homo-interneticus, les utilisateurs ne seront plus sur le web mais dans le web en permanence, presque omniscient d'ici une dizaine d'années. Les objets, les pièces et les humains eux mêmes seront dotés de puces et formeront le symbionet.

Dans cette partie, nous introduirons tout d'abord les principes du WEB SÉMANTIQUE [Berners-Lee 1998, Berners-Lee *et al.* 2001, Antoniou & Harmelen 2008, Hendler 2008] et des formalismes de représentation associés [Euzenat 2002], tout en revenant plus particulièrement sur certains aspects qui nous paraissent fondamentaux pour la bonne compréhension de cette thèse. Nous présenterons le langage RDF [Carroll & Klyne 2004, Hayes 2004, Lassila & Swick 1999, Manola & Miller 2004, Beckett 2009b] (Section 2.2.2.1), qui permet de représenter les données sur le WEB SÉMANTIQUE [Berners-Lee 1998, Berners-Lee *et al.* 2001, Antoniou & Harmelen 2008, Hendler 2008] selon la vision du W3C, avant d'aborder les notions de vocabulaires et d'ontologies [Gruber 1993] ainsi que les langages associés, à savoir RDFS [Munoz *et al.* 2009] et OWL [McGuinness & Harmelen 2004, Patel-Schneider *et al.* 2004] (Section 2.2.3). Si ces différents points sont relatifs à la production de données, ou annotation sémantique, il est nécessaire de présenter les mécanismes relatifs à leur interrogation. Nous expliciterons ainsi l'utilisation du langage et protocole SPARQL [Franconi & Tessaris 2005, Prud'hommeaux & Seaborne 2008, Arenas & Pérez 2011, Pérez *et al.* 2006] (Section 2.2.4), celui-ci jouant un rôle important dans l'avènement du Web sémantique. Enfin, nous aborderons l'initiative des DONNÉES LIÉES [Heath & Bizer 2011, Berners-Lee 2009, Bizer *et al.* 2009, Berners-Lee 2006] (Linked Data) (Section 2.2.5), qui vise à interconnecter un grand nombre de données RDF [Carroll & Klyne 2004, Hayes 2004, Lassila & Swick 1999, Manola & Miller 2004, Beckett 2009b] présentes sur le Web, dans une vision plus pragmatique du Web Sémantique et de ce que l'on appelle maintenant plus communément Web de données (Web Of data).

## 2.2 Le web sémantique

---

Dans sa forme originelle, le Web consiste en un système d'information décentralisé contenant un ensemble de documents inter-reliés par des liens hypertexte. Toutefois, l'explosion de la taille du web et de la diversité de ses contenus a privilégié la mise en disposition des moteurs de recherche, un moyen rapide d'accès aux informations contenues sur le web. Néanmoins, les moteurs de recherche démontrent leurs limites face aux requêtes complexes par exemple ceux qui nécessitent la fusion d'informations contenues dans des documents distincts car les machines n'appréhendent pas le sens des documents qu'elles traitent. L'avènement du WEB SÉMANTIQUE [Berners-Lee 1998, Berners-Lee *et al.* 2001, Antoniou & Harmelen 2008, Hendler 2008], un projet initié par Tim Berners-Lee en 2001 est une évolution du web documentaire qui interconnecte non plus seulement des documents mais des ressources en tous genres, en ajoutant une couche de sémantique au web de manière à rendre l'information interprétable par les ordinateurs afin de faciliter la découverte, l'intégration, l'exploitation et le partage d'informations. Face à la mutation du web, les moteurs de recherche doivent s'adapter sans cesse pour rester concurrentiels. Pour cela, le WEB SÉMANTIQUE [Berners-Lee 1998, Berners-Lee *et al.* 2001, Antoniou & Harmelen 2008, Hendler 2008] se fonde sur l'héritage des travaux

## 2.2. LE WEB SÉMANTIQUE

de recherche en représentation des connaissances [Baget 2001] et utilise le formalisme des ontologies [Gruber 1993] pour définir et lier formellement les données (voir Figure 5.8).

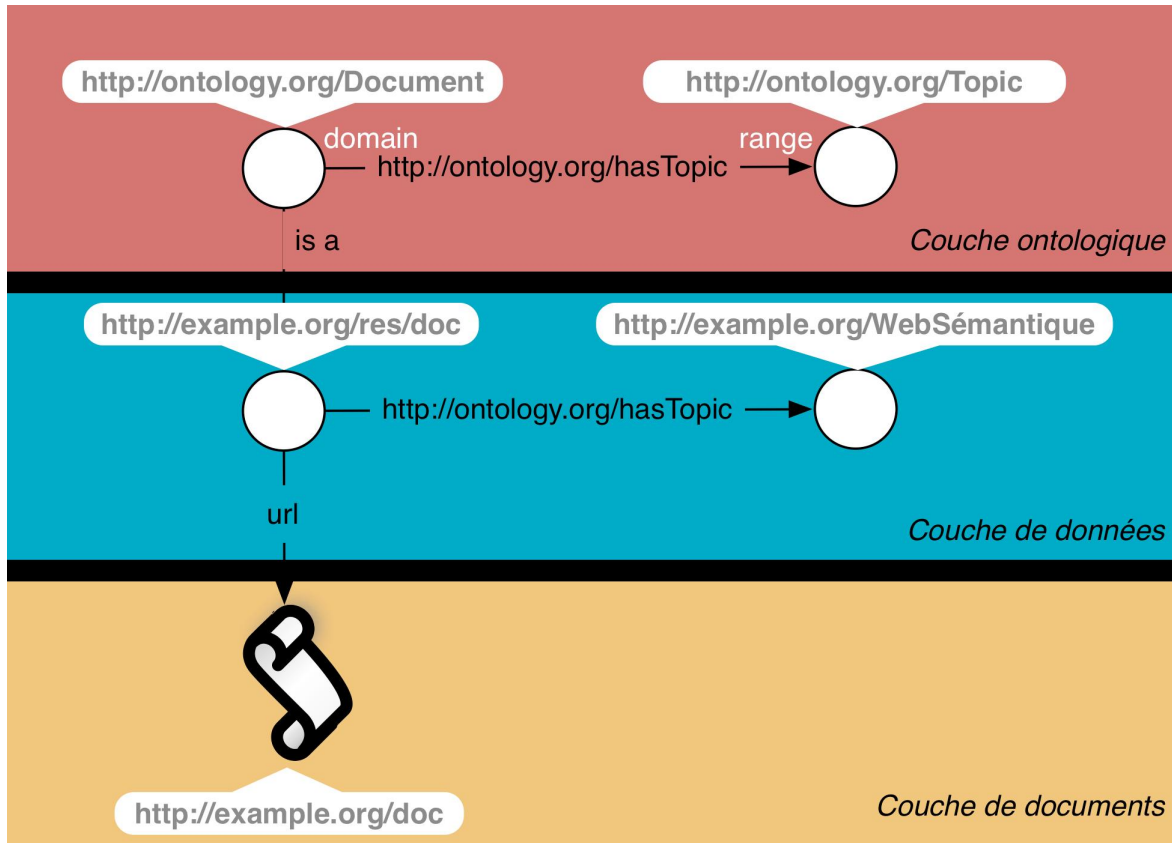


FIGURE 2.1 – Les Trois couches du Web Sémantique [Hitzler *et al.* 2009]

Là encore, les moteurs de recherche s'adaptent et apprennent peu à peu à exploiter le potentiel de ces nouveautés. Google par exemple, s'est enrichi depuis quelques années par le Knowledge Graph<sup>2</sup> (voir Figure 2.2) et a déployé une mise à jour majeure de son algorithme de recherche, nommée Hummingbird<sup>3</sup> (voir Figure 2.3) visant à retourner directement, en plus d'une liste de documents pertinents, la réponse à la requête posée par l'utilisateur<sup>4</sup>.

Le principal objectif du WEB SÉMANTIQUE [Berners-Lee 1998, Berners-Lee *et al.* 2001, Antoniou & Harmelen 2008, Hendler 2008] est de développer la sémantique des données en utilisant des annotations autres que celles utilisées précédemment qui sont simplement textuelles et "écrites à la main", offrant ainsi une plate-forme grâce à laquelle les données peuvent être partagées et réutilisées par les applications, les entreprises et les communautés.

Tim Berners-Lee a proposé un schéma appelé "Semantic Web Layer Cake" (Figure 2.4<sup>5</sup>)

2. <http://www.google.com/insidesearch/features/search/knowledge.html>

3. <http://searchengineland.com/google-hummingbird-172816>

4. [http://www.lemonde.fr/technologies/article/2013/09/27/pour-ses-15-ans-google-presente-s-discretionary-{}-{}-on-nouvel-algorithme\\_3486330\\_651865.html](http://www.lemonde.fr/technologies/article/2013/09/27/pour-ses-15-ans-google-presente-s-discretionary-{}-{}-on-nouvel-algorithme_3486330_651865.html), septembre 2013

5. <http://www.w3.org/2001/sw/>

## Google Search Timeline

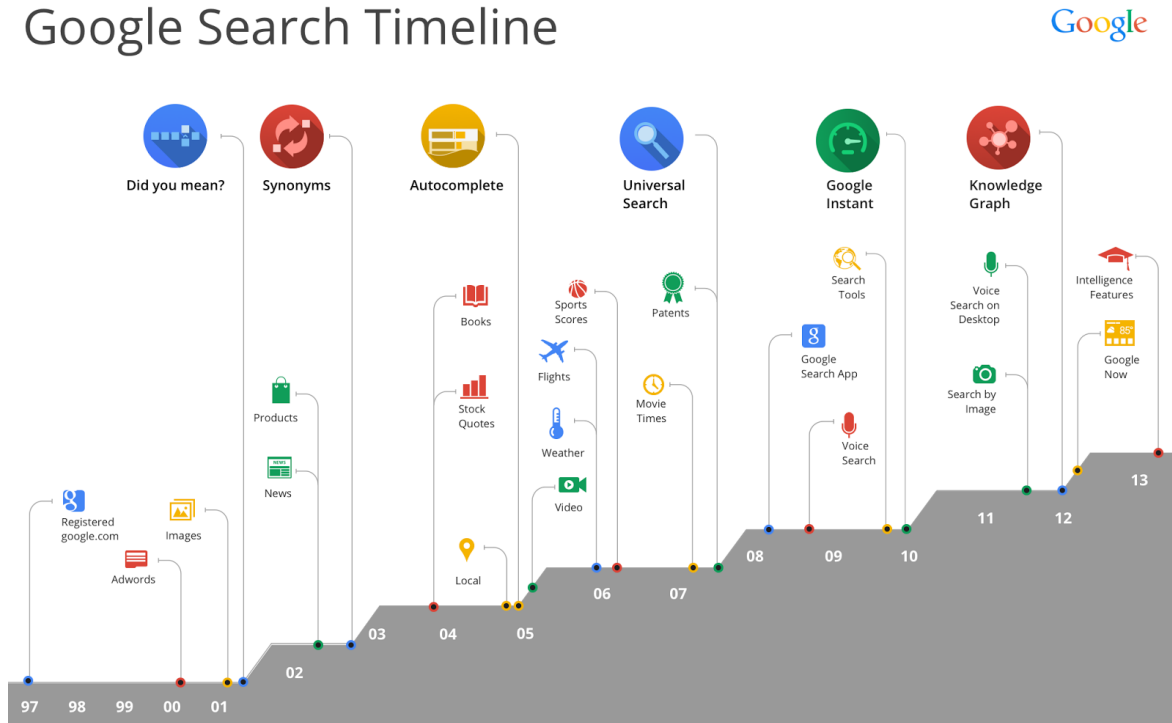


FIGURE 2.2 – Evolution de la recherche Google [Lardinois 2013]

qui présente les technologies et les standards [Hitzler *et al.* 2009] nécessaires à la vision du WEB SÉMANTIQUE [Berners-Lee 1998, Berners-Lee *et al.* 2001, Antoniou & Harmelen 2008, Hendler 2008] ainsi que leurs interdépendances où chaque couche de la pile dépend des couches qui sont au-dessous. Si les couches basses correspondent à des standards et des technologies existantes, les couches supérieures évoquent les capacités susceptibles d’être développées dans l’avenir, telles que les vérifications automatiques de faits (en. proof) et la gestion de la confiance par des applications Web d’une manière qui ressemble à la notion de confiance dans les relations humaines.

Le Web de données est basé sur deux technologies du Web : le protocole HTTP, les URI [Berners-Lee *et al.* 1998].

- **Hyper Text Transfer Protocol, ou http**, est le protocole le plus largement utilisé pour le transfert de documents sur Internet et la consultation de sites Web.
- Une Uniform Resource Identifier, ou URI, est une chaîne de caractères permettant d’identifier une ressource sur un réseau de manière permanente, même si celle-ci venait à être déplacée ou supprimée. On retrouve deux ensembles quasi-disjoints au sein des **URI** : les URL (Uniform Resource Locator) et les URN (Uniform Resource Name).
  - Une **URL** est la forme la plus connue des URI, elle indique comment obtenir plus d’informations au sujet de la ressource [www.w3c.org](http://www.w3c.org) en utilisant le protocole http <http://www.w3c.org>.
  - Une **URN** est une forme moins connue d’URI, elle identifie une ressource au sein d’un espace de noms.

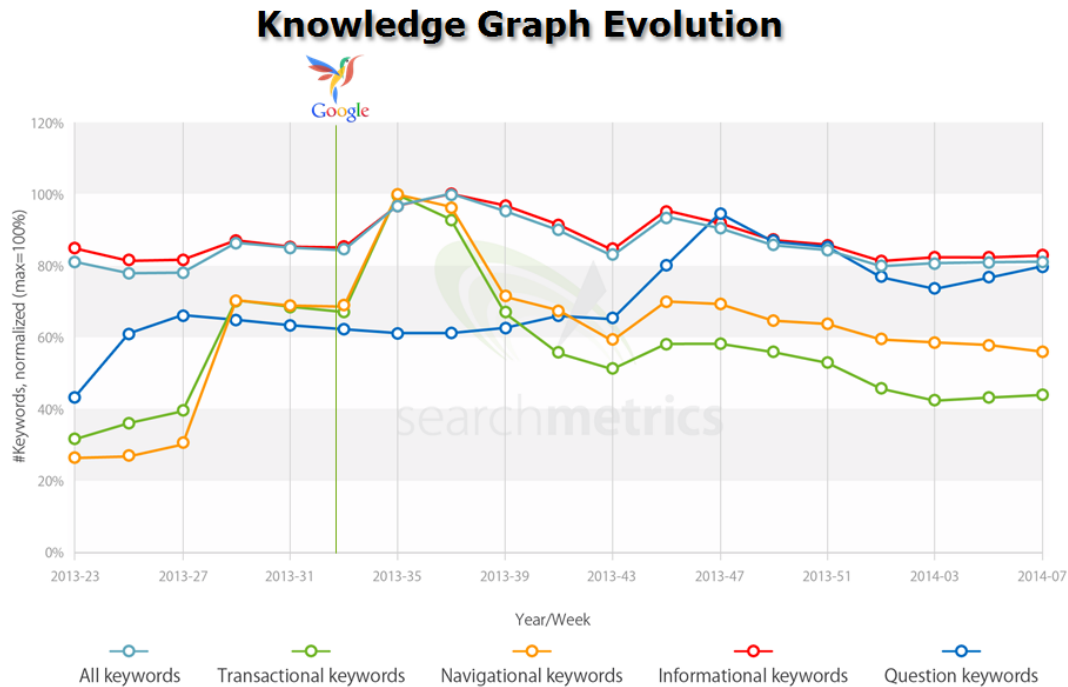


FIGURE 2.3 – Evolution du hummingbird knowledge graph [Lardinois 2013]

### 2.2.1 Données, Information et Connaissance

Cependant, il convient de marquer les différences entre les notions de **donnée**, d'**information** et de **connaissance** dont les frontières ne sont cependant pas claires et peuvent varier en fonction du contexte. Une donnée est le résultat d'une mesure dénué de sens, elle est collectée par un capteur, une personne, ou extraite d'une base de données. Sa finalité est d'être transmise, stockée et surtout traitée. Un ensemble de données peuvent être structurées pour produire de l'information qui porte un sens, soit pour le système, soit pour l'humain. L'information devient connaissance lorsqu'elle est reçue dans un contexte donné et interprétée par un humain sur la base des connaissances qu'il possède déjà suite à son expérience personnelle. Cette représentation de l'information fournit donc des données semi-structurées, des données structurées ou non structurées.

Dans le domaine de l'Intelligence Artificielle (IA), l'information devient connaissance lorsqu'elle sert de fondement à une inférence menant à de nouvelles connaissances. [Bahoul 2006, Charlet 2000] définit une inférence comme une façon générique de désigner l'ensemble des mécanismes par lesquels des entrées perceptives ou non sont combinées à des connaissances préalables afin d'obtenir des comportements élaborés. On peut distinguer deux types de connaissances : les connaissances assertionnelles des connaissances ontologiques. Les connaissances assertionnelles (ou factuelles) décrivent des faits précis et établis, alors que les connaissances ontologiques de description [Grosz *et al.* 2003, Decker *et al.* 2000, Baader *et al.* 2003](ou terminologiques, ou intentionnelles) décrivent le monde à un plus haut niveau

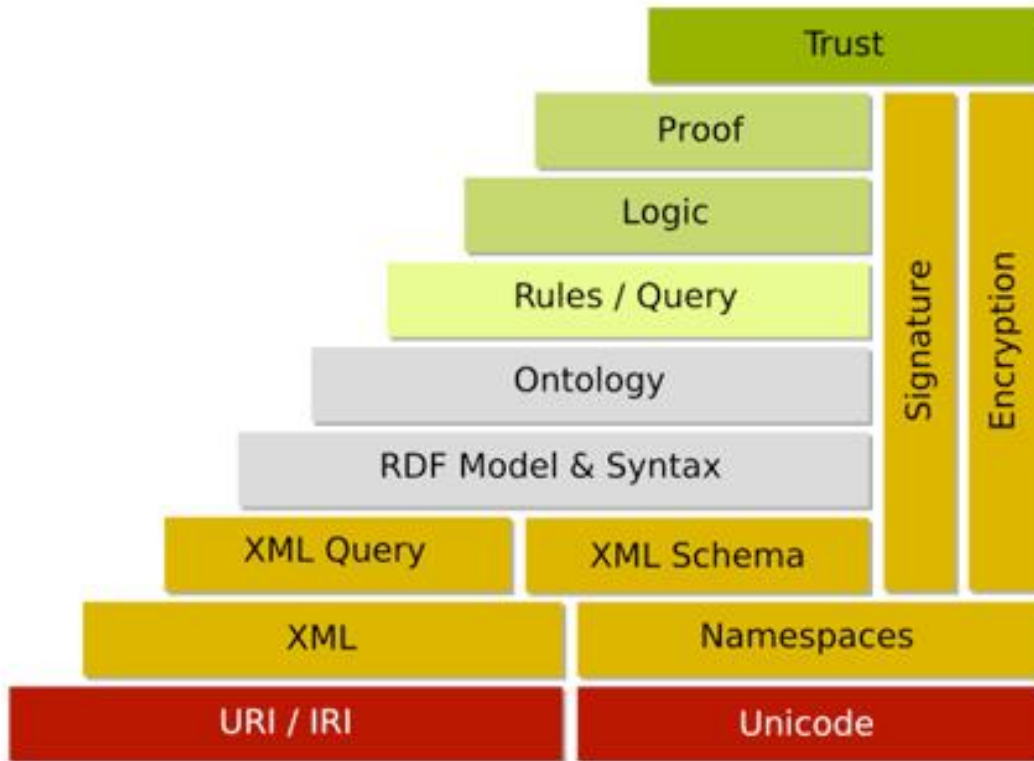


FIGURE 2.4 – Semantic Web Layer Cake, repris de Tim Berners-Lee [Antoniou & Harmelen 2008]

d'abstraction.

## 2.2.2 Représentation des connaissances avec RDF(S) et OWL

Nous présenterons ici uniquement des formalismes proposés ou standardisés via les activités du W3C, formalismes que nous utilisons par ailleurs au sein des différents travaux présentés dans cette thèse.

### 2.2.2.1 Représentation des ressources : RDF(S)

**Resource Description Framework** (RDF [Carroll & Klyne 2004, Hayes 2004, Lassila & Swick 1999, Manola & Miller 2004, Beckett 2009b]) est un modèle de données générique basé sur la théorie des graphes [Corby *et al.* 2000, Chein & Mugnier 2009].

**Définition 2.2.1** (Un graphe RDF ). *est un graphe dirigé étiqueté (directed labeled graph) permettant de représenter de l'information avec des triplets de la forme  $\langle \text{ sujet, prédicat, objet} \rangle \in (\mathcal{U} \cup \mathcal{B}) \times (\mathcal{U}) \times (\mathcal{U} \cup \mathcal{B} \cup \mathcal{L})$  où  $\mathcal{U}$  dénote les URI arbitraires par exemple  $ab:Oran$ ,  $\mathcal{B}$  les nœuds blancs préfixé par  $_$ : par exemple  $_:w$ ,  $\mathcal{L}$  les littéraux par exemple "Alger" et l'union de ces trois ensembles constitue la terminologie  $\mathcal{T}$ . Le vocabulaire  $\mathcal{N}$  souligne l'ensemble des noms, i.e.,  $\mathcal{N} = \mathcal{U} \cup \mathcal{L}$ .*

- Le sujet est la ressource au sujet de laquelle une information va être définie. Il est représenté par une URI.
- Le prédicat est le type d'information qui va être définie à propos du sujet.
- L'objet est la valeur du prédicat du sujet.

Assertion	Triplet	Notation relationnelle
Classe	s rdf:type o	o(s)
Propriété	s p o	p(s, o)

TABLE 2.1 – Déclarations RDF

Pour des raisons de lisibilité, nous adoptons dans la suite de ce travail le mécanisme de préfixe permettant d'alléger la notation de ces URI. Inspirée des espaces de nommage XML [Bray *et al.* 2006, Maler *et al.* 2004], cette méthode permet de substituer le début d'un URI<sup>6</sup> par un identifiant plus court. Par exemple, en définissant un préfixe qui a pour nom Oran et pour valeur <http://dbpedia.org/resource/> le nom qualifié (*Qualifiedname* ou *Qname*) db:Oran est une version courte de l'URI <http://dbpedia.org/resource/Oran>.

Comme montré en Figure 2.5, qui décrit la ville d'Oran ("db:Oran" ou bien son identifiant geonames "geonamesid:2485926") en détaillant sa géo-localisation ( sa longitude "wgs84\_pos:long" et latitude "wgs84\_pos:lat"), sa capitale ("dbprop:capital") et sa population ("gn:population") . Il existe également des quadruplets (quads), une extension des triplets rajoutant le contexte (la provenance de l'information) comme quatrième information au triplet.

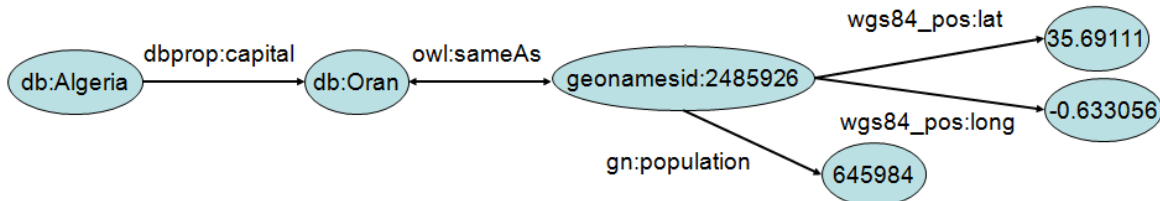


FIGURE 2.5 – Représentation graphique d'un exemple de triplets RDF (Graphe RDF)

**Définition 2.2.2** (L'Interprétation RDF). Soit  $V \subseteq (\mathcal{U} \cup \mathcal{L})$  un vocabulaire. Une interprétation RDF de  $V$  est un tuple  $I = \langle I_R, I_P, I_{EXT}, \iota \rangle$  tel que :

- $I_R$  est l'ensemble de ressources contenus dans  $V \cap \mathcal{L}$  ;
- $I_P \subseteq I_R$  est l'ensemble de propriétés ;
- $I_{EXT} : I_P \rightarrow 2^{I_R \times I_R}$  associe pour chaque propriété une paire de ressources appelé l'extension de la propriété ;
- l'interprétation est une fonction  $\iota : V \rightarrow I_R$  qui associe pour chaque nom de  $V$  une ressource  $I_R$ , tel que Si  $v \in \mathcal{L}$ , then  $\iota(v) = v$ .

**Définition 2.2.3** (Un modèle RDF). Soit  $V \subseteq \mathcal{V}$  un vocabulaire, et  $G$  un graphe RDF tel que  $\text{voc}(G) \subseteq V$ . Une interprétation  $I = \langle I_R, I_P, I_{EXT}, \iota \rangle$  of  $V$  est un modèle RDF de  $G$  Ssi

6. qui est le plus souvent dans un contexte local peu pertinent et commun à plusieurs URI

il existe un matching  $\iota' : \mathcal{T}(G) \rightarrow I_R$  étendant  $\iota$ , i.e.,  $t \in V \cap \mathcal{T}(G) \Rightarrow \iota'(t) = \iota(t)$ , tel que pour chaque triplet  $\langle s, p, o \rangle \in G$ ,  $\iota'(p) \in I_P$  et  $\langle \iota'(s), \iota'(o) \rangle \in I_{EXT}(\iota'(p))$ .

Une caractéristique importante de RDFS [Munoz et al. 2009] est la modélisation de triplets implicites. Le W3C nomme RDF entailment [Baget 2005, Munoz et al. 2007] le mécanisme par lequel les triplets implicites sont dérivés (ou engendrés) à partir des triplets explicites d'un graphe et de règles d'entailment.

**Définition 2.2.4** (RDF entailment). *Un graphe  $G$  RDF-entails un graphe  $P$  (noté  $G \models_{RDF} P$ ) Ssi  $\forall \langle s, p, o \rangle \in P$ , alors  $\langle s, p, o \rangle \in G$  i.e., il existe une séquence d'entailments immédiats menant de  $G$  à  $P$  (o à chaque pas de la séquence, les triplets précédemment engendrés sont aussi pris en compte). Nous notons  $\models_{RDF}^i$  un entailment immédiat, i.e., la dérivation de triplets par l'application d'une règle d'entailment.*

**RDF Schema** (RDFS [Munoz et al. 2009] [Horst 2005, Munoz et al. 2009, Brickley & Guha 1999, Brickley & Guha 2004]) permet de déclarer des contraintes sémantiques [Baget & Mugnier 2002] entre les classes et les propriétés dont le but d'enrichir les descriptions de ressources dans les graphes. Ceux-ci laissent notamment la possibilité de créer des règles d'inférence, via un mécanisme de classes et de propriétés 2.3, permettant ainsi de générer une grande quantité d'information et de triplets dérivés des triplets existants. Cela permet d'augmenter les capacités de raisonnement que peuvent avoir les machines et les programmes traitant les données du Web sémantique. RDFS [Munoz et al. 2009] permet notamment de définir des classes et sous-classes de ressources [Munoz et al. 2009]. Ainsi, si l'on définit que B est une sous-classe de A, alors toute ressource X ayant pour triplet (X rdf:type B), i.e., étant de type B, est automatiquement également de type A. De même, il est possible de définir le type de donnée du sujet et de l'objet d'un prédicat. La Figure 3.1 montre les contraintes autorisées et comment les déclarer; le domaine et le co-domaine désignent respectivement le premier et le second attribut de toute propriété. Enfin, signalons que rien n'oblige l'ensemble des triplets concernant une ressource donnée à être stocké au sein du même graphe ou document, il est tout à fait possible de définir ces informations dans plusieurs documents, l'identification des sources permettant par la suite de tracer l'origine de chaque assertion.

Contrainte	Triplet	Interprétation
Sous-classe	s rdfs :subClassOf o	$s \subseteq o$
Sous-propriété	s rdfs :subPropertyOf o	$s \subseteq o$
Typage de domaine	s rdfs :domain o	$\prod_{domain}(s) \subseteq o$
Typage de range	s rdfs :range o	$\prod_{range}(s) \subseteq o$

TABLE 2.2 – Déclarations RDFS

Le standard RDF exprime des assertions de classe (relation unaire) et de propriété (relation binaire) via les espaces de noms normalisés rdf : et rdfs : e.g., la propriété rdf:type permet d'indiquer à quelles classes des ressources appartient, comme montré sur les deux tables 3.1, 2.2.



rdfs :domain[dom]	rdfs :Container[cont]	rdf :Bag[bag]
rdfs :range[range]	rdfs :isDefinedBy[isDefined]	rdf :Seq[seq]
rdfs :Class[class]	rdfs :Literal[literal]	rdf :List[list]
rdf :value[value]	rdfs :subClassOf[sc]	rdf :Alt[alt]
rdfs :label[label]	rdfs :subPropertyOf[sp]	rdf :1[1]
rdf :nil[nil]	rdfs :comment[comment]	...
rdf :type[type]	rdf :predicate[pred]	rdf :i[2]
rdf :object[obj]	rdf :Statement[stat]	rdf :first[first]
rdf :subject[subj]	rdfs :seeAlso[seeAlso]	rdf :rest[rest]
rdf :Property[prop]	rdfs :Datatype[datatype]	rdfs :member[member]
rdfs :Resource[res]	rdf :XMLLiteral[xmlLit]	
	rdfs :ContainerMembershipProperty[contMP]	

TABLE 2.3 – Le vocabulaire RDF Schema.

**Définition 2.2.5** (L'interprétation RDFS). *Une Interprétation RDFS du vocabulaire  $V$  est un tuple  $\langle I_R, I_P, Class, I_{EXT}, I_{CEXT}, Lit, \iota \rangle$  tel que :*

- $\langle I_R, I_P, I_{EXT}, \iota \rangle$  est une interprétation RDF ;
- $Class \subseteq I_R$  est un sous-ensemble distinct de  $I_R$  qui identifie une ressource dénotant une classe de ressources ;
- $I_{CEXT} : Class \rightarrow 2^{I_R}$  est un matching assignant un ensemble de ressources pour toute ressource dénotant une classe ;
- $Lit \subseteq I_R$  est l'ensemble de valeurs littéraires,  $Lit$  contient les littéraires textuelles de  $\mathcal{L} \cap V$ .

**Définition 2.2.6** (Le modèle RDFS). *Soit  $G$  un graphe RDFS, et  $I = \langle I_R, I_P, Class, I_{EXT}, I_{CEXT}, Lit, \iota \rangle$  est une interprétation RDFS du vocabulaire  $V \subseteq rdfsV \cup \mathcal{V}$  tel que Si  $\mathcal{V}(G) \subseteq V$ . alors  $I$  est un modèle RDFS de  $G$  SSi  $I$  satisfait les conditions suivantes :*

1. *Sémantique simple :*
  - (a) *Il existe une extension  $\iota'$  de  $\iota$  vers  $\mathcal{B}(G)$  tel que pour chaque triplet  $\langle s, p, o \rangle$  of  $G$ ,  $\iota'(p) \in I_P$  et  $\langle \iota'(s), \iota'(o) \rangle \in I_{EXT}(\iota'(p))$ .*
2. *Sémantique RDF :*
  - (a)  $x \in I_P \Leftrightarrow \langle x, \iota'(prop) \rangle \in I_{EXT}(\iota'(type))$ .
  - (b) *Si  $\ell \in term(G)$  est un type XML littéral avec la forme lexical  $w$ , alors  $\iota'(\ell)$  un type XML de valeurs littérales  $w$ ,  $\iota'(\ell) \in Lit$ , et  $\langle \iota'(\ell), \iota'(xmlLit) \rangle \in I_{EXT}(\iota'(type))$ .*
3. *Classes RDFS :*
  - (a)  $x \in I_R, x \in I_{CEXT}(\iota'(res))$ .
  - (b)  $x \in Class, x \in I_{CEXT}(\iota'(class))$ .
  - (c)  $x \in Lit, x \in I_{CEXT}(\iota'(literal))$ .
4. *Sous-propriétés RDFS :*
  - (a)  $I_{EXT}(\iota'(sp))$  est transitive et reflexive sur  $I_P$ .

(b) if  $\langle x, y \rangle \in I_{EXT}(\iota'(sp))$  alors  $x, y \in I_P$  et  $I_{EXT}(x) \subseteq I_{EXT}(y)$ .

5. Sous-classe RDFS :

(a)  $I_{EXT}(\iota'(sc))$  est transitive et réflexive sur la Class.

(b)  $\langle x, y \rangle \in I_{EXT}(\iota'(sc))$ , alors  $x, y \in Class$  et  $I_{CEXT}(x) \subseteq I_{CEXT}(y)$ .

6. Types RDFS :

(a)  $x \in I_{CEXT}(y)$ ,  $(x, y) \in I_{EXT}(\iota'(type))$ .

(b) Si  $\langle x, y \rangle \in I_{EXT}(\iota'(dom))$  et  $\langle u, v \rangle \in I_{EXT}(x)$  alors  $u \in I_{CEXT}(y)$ .

(c) Si  $\langle x, y \rangle \in I_{EXT}(\iota'(range))$  et  $\langle u, v \rangle \in I_{EXT}(x)$  alors  $v \in I_{CEXT}(y)$ .

7. RDFS Additionnelles :

(a) Si  $x \in Class$  alors  $\langle x, \iota'(res) \rangle \in I_{EXT}(\iota'(sc))$ .

(b) Si  $x \in I_{CEXT}(\iota'(datatype))$  alors  $\langle x, \iota'(literal) \rangle \in I_{EXT}(\iota'(sc))$ .

(c) Si  $x \in I_{CEXT}(\iota'(contMP))$  alors  $\langle x, \iota'(member) \rangle \in I_{EXT}(\iota'(sp))$ .

**Définition 2.2.7** (RDFS entailment [Baget & Mugnier 2002]). Soit  $G$  et  $P$  deux graphes RDFS, alors  $G$  RDFS-entails  $P$  (noté  $G \models_{RDFS} P$ ) Ssi tout modèle RDFS de  $G$  est aussi un modèle RDFS de  $P$ .

$\frac{\langle p, sp, q \rangle \quad \langle q, sp, r \rangle}{\langle p, sp, r \rangle}$	$\frac{\langle A, sc, B \rangle \quad \langle B, sc, C \rangle}{\langle A, sc, C \rangle}$	
$\frac{\langle p, sp, q \rangle \quad \langle x, p, y \rangle}{\langle x, q, y \rangle}$	$\frac{\langle A, sc, B \rangle \quad \langle x, type, A \rangle}{\langle x, type, B \rangle}$	
$\frac{\langle p, sp, q \rangle \quad \langle q, dom, A \rangle}{\langle p, dom, A \rangle}$	$\frac{\langle p, dom, A \rangle \quad \langle A, sc, B \rangle}{\langle p, dom, B \rangle}$	$\frac{\langle p, dom, A \rangle \quad \langle x, p, y \rangle}{\langle x, type, A \rangle}$
$\frac{\langle p, sp, q \rangle \quad \langle q, range, A \rangle}{\langle p, range, A \rangle}$	$\frac{\langle p, range, A \rangle \quad \langle A, sc, B \rangle}{\langle p, range, B \rangle}$	$\frac{\langle p, range, A \rangle \quad \langle x, p, y \rangle}{\langle y, type, A \rangle}$

TABLE 2.4 – Règles d'inférence en RDFS

### 2.2.2.2 Sérialisation de triplets RDF

Après avoir décrit le modèle conceptuel à la section précédente, il est utile de s'intéresser à la manière de sérialiser les triplets RDF, le processus d'encodage d'une information (comme un modèle ou un graphe RDF) propose plusieurs syntaxes de sérialisation des graphes RDF : RDF/XML, N-Triple, Turtle [Beckett 2006], et Notation3 (N3). RDF/XML est une représentation d'un graphe RDF sous forme XML dont la syntaxe est difficilement lisible par des humains, mais parsable par une machine. Turtle étant le plus utilisé, c'est lui qui va être décrit ici. Son écriture est beaucoup plus directe et plus proche du modèle conceptuel de triplets sachant que le N-Triple est un sous-ensemble de Turtle, qui est lui-même un sous-ensemble de Notation3. Voir Figure 5.11.

```

@PREFIX db: <http://dbpedia.org/resource/>
@PREFIX dbprop: <http://dbpedia.org/property/>
@PREFIX geonamesid: <http://sws.geonames.org/>
@PREFIX gn: <http://www.geonames.org/ontology#>
@PREFIX owl: <http://www.w3.org/2002/07/owl#>
@PREFIX wgs84_pos: <http://www.w3.org/2003/01/geo/wgs84_pos#>

db:Algeria dbprop:capital db:Oran;
db:Oran owl:sameAs geonamesid:2485926/;
geonamesid:2485926/ gn:population 645984;
geonamesid:2485926/ wgs84_pos:lat 35.69111;
geonamesid:2485926/ wgs84_pos:long -0.633056;

```

Listing 2.1 – Exemple de triplets RDF en format turtle

Au dessus de la couche "vocabulaire RDF" qu'est RDFS [Munoz *et al.* 2009], il existe des ontologies [Gruber 1993, Guarino *et al.* 1994] pour le Web sémantique, comme expliqué précédemment, offrant encore plus de possibilités d'inférence de données. Le langage d'ontologies [Gruber 1993, Guarino *et al.* 1994] le plus connu est OWL et sera détaillé à la section suivante.

### 2.2.3 Ontologie, sens de données et raisonnement

Les ontologies [Gruber 1993, Guarino *et al.* 1994], servent à la fois de vocabulaire commun pour échanger des données structurées entre plusieurs acteurs de manière précise et de support au raisonnement automatique et à l'inférence de faits. OWL (Ontologie Web Language) [Smith *et al.* 2004, Dean & Schreiber 2004, McGuinness & Harmelen 2004, Patel-Schneider *et al.* 2004] est un langage d'une expressivité plus élevée et graduée, passé au statut de recommandation du W3C en 2004, reprend ainsi les notions de classes et de propriétés définies en RDFS [Munoz *et al.* 2009] en les précisant respectivement par owl :Class (sous-classe de RDFS RDFS [Munoz *et al.* 2009] :Class) et owl :datatypeProperty et owl :objectProperty (sous-classe de rdf :Property) distinguant ainsi les attributs (types primitifs) des relations (liens vers d'autres classes). Surtout, OWL ajoute de nouveaux constructeurs et axiomes permettant d'accroître l'expressivité des ontologies avec une sémantique plus poussée que celle de . Plusieurs versions d'OWL ont différentes capacités expressives. Plus une version d'OWL est expressive, moins il est possible d'automatiser le raisonnement avec des données exprimées.

- OWL-Lite qui étend RDFS [Munoz *et al.* 2009] et ajoute de nouveaux constructeurs comme la symétrie des propriétés et des contraintes de cardinalité (uniquement 0 ou 1);
- OWL-DL dont le nom est hérité des logiques de description [Baader *et al.* 2003, Grosz *et al.* 2003] et qui ajoute des constructeurs supplémentaires comme les combinaisons booléennes de classes (union ou intersection), des axiomes de classes (disjonction) et étend les contraintes de cardinalité d'OWL-Lite;

- OWL-Full qui n'ajoute pas de constructeur par rapport à OWL-DL mais qui les interprète différemment offrant ainsi une expressivité plus forte (toute classe est vue à la fois comme une classe, un individu et un ensemble d'individus) mais sans garantie de calculabilité, OWL-Full n'étant pas décidable.

Sachant qu'OWL-Lite est Exp-complet, OWL-DL Nexp-complet et OWL-Full est indécidable. Les différents éléments définis au sein des ontologies sont :

- les concepts, ou classes, définissant des ensembles d'objets, abstraits ou concrets, que l'on souhaite modéliser pour un domaine donné. Les termes associés au concept, la notion (i.e. l'intention du concept) et l'objet (i.e. son extension).
- des propriétés attribuées à ces concepts parmi lesquelles on distingue généralement (1) les relations qui peuvent exister entre concepts ou instances de ces concepts et (2) les attributs primitifs qu'il est possible d'associer aux différents concepts ou à leurs instances (chaîne de caractère, entier ...). Un exemple particulier de relation fréquemment utilisée dans les ontologies est la relation de subsomption qui permet d'établir des hiérarchies de concepts ;
- des axiomes, qui permettent de modéliser des assertions logiques de description et qui sont utilisés dans la définition de concepts ou de propriétés afin d'affiner celles-ci. Associés à des raisonneurs, ils permettent d'établir de nouveaux faits à partir des connaissances de base ou de vérifier la consistance d'un ensemble d'assertions. Ces axiomes peuvent être pris en compte dans un processus de raisonnement avec des systèmes comme Pellet<sup>7</sup> [Sirin *et al.* 2007] ou Racer<sup>8</sup> [Haarslev & Möller 2001]. Ceux-ci peuvent être utilisés par exemple (1) pour la classification automatique d'instances en fonction de leurs propriétés et des axiomes définis dans l'ontologie ou (2) pour la création de nouvelles relations entre instances en fonction de l'état initial d'une base de connaissance [Charlet 2000, Euzenat 2002, Bahloul 2006].

**Définition 2.2.8** (ontologie). *Formellement une structure d'ontologies [Gruber 1993] est un quintuplet :  $O := \{C, R, H^C, rel, A^O\}$  avec :*

- $C$  et  $R$  des ensembles disjoints de concepts et de relations,
- $H^C$  une hiérarchie (taxonomie) de concepts :  $H^C \subseteq C \times C$ ,  $H^C(C_1, C_2)$  signifie que  $C_1$  est un sous-concept de  $C_2$  (relation orientée)
- $rel$  : relation,  $R \rightarrow \mathcal{P}(C \times C)$  définit des relations sémantiques non taxonomiques avec deux fonctions associées
  - $dom : R \rightarrow \mathcal{P}(C)$  avec  $dom(R) := \prod_1(rel(R))$
  - $range : R \rightarrow \mathcal{P}(C)$  avec  $range(R) := \prod_2(rel(R))$ , co-domaine
- $(C_1, C_2) \in rel(R)$  s'écrit aussi  $R(C_1, C_2)$
- $A^O$  un ensemble d'axiomes, dans un langage logique adapté en logique de description ou logique du premier ordre.

Les logiques de descriptions (LD) [Baader *et al.* 2003, Grosz *et al.* 2003], issues des travaux de Brachman [Brachman & Levesque 1985], adoptent un point de vue plus logique et

---

7. <http://pellet.owldl.com/>

8. <http://www.racer-systems.com/>

se focalisent sur l'axiomatisation de l'ontologie . Les LD sont des fragments (le plus souvent) décidables de la logique du premier ordre (LPO), et forment donc une famille de langages offrant différents niveaux de compromis entre expressivité et complexité algorithmique. L'idée générale consiste à décrire des concepts de façon à pouvoir ensuite classifier automatiquement des instances dans ces concepts. La palette d'opérateurs autorisés pour la définition de ces concepts dépend du langage de *LD* choisi. Une base de connaissances en *LD* est composée d'une *Tbox* (partie ontologique) et d'une *Abox* (partie assertionnelle). Une *TBOX* permet de définir (équivalence de concepts) et de spécifier (inclusion de concepts) des concepts. La *Tbox* peut être vue comme une ontologie; elle définit un modèle pour la *Abox* qui contient les données assertionnelles liées à cette ontologie. Une *Abox* est un ensemble de connaissances établissant que certains individus satisfont certains concepts définis dans la *Tbox*. Une *Abox* permet d'asserter qu'un individu est une instance de concept, et que deux individus sont membres d'un rôle.

L'univers décrit en *LD* est constitué d'individus (ou instances), appartenant à des concepts (ou classes), et reliés entre-eux par des rôles (ou propriétés). Un individu représente un élément et correspond à un terme en *LPO*; Algeria, Oran sont des exemples d'individus. Un concept représente un ensemble d'éléments et correspond en *LPO* à un prédicat unaire; Country, City sont des exemples de concepts et Country(Algeria) exprime le fait que l'élément Algeria appartient à l'ensemble des pays. Un rôle représente une relation binaire entre éléments et correspond en *LPO* à un prédicat binaire; géo-localisation (sa longitude "wgs84\_pos :long" et latitude " wgs84\_pos :lat") sont des exemples de rôles.

**Définition 2.2.9** (Les logiques de description). *Des concepts complexes peuvent être définis via deux types d'opérateurs :*

- *Des concepts complexes peuvent être définis via deux types d'opérateurs :*
- *les constructeurs ensemblistes :*
  - *le concept universel (top)  $T$  correspond à l'ensemble de l'univers  $\Delta$ ,*
  - *le concept absurde (bottom)  $\perp$  correspond à l'ensemble vide  $\emptyset$ ,*
  - *le complément  $\neg C$  correspond à l'ensemble  $\{x | \neg C(x)\}$ ,*
  - *l'union  $C \cup D$  correspond à l'ensemble  $\{x | C(x)\} \cup \{x | D(x)\}$ ,*
  - *l'intersection  $C \cap D$  correspond à l'ensemble  $\{x | C(x)\} \cap \{x | D(x)\}$ ,*
  - *l'extension  $\{a_1, a_2, \dots, a_n\}$  correspond à l'ensemble  $\{a_1, a_2, \dots, a_n\}$ ,*
- *les restrictions :*
  - *la restriction existentielle  $\exists rC$  correspond à l'ensemble  $\{x | \exists y, r(x, y) \wedge C(y)\}$ ,*
  - *la restriction universelle  $\forall rC$  correspond à l'ensemble  $\{x | \forall y, r(x, y) \rightarrow C(y)\}$ ,*
  - *la restriction de cardinalité  $= nrC$  correspond à l'ensemble  $\{x | \|\{y | r(x, y) \wedge C(y)\}\| = n\}$ ,*
  - *la restriction de cardinalité minimale  $\geq nrC$  correspond à l'ensemble  $\{x | \|\{y | r(x, y) \wedge C(y)\}\| \geq n\}$ ,*
  - *la restriction de cardinalité maximale  $\leq nrC$  correspond à l'ensemble  $\{x | \|\{y | r(x, y) \wedge C(y)\}\| \leq n\}$ .*
- *Des rôles complexes peuvent également être définis :*
  - *le rôle inverse  $r^-$  correspond à la relation  $\{(x, y) | r(y, x)\}$  ,*

- le rôle composé  $r \circ s$  correspond à la relation  $\{(x, y) | \exists z, r(x, z) \wedge s(z, y)\}$ ,
- le complément  $\neg r$  correspond à la relation  $\{(x, y) | \neg r(x, y)\}$ ,
- Enfin, les axiomes d'inclusion permettent de déclarer des relations de subsumption entre concepts et propriétés :
  - l'inclusion de concept  $C \sqsubseteq D$  exprime que le concept  $D$  subsume le concept  $C$  :  
 $\forall x, C(x) \rightarrow D(x)$
  - l'inclusion de rôle  $r \sqsubseteq s$  exprime que le rôle  $s$  subsume le rôle  $r$  :  $\forall x, y, r(x, y) \rightarrow s(x, y)$

Il existe des applications des ontologies dans des domaines variés bien antérieurs au Web sémantique : la finance, le transport, le tourisme, l'immobilier, la médecine, le commerce électronique, le tourisme. . . Nous pouvons citer par exemple :

- **Cyc 1984** : base de connaissance<sup>9</sup> (ontologie générique).
- Informatique Médicale : SNOMED (Systematized Nomenclature of Medicine) est une ontologie issue d'un projet américain<sup>10</sup> commencé en 1965 avec aujourd'hui plus de 450000 concepts.
- **Bioinformatique** : The Gene Ontology (GO) [Gen] comporte plus de 17000 concepts. Son but est d'aider à annoter des génomes.
- **Enterprise Ontology**<sup>11</sup>, est appliquée à la modélisation d'entreprise.
- **Suo**<sup>12</sup> est destinée au domaine des systèmes d'information et vise à décrire les données manipulées par les applications
- **DOLCE**<sup>13</sup> a été la première réalisation de WonderWeb (projet qui a pris fin en 2004).
- **FOAF**<sup>14</sup> représente le concept d'organisation qui peut avoir différentes propriétés et être également lié à d'autres concepts également interprétables.
- YAGO (Yet Another Great Ontology)<sup>15</sup> est une énorme base de connaissances, développée à l'Institut d'informatique Max Planck de Sarrebruck, décrivant plus de 2 millions d'entités, des personnes, des organisations, des villes et contenant plus de 20 millions de faits sur ces entités. Ses données proviennent de Wikipedia et sont structurées à l'aide de WordNet. De par sa taille et ses multi-thématiques, YAGO est difficile à utiliser directement pour des applications particulières. Afin de tirer parti plus aisément de ce gigantesque travail collectif.

**Wordnet**<sup>16</sup> n'est pas exactement une ontologie, c'est plutôt une taxonomie, un thésaurus, c'est-à-dire un réseau de relations entre mots. Wordnet est composé d'un ensemble de synonymes et d'un ensemble de relations entre ces synonymes qui n'ont pas vraiment de sémantique. Les ontologies ont un pouvoir d'expression relativement fort qui en augmente grandement la complexité de calculs et la décidabilité. Au vu du nombre de données disponibles sur le Web de données, et puisque ce nombre est amené à grandir exponentiellement

---

9. <http://en.wikipedia.org/wiki/Cyc>

10. <http://www.snomed.org/>

11. <http://www.aiai.ed.ac.uk/project/enterprise/>

12. <http://suo.ieee.org/>

13. <http://www.loa-cnr.it/DOLCE.html>

14. <http://XMLns.com/foaf/0.1/Organization>

15. <https://www.mpi-inf.mpg.de/yago-naga/yago/>

16. <http://wordnet.princeton.edu/>

dans les années à venir, l'utilisation de RDFS [Munoz *et al.* 2009] est privilégiée pour le Web sémantique, même si cela réduit les possibilités d'exploitation du potentiel du Web de données. En complément de RDFS [Munoz *et al.* 2009], OWL 2 [Motik *et al.* 2009b, Motik *et al.* 2009a, Bao *et al.* 2009, Beckett 2009a] permet de définir des propriétés supplémentaires, comme déclarer des classes disjointes, des propriétés spécifiques sur les prédicats, comme la transitivité, l'unicité, l'inverse ou la symétrie. Le langage comprend 3 versions, appelées profils, OWL EL, OWL QL et OWL RL. L'usage des éléments du langage restreint à ceux appartenant à OWL EL garantit les meilleures performances de raisonnement en échange d'une expressivité réduite. OWL QL est consacré aux usages dans les ontologies qui représentent une structure de données similaire à celle des bases de données relationnelles, et OWL RL garantit des bonnes performances de raisonnement avec OWL dans le cadre de systèmes de raisonnement à base de règles.

#### 2.2.4 Interrogation du Web Sémantique

Afin d'accéder à la quantité énorme d'information sémantique qui est stockée de manière structurée, il est nécessaire d'avoir un langage de requêtes sémantiques [Corby *et al.* 2004, Mendelzon *et al.* 1997], semblable à SQL et dans la continuité de RDQL [Seaborne 2004], SPARQL [Prud'hommeaux & Seaborne 2008, Franconi & Tessaris 2005] acronyme pour Simple Protocol And RDF Query Language qui est à la fois un langage et un protocole pour interroger des données modélisées en RDF.

Le composant de base d'une requête SPARQL est un motif de graphe basique (Basic Graph Pattern), qui peut être vu comme un ensemble de morceaux de requête élémentaires, appelés motif de triplet (Triple Pattern). Un motif de triplet s'exprime sous la forme d'un triplet RDF, pouvant contenir une ou plusieurs variables  $V$  en tant que  $\langle \text{ sujet, prédicat, objet } \rangle \in (T \cup V) \times (U \cup V) \times (T \cup V)$ . Une variable est caractérisée par le fait qu'elle commence par un point d'interrogation dans le motif de triplet.

Il utilise d'ailleurs une syntaxe proche de SQL [Harris & Shadbolt 2005], et notamment la structure générale SELECT WHERE d'une requête. Le contenu de la clause *WHERE* est un motif de graphe RDF dont chaque occurrence dans la base de triplets interrogée constitue une réponse à la requête. Une requête peut interroger un ou plusieurs documents RDF [Carroll & Klyne 2004, Hayes 2004, Lassila & Swick 1999, Manola & Miller 2004, Beckett 2009b], soit par l'utilisation d'un attribut *FROM* en début de requête, soit par l'intermédiaire d'APIs - Application Programming Interface - qui permettent de considérer simultanément plusieurs sources, soit via l'utilisation d'entrepôts de données RDF [Carroll & Klyne 2004, Hayes 2004, Lassila & Swick 1999, Manola & Miller 2004, Beckett 2009b] associés à des points d'accès (ou endpoints). *SPARQL* dispose des quatre opérateurs suivants :

- SELECT qui comme son nom l'indique va sélectionner différents éléments selon un pattern de requête particulier.
- CONSTRUCT qui permet de transformer un graphe RDF en un autre graphe. On peut ainsi voir cet opérateur comme le XSLT [Clark 1999] du Web Sémantique.

- ASK permet de répondre à une requête, en identifiant si oui ou non le pattern recherché est présent dans le graphe interrogé.
- DESCRIBE qui renvoie sous forme d'un graphe RDF une description de la ressource passée en argument. Cette description est laissée à la discrétion des outils de moteurs SPARQL et peut pas exemple retourner l'ensemble des triplets ayant pour sujet cette ressource.

Dans ce BGP, les triplets s'expriment dans une syntaxe très proche de Turtle, qui autorise en plus l'utilisation de variables; les variables commencent par ? ou \$ directement suivi de l'identifiant de la variable, comme par exemple ?maVariable. Un motif de graphe M est formé de l'un des éléments suivants :

- Un motif de triplet (s p o.) constitué de termes RDF et de variables,
- Une jointure de deux motifs ( $M_1 M_2$ ),
- Une union de deux motifs ( $M_1 \text{ UNION } M_2$ ),
- Un motif optionnel ( $\text{OPTIONAL } M_1$ ),
- Un motif de filtre ( $\text{FILTER } F$ ) où F est soit une expression booléenne fondée sur des opérateurs prédéfinis, soit une négation de motif ( $\text{NOT EXIST } G_1$ ),
- Un motif de graphe nommé ( $\text{GRAPH } g \ G_1$ ) où g est l'URI, le nom qualifié ou une variable faisant référence à un graphe nommé,
- Une sous-requête ( $\text{SELECT ... WHERE } G_1$ ).

Les préfixes se déclarent comme en Turtle, mais en amont de la clause WHERE, avant le mot-clé SELECT, et sans le caractère .

**Définition 2.2.10** (motif de graphe SPARQL). *Un motif de graphe SPARQL est un motif de graphe basique, définie comme suit Ssi  $P, P'$  sont deux motif de graphe basique et  $K$  une contrainte SPARQL, alors  $(P \text{ AND } P')$ ,  $(P \text{ UNION } P')$ ,  $(P \text{ OPT } P')$ , et  $(P \text{ FILTER } K)$  sont des motifs de graphe SPARQL.*

Le traitement d'une requête (c'est-à-dire la recherche des résultats de cette requête) se fait par appariement de graphe (graph pattern matching) [Glimm & Krötzsch 2010]. Pour répondre à une requête, un moteur SPARQL détermine l'ensemble des valeurs que peuvent prendre les variables du pattern de graphe de sorte que le graphe obtenu après instanciation de ces variables soit un sous-graphe partiel du graphe cible; chaque appariement variables-valeurs menant à un sous-graphe partiel du graphe cible est une réponse à la requête. Le contenu de la clause SELECT liste l'ensemble des variables dont la valeur doit être retournée pour chaque réponse.

**Définition 2.2.11** (Map). *Soit  $V_1 \subseteq \mathcal{T}$ , et  $V_2 \subseteq \mathcal{T}$  deux terminologies. Un map de  $V_1$  à  $V_2$  est une fonction  $\sigma : V_1 \rightarrow V_2$  tels que  $\forall x \in (V_1 \cap \mathcal{V}), \sigma(x) = x$ .*

**Définition 2.2.12** (Répondre à une requête). *Soit  $\models_{RDF}$  une relation d'entailment sur les motif de graphes basiques,  $P, P'$  est un motif de graphe SPARQL,  $K$  est une contrainte*



SPARQL, et  $G$  un graphe RDF. L'ensemble  $\mathcal{S}(P, G)$  des réponses de  $P$  dans  $G$  est l'ensemble des maps de  $\mathcal{B}(P)$  vers  $\mathcal{T}(G)$  est définie intuitivement comme suit :

$$\mathcal{S}(P, G) = \{\sigma|_{\mathcal{B}(P)} \mid G \models_{RDF} \sigma(P)\} \quad \text{Si } P \text{ est un motif de graphe basique} \quad (2.1)$$

$$\mathcal{S}((P \text{ AND } P'), G) = \mathcal{S}(P, G) \bowtie \mathcal{S}(P', G) \quad (2.2)$$

$$\mathcal{S}(P \text{ UNION } P', G) = \mathcal{S}(P, G) \cup \mathcal{S}(P', G) \quad (2.3)$$

$$\mathcal{S}(P \text{ OPT } P', G) = (\mathcal{S}(P, G) \bowtie \mathcal{S}(P', G)) \cup (\mathcal{S}(P, G) \setminus \mathcal{S}(P', G)) \quad (2.4)$$

$$\mathcal{S}(P \text{ FILTER } K, G) = \{\sigma \in \mathcal{S}(P, G) \mid \sigma(K) = \top\} \quad (2.5)$$

Dans cet exemple, le premier motif de triplet est (`<http://dbpedia.org/resource/Algeria> owl:sameAs ?x`) introduisant une première variable `?x` pouvant (à cette étape-ci de la requête) prendre la valeur de n'importe quelle ressource du graphe interrogé équivalente à la ressource Algérie dont l'URI des recherché contient la sous-chaine " geo ". Voici le code de cette requête  $Q$  :

[Requête Q]

```
PREFIX owl:<http://www.w3.org/2002/07/owl#>
select ?x
{
<http://dbpedia.org/resource/Algeria> owl:sameAs ?x
filter regex(str(?x), "geo")
}
```

Listing 2.2 – Exemple de requête Q de filtrage textuel SPARQL

X
<code>&lt;http://linkedgeodata.org/triplify/node432424926&gt;</code> <code>&lt;http://gadm.geovocab.org/id/065 &gt;</code> <code>&lt;http://sws.geonames.org/2589581/&gt;</code>

TABLE 2.5 – Résultat de la requête Q (Filtrage textuel)

La requête  $Q'$  permet de lister au plus 6 noms de ville ou village faisant partie de l'Algérie :

[Requête Q']

```
PREFIX dbpedia: <http://dbpedia.org/resource/>
PREFIX dbp-ont: <http://dbpedia.org/ontology/>
select *
{
?location ?locatedIn dbpedia:Algeria .
```

```
?location a ?type filter (?type=dbp-ont:City || ?type=dbp-ont:Village)
} limit 6
```

Listing 2.3 – Exemple de requête Q’ de filtrant les ressources cibles SPARQL

Location	locatedIn	type
:Algiers	dbpedia :ontology/country	dbpedia :ontology/City
:Adrar	dbpedia :ontology/country	dbpedia :ontology/City
:Annaba	dbpedia :ontology/country	dbpedia :ontology/City
:Aokas	dbpedia :ontology/country	dbpedia :ontology/City
:Barika	dbpedia :ontology/country	dbpedia :ontology/City
:Biskra	dbpedia :ontology/country	dbpedia :ontology/City

TABLE 2.6 – Résultat de la requête Q’

### 2.2.5 Données liées- Le Web sémantique d’aujourd’hui

Le web de données est une initiative du W3C visant à encourager la publication de données structurées reliées entre-elles pour augmenter leur utilité par l’utilisation de schémas ou de ressources communes. Il peut également être vu comme une étape intermédiaire vers la mise en place du web sémantique.

L’objectif des DONNÉES LIÉES [Heath & Bizer 2011, Berners-Lee 2009, Bizer *et al.* 2009, Berners-Lee 2006] (Linked Data)<sup>17</sup> est d’utiliser le Web pour faciliter la création de liens entre des données connexes qui n’étaient pas encore liées, ou qui l’étaient déjà via d’autres méthodes. On y retrouve un ensemble de pratiques recommandées pour exposer, partager et connecter des morceaux de données, informations et connaissances sur le Web sémantique grâce à l’utilisation des URI et de RDF. Les DONNÉES LIÉES [Heath & Bizer 2011, Berners-Lee 2009, Bizer *et al.* 2009, Berners-Lee 2006] décrivent une méthode de publication de données semi-structurées sur le Web de données, afin de lier les données et de les rendre plus accessibles, augmentant ainsi l’utilisation que l’on peut en faire. La méthode de publication peut se résumer en quatre principes introduits par Tim Berners-Lee :

- Utiliser des URI pour identifier les choses ;
- Les URI doivent utiliser le protocole http afin de rendre les ressources accessibles via ce protocole ; on parle d’URI déréférencables ;
- Adapter le format des informations servies en cas de déréférencement d’un URI en fonction de la nature de l’agent à l’origine de la requête ; par exemple, si un humain a effectué cette requête à l’aide d’un navigateur web, les informations sont envoyées sous la forme d’un document HTML, ou s’il s’agit d’un agent logiciel, les informations sont envoyées en RDF [Carroll & Klyne 2004, Hayes 2004, Lassila & Swick 1999, Manola & Miller 2004, Beckett 2009b] ;

17. <http://linkeddata.org/>

- Ajouter des liens à des éléments de jeux de données externes en utilisant leurs URI.

Les premier et deuxième principes indiquent qu'une entité (ressource) doit être identifiée par une unique URI basée sur le protocole http qui sert d'identifiant unique pour la ressource, mais elle offre également, un accès HTTP à une représentation de données structurées au sujet de ladite ressource. Ainsi, il est possible d'effectuer un look-up (ou déréférencement) de cette URI, *i.e.*, qu'une requête HTTP vers cette URI recevra en réponse de l'information concernant la ressource identifiée par l'URI. Le troisième principe veut que cette information soit représentée de manière interprétable par les machines, *i.e.*, selon le standard RDF pour la représentation des données, et selon le standard SPARQL pour l'adressage de requêtes. De même, selon le quatrième principe, il est demandé de fournir des références (liens) vers d'autres URI dans les résultats renvoyés, de manière à permettre aux utilisateurs et aux machines d'effectuer un processus itératif lors de leur recherche d'information.

Toujours dans le but de pouvoir lier les informations entre elles, il est fort probable, dans un système en pleine évolution comme celui-ci, d'avoir deux sources de données (ou plus) possédant des informations sur des ressources représentant un même concept réel. Ces ressources se trouvant sur des espaces de noms différents, elles possèdent des URI différentes et sont donc virtuellement considérées comme des concepts différents, alors qu'en réalité elles représentent toutes un seul et même concept réel. Afin de pallier ce problème, des liens spéciaux tels que `sameAs` et `refersTo` ont été définis [Hogan *et al.* 2010] dans le langage d'ontologies OWL, permettant d'indiquer qu'une ressource représente en réalité le même concept physique qu'une autre ressource. A terme, l'application de ces principes en vue de faire grandir et d'alimenter le Web de données devra permettre l'apparition de navigateurs Linked Data, de mashups Linked Data et de moteurs de recherche Linked Data.

### 2.2.5.1 Données liées ouvertes (Linked Open Data)

Nous appelons le graphe de données résultant de la pratique de la publication qui suit ces principes, graphe de données liées, ou simplement " DONNÉES LIÉES [Heath & Bizer 2011, Berners-Lee 2009, Bizer *et al.* 2009, Berners-Lee 2006] " (en. Linked Data). Les données liées ouvertes (en. Linked Open Data, LOD par la suite) représentent le sous-graphe de Données Liées qui est accessible de manière publique libres de droits sur le Web de données selon les principes de Données Liées, en interconnectant des données de différentes sources de données. Des représentations graphiques de ces ensembles de données interconnectés (linked datasets, ou LDS) ont été réalisées. La quantité de Données Liées ouvertes a connu une croissance exceptionnelle ces dernières années. Chaque flèche liant deux sources de données indique que ces sources ont été liées entre elles, et ont défini un ensemble de liens `sameAs`, afin de signifier à l'utilisateur qu'une ressource définie par la première source est conceptuellement équivalente à une autre ressource définie par la deuxième source.

La Figure 2.6 démontre une croissance du nombre de jeux de données liées ouvertes depuis l'année 2007.

Ce projet vise à connecter un maximum de sources de données entre elles, afin d'obtenir "UN" grand et unique Web de données. Les sources de données peuvent cependant être regroupées par nœuds ou clusters, *i.e.*, par thématique. Par exemple, toutes les sources

## CHAPITRE 2. ÉVOLUTION DU WEB

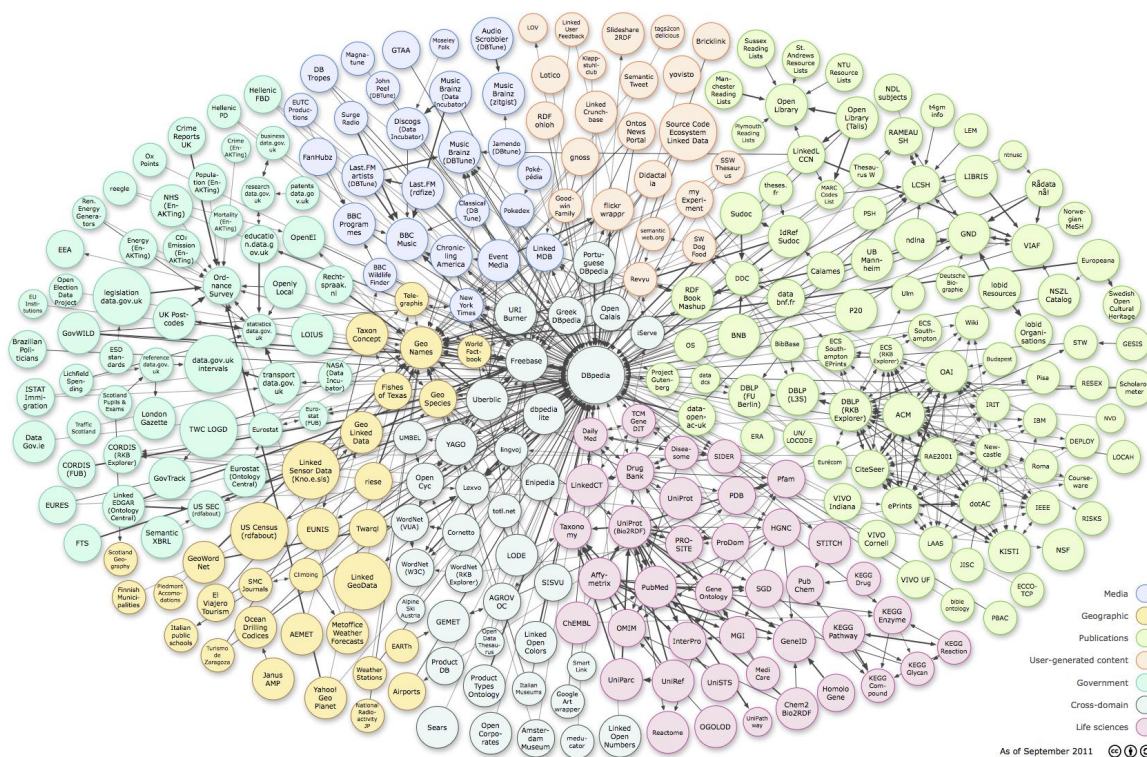


FIGURE 2.6 – Représentation des sources de données interconnectées grâce à Linking Open Data (en Aout 2014).

Mois et année	Nombre de triplets RDF	Nombre de liens RDF
Mai 2007	500.000.000	120.000
Avril 2008	2.000.000.000	3.000.000
Septembre 2011	31.000.000.000	504.000.000
Aout 2014	85,567,007,302	630,631,700

TABLE 2.7 – Evolution des ensembles de données publiées grâce aux Données Liées Ouvertes

contenant de l'information sur les génomes peuvent être regroupées en un cluster propre à ce thème, toutes les sources à vocation cinématographique dans un cluster spécifique au cinéma, etc. Il est fort intéressant d'observer l'évolution au cours du temps des sources de données interconnectées grâce au projet Données liées ouvertes, illustrée à la table<sup>18</sup> 5.5.2.2. Ce qui est très intéressant à remarquer, c'est la croissance exponentielle du nombre de sources de données interconnectées, et donc du nombre de données publiées sur le Web de données. La Figure 2.6 montre la version la plus récente du nuage du web de données, schéma emblématique du mouvement du web de données, mis à jour en 2014-08-30. La composition actuelle du graphe de données liées ouvertes est présentée en Figure 2.6 qui présente une densité variable de liens entre des jeux dans les différentes parties de ce graphe.

18. <http://stats.lod2.eu/stats>

Des parties du graphe qui représentent des domaines avec plusieurs jeux de contenu similaire ont, naturellement plus de connexions. On y voit les principaux jeux de données qui forment le web de données, ainsi que les relations qui les unissent. Il est particulièrement important de souligner le rôle central de DBpedia<sup>19</sup> dans ce graphe, et sa connectivité avec un grand nombre d'autres sources.

La base de connaissances DBpedia est au cœur de la toile tissée par les liens entre chaque jeu de données. Cette popularité s'explique par la largeur du spectre qu'elle couvre. En effet, DBpedia est le résultat d'efforts universitaires et communautaires d'extraction de données sur la base de Wikipedia<sup>20</sup>. D'autres jeux de données jouent un rôle dans la cohésion du web de données de par l'omniprésence des domaines qu'ils modélisent dans de multiples autres domaines ; par exemple, FOAF permet de décrire des personnes, et GeoNames<sup>21</sup> d'exprimer des informations géospatiales. Parmi les autres jeux de données notables, on en dénombre quelques-uns provenant des communautés de recherche en sciences de la vie qui, comme nous l'avons dit plus haut, entretiennent une longue tradition d'exploitation des méthodes de représentation des connaissances. D'autres, construits par la communauté ou par des acteurs privés, ciblent des domaines d'intérêt du grand public, comme la musique pour Musicbrainz. Enfin, d'autres encore, plus récents, ont été mis en ligne par des collectivités publiques (états, régions, communes. .) ; ils sont le fruit de courants politiques (principalement dans les pays anglo-saxons) visant à promouvoir la transparence dans le fonctionnement des organisations publiques et à encourager toute nouvelle idée d'applications exploitant ces données. L'initiative du web de données a rencontré un franc succès.

### 2.2.5.2 Les Triple Stores

Un Triple Store est une base de données conçue pour le stockage de triplets RDF. On pourrait comparer cela à une table d'une base de données relationnelle traditionnelle, contenant simplement trois colonnes (sujet - prédicat - objet), et une grande quantité d'entrées. Cependant, contrairement à une base de données traditionnelle, le Triple Store est optimisé pour un fonctionnement basé sur le stockage et la récupération des triplets RDF, avec notamment les mécanismes d'indexation qui en découlent. Son interfaçage est cependant relativement semblable à celui des bases de données traditionnelles : afin d'écrire ou de récupérer des données du Triple Store, il est nécessaire de formaliser la requête en SPARQL.

Un triple store permet de stocker de l'information semi-structurée pour le Web sémantique, mais encore faut-il avoir la possibilité de l'interroger afin de récupérer cette information [Verborgh *et al.* 2014, Arenas & Pérez 2011]. Cela peut notamment se faire via un SPARQL endpoint jouant le rôle d'interface de requêtes. Parmi les implémentations de Triple Stores, on peut citer entre autres Virtuoso<sup>22</sup> (proposant une version open-source de son outil),

---

19. <http://wiki.dbpedia.org/>

20. <http://www.wikipedia.org/>

21. <http://www.geonames.org/>

22. <http://virtuoso.openlinksw.com/>

Jena SDB<sup>23</sup>, AllegroGraph<sup>24</sup> ou encore Mulgara<sup>25</sup>.

### 2.2.5.3 Web des données et interrelations

Le web des données consiste à publier des données sur le web de telle sorte qu'elles puissent être interprétées et connectées entre elles. Les sources liées du web de données reposent sur des formats, une architecture et des langages de requêtes standardisés. RDF(S) & SPARQL sont notamment deux langages du Web sémantique et du Web de données utilisés pour l'Open Data<sup>26</sup>.

De grandes quantités de données structurées ont été publiées, notamment dans le cadre du projet Linking Open Data. Cependant chaque source varie par les schémas, les domaines et même les langues naturelles utilisées dans son contenu. Ces différences appellent la mise en place de liens entre ces données à la fois pour le web des données et pour le WEB SÉMANTIQUE [Berners-Lee 1998, Berners-Lee *et al.* 2001, Antoniou & Harmelen 2008, Hendler 2008] qu'il contribue à nourrir. Les jeux de données sont exprimés en fonction d'une ou de plusieurs ontologie [Gruber 1993] permettant de fixer le vocabulaire dans lequel les données sont exprimées. Dans de nombreux cas, des jeux de données décrivant des ressources d'un domaine similaire sont publiées en utilisant des ontologies [Gruber 1993] différentes.

Au vu de l'immense quantité de données publiées, il est nécessaire de fournir des méthodes de liage automatique de ces données afin d'en extraire les alignements sémantiques spécifiant les ressources susceptibles d'être liées. Plusieurs outils ont récemment été proposés pour résoudre ce problème, chacun ayant ses propres caractéristiques.

Le web des données requiert de lier entre elles les différentes sources de données publiées et par conséquent découvrir les correspondances entre entités provenant de diverses ontologies. Nous présentons un cadre général dans lequel s'inscrivent les différentes techniques utilisées pour établir ces liens et nous montrons comment elles s'y insèrent.

Bien que l'alignement d'ontologies [Euzenat *et al.* 2007, Zimmermann & Euzenat 2006, Cheatham & Hitzler 2014] et le liage de données peuvent être similaires dans un certain sens (tous deux relient des entités définies formellement), il y a d'importantes différences telles qu'illustrées dans le cadre précédent. En particulier, l'un agit au niveau du schéma alors que l'autre travaille au niveau des données. Ces différences sont reflétées dans le type de spécification de ces processus :

- Une assertion `sameAs` dit quelle Hotel de wikipedia correspondent à Hotel de Schema,
- Une spécification de liage indique comment trouver des tels liens, par exemple pour lier un `dbpedia :Hotel` à `Schema :Hotel`, évaluer comment l'étiquette (label) du premier est proche du nom (name) du second à l'aide d'une mesure appropriée (par exemple,

---

23. <http://jena.apache.org/documentation/sdb/>

24. <http://www.franz.com/agraph/allegrograph/>

25. <http://mulgara.org/>

26. <http://lov.okfn.org/dataset/lov/>

jaroSimilarity), évaluer la proximité entre la populationTotal du premier population du second avec une autre mesure (par exemple, numSimilarity), calculer la moyenne des deux valeurs et si le résultat est supérieur à 0.9, alors engendrer le lien sameAs.

- Un alignement d'ontologies précise quels composants d'une ontologie correspond à quels composants de l'autre ontologie. Par exemple, dbpedia :City est une sorte de geonames :P et dans ce contexte, label est équivalent à name et populationTotal est équivalent à population.

**Définition 2.2.13** (Une source de données). Une source de données  $\sigma_x$  est un ensemble de triplets RDF qui son publiés, maintenues et agrégées par un seul fournisseur.  $\sigma_x$  est un sous-graphe du Web de données (Données Liées Ouvertes);  $\sigma_x \subset G_{lod}$  dont les ressources sont incluses dans  $\sigma_x$ .

Ceci résulte en deux spécifications de processus - liage et alignement - et leurs résultats - linksets entre données et alignements entre ontologies.

**Définition 2.2.14** (Liage de données). Liage de données consiste à détecter que deux descriptions d'entités réfèrent au même objet du monde réel (e.g., même personne, même article, même gène).

- Chaque entité est décrite par une URI et par une description.
- $SameAs(i_1, i_2)$  : exprime que les deux descriptions de  $i_1$  et de  $i_2$  réfèrent au même objet du monde réel.
- Soient  $I_1, I_2$  deux ensembles URIs correspondant aux descriptions d'entités de deux sources  $S_1$  et  $S_2$ .
- Le problème de liage de données consiste à trouver toutes les paires d'URIs  $(i_1, i_2)$  de  $I_1 \times I_2$  telles que :  $owl :SameAs(i_1, i_2)$  ou  $owl :differentFrom(i_1, i_2)$

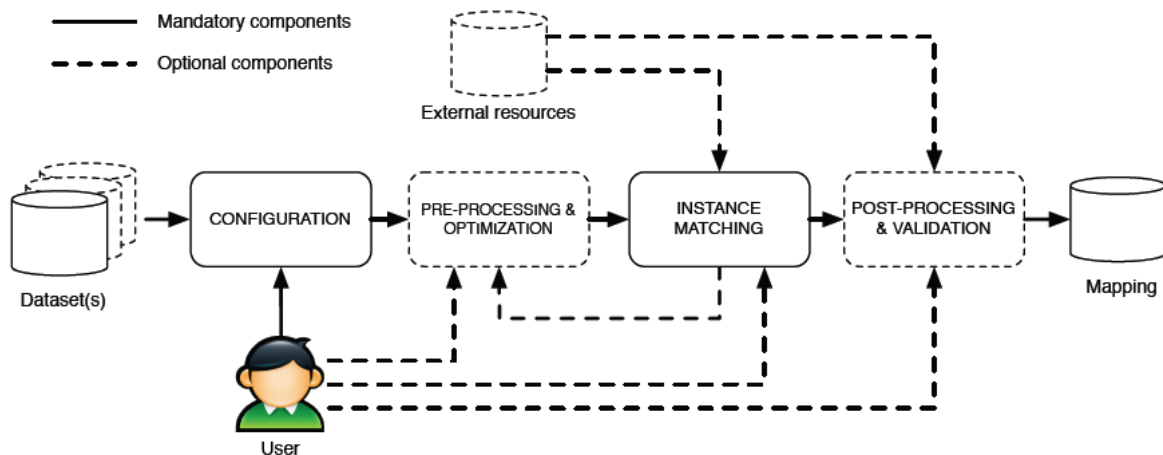


FIGURE 2.7 – Processus de Liage de données [Heath & Bizer 2011]

**Définition 2.2.15** (Linkset). Un Linkset ( $LS$ ) est un ensemble de triplets rdf, triplet  $t_i = (s_i, p_i, o_i) \in LS$ , dont les sujets  $s_i \in \sigma_{source}$  est décrit dans l'une des sources de données et les objets dans un autre  $o_i \in \sigma_{cible}$

Ces outils prennent en entrée deux jeux de données et produisent ultimement un "link-set", un graphe de liens décrivant les correspondances entre deux ressources. Ils se basent principalement sur des techniques d'alignement de chaînes de caractères [Moreau *et al.* 2008], ainsi que sur des techniques plus élaborées utilisant des outils linguistiques et exploitant la structure de graphe des jeux de données. Ces jeux de données sont reliés par une méthode de comparaison des ressources et sont décrits chacun par une ontologie<sup>27</sup>. Le résultat de la méthode, automatique est un ensemble de relations owl :sameAs entre ces ressources :

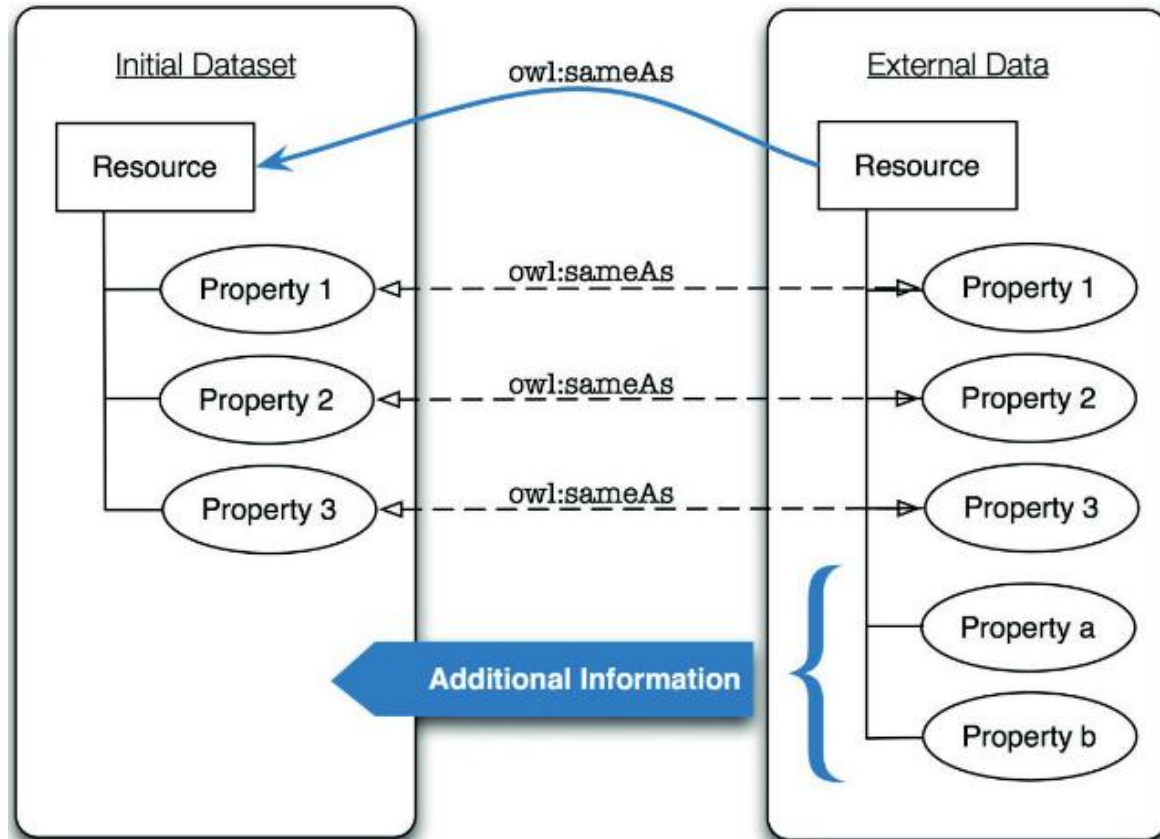


FIGURE 2.8 – Techniques de Liage [Jentzsch *et al.* 2010]

- **Mise en correspondance des identifiants** : On applique un ensemble de règles bien défini pour identifier les ressources équivalentes à partir de leur identifiant sur la base de leur motif.
- **Ontologie hétérogène et alignement implicite** : Un alignement entre les ontologies est utilisé pour indiquer au système d'alignement des données les correspondances entre les entités des ontologies
- **Alignement de données avec ontologie commune** : Le rôle du système d'alignement des données est d'inspecter les ressources de même type afin de détecter celles qui sont équivalentes. Pour cela le système va comparer les propriétés des ressources et construire une mesure de similarité. Les systèmes de cette catégorie sont para-

27. n fait chacun peut être décrit par plusieurs ontologies qui peuvent différer d'un ensemble à l'autre.



métrés par les propriétés à comparer, le type d'algorithme de comparaison à utiliser pour chaque propriété, et la façon dont la mesure de similarité entre les ressources est construite en fonction de la similarité entre les propriétés.

- **Ontologies différentes et alignement explicite** : La seule différence est qu'un alignement est disponible entre les ontologies et utilisé par l'outil. L'utilisateur n'a pas dans ce cas à décrire d'alignement.

Les Types d'approches de Liage peuvent être classifiés :

- **Approche locale (instance-based)** : exploitation des attributs
- **Approche globale (graph-based)** : exploitation des attributs et des relations pour propager des similarités/décisions (liage collectif des entités)
- **Approche supervisée** : nécessite l'intervention d'un expert (échantillon de données liées, approche manuelle interactive)
- **Approche informée** : exploite les connaissances déclarées dans une ontologie (ou sous une autre forme, par un expert)

La capacité à structurer de grandes quantités d'informations provenant de sources différentes est un enjeu technologique essentiel, et cette capacité réside notamment dans la possibilité de classer l'information par entité. Pour améliorer la recherche d'information, il est donc nécessaire de savoir reconnaître la même entité lorsqu'elle apparaît sous des formes différentes.

Cette partie le problème d'identification de séquences de mots représentant une même entité nommée (EN), dans un cadre où l'on dispose des entités elles-mêmes et du contexte dans lequel elles apparaissent. La difficulté porte sur les nombreuses variations textuelles possibles d'une EN : ces variations peuvent être volontaires et/ou naturelles (écriture différente selon la langue, abréviations ou extensions, surnoms, etc.) ou involontaires (erreurs typographiques ou orthographiques, erreurs d'OCR, etc.). On s'intéresse ici particulièrement aux variations dues aux translittérations (traductions entre systèmes d'écriture différents). L'hypothèse de départ réside donc dans l'idée qu'un même référent conduit généralement à des formes similaires, ainsi qu'à des ressemblances de leurs contextes d'occurrence.

L'appariement d'entités nommées coréférentes est notamment étudié dans la problématique du liage des Données Liées, qui consiste à repérer deux instances distincts représentant la même ressource. Les mesures de similarité (ou de distance) textuelle sont les principaux outils utilisés pour ce type de tâche. On peut grossièrement classer les différents types de mesures existants en trois classes :

- Les **méthodes basées sur les séquences de caractères**, qui définissent la similarité par la présence de caractères identiques à des positions similaires (e.g. Levenshtein, Jaro).
- Les **méthodes de type sac de mots**, qui sont basées sur le nombre de mots en commun entre les deux chaînes, indépendamment de leur position. Notons que ces types de mesures sont également applicables aux n-grammes de caractères au lieu des mots.
- Les **méthodes hybrides**, qui combinent les caractéristiques des deux précédents types

de mesures.

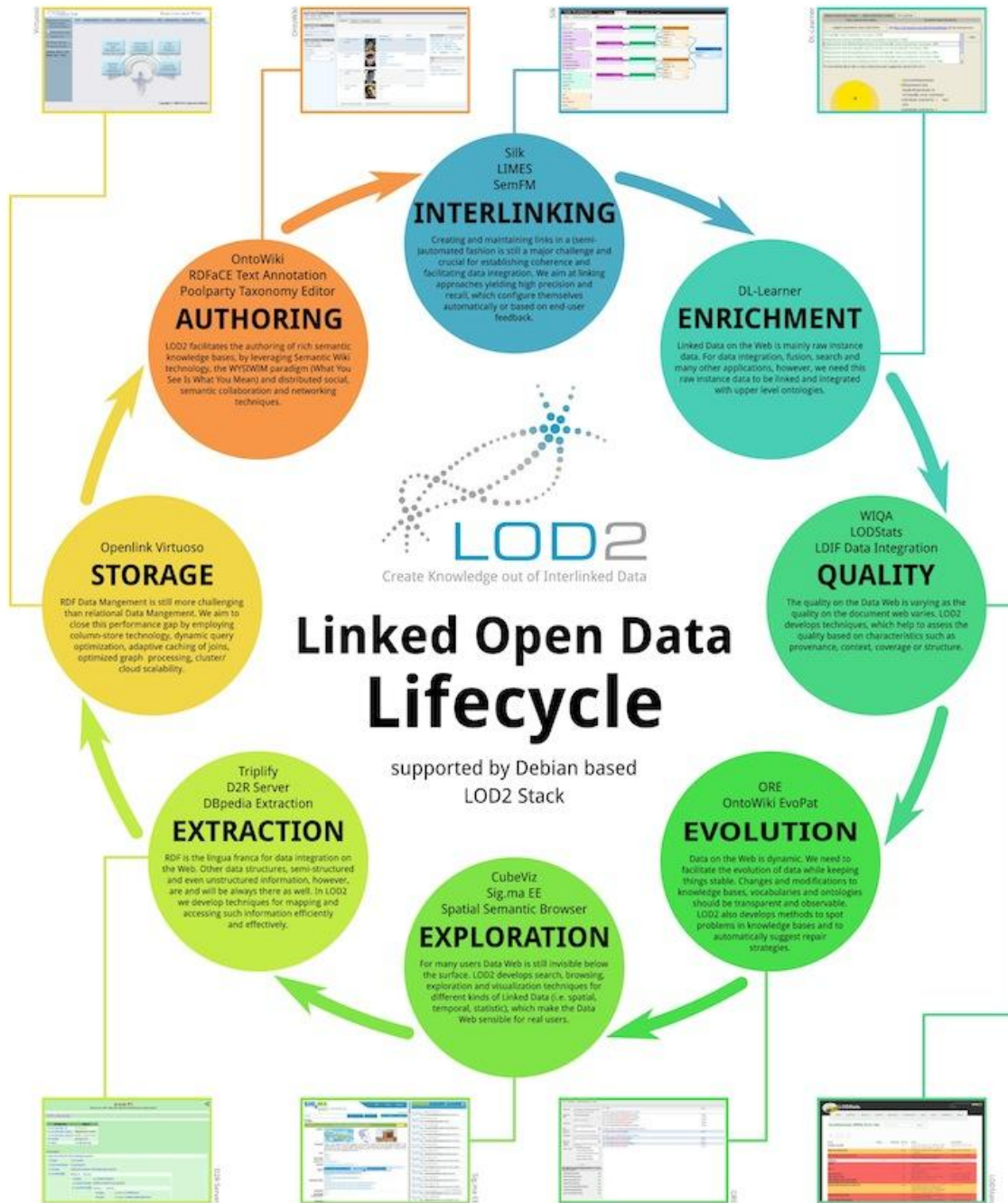


FIGURE 2.9 – Cycle de vie des données liées

Nous présentons ci-dessous quelques-unes des principales mesures fréquemment utilisées pour l'appariement d'EN.

- *La distance d'édition de Levenshtein (et variantes)*. Cette mesure de distance  $d$  représente le nombre minimal d'insertions, suppressions ou substitutions nécessaires pour

transformer une chaîne  $x$  en une chaîne  $y$ . La similarité  $s(x, y)$  normalisée sur  $[0, 1]$ , est définie par :

$$s = 1 - d/\max(|x|, |y|).$$

- *La métrique de Jaro.* Cette mesure est basée sur le nombre et l'ordre des caractères communs entre deux chaînes. Étant données deux chaînes  $x = a_1 \dots a_n$  et  $y = b_1 \dots b_m$ , soit  $H = \min(n, m)/2$  : un caractère  $a_i$  de  $x$  est en commun avec  $y$  s'il existe  $b_j$  dans  $y$  tel que  $a_i = b_j$  et  $i - H \leq j \leq i + H$ . Soit  $x' = a'_1 \dots a'_i$  (respectivement  $y' = b_1 \dots b'_k$ ) la séquence de caractères de  $x$  (resp.  $y$ ) en commun avec  $y$  (resp.  $x$ ), dans l'ordre où les caractères apparaissent dans  $x$  (resp.  $y$ ). Toute position  $i$  telle que  $a'_i \neq b'_i$  est appelée une transposition. Soit  $T$ , le nombre de transpositions entre  $x'$  et  $y'$  divisé par 2, la mesure de similarité de Jaro (Il est utile de noter que cette mesure n'est pas symétrique. On trouve dans la littérature et dans les implémentations existantes diverses variantes pour contourner ce problème.) est définie par :

$$Jaro(x, y) = \frac{1}{3} \times \left( \frac{\|X\|}{\|X'\|} + \frac{\|Y\|}{\|Y'\|} + \frac{\|Y'\| - T}{\|Y'\|} \right)$$

- Les mesures de type sac de mots ou de n-grammes de caractères. Pour ces mesures, chaque entité est traitée comme un ensemble d'éléments (les mots ou les n-grammes). Soient  $X = x_1 \dots x_n$  et  $Y = y_1 \dots y_m$  les ensembles représentant les EN  $x, y$  à comparer. Les mesures les plus simples ne prennent en compte que le nombre d'éléments en commun, par exemple :

$$Jaccard(x, y) = \frac{|X \cap Y|}{|X \cup Y|}; \text{Overlap}(x, y) = \frac{|X \cap Y|}{\min(X, Y)}; \text{Cos}(x, y) = \frac{|X \cap Y|}{\sqrt{|X|} \sqrt{|Y|}};$$

- Certaines mesures plus élaborées s'appuient sur une représentation vectorielle des ensembles  $X$  et  $Y$ , qui peut tenir compte de paramètres extérieurs aux ensembles eux-même. Soient  $A = (a_1, \dots, a_{|\Sigma|})$  et  $B = (b_1, \dots, b_{|\Sigma|})$  ces vecteurs, où  $|\Sigma|$  est l'ensemble des éléments considérés (e.g. tous les mots apparaissant dans au moins une EN), la similarité définie par le cosinus de l'angle  $T$  formé par  $A$  et  $B$  est fréquemment utilisée :  $\cos(A, B) = \frac{A^T B}{\|A\| \times \|B\|}$ . La représentation des éléments par leurs poids TF-IDF (Term Frequency-Inverse Document Frequency) est l'une des plus classiques. Il s'agit dans notre cas de mesurer l'importance d'un élément  $w$  pour une EN  $x$  parmi un ensemble  $E$  d'entités :

$$tf_{w,x} = \frac{n_{w,x}}{\sum_{w' \in \Sigma} n_{w',x}}, \text{idf}_w = \log \frac{\|E\|}{|\{x \in E | w \in x\}|}, \text{tfidf}_{w,x} = tf_{w,x} \times \text{idf}_w.$$

- Les combinaisons de mesures. Leur principe est la combinaison des propriétés des différents types de mesures présentés ci-dessus. Il s'agit généralement d'appliquer une sous-mesure simt aux mots des deux EN à comparer, puis d'en déduire un éventuel alignement optimal des EN. Il s'agit donc d'appliquer une méthode de type sac de mots, mais sans subir la rigidité d'un test d'identité entre mots. La sous-mesure doit bien sûr être choisie judicieusement.
- La mesure de Monge-Elkan calcule simplement la moyenne des meilleurs paires de mots  $n$  trouvés :

$$\text{sim}(x, y) = \frac{1}{n} \sum_{i=1}^n \max_{j=1}^m (\text{sim}'(x_i, y_j))$$

- *La mesure Soft-TFIDF* est une forme assouplie du cosinus sur les vecteurs de poids TF-IDF : grossièrement, deux mots différents peuvent être considérés comme identiques selon que leur score de sous-mesure dépasse ou non un seuil. Enfin, on peut mesurer la similarité des contextes des EN. On nomme contexte d'une occurrence d'une EN

l'ensemble des  $n$  mots qui la suivent et qui la précèdent, et le contexte (global) d'une entité distincte est formé par l'union des contextes de toutes ses occurrences. De façon classique, nous calculons l'ensemble des vecteurs représentant le contexte de chaque entité, chaque vecteur contenant les poids TF-IDF des mots de ce contexte. La similarité entre les contextes de deux EN est alors le cosinus de leurs vecteurs respectifs.

Ces mesures sont disponibles via deux bibliothèques publiques : SimMetrics<sup>28</sup> de S. Chapman et Second-String<sup>29</sup> de W. Cohen, P. Ravikumar et S. Fienberg.

Les outils existants développés pour la recherche de liens permettent de clarifier les interactions entre liage de données et alignement d'ontologies. Ceci présenterait plusieurs avantages : en particulier il devient possible de distribuer, partager et améliorer les spécifications de liage. Il est aussi possible de les réutiliser ou les étendre au lieu de les recalculer à chaque fois qu'un jeu de données est modifié. Finalement, cela permettrait de composer les spécifications de liage. On considère ci-après la manière de mener cette intégration. Comme on a la possibilité de faire inter-opérer plusieurs outils ensemble permettant ainsi de composer ces spécifications de liens afin de pouvoir passer d'un jeu de données directement à un autre sans passer par un ensemble intermédiaire. Parmi les outils les plus utilisées supportant plusieurs ontologies et non une seule, on cite :

- Silk [Jentzsch *et al.* 2010] est une approche locale et est paramétré par un langage de spécification de liens, le Silk Link Specification Language. L'utilisateur spécifie quelles entités doivent être reliées et indique la mesure de similarité à utiliser (e.g., Jaccard, Jaro). Silk utilise plusieurs méthodes d'alignement de chaînes de caractères, des méthodes mesurant les similarités numériques, de dates, la distance entre concepts dans une taxonomie, et la similarité d'ensembles (fonctions d'agrégation (e.g. max, moyenne)) Des transformations peuvent être spécifiées préparant le jeu de données avant le processus d'alignement afin d'en améliorer ses résultats. Silk prend en entrée deux jeux de données accessibles à travers un point d'accès SPARQL. Il fournit en sortie des triplets owl :sameAs ou d'autres prédicats spécifiés par l'utilisateur. Silk a été expérimenté sur plusieurs jeux de données indiqués sur la page de présentation du projet<sup>30</sup> et est implémenté en Python.
- Knofuss [Nikolov *et al.* 2008] est une approche Locale, non supervisée, informée et est destinée à fusionner des jeux de données. Une des particularités de Knofuss est sa capacité à aligner des jeux de données décrits en fonction d'ontologies [Gruber 1993] hétérogènes. Le processus d'alignement des données est dirigée par une ontologie dédiée spécifiant les ressources à comparer, ainsi que les techniques d'alignement adéquates à utiliser. Les ressources sont sélectionnées en spécifiant des requêtes SPARQL adressées aux jeux de données. L'outil propose un ensemble d'algorithmes d'alignement de chaînes de caractères. Lorsque les deux jeux de données à aligner sont décrits par des ontologies différentes, il est possible de spécifier un alignement dans le format d'alignement, permettant d'utiliser un des nombreux systèmes d'alignement

---

28. <http://www.dcs.shef.ac.uk/~sam/stringmetrics.html>

29. <http://secondstring.sourceforge.net>

30. <http://www.wiwiss.fu-berlin.de/bizer/silk/>



FIGURE 2.10 – Les mesures de similarité utilisées par SILK [Jentzsch *et al.* 2010]

d'ontologies disponible. Knofuss étant à l'origine dédié à la fusion de données, il est possible de spécifier une stratégie de fusion. Une étape post-opératoire est effectuée par l'outil pour vérifier la consistance du jeu de données résultant de l'opération de fusion. Cet outil utilise des copies locales des jeux de données et est implémenté en Java.

Metric	Description
jaroSimilarity	String similarity based on Jaro distance metric
jaroWinklerSimilarity	String similarity based on Jaro-Winkler metric
qGramSimilarity	String similarity based on q-grams
stringEquality	Returns 1 when strings are equal, 0 otherwise
numSimilarity	Percentual numeric similarity
dateSimilarity	Similarity between two date values
uriEquality	Returns 1 if two URIs are equal, 0 otherwise
taxonomicSimilarity	Metric based on the taxonomic distance of two concepts

FIGURE 2.11 – Silk Link Specification Language [Jentzsch *et al.* 2010]

- RKB-CRS Le système de résolution de co-référence (CRS) de la base de connaissances RKB [Nikolov *et al.* 2008] consiste en des listes d'équivalence entre URI. Ces listes sont construites en utilisant un programme Java sur mesure pour le domaine spécifique des conférences/universités. Un nouveau programme doit donc être réécrit pour chaque jeu de données. Chaque programme consiste à sélectionner les ressources à aligner, et à les comparer en appliquant des algorithmes de similarité de chaînes de caractères sur leurs attributs.

Dans notre thèse, nous adopterons un outil automatique de liages de données implémentant un langage expressif de représentation d'alignement entre ontologies aidant ainsi à renforcer les interrelations des données sur le web ainsi assurant une meilleure performance et efficacité de la découverte orientée composition.

### 2.3 Description de Services Web sémantiques

L'évolution des systèmes d'information et des processus en e-Business (réseau d'entreprise) passe par le développement de nouvelles applications qui requiert l'accès et l'intégration de ressources (applications et bases de données) distribuées et hétérogènes à moindre coût

### 2.3. DESCRIPTION DE SERVICES WEB SÉMANTIQUES

	<b>Silk</b>	<b>Knofuss</b>	<b>RKB CRS</b>
Ontologies	multi	multi (not considered)	multi (not considered)
Automatisation	semi	semi	semi
Spec. utilisateur	specification liens methode d'alignement	fusion onto.	programme sur mesure
Format entrée	Silk-LSL (XML)	OWL	Java
Techniques d'alignement	chaînes de car.	chaînes de car. apprentissage adaptatif	chaînes de car. alignement
Alignement onto.	non	oui, en entrée	non
Sortie	linkset	format d'alignement jeu de données fusioné	owl:sameAs
Accès aux données	SPARQL	copie locale	API
Domaine	multiple	multiple	publications
Post-opérateur	non	résolution de consistance	non

FIGURE 2.12 – La comparaison entre les trois outils

en utilisant la technologie des Services Web. Ces SERVICES WEB [Haas & Brown 2004] autorisent une plus grande flexibilité dans les relations inter-partenaires puisque chaque partenaire a la possibilité de posséder ses propres composants dans un système d'intégration faiblement couplé et décentralisé. Ils peuvent fournir des informations statiques ou dynamiques modifiant les états des systèmes dans lesquels ils sont immergés. Plusieurs définitions ont été données des SERVICES WEB [Haas & Brown 2004] mais la définition la plus aboutie est celle du W3C "Ce sont des composants logiciels qui fournissent des services distants, distribués et accessibles via des protocoles standardisés du Web [Jones 2005, Haas & Brown 2004]".

L'émergence des architectures et des méthodologies "orientées services" a donné naissance au Service oriented Computing (SOC- PROGRAMMATION ORIENTÉE SERVICE [Papazoglou 2003]) et son architecture associée Service Oriented Architecture (SOA- ARCHITECTURE ORIENTÉE SERVICE [Erl 2005]) permettant de réduire les coûts d'intégration en déployant les Services Web. Le domaine de SERVICES WEB [Haas & Brown 2004] trouve de plus en plus de l'intérêt dans le domaine du WEB SÉMANTIQUE [Berners-Lee 1998, Berners-Lee *et al.* 2001, Antoniou & Harmelen 2008, Hendler 2008] puisque ni les normes [Walsh 2002] (telles que SOAP [Mitra & Lafon 2007] et UDDI), ni les langages de description ou de modélisation de processus métiers [Juric 2010] (tels que WSDL et WS-BPEL) ne considèrent la sémantique ce qui pénalise à la fois la découverte et la composition automatique. Toutefois, une intégration automatisée, ad hoc n'est possible qu'au prix d'une modélisation formelle, sémantique et comportementale des interfaces de ces "services" afin de la rendre dynamique

*i.e.*, exécutable au moment voulu. C'est ainsi qu'a émergé naturellement l'extension SERVICES WEB SÉMANTIQUES [Jorge 2006, McIlraith *et al.* 2001] (SWS, en anglais Semantic Web Services) qui se différencie de son homologue les services web actuels, non sémantique. Pour y remédier et améliorer l'interopérabilité [Euzenat 2001], le web sémantique propose l'enrichissement sémantique des mécanismes d'annotation du standard de la description de services Web spécifié. Ce qui permet, d'une part, une description sémantique explicite de leurs fonctionnalités, et d'autre part, une interprétation correcte des informations envoyées et reçues dans le cadre de la découverte, l'invocation et surtout la composition. Pour cela, Nous détaillerons quatre descriptions sémantiques des Services Web qui se distinguent par la sémantique qu'elles induisent pour le service associé à savoir :

- La *description abstraite ou conceptuelle* d'un service, qui regroupe les informations décrivant la fonctionnalité du service par rapport au domaine d'activité associé et représenté de deux manières conjointes : (i) texte descriptif en langue naturelle utilisant des mots-clés sensible à de multiples variations interprétatives, notamment la description par mots-clés dans UDDI et (ii) une représentation formelle utilisant un formalisme rigoureux interprétant comme l'ensemble des buts et ou des effets du service abstrait issus d'une ontologie regroupant les définitions logiques de description des concepts du domaine. Cependant, La description abstraite n'est pas spécifique aux Services Web Sémantiques.
- La *description fonctionnelle* d'un service est la description du service vue comme une "boîte noire" détaillant les entrées les sorties, les pré-conditions et les effets associés au service en utilisant une ontologie du domaine. Elle peut être complétée par des propriétés non fonctionnelles, par exemple temps de réponse du service, éventuelles contraintes, qualité de Service (QoS) On distingue les descriptions fonctionnelles paramétriques des descriptions fonctionnelles instanciées. Une description paramétrique décrit les entrées, sorties, pré-conditions et effets sous la forme d'un concept alors qu'une description fonctionnelle instanciée les décrit avec une valeur précise.
- La *description comportementale* d'un service, explicite le comportement dynamique du service utilisant une ontologie des processus et des actions décrivant le protocole de dialogue ou encore sa politique de contrôle d'états. Cette description sert de plan d'exécution ainsi que d'un point de dialogue avec le service. La sémantique de cette description est donc opérationnelle, contenant ainsi les actions à réaliser, leur enchaînement, les flux de données échangées,
- La *description exécutable* d'un service, ou "service exécutable" est la description de son interface d'appel (nom du service, nom de la procédure invocable à distance, paramètres, adresse) correspondant aux appels possibles de ce service.

Ainsi, les technologies sémantiques appliquées au Web ont permis de construire un "web de données" que les applications pourront partager sur le Web pour faciliter leur interopérabilité de même l'idée s'est prolongée vers la construction d'un "web de services" que les applications pourront intégrer et partager grâce à des standards de modélisation et de représentation. Les SERVICES WEB SÉMANTIQUES [Jorge 2006, McIlraith *et al.* 2001] (SWS) sont présentés et définis dans le contexte du paradigme du Web sémantique que du Web Service. Ce sont des programmes dont les effets sur leur environnement sont connus et dont les données manipulées



possèdent une sémantique. Pour cela, de nombreux langages et approches ont été développés dans l'objectif de décrire explicitement les SERVICES WEB SÉMANTIQUES [Jorge 2006, McIlraith *et al.* 2001], les plus connues parmi eux à savoir SAWSDL [Kopecký *et al.* 2007], OWL-S [Burstein *et al.* 2004], WSMO [Bruijn *et al.* 2005] et USDL [Barros & Oberle 2012]. Par la suite, on présentera les principes et les détails du standard des Services Web (). Ensuite, On décrira les approches SAWSDL (section 2.3.2.1), USDL (section 2.3.2.1), OWL-S (section 2.3.2.2), et l'approche WSMO (section 2.3.2.2). Enfin, on propose une synthèse de l'ensemble de ces langages.

### 2.3.1 Description syntaxique des Services Web

Les applications peuvent communiquer automatiquement grâce aux SERVICES WEB [Haas & Brown 2004] à travers des messages échangés qui reposent sur une grammaire XML [Bray *et al.* 2006, Maler *et al.* 2004] bien structurée grâce au standard "Web Service Description Language" (WSDL). Ce dernier fournit un modèle de description de services Web au format XML séparant la description abstraite de la fonctionnalité offerte de ses détails concrets. Les éléments de la partie abstraite décrivent principalement le service en termes de messages envoyés et reçus totalement indépendamment par un schéma XML [Consortium 2004] (au niveau d'une balise XML notée "types"). Cette partie inclut aussi des éléments "operation" dont chacun associe un ou plusieurs messages à un modèle d'échange de messages en spécifiant le nombre et l'ordre ainsi que la cible et la source abstraite des messages. Ces opérations sont groupées dans un élément "interface / portType" sans en préciser le protocole. Quant à la partie concrète, elle précise les détails sur le moyen d'échange de messages d'une ou plusieurs interfaces spécifié par les éléments "binding" similaire à la structure d'un élément "interface/portType" mais offre deux types de transmission de messages "binding" entre client et services Web : soit intégrés dans des messages du protocole SOAP sur le protocole HTTP, soit directement dans les messages du protocole HTTP. Ainsi, un service Web peut être accessible depuis plusieurs adresses réseau "endpoint/port"s implémentant une même interface où chacun est potentiellement associé à un "binding" différent. Ces "endpoint/port" s sont groupés sous l'élément "service".

Initialement, la version WSDL 1.0 fut développée par IBM, Microsoft et Ariba en septembre 2000 qui combinait alors deux langages de description de services : NASSL (Network Application Service Specification Language) de IBM et SDL (Service Description Language) de Microsoft. Publiée en mars 2001, la version WSDL 1.1 est devenue largement utilisée même si elle n'a pas été approuvée par le W3C. Cette spécification englobe l'élément <definitions> qui contient six principaux éléments à savoir : types, message, portType, binding, port et service. Pour les endpoints, le WSDL 1.1 définit quatre primitives de transmission :

- **One-way** : l'endpoint reçoit un message,
- **Request-response** : l'endpoint reçoit un message et retourne un message en réponse,
- **Solicit-response** : l'endpoint envoie un message et reçoit un message en réponse,
- **Notification** : l'endpoint envoie un message

En juin 2003, une version plus flexible à vu le jour le WSDL 1.2, sauf que cette version n'était compatible avec la plupart des serveurs des fournisseurs de services. A noter que jusqu'ici, aucune des versions ne prévoient un élément ou attribut d'ordre sémantique pour la

## CHAPITRE 2. ÉVOLUTION DU WEB

description hors la version WSDL 2.0 qui se distingue des autres versions par la possibilité de l'étendre si besoin en ajoutant des éléments et/ou d'attributs à des fins sémantique. Cette dernière version est devenue une recommandation W3C depuis juin 2007 offrant une syntaxe plus riche et est dotée de l'élément englobant <description> et est définie par cinq constructions : types, interface, binding, endpoint et service. Les deux premiers éléments concernent la définition abstraite du service tandis que les trois autres sont relatifs à l'implémentation du service. La différence entre cette version et l'ultérieur réside dans la suppression des éléments " message", et le remplacement de l'élément XML " portType" par l'élément XML "interface" et l'attribut XML " port " par l'élément XML "endpoint ". En conclusion, L'annotation sémantique des documents WSDL est possible grâce à l'extensibilité de WSDL 2.0.

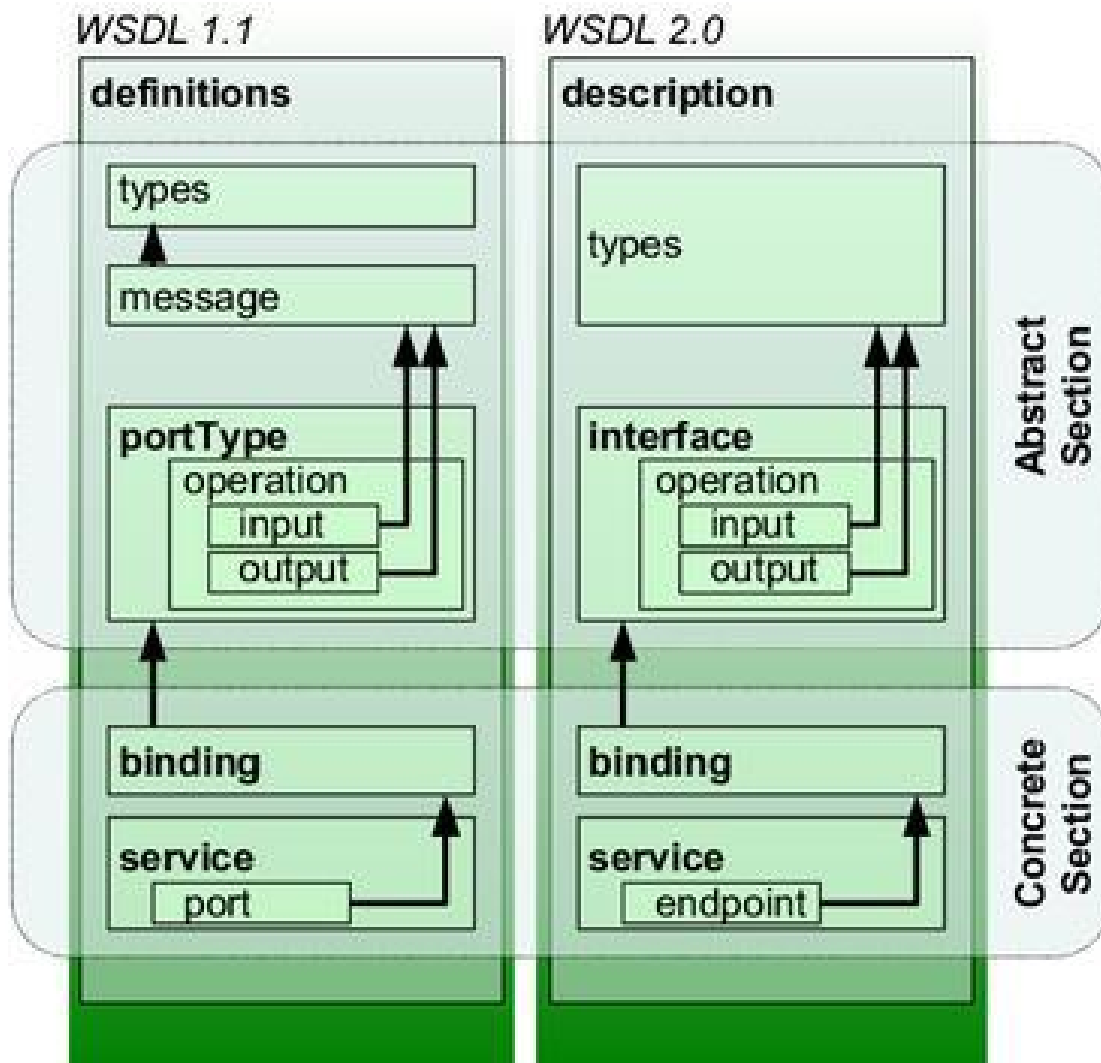


FIGURE 2.13 – Les versions de WSDL

Lorsque des services décident de collaborer, afin de réaliser une tâche complexe, qui ne

peut être effectuée que partiellement par chaque service, il faut que ces services soient chorégraphiés *i.e.*, ces services doivent respecter certaines règles et contraintes concernant les messages observables qu'ils échangent en utilisant le langage de description de chorégraphie comme WSCDL. Une fois la description WSCDL établie, il faut que la description WSCI de chaque membre de la chorégraphie respecte ces règles. Ainsi, la description WSCI de chaque service définit la description d'une partie des échanges de messages le concernant et de leurs dépendances dans la chorégraphie. Lorsque la collaboration est effectuée entre différents partenaires, possédant chacun un ensemble de services, les règles vont être établies après un accord entre ces derniers. Contrairement à WSCDL (Web Services Choreography Description Language) et WSCI (Web Service Choreography Interface) qui ne sont pas exécutables, BPEL4WS (Business Process Execution Language for Web Services) est un langage d'implémentation de processus qui définit le comportement interne du nouveau service, en plus de la description des échanges de messages entre ce nouveau service et les services existants en utilisant WSDL. La différence avec WSCDL est que le service décrit en BPEL4WS est celui qui orchestre les échanges de messages. Alors que, BPEL décrit des processus comportant des activités primitives et des activités composées et est basé sur le standard WSDL.

### 2.3.2 La nébuleuse des langages sémantiques

#### 2.3.2.1 (A) Annotations sémantiques

L'annotation sémantique consiste à enrichir et à compléter la description d'un service. Elle établit des correspondances entre des éléments de la description et des concepts d'un ensemble d'ontologies de référence. Une ontologie de référence permet de représenter un domaine par des structures interprétables par une machine. Nous présentons les approches qui s'inspirent de WSDL : SAWSDL [Kopecký *et al.* 2007] et USDL [Barros & Oberle 2012].

(1) **SAWSDL : Semantic annotation for WSDL** SAWSDL [Kopecký *et al.* 2007] est un langage sémantique de description de Service Web qui est défini au dessus des standards des Services Web existants et compatible avec l'existant. SAWSDL est une approche incrémentale puisque les outils existants autour de WSDL sont facilement modifiable et adaptable dessus et fournit un mécanisme permettant d'annoter sémantiquement les types de données, les opérations, les entrées et les sorties de WSDL. C'est un langage évolutif augmentant l'expressivité sémantique du langage WSDL en utilisant des concepts analogues à ceux utilisés dans OWL-S [Burstein *et al.* 2004] en y ajoutant des éléments et des attributs pour spécifier les pré-conditions, les effets et les catégories des services Web. Pour cela, ce langage fournit des mécanismes pour référencer des concepts de modèles définis à l'extérieur du document WSDL grâce à l'attribut "sawSDL". Il existe trois formes d'extensions de cet attribut. Le `modelReference` qui associe un composant WSDL (interfaces, opérations, entrées/sorties) ou XML Schema [Consortium 2004] (les types XML simples) à un concept d'une ontologie. Les deux autres `liftingSchemaMapping` et `loweringSchemaMapping` permettent de spécifier la correspondance (Mapping ou matching) entre les données sémantiques et les éléments XML. Ces derniers sont utiles pour établir la correspondance entre les structures des entrées et des sorties lorsque les structures XML demandées par le client et celles fournies par le service sont différentes. Par contre, l'annotation des types de données XML complexes nécessite en

plus un SchemaMapping du fait que deux services Web distincts peuvent manipuler le même type complexe mais avec deux structures différentes.

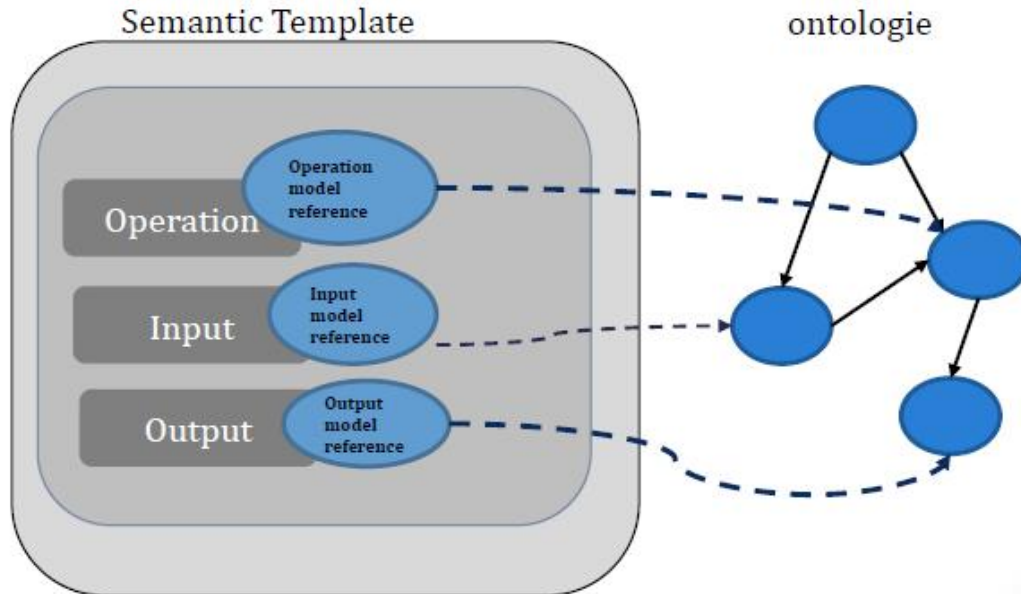


FIGURE 2.14 – Mécanisme d’annotation sémantique des types de données, des opérations, des entrées et des sorties

(2) **USDL : Unified Service Description Language** développé au sein du laboratoire de recherche ALPS (Applied Logic Programming-Languages and Systems) de l’Université du Texas, offre un langage qui décrit formellement et sémantiquement les Services Web en utilisant les concepts issus de l’ontologie OWL WordNet<sup>31</sup> au-dessus de la description syntaxique WSDL 1.1 intervenant sur les balises XML "Messages" afin d’en décrire les paramètres du service et "PortType" afin de décrire ces opérations en les associant à des concepts basiques, qualifiés, concepts inversés, conjonctif et disjonctifs (voir Figure fig :Dhjk). La description USDL garde pratiquement la même hiérarchie des éléments que WSDL mais apporte quelques modifications et de nouveaux éléments XML où les inputs et outputs sont définis sous forme d’éléments Message. Chaque Message pointe vers un élément BasicConcept qui fait référence à un concept provenant de WordNet. Pour cela, USDL propose une description sémantique basée sur des concepts en utilisant la logique de proposition, dont les principaux éléments sont :

- La classe Concept : est une classe générique pour modéliser des concepts du monde réel,
- La propriété Affects : est une classe générique pour décrire des effets du service sur le monde réel,

31. C’est une ontologie universelle donnant une représentation commune aux concepts du monde réel et fournit une abondance de relations sémantiques telles que l’hyponymie, l’antonymie, la synonymie, etc. entre les termes y figurant.

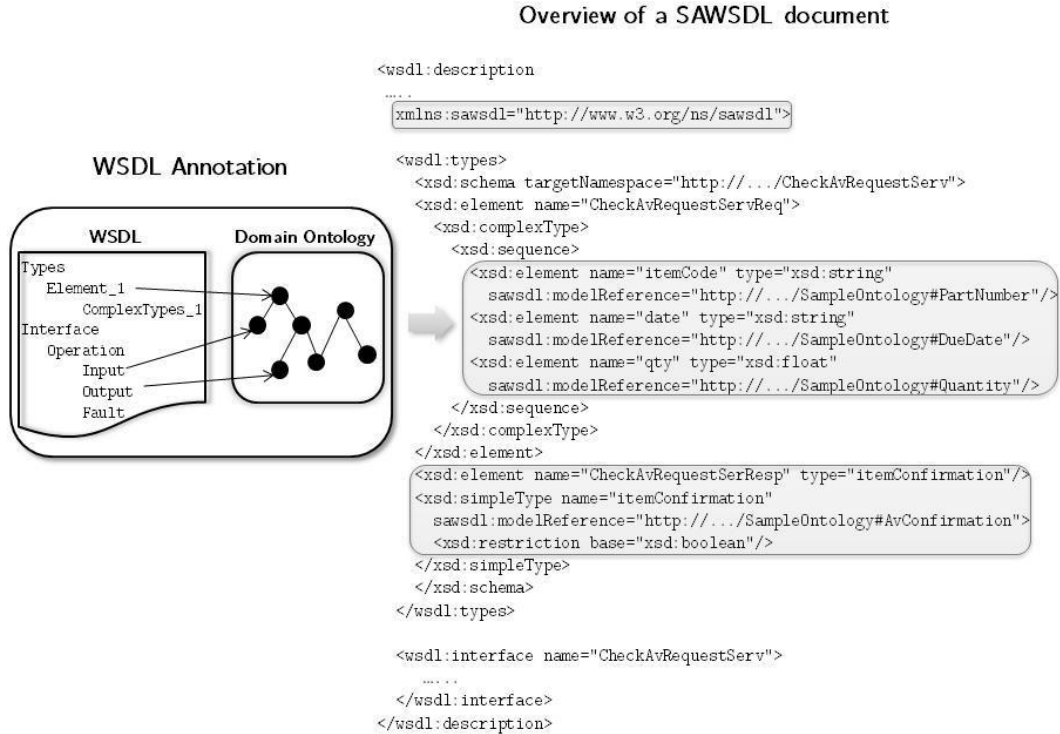


FIGURE 2.15 – Mécanisme d’annotation sémantique

— La classe Conditions : est une classe générique pour décrire des contraintes pour le service.

USDL limite sa considération aux propriétés sémantiques des effets secondaires suite à une opération de création, de mise à jour, de suppression ou de recherche (voir Figure 2.16). Par ailleurs, si aucun de ces effets n’est appliqué, alors un effet générique est utilisé. Toute application qui aurait besoin de faire appel à un service USDL doit impérativement avoir la capacité de raisonner avec le lexique et les termes spécifiques de WordNet appelés atomes (tels que les lemmas, senses, lexical semantic relations, etc.)

### 2.3.2.2 (B) Ontologies de services

**(A) WSMO : Web Service Modeling Ontology** est un autre standard émergent en concurrence avec OWL-S où il est possible d’exprimer les pré-conditions et les effets sur les inputs et les outputs des Web services et sur leur environnement. WSMO [Bruijn *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 et l’invocation et est basée sur WSMF (Web Service Modelling Framework) qui spécifie les éléments principaux pour décrire les services Web sémantiques.

Les facettes de WSMO [Bruijn *et al.* 2005] sont les ontologies, les médiateurs, les services Web et les objectifs.

— Les ontologies définissent la terminologie, utilisée par les autres éléments, en termes

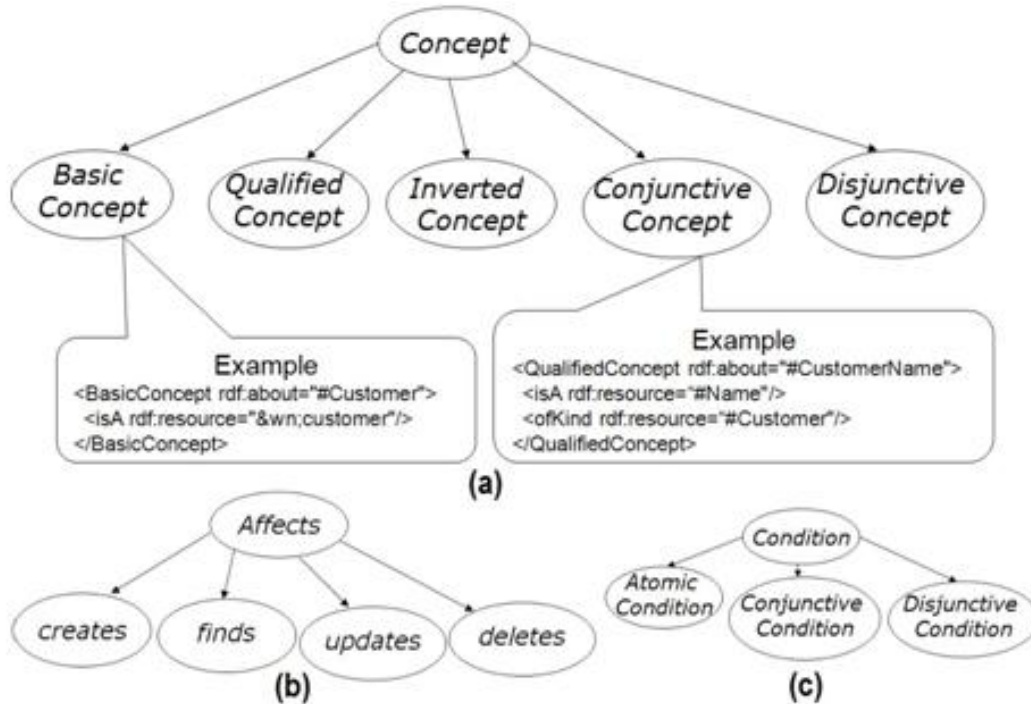


FIGURE 2.16 – Principaux éléments de description dans USDL

de concepts, relations, fonctions, instances et axiomes et fournissent la sémantique compréhensible par une machine pour les informations utilisées par tous les acteurs d'un service Web. WSMO permet d'importer une ontologie dans une autre ontologie soit directement, quand il n'y pas de conflits entre les concepts, soit indirectement, par le biais d'un médiateur qui va résoudre les conflits possibles. Les blocs de base d'une ontologie sont concepts, relations, fonctions, instances et axiomes. Un ensemble de propriétés non fonctionnelles est donné généralement au début de chaque définition d'une ontologie.

- Les médiateurs permettent de lier différentes ressources hétérogènes et résoudre les incompatibilités à plusieurs niveaux. Il peut y avoir une incompatibilité entre les données ou entre les processus. Dans le premier cas, la médiation sert à établir la correspondance entre les différentes terminologies. Le deuxième type d'incompatibilité apparaît au moment de la communication entre différents services Web hétérogènes et dans ce cas, le médiateur fournit la fonctionnalité pour une analyse d'exécution de deux services Web donnés, et compense les éventuelles disparités.
- Les services Web WSMO sont composés d'un ensemble optionnel de propriétés non fonctionnelles, un mécanisme permettant d'importer des ontologies soit directement en utilisant `importsOntology`, soit indirectement via un médiateur, une capacité décrivant la fonctionnalité du service en termes de pré-conditions et effets et une interface décrivant le comportement du service vis-à-vis de ses partenaires.
- Les buts indiquent ce que l'utilisateur attend du service. La description du service Web définit les fonctionnalités offertes par le service. Les médiateurs lient les différents

---

## 2.3. DESCRIPTION DE SERVICES WEB SÉMANTIQUES

---

éléments afin de permettre l'interopérabilité entre les composants hétérogènes. Le langage WSMML est utilisé pour décrire formellement tous les éléments de WSMO et l'environnement d'exécution WSMX permet la découverte, la sélection, la médiation, l'invocation et l'interopérabilité des services Web sémantiques. WSMO partage avec OWL-S les mêmes motivations à savoir la découverte, l'invocation et la composition automatiques des services Web. Cependant WSMO ajoute à celles-ci l'objectif suivant : Un découplage fort entre les composants et un rôle central pour la médiation. 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 veut distinguer comment le client formule sa demande et comment le fournisseur expose son service. 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.

**(B) OWL-S : Ontology Web Language for Services** OWL-S [Burststein *et al.* 2004] est un langage permettant de décrire les services Web de façon non ambiguë et interprétable par des programmes et est basé sur le langage d'ontologie du Web (OWL). Il est considéré lui-même comme une OWL assez particulière. OWL-S s'inscrit dans les approches de la composition automatique des services Web. L'accomplissement d'une tâche complexe implique la sélection, la composition automatique des services Web. Pour cela, ce langage définit une ontologie supérieure pour la description, l'invocation et la composition des services Web. La structuration de l'ontologie supérieure de OWL-S est motivée par la nécessité de fournir trois types d'information essentiels pour un service, à savoir :

- Que réalise le service : cette information est donnée dans le Service Profile. Ce dernier permet la description, la publication et la découverte des services, en spécifiant une description textuelle à destination des utilisateurs "humains", des propriétés fonctionnelles et des propriétés non fonctionnelles. La section Profile est utilisée à la fois par les fournisseurs pour publier leurs services et par les clients pour spécifier leurs besoins. Par conséquent, elle constitue l'information utile pour la découverte et la composition de services. Les recherches de services peuvent se baser sur n'importe quel élément de Profile comme critère.
- Comment utilisons-nous le service : cette information est fournie dans le Service Model qui est utilisé pour, entre autres, composer les services. OWL-S modélise les services en tant que processus et celui-ci est défini par ses entrées/sorties. Trois types de processus existent : les processus atomiques (AtomicProcess), simples (SimpleProcess) et composites (CompositeProcess). Un processus atomique représente le niveau le plus fin pour un processus et correspond à une action que le service peut effectuer en une seule interaction avec le client. Les processus composites sont décomposables en d'autres processus (composés ou non) ; leur décomposition peut être spécifiée en utilisant un ensemble de structures de contrôles tels que : Sequence, Split, If-Then-Else etc. Les structures de contrôle proposées par OWL-S sont :
  - Sequence : la séquence est une liste de processus qui doivent être exécutés dans l'ordre ;
  - Split : appelle des sous-processus de façon concurrente ;

- Split+join : consiste à lancer différents processus de façon concurrente mais de les synchroniser en attendant que toutes les sous-procédures du split+join soient achevées pour les joindre ;
- Any-order : définit l'exécution d'un ensemble de tâches dans un ordre quelconque ;
- Choice : le choix consiste à choisir entre plusieurs sous-procédures équivalentes ;
- If-Then-Else : correspond au if-then-else habituel, c'est-à-dire si la condition du if est réalisée, alors la sous-procédure du then est exécutée, sinon la sous-procédure du else est exécutée ;
- Iterate : boucle continuellement avant d'être interrompu par whileCondition ou untilCondition ;
- Repeat-While et Repeat-Until : sont des boucles conditionnelles qui répètent une sous-procédure tant que (Repeat-While) la condition de la boucle n'est pas satisfaite ou jusqu'à ce que (Repeat-Until) la condition de la boucle soit réalisée. Enfin le ServiceGrounding définit les entrées et les sorties du service Web ainsi que les informations nécessaires pour appeler le service Web. Il permet de lier les AtomicProcess avec des services Web existants.

Un processus atomique est utilisé pour fournir une vue d'un processus atomique ou une représentation simplifiée d'un processus composite. Les Processus simples ne peuvent pas être appelés directement comme les Processus atomiques, ce sont des procédures ne comportant qu'une seule étape et pouvant être réalisées par un Processus atomique ou composite. Ils sont en fait une vue alternative des deux autres procédures.

- Comment accédons-nous au service : cette information est donnée dans le Service Grounding. Celui-ci indique comment accéder concrètement au service et fournit les détails concernant les protocoles, les formats de messages et les adresses physiques. Cette information est particulièrement utile pour l'invocation automatique de services.

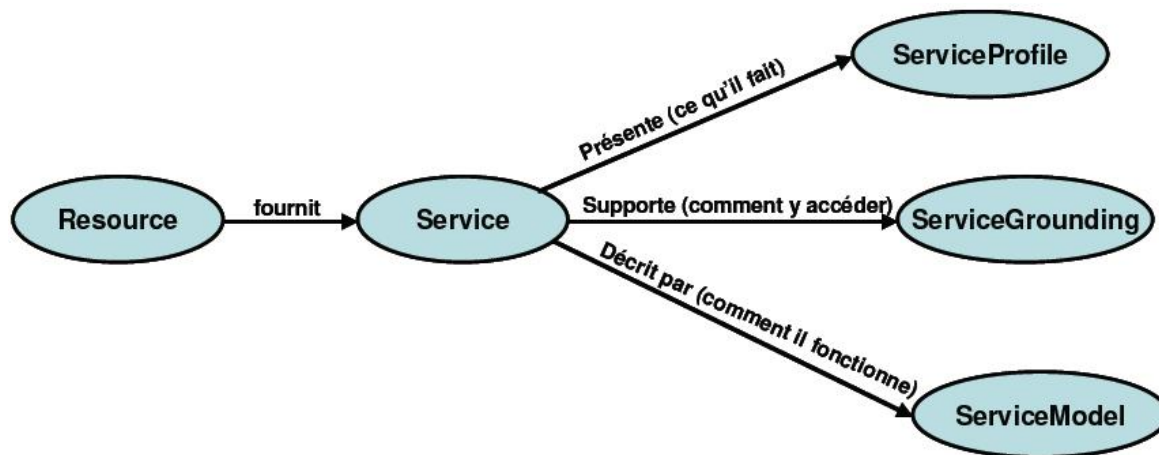


FIGURE 2.17 – L'ontologie supérieure d'OWL-S

La Figure 2.18 présente le processus du service (service de réservation de chambre d'hôtel). Nous l'avons simplifié pour plus de clarté ; Nous proposons ici qu'il soit composé de trois processus atomiques collectInfoHotel, selectHotel et Payment. Ce service de réservation de chambre d'hôtel prend en entrée un le nom de la ville, du pays, le nom de l'hôtel, la date d'en-



## 2.3. DESCRIPTION DE SERVICES WEB SÉMANTIQUES

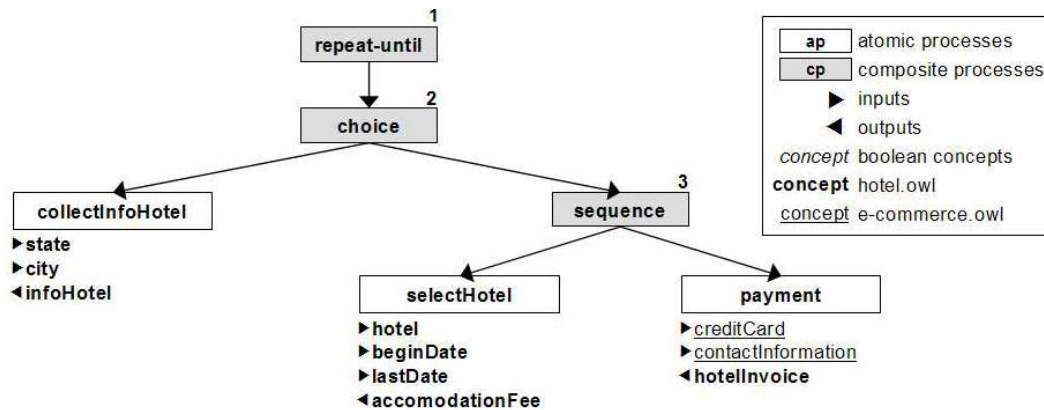


FIGURE 2.18 – Service OWL-S de réservation d’hôtel

trée et sortie ainsi que les informations concernant le consoammteur et fournit en sortie les informations pour s’y rendre à l’hôtel, le tarif de la chambre désiré ainsi que sa facturation. Ici ce processus composite contient un choix d’actions : soit on cherche des informations concernant les hotels se trouvant dans une ville spécifique (Le processus atomique CollectInfoHotel) ou bien l’on passe à la réservation d’une chambre d’hôtel directement (par le séquençement de deux processus atomiques, selectHotel suivi de Payment). Ce processus composite contient le constructeur de répétition dans le cas ou le consommateur n’a pas décidé de l’hôtel ou il souhaite séjourner, initialement il effectue des recherches puis passe à la réservation.

### 2.3.2.3 Synthèse & discussion : Le choix du OWL-S

En examinant l’ensemble des travaux, SAWSDL, USDL, OWL-S et WSMO qui proposent des mécanismes alternatifs permettant d’intégrer et d’utiliser des ontologies spécifiques de domaine.

SAWSDL [Kopecký *et al.* 2007] permet d’utiliser tous types d’ontologies (OWL, WSML et UML) pour annoter les modèles de descriptions existants avec des données sémantiques en utilisant un seul outil existant et propre à lui, le SAWSDL Tool Annotation (un outil pour l’annotation sémantique). L’avantage est qu’il est indépendant d’un langage d’ontologie en particulier et s’appuie sur RDF [Carroll & Klyne 2004, Hayes 2004, Lassila & Swick 1999, Manola & Miller 2004, Beckett 2009b] et XML [Maler *et al.* 2004, Bray *et al.* 2006]. Ces annotations sémantiques concernent les parties interface, opération, input et output ne couvrant pas la totalité des éléments. Celui-ci s’intéresse à la découverte et l’invocation automatique des services cependant il ne s’occupe pas de la composition.

Alors que l’approche USDL présente un inconvénient majeur puisqu’elle utilise exclusivement l’ontologie OWL WordNet et cela la classe dans le rang des approches ultra fermées. Cette approche nécessite la normalisation pour réaliser leurs descriptions du fait que sa construction fait intervenir beaucoup d’éléments et de codes descriptifs vu qu’elle est très verbeuse par rapport à WSDL.

On constate que WSMO et OWL-S sont des langages décrivant à la fois les services Web et leur sémantique dans une même structure.

(Absent - Mauvais - Bon)		OWL-S	WSMO
Définition - Description	<b>Intégration avec WSDL</b>	Bon, à travers le « service grounding »	Mauvais, pas encore complètement défini
	<b>L'ontologie principale illustre les aspects essentiels des services</b>	Bon, définit une ontologie avec quatre concepts de haut niveau (service, service profile, service model et service grounding)	Bon, définit un méta modèle basé sur une ontologie avec quatre composants (ontologies, web services, goals et mediators)
	<b>Le modèle supporte les ontologies externes</b>	Absent, tous les éléments nécessaires à la description d'un service sont fournis dans l'ontologie OWL-S. Aucun mécanisme d'extension n'est fourni	Bon, à tous les niveaux des « imported ontologies » peuvent être référencées et les médiateurs sont utilisés pour résoudre les conflits
	<b>Le modèle permet de définir des propriétés non fonctionnelles</b>	Mauvais, certains travaux complémentaires étudient les possibilités d'intégration des propriétés non fonctionnelles dans le « service profile »	Bon, chaque composant de WSMO a des propriétés non fonctionnelles.
	<b>Le modèle supporte la médiation</b>	Absent, owl-s ne supporte pas directement la médiation. Les médiateurs ne sont pas décrits par l'ontologie, ils font partie de l'infrastructure sous-jacente	Bon, quatre types de médiateurs sont décrits par WSMO
	<b>Le modèle garantit une indépendance entre les modules de l'ontologie principale</b>	Mauvais, l'ontologie est basée sur la « service class » de laquelle sont dérivés « service, service profile, service model et service grounding » mais ils ne peuvent pas être développés indépendamment	Bon, un des objectifs de WSMO est la modularité qui permet de définir les ressources séparément et indépendamment des interactions possibles
	<b>Le modèle décrit l'interaction avec d'autres services</b>	Bon, le « service process model » propose des processus composites abstraits mais aussi exécutables	Bon, l'interface d'un « WSMO web service » peut être représentée par une chorégraphie et son interaction avec d'autres services par une orchestration
	<b>Expressivité du langage sémantique et puissance de raisonnement logique</b>	Bon, OWL et SWRL (pour la logique de type Horn)	Bon, WSML supporte la logique descriptive et la F-Logic à travers cinq variantes : WSML-Core, WSML-DL, WSML-Flight, WSML-Rule et WSML-Full
Implémentation	<b>Prototypes et outils disponibles</b>	Une liste d'outils est disponible à l'adresse suivante : <a href="http://www.daml.org/services/owl-s/tools.html">http://www.daml.org/services/owl-s/tools.html</a>	Le « WSMO studio » est disponible à l'adresse suivante : <a href="http://www.wsmo.org/wsmo_tools.html">http://www.wsmo.org/wsmo_tools.html</a>
	<b>Contribution à la découverte et l'indexation de services</b>	Le « OWL-S service profile » a ouvert la voie à la nouvelle tendance de décrire sémantiquement les services Web et par suite à automatiser la découverte	WSMO différencie un service accompli d'un service web. La description des « capabilities » d'un service a marqué une nouvelle étape dans la description sémantique des services Web

FIGURE 2.19 – Comparaison entre OWL-S et WSMO

OWL-S [Burstein *et al.* 2004] accepte seulement les ontologies OWL facilitant l'adaptation des descriptions déjà développées par les fournisseurs aux annotations et aux solutions proposées. En termes de réutilisation et d'outillage, OWL-S tire partie de son ancienneté et ainsi il dispose de plusieurs outils reposant sur RDF [Carroll & Klyne 2004, Hayes 2004, Lasila & Swick 1999, Manola & Miller 2004, Beckett 2009b] et XML [Maler *et al.* 2004] allant du simple éditeur au composeur semi-automatique en passant par les outils d'appariement et de validation. Cette approche associe plusieurs éléments descriptifs au service lui même (modèle, version, profil, actor, etc). La description proposée par OWL-S permet une découverte basée sur la sémantique qui est faite grâce à des outils, implémentant des algorithmes d'appariement existants (voir section suivante).

WSMO [Bruijn *et al.* 2005] accepte uniquement les ontologies WSML et ne disposant que d'outils pour l'édition, WSML Editor qui s'avèrent plus difficiles à développer car celui-ci se base sur WSML. Cette approche ne lui associe qu'un élément dit "Goal" (Objectif) du service.

Vu la publication sur le web de jeux de données RDF liées les uns aux autres par l'utilisation de schémas et vocabulaires (ontologies en OWL) hétérogènes. Il convient de rappeler que parmi les approches examinées et en les évaluant par rapport aux exigences et préférences dans le domaine touristique : assurer une découverte et composition automatique et

dynamique, Nos objectifs sont totalement pris en compte par OWL-S.

	USDL	WSMO	OWL-S	SAWSDL	WSDL
<b>Service</b>	Service (identifier / name)	x	x	x	x
	Documentation / Description / Comment	x		x	x
	Type / Model / Category	x		x	
	Product / Version			x	
	Profile			x	
	Goal / Purpose		x		
	Actor			x	
<b>Interface / Operation</b>	Inputs (parameters)	x	x	x	x
	Outputs (parameters)	x	x	x	x
	Range / Occurrence (of parameters)			x	x
	Preconditions		x	x	
	Postcondition		x		
	Effects	x	x	x	
	Result			x	
	Condition	x		x	
	Output Constraints			x	
	Constraint	x			
	Assumption		x		
	Operation / Activity	x	x		x
	Interface	x	x		x
	Capability / Capacity		x		
Process			x		

FIGURE 2.20 – Récapitulatif des éléments de description dans les approches étudiées [Jorge 2006]

## 2.4 Conclusion

Dans ce chapitre, nous avons présenté l'état de l'art sur le web de données et les technologies sous-jacentes ainsi que de la nébuleuse des descriptions de Services Web Sémantiques afin d'identifier les objectifs par rapport aux contributions en termes de disponibilité d'information touristiques des descriptions de services Web en temps réel par rapport aux spécificités et l'expressivité de leurs langages de description et de leurs annotations sémantiques. Nous avons fait l'étude des différents modèles de description des services, il en ressort que les web services sémantiques basé sur OWL-S décrivent le comportement d'un service en précisant l'ordre d'invocation de ses opérations. L'objectif de notre thèse est d'intégrer les jeux de données disponibles étant accessible via un point d'accès SPARQL tout en intégrant les descriptions de service exprimé en OWL-S qui reposent sur l'utilisation de l'ontologie OWL servant de pivot pour instancier les données liées ouvertes d'une part et d'explicitier les des-

## CHAPITRE 2. ÉVOLUTION DU WEB

---

criptions sémantiques fonctionnelles en se référant à leur schémas d'autre part. Par la suite, nous aborderons uniquement les approches de découverte et composition des Services en OWL-S.

# 3

## Coordination de Services Web Sémantiques

### Sommaire

---

<b>3.1 Introduction</b> . . . . .	<b>61</b>
<b>3.2 Découverte de Services Web Sémantiques</b> . . . . .	<b>62</b>
3.2.1 Mise en correspondance de Services Web . . . . .	63
3.2.2 Les approches de la découverte sémantique . . . . .	66
3.2.3 Discussion . . . . .	73
<b>3.3 Composition des Services Web Sémantiques</b> . . . . .	<b>74</b>
3.3.1 Approches basées sur les automates et réseaux de pétri . . . . .	76
3.3.2 Approches basées sur les techniques de planification . . . . .	77
3.3.3 D'autres approches . . . . .	85
3.3.4 Discussion . . . . .	85
<b>3.4 Conclusion</b> . . . . .	<b>88</b>

---

### 3.1 Introduction

---

La PROGRAMMATION ORIENTÉE SERVICE [Papazoglou 2003] (SOC, Service Oriented Copmuting) est un paradigme de programmation utilisant les services comme constituants élémentaires à partir desquels sont réalisées des applications réparties défini sur une ARCHITECTURE ORIENTÉE SERVICE [Erl 2005](SOA, Service Oriented Architecture), une architecture orientée composant reposant sur trois éléments : Les services fournisseurs qui sont des services qui proposent des fonctionnalités accessibles par le maximum d'utilisateurs et utilisable par d'autres services. Ces services sont répertoriés dans un service particulier appelé annuaire. Dans cet annuaire se trouve l'adresse et la description des services fournisseurs répertoriés. Les utilisateurs des services fournisseurs sont appelés services clients. ARCHITECTURE ORIENTÉE SERVICE [Erl 2005] permet d'organiser et d'utiliser des capacités distribuées qui peuvent être sous le contrôle de plusieurs domaines de propriété. Ainsi la ARCHITECTURE ORIENTÉE SERVICE [Erl 2005] fournit une puissante architecture pour l'appariement de besoins (requêtes) et d'offres (ou capacités-Service Web) très souvent combinées pour répondre à ces besoins. L'impact potentiel envisagé par l'apport de la sémantique aux SERVICES WEB [Haas & Brown 2004] pour les applications commerciales des entreprises, couvre d'une part l'intégration sémantique en B2B assurant le partage des données et services entre entreprises partenaires et d'autres part, l'Interopérabilité sémantique offrant un échange flexible et

dynamique de données et de services entre Système d'informations d'entreprises par le biais de métadonnées et ontologies développant ainsi des systèmes d'interrogation de données, de découverte et d'intégration de services automatisée et très fiable.

Le développement et le déploiement de SERVICES WEB SÉMANTIQUES [Jorge 2006, McIlraith *et al.* 2001] à base de services ARCHITECTURE ORIENTÉE SERVICE [Erl 2005] permet d'offrir les fonctionnalités de base suivantes : description, découverte, adaptation, sélection, médiation, composition, chorégraphie, orchestration et exécution de services Web.

- La découverte de services Web consiste à faire une recherche spécialisée sur la base de la description sémantique de leur interface. Ainsi le moteur de recherche en ressources de type " services web " va extraire de l'ensemble de l'offre de services Web publiés, les services qui répondent " le mieux " aux besoins de cette recherche en tenant compte de contraintes de typage sémantique (e.g. compatibilité des données échangées) mais aussi de contraintes de contextes d'usage plus globaux (e.g, Qualité de Services (QoS), contexte d'utilisation du service, stratégie de l'utilisateur, politique privée d'usage). Trier les services web les plus pertinents revient à filtrer les services découverts au moyen de conditions et de contraintes d'utilisation, permettant d'établir un classement de ces services en fonction de leur adéquation aux besoins, nommé, la phase de sélection de services Web.
- La Composition de services permet de créer de nouveaux services à partir de services existants dans le but d'obtenir une nouvelle fonctionnalité combinant deux aspects i.e. (i) fonctionnel en prenant en compte les contraintes liées au typage sémantique de la nature des messages à échanger et (ii) comportemental lié aux interactions dialogiques des services à assembler.

Comme on l'a démontré au premier chapitre, OWL-S est un langage assez riche et expressif dont le Process Model fournit des descriptions sémantiques concises et non ambiguës par rapport aux services profil (des services atomiques). De là, dans cette thèse et dans le but d'assurer une découverte orienté composition automatique la nécessité d'exploiter uniquement le Process Model d'OWL-S est primordial.

Une pléthore d'approches de composition de services existe qui mettent en évidence que, quel que soit l'approche développée, une tâche centrale doit être fréquemment effectué tout au long du processus de composition, c'est la découverte de services appropriés à combiner. Pourtant, en dépit de la forte dépendance entre eux, la recherche et le développement dans ces deux activités a évolué pour la plupart indépendamment. Ce chapitre contient deux parties, une détaillera les approches de la phase de découverte sémantique et l'autre explicitera les approches de compositions sémantiques existantes. On abordera quelques approches combinant les deux.

### 3.2 Découverte de Services Web Sémantiques

---

Etant donné le nombre croissant de Services Web disponibles et l'évolution constante des besoins/contraintes des utilisateurs, l'identification d'un ou plusieurs service(s) particulier(s) est devenue une tâche complexe. Dans ce contexte, diverses approches ont été proposées pour la découverte des Services Web Sémantiques formalisé en OWL-S [Burstein *et al.* 2004] . Dans cette partie, nous comparons ces différentes classes d'approches de découverte afin d'évaluer

leur maturité et de leur adaptation aux exigences actuelles du domaine. L'évaluation comparative nous aidera à identifier les mécanismes qui constituent les points forts des approches existantes, et par la suite servira de directive pour déterminer la base d'une approche de découverte plus adaptée au contexte actuel des services Web.

### 3.2.1 Mise en correspondance de Services Web

Communément, les procédures de découverte de SWS sont basées sur la recherche de mots clés et sont aussi guidés par intervention manuelle. En réponse aux besoins identifiés dans sa requête, le client reçoit une liste de descriptions de WEB SÉMANTIQUE [Berners-Lee 1998, Berners-Lee *et al.* 2001, Antoniou & Harmelen 2008, Hendler 2008] qui devraient être analysées manuellement pour sélectionner les services qui répondent exactement à ses besoins. Cependant, dans un environnement conçu pour une intégration dynamique des systèmes distribués, une recherche rapide, automatique et sémantique pour les SERVICES WEB [Haas & Brown 2004] composables est fortement recommandée. La découverte automatique des SERVICES WEB SÉMANTIQUES [Jorge 2006, McIlraith *et al.* 2001] cherche à identifier les services qui peuvent répondre à une requête donnée en effectuant une correspondance des éléments de requête avec ceux correspondants dans les descriptions de services stockées dans un registre. Ce mécanisme consiste généralement à identifier un degré de similitude entre les concepts sémantiques décrivant le service requis (requête) et ceux correspondants dans les services fournis (services publiés).

Dans le paradigme PROGRAMMATION ORIENTÉE SERVICE [Papazoglou 2003] (Service Oriented Computing), la recherche de similitude est employée dans trois activités principales du cycle de vie des services web, la découverte, la composition et la publication :

- Pendant la publication, les services Web peuvent être organisés selon des critères de similitude. L'organisation qui découle de ce processus de classification peut être ensuite utilisée pour améliorer la découverte, ou à des fins de substitution pour détecter les services similaires. La substitution intervient pour le remplacement d'un service par un autre service équivalent pour des raisons d'indisponibilité ou d'insatisfaction du service proposé. La substitution peut être un processus automatique mis en œuvre pendant l'exécution de la composition. Elle peut aussi prendre la forme d'une recommandation.
- La découverte exploite les informations liées aux services Web dans le but de localiser des services capables de répondre à une requête particulière avec la meilleure adéquation possible en se basant sur la mise en correspondance. Dans le processus de découverte, elle consiste à rechercher les similitudes potentielles entre la requête service et les services Web offerts à travers la comparaison des propriétés de la requête avec les propriétés des services Web disponibles. Pour cela, les services recherchés doivent produire les mêmes buts que le service demandé et exiger les mêmes entrées.
- Dans le processus de composition, une fonction de mise en correspondance intervient pour mettre à jour des similitudes entre deux services qui vont s'enchaîner. Elle

s'applique à déterminer la compatibilité entre les buts d'un premier service et les entrées d'un second service.

L'aspect central de la mise en correspondance de services Web est de trouver la notion de similitude la plus pertinente. De ce fait, la façon dont la similitude est définie apparaît comme un élément fondamental pour déterminer la façon dont les services correspondent à une requête, comment ils peuvent être composés ou encore comment ils peuvent être classifiés. Les mesures de similitude ont été largement utilisées dans plusieurs domaines. Nous pouvons notamment citer les systèmes d'information, la science cognitive, les bases de données, le génie logiciel et l'intelligence artificielle.

La recherche de similitude s'opère par le biais d'une fonction de mise en correspondance entre un objet donné, l'objet de la requête, et un ensemble d'objets d'une collection dans le but de récupérer ceux qui lui ressemblent le plus. Considérons deux objets  $x$  et  $y$ , deux types de correspondances peuvent être envisagés entre ces deux objets, la correspondance syntaxique et la correspondance sémantique.

### 3.2.1.1 Correspondance syntaxique :

La mise en correspondance syntaxique vise à appairer des paramètres à partir de leurs orthographes respectives. On peut distinguer la correspondance égale et la correspondance approximative.

- La correspondance égale ( $=$ ) utilise l'équivalence syntaxique stricte. Deux objets sont dits similaires si et seulement si ils possèdent une orthographe identique. Cependant, deux objets ayant des représentations sensiblement différentes, comme GOVERNEMENT et GOVERNMENT ne peuvent pas être appariés par ce type de correspondance stricte.
- La correspondance approximative ( $\approx$ ) utilise des fonctions de distance  $d(x, y)$  pour quantifier la similitude entre deux chaînes de caractères  $x$  et  $y$ . Si la distance entre deux objets est au dessus d'un certain seuil, ces objets sont dits similaires. La mise en correspondance approximative est plus flexible que la mise en correspondance égale. En effet, deux termes avec des orthographes différentes peuvent avoir la même sémantique et par conséquent être interchangeables. Les vocabulaires utilisés peuvent comporter des erreurs typographiques comme par exemple dans les deux noms de paramètres GOVERNEMENT et GOVERNMENT, comporter des abréviations comme par exemple PhysicianPassword et PhysicianPwd. Ce n'est cependant pas suffisant pour détecter que deux termes comme PRIX et TARIF peuvent être interchangeables. Ce problème est résolu avec le deuxième type de mise correspondance grâce à la comparaison des concepts ontologiques.

### 3.2.1.2 Correspondance sémantique en utilisant des ontologies

La mise en correspondance sémantique vise à appairer des concepts organisés dans une relation hiérarchique de subsomption. On peut distinguer deux types d'approches pour évaluer la similitude sémantique entre concepts dans une ontologie : les approches basées sur la



distance, *i.e.*, sur la structure de l'ontologie, et les approches utilisant le contenu informatif des concepts. Dans le premier cas, la similitude est évaluée par la distance qui sépare les concepts dans l'ontologie. Dans le second cas, on associe aux concepts l'information véhiculée par ce concept au sens de la théorie de l'information. La similitude entre deux concepts est alors mesurée par la quantité d'information qu'ils partagent.

Pour comparer les entrées (Inputs) et les sorties (Outputs) d'une requête aux sorties d'un service publié, quatre degrés de mise en correspondance sont utilisés. Ces quatre degrés de mise en correspondance sont nommés exact, plug-in, subsumes et fail. Nous en donnons la signification ci-après en notant pour des questions de commodité  $C_R$  le concept associé à la requête et  $C_S$  le concept associé au service publié.

- La correspondance exact ( $\equiv$ ) comporte deux clauses. La première clause est l'équivalence des concepts. Si  $C_R$  et  $C_S$  sont deux concepts équivalents, alors la correspondance est dite exact. La deuxième clause est une relation de subsomption. Si  $C_R$  est une sous classe de  $C_S$ , alors les deux concepts sont également dits en correspondance exacte. Cela suppose qu'en publiant  $C_S$ , le fournisseur s'engage à fournir des sorties compatibles avec la sous classe immédiate de  $C_S$ .
- La correspondance plug-in ( $\sqsubseteq$ ) correspond à la situation où le concept du service est une généralisation du concept de la requête. Autrement dit si  $C_S$  subsume  $C_R$ , on parle de correspondance plug-in. Dans ce cas,  $C_S$  est un ensemble qui généralise  $C_R$ . Dans ce cas, la relation entre  $C_R$  et  $C_S$  est plus fragile que dans le deuxième cas de figure de la relation exact.
- La correspondance subsumes ( $\sqsupseteq$ ) correspond à la situation où le concept du service est une spécialisation du concept de la requête. Autrement dit si  $C_R$  subsume  $C_S$ , on parle de correspondance subsumes. Dans ce cas, le service ne satisfait pas complètement la requête. Ce service peut être utilisé pour atteindre partiellement le but de la requête. Un ou plusieurs services supplémentaires devront éventuellement être utilisés pour satisfaire l'intégralité des buts de l'utilisateur.
- Le cas fail ( $\perp$ ) traduit le fait qu'il n'existe aucune relation de subsomption entre  $C_R$  et  $C_S$ . Il n'existe par exemple aucune relation de subsomption entre roman de fiction et roman historique.

En terme de satisfaction de la requête, ces quatre degrés de mise en correspondance sémantique peuvent être ordonnés selon une échelle de préférence comme suit : Exact  $\succ$  Plug-in  $\succ$  Subsumes  $\succ$  Fail.

La mesure de similitude joue un rôle important dans la recherche de services pour la découverte et la composition ainsi que dans la classification. Pour la découverte, la mise en correspondance peut se faire sur des informations textuelles et sur les paramètres en entrée et les paramètres en sortie par une mise en correspondance verticale. Pour la composition, la mise en correspondance est horizontale ; elle s'applique sur les paramètres en sortie du premier service comparé et sur les paramètres en entrée du deuxième service comparé. Par ailleurs, selon la nature des objets à traiter, la mise en correspondance peut se décliner selon une mise en correspondance syntaxique ou une mise en correspondance sémantique. Notons néanmoins que la mise en correspondance sémantique peut être utilisée sur des descriptions syntaxiques en enrichissant les descriptions pour le traitement. Diverses solutions ont été proposées dans la littérature telle que l'utilisation d'outils comme la base de données lexicale

WordNet [Miller 1995] ou des méthodes comme l'analyse sémantique latente [Ataa-Allah et al. 2008].

### 3.2.2 Les approches de la découverte sémantique

La découverte de SERVICES WEB SÉMANTIQUES [Jorge 2006, McIlraith et al. 2001] est un processus qui consiste à identifier automatiquement les descriptions sémantiques des services Web offerts (préalablement inconnus) et correspondant à certains critères répondant à la requête utilisateur. Ce processus prend en entrée une requête (service) utilisateur et fournit en sortie une liste de Services Web Sémantiques en utilisant un mécanisme de comparaison automatique entre la requête et les descriptions sémantiques des Services Web publiés. Les résultats issus du mécanisme de comparaison sont triés selon les besoins de l'utilisateur.

"Web service discovery is the act of locating a machine-processable description of a Web service that may have been previously unknown and that meets certain functional criteria. It involves matching a set of criteria with a set of Web service descriptions. The goal is to find an appropriate Web service." W3C definition

Le principe de la découverte des services est d'identifier les services publiés dans un annuaire qui peuvent répondre à un besoin présenté sous forme d'un service requête. Le processus de découverte identifie un degré de similitude entre les différents concepts sémantiques qui décrivent le service requête (requis) et celles des services offerts ou publiés. Ce mécanisme d'évaluation pour la découverte est appelé : Matching (Appariement). Dans la littérature [Trastour et al. 2001], les approches de découverte de services Web sont classées selon les critères suivants, Premièrement, l'aspect traitement des données qui indique le degré d'automatisation de la découverte. Deuxièmes, l'aspect mise en correspondance ou "matching" qui compare la requête utilisateur aux services répertoriés :

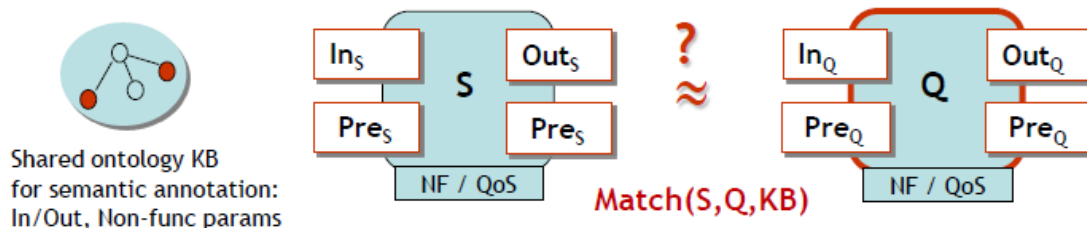


FIGURE 3.1 – Mise en correspondance de Services Web

- Les approches pour la mise en correspondance dépendent de l'information considérée, *i.e.*, de la partie de la description utilisée. Une première catégorie d'approches se concentre sur l'aspect dynamique, *i.e.*, les processus. Une deuxième catégorie prend en compte les aspects non fonctionnels comme la qualité de service, des informations générales comme la catégorie, le fournisseur, la description textuelle des services. Enfin une troisième catégorie s'intéresse aux aspects fonctionnels en considérant les interfaces avec le nom des opérations, les noms, types et concepts des paramètres. On distingue principalement trois sous-catégories de correspondances pour cette catégorie. La première catégorie concerne la mise en correspondance des entrées et sorties

(appelé la signature). La deuxième catégorie concerne la mise en correspondance des pré-conditions et effets. La troisième catégorie concerne à la fois les entrées et sorties et les pré-conditions et effets.

- L'algorithme d'appariement employé, en trois catégories : les approches logiques (mise en correspondance sémantique), les approches non-logiques (mise en correspondance syntaxique) et les approches hybrides en fonction des mises en correspondance utilisés définis dans la section suivante. Notons que des approches hybrides ont été proposées. Dans le cadre des services Web, la similitude de concept s'apparente aux approches basées sur la distance.

Cependant, il y a l'aspect localisation des services qui consiste à trouver l'adresse à laquelle est fournie la description d'un service. La découverte de services consiste, en premier, à localiser les fournisseurs proposant des services répondant à une requête utilisateur. Les approches de localisation de services récurrentes dans la littérature sont classées en deux catégories, à savoir les approches centralisées et les approches distribuées.

La S3 contest editions <sup>1</sup> [Klusch 2012] a reçu 14 outils d'appariement OWL-S de 2007 à 2013 <sup>2</sup> (voir Figure 3.2), ces outils :

ALIVE (University of Bath, UK), OWLS-iMatcher1 and OWLS-iMatcher2 (University of Zurich, Switzerland), Opossum (Technion, Israel), SPARQLent (Hewlett-Packard Research, Italy), XSSD (Beihang University, China), EMMA (University of Seville, Spain), JIAC-OWLSM and SeMa2 (Berlin University of Technology, Germany), OWLS-SLR lite (Aristotle University of Thessaloniki, Greece), et OWLS-MX1, OWLS-MX2, OWLS-MX3 and iSeM (DFKI, Germany). La plupart des Matchmaker existants dans la littérature effectue une découverte sémantique hybride notamment (SeMa2, OWLS-SLR Lite, XSSD, Opossum, ALIVE [Vázquez-Salceda *et al.* 2010]). Par contre, les matchmaker OWLS-MX3, ISEM et OWLS-iMatcher2 effectuent une sélection de service sémantique hybride et adaptative, qui appliquent des techniques d'apprentissage machine afin de déterminer les services les plus pertinents aux ceux demandés tandis que seulement trois d'entre eux se concentrent la sélection sémantique non logique (OWLS-iMatcher) et logique (EMMA, SPARQLent) auxquels se rajoutent les prédécesseurs.

Nous identifions un peu plus loin dans ce chapitre nos propres critères d'examen et de comparaison des approches de découverte dans cet état de l'art. Nous n'évoquons cette classification que pour la présentation des approches afin que le lecteur puisse retrouver des repères clairs tout au long de la première partie du chapitre.

### 3.2.2.1 Les approches non-logiques

La découverte basée sur l'appariement non logique, trouve ses origines dans les mécanismes de recherche d'information se basant souvent sur des hypothèses et font face à leurs limites dès que les descriptions de services ne respectent pas les hypothèses posées. Les approches de découverte non logiques exploitent la sémantique implicite plutôt que la sémantique explicite, de ce fait l'appariement non logique repose sur des mécanismes syntaxiques, structurels et numériques issus des mécanismes de recherche d'information... Pour assurer cela, de tels

---

1. [www-ags.dfki.uni-sb.de/~klusch/s3/](http://www-ags.dfki.uni-sb.de/~klusch/s3/)

2. Dernière conférence en date

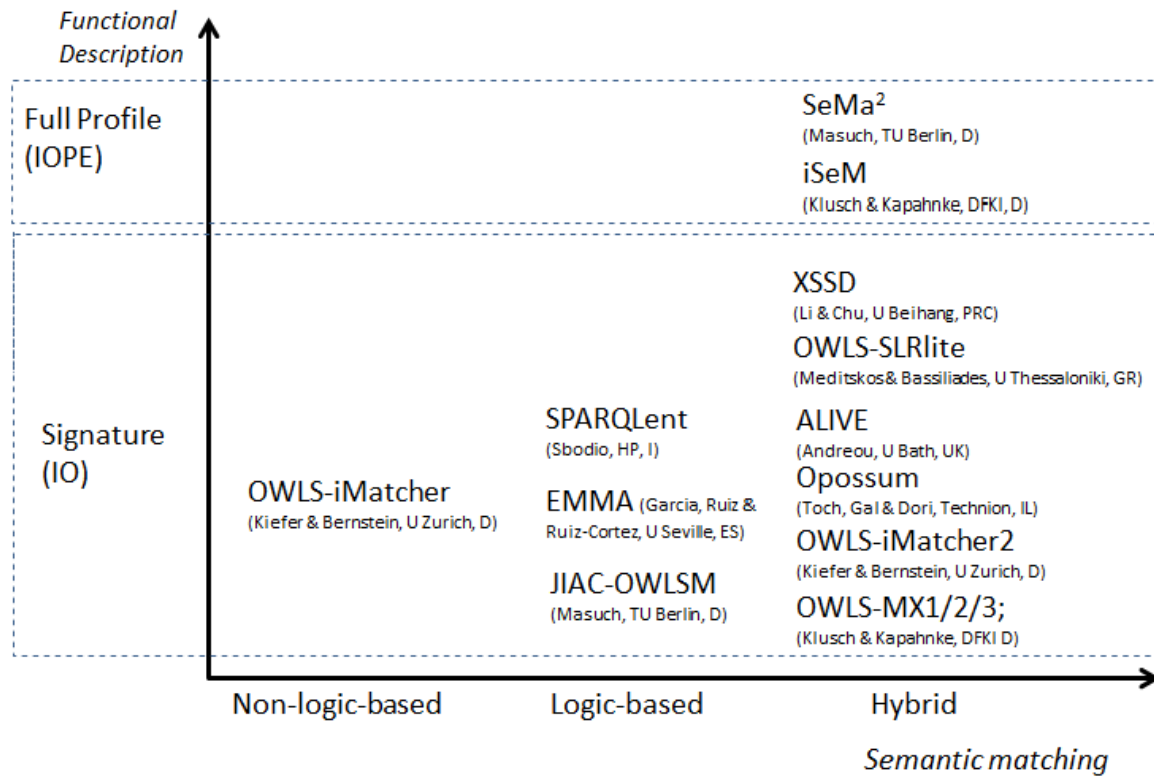


FIGURE 3.2 – Classification des approches pour la mise en correspondance [Klusch 2012]

mécanismes d'appariement utilisent les fréquences des termes, la distance numérique entre concepts, les appariements de graphes structurés, sous-graphes, etc.

OWLS-iMatcher1 [Kiefer *et al.* 2007b] est un matchmaker syntaxique basé sur la signature du service (entrée/sortie). Ces derniers sont stockés sous forme de graphes RDF sérialisés dans une base de données RDF interrogeable avec l'extension de RDQL [Seaborne 2004], appelée iRDQL [Bernstein & Kiefer 2005]. Le calcul du degré de correspondance est effectué à partir de quatre métriques de similarité syntaxique issues de domaine de la recherche d'information [Moreau *et al.* 2008] : TFIDF (Term Frequency-Inverse Document Frequency), la distance de similarité de Levenshtein, la mesure du vecteur Cosinus, la mesure de divergence de Jensen-Shannon. Les scores numériques issus de ces mesures de similarités syntaxiques sont classés en fonction d'un seuil prédéfini par l'utilisateur. OWLS-iMatcher a été développé par Christoph Kiefer et Abraham Bernstein (Université de Zurich, Suisse) et a réalisé sa meilleure précision. Le successeur hybride d'iMatcher1, est iMatcher2 qui sera présenté plus tard.

### 3.2.2.2 Les approches logiques

Ces approches de découverte utilisent des descriptions sémantiques de services et des requêtes spécifiées en langages basés sur des formalismes logiques, telles que la logique de description, la logique de premier ordre, etc. Cette catégorie utilise des concepts d'ontologies et des règles logiques. Les degrés d'appariement sont déterminés de différentes façons et en fonction de la sémantique des éléments des descriptions à apparier. Il existe principalement

trois approches d'appariement :

- La signature (IO-matching) : Ce type d'appariement est déterminé à partir des données sémantiques des paramètres de service : les entrées : inputs (I) et les sorties : outputs (O).
- La spécification (PE-matching) : Ce type d'appariement est déterminé à partir d'appariements sur des préconditions (P) et des effets (E) des services et des requêtes.
- service profile intégrale (IOPE-matching) : Ce type d'appariement est déterminé à partir d'appariements sur les données sémantiques des inputs (I), des outputs (O), des préconditions (P) et des effets (E) des services et des requêtes.

Le matching basé sur la similarité consiste à établir des correspondances entre deux ou plusieurs services en calculant des valeurs de similarité indiquant le degré de rapprochement entre leurs descriptions respectives. Le calcul de similarité peut être basé sur des données syntaxiques ou sémantiques plus expressives. Dans la suite nous résumons les approches décrites dans la littérature.

Dans [Pilioura *et al.* 2003], les auteurs étendent le modèle UDDI pour prendre en compte des descriptions OWL-S. L'objectif est de permettre le stockage des fonctionnalités des services désignées par capacités afin de rendre la découverte plus précise. Bien que UDDI permet une grande variété de recherches par catégories (par nom, localisation, métier ou autres), UDDI est limité à la recherche par mots clés, ne permettant aucun raisonnement ni inférence par rapport à des taxonomies sur la possibilité de correspondance entre une description et une requête utilisateur. Ils présentent un composant, qui assure le raisonnement logique nécessaire pour inférer la correspondance entre le profil d'un service publié par UDDI et le profil d'un besoin décrivant la requête utilisateur. Ils décrivent une implémentation basée sur jUDDI3 et le raisonneur Racer [Haarslev & Möller 2001]. Quelques résultats préliminaires sont proposés. Cette approche bien que riche sémantiquement ne traite pas l'aspect automatisation de la découverte et repose sur un UDDI centralisé. Dès lors, [Li & Horrocks 2003] étendent l'algorithme de [Pilioura *et al.* 2003] en insérant un cinquième degré de correspondance intitulé intersection, ce dernier permet de repérer les services ayant des concepts chevauchant ceux de la requête. Dans [Srinivasan *et al.* 2006], ils présentent une approche distribuée pour la découverte de services qui sont supposés être décrits par des service profils de l'ontologie OWL-S. Le protocole pair-à-pair adopté pour la recherche d'information est le Gnutella. En combinant ces deux modèles, OWL-S et Gnutella, les auteurs estiment pouvoir allier la précision de description de l'approche OWL-S et l'efficacité de la technique de recherche et de localisation Gnutella. Les détails de l'implémentation ne sont pas discutés.

[Benatallah *et al.* 2005] proposent de formaliser la découverte comme un problème de réécriture de concepts en logique descriptive. Ils décrivent un algorithme pour découvrir la meilleure couverture sémantique d'une requête par les concepts d'une ontologie donnée. Ils démontrent que le problème de découverte de la meilleure couverture sémantique présente des similitudes avec le problème de calcul, à coût minimal, des transversales minimales d'un hypergraphe attribué (computing the minimal transversals with minimum cost of a weighted hypergraph). Les résultats expérimentaux sont présentés dans [Benatallah *et al.* 2005]. Les auteurs affirment que l'approche décrite n'est pas encore adaptée pour traiter un grand nombre d'ontologies hétérogènes, néanmoins les premières expériences présentées sont prometteuses.

Dans [Mandell & McIlraith 2003], les auteurs réutilisent les technologies service existantes et les combine avec celle du web sémantique en étendant BPEL [Juric 2010]. Dans le document BPEL, une description OWL-S de type service profile remplace la référence statique aux services impliqués. Ils mettent en place un service de découverte sémantique (SDS) décrit par un service profile OWL-S. SDS utilise le raisonneur JTP (Java Theorem Prover) pour générer des requêtes DQL. Au cours de l'interprétation du document BPEL, le service SDS intervient et envoie des requêtes DQL générées à partir des services profile du process BPEL vers un annuaire contenant des services sémantiquement décrits en OWL-S. De cette façon chaque service sera découvert juste avant son invocation durant l'exécution du processus composite.

Dans [Sabou & Pan 2005], les auteurs étudient les lacunes des systèmes de découverte de services proposés et présentent les annuaires disponibles en ligne. Ils concluent que la recherche de services dans ces annuaires n'est pas suffisamment précise car elle consiste à faire correspondre les mots clés d'une requête avec une classification des services publiés. Ils ajoutent qu'il est difficile de maintenir à jour les classifications des services car ils évoluent rapidement. Ils proposent alors les bases d'une approche semi-automatique pour la découverte de services qui repose sur des techniques empruntées au web sémantique. Elle consiste, d'abord, à adapter les techniques d'apprentissage à base d'ontologies pour permettre une mise à jour automatique des schémas de classification des services. Des ontologies seront générées à partir de ces schémas. Un utilisateur "expert" peut intervenir. A ce niveau pour vérifier la cohérence des ontologies générées. Elle consiste, ensuite, à réutiliser les techniques de matching entre ontologies pour comparer ces schémas et les intégrer en méta-schémas. Elle utilise, finalement, des méthodes de visualisation pour représenter les services par rapport aux méta-schémas générés.

Dans [Jaeger *et al.* 2005] Offrent un appariement IOPE-matching qui recherche les correspondances sémantiques entre les paramètres fonctionnels définis dans les descriptions OWL-S et les paramètres introduits dans la requête. Le processus d'appariement comporte quatre tâches : entrées correspondants, sorties correspondants, les catégories de services correspondant et un quatrième tâche au cours de laquelle les contraintes et les caractéristiques prédéfinies sont appliquées par l'utilisateur. L'algorithme calcule le degré de correspondance dans chacune des trois premières tâches et effectue le quatrième enfin agréger les résultats et de retourner le rang des WS appariés si elle est choisie comme un service qui répond à la requête. Ce cours de classement le service en fonction de son degré de correspondance entre tous WS découvert.

Dans [Stollberg *et al.* 2005], les auteurs proposent un modèle de découverte sémantique et automatiques appelée SLA (localisation automatique des Services). Ce modèle utilise des objectifs et des abstractions de services prédéfinis pour finalement découvrir les services concrets répondant à une requête. L'approche proposée distingue cinq niveaux de diplômes correspondants appelés degrés de correspondance intentionnelle (Match, parMatch, PossMatch, No-Match et PossParMatch). GR (Graded pertinences) est une extension de la SLA en termes de degrés correspondant. GR ajoute deux autres degrés de correspondance : RelationMatch et ExcessMatch.

CASD (Context Aware de Service Discovery) [Doukeridis *et al.* 2005] est une approche de découverte orienté contexte qui utilise une ontologie de domaine permettant de filtrer sémantiquement les services en fonction de la requête utilisateur. Pour cela, l'utilisateur fournit

## 3.2. DÉCOUVERTE DE SERVICES WEB SÉMANTIQUES

une requête (Q) en termes de mots-clés nécessaire à la recherche et exprimé en SPARQL. SPARQL (SPARQL Protocole et RDF Query Language) est un langage d'interrogation compatible avec les ontologies OWL. L'architecture CASD génère une nouvelle requête enrichie (Q') servant de filtre à la sélection de services et répondant au contexte de l'utilisateur. L'approche Schulzrinne et Arabshian propose une version distribuée de CASD appelé DCAASD (Contexte Distributed Service Aware Discovery Agent pour l'architecture).

SPARQLent [Sbodio 2012] est un matchmaker basé sur le profil complet du service qui se focalise sur les préconditions et les effets principalement et sont formalisés en graphe SPARQL. Le calcul de similarité se base sur le confinement de la requête en appliquant l'entailment RDF sur les concepts d'Entrées/Sorties à base d'ontologies OWL. SPARQLent a été développé par Marco Luca Sbodio (Hewlett-Packard CPN, Italie) et atteint une précision moyenne élevée avec notamment un faible temps de réponse moyen.

Le matchmaker EMMA [García *et al.* 2012] procède par contre à un pré-filtrage en proposant deux filtres exprimés en requêtes SPARQL concrètes le premier filtre cherchant uniquement le web service qui satisfait toutes les exigences de la requête par contre le deuxième filtre permet de rechercher un ensemble de services répondant à la requête de l'utilisateur en vue d'une composition future permettant ainsi de réduire l'espace de recherche et optimisant ainsi le temps d'exécution logique de services avant d'appeler un matchmaker externe notamment le OWLS-MX.

### 3.2.2.3 Approches déductive/Hybrides (Logiques et Non Logiques)

L'approche hybride combine les deux mécanismes à savoir les méthodes déductives tout en intégrant le calcul de la distance. L'idée est de surmonter les limitations de chacun de ces deux mécanismes. Plusieurs études travaux optent pour cette approche. Cette catégorie utilise les concepts des ontologies ainsi que des règles logiques. Les degrés d'appariement sont déterminés de différentes façons et en fonction de la sémantique des éléments des descriptions à appairer. Il existe principalement trois approches d'appariement : L'idée est de remédier à certaines limites de chacun de ces deux mécanismes grâce à différentes combinaisons hybrides qui réussissent là où chacun de ces deux mécanismes échoue.

[Benatallah *et al.* 2005] propose une approche hybride basée sur les logiques de description (LD). Le processus de découverte consiste à rechercher la meilleure couverture de la description d'un concept en utilisant une terminologie (ontologie). Pour chaque description Q et une terminologie T, il recherche une description E, qui couvre la plupart Q par l'utilisation de T. Pour ce faire, l'auteur utilise le travail de Teege qui propose une classe de langages pour lesquelles la différence entre les concepts peuvent être calculés de la même manière que la différence entre les séries au moyen de clauses de descriptions de formulaire. Il choisit un langage de représentation de cette classe de modéliser WS sur lequel il applique ce raisonnement. Pour trouver la meilleure couverture de concept en utilisant une terminologie, l'auteur utilise un algorithme dérivé de la théorie Hyper-Graph. L'algorithme effectue une recherche qui est équivalente à la recherche transversale minimale avec un coût minimal dans un Hypergraphe où les sommets sont l'envers et les bords sont les clauses de requête.

OWLS-MX [Klusch *et al.* 2006] est une approche hybride de l'appariement de signature des services profils basée sur le IOPE-matching. Partant d'un appariement entre deux profils

de service spécifiés en OWL-S, le processus d'appariement prend en compte les inputs, les outputs, la catégorie de service et des éléments prédéfinis par l'utilisateur qui comporte quatre tâches : l'appariement des inputs, l'appariement des outputs, et l'appariement de la catégorie du service. L'algorithme calcule dans chaque tâche le degré d'appariement. Une quatrième tâche pendant laquelle des contraintes et des fonctionnalités prédéfinies par l'utilisateur s'appliquent avant que les résultats soient agrégés afin de retourner le résultat final. Le résultat est soit un échec s'il y a échec d'appariement dans l'une des précédentes tâches, soit une valeur numérique (dite RANK ) déterminée par l'agrégation des degrés d'appariements de chaque tâche ,et représentant le rang du service Web apparié dans l'ensemble des services offerts. Pour évaluer l'appariement, les auteurs proposent quatre types de degrés d'appariement avec quelques différences d'interprétation d'un élément du profil à un autre :

- Degrés d'appariement pour les inputs / outputs :
  - FAIL : si au moins un input / output requis n'a pas été apparié,
  - UNKNOWN : si l'algorithme d'appariement ne peut pas catégoriser un input / output,
  - SUBSUMES : si l'input / l'output offert est plus spécifique que l'input / l'output requis,
  - EQUIVALENT : en cas d'équivalence entre inputs / outputs.
- Degrés d'appariement pour les catégories de service :
  - FAIL : si les concepts des catégories de service ne s'apparient pas,
  - UNKNOWN : si l'algorithme d'appariement ne peut pas catégoriser l'un des services,
  - SUBSUMES : si le service offert est plus spécifique que le service requis,
  - EQUIVALENT : en cas d'équivalence entre catégories de service.

OWLS-MX3 adopte la notion d'appariement sur la base de quatre filtres de ses prédécesseurs OWLS-MX2 et les OWLS-MX1, et applique quatre autres textes de similarité. Il calcule les scores numériques pour les similarités structurelles entre concepts. L'agrégation pondérée optimale basé sur le SVM pour les décisions de pertinence en utilisant une base logique de classificateur. OWLS-MX3 a été développé par Matthias Klusch et Patrick Kapahnke (DFKI, Allemagne).

Dans la FC-Match (fonctionnelle Comparaison), les concepts de services et de requête pour être appariés sont présentés dans OWL-DL (Web Ontology Language logiques de description). Les concepts utilisés pour décrire un service fournissent des informations sur les éléments : Catégorie, fonctionnement, entrée et sortie. Cette approche opte également pour un appariement sémantique déductif et une correspondance algébrique basée sur le calcul de la similarité syntaxique. Le calcul de la correspondance de la valeur totale est réalisé en additionnant le résultat de la mise en correspondance basée sur des concepts subsumption logique entre le service et la demande de recherche et la valeur du coefficient de la similarité syntaxique.

Le matchmaker Opossum [Kuster *et al.* 2008] est un matchmaker sémantique hybride qui combine un appariement sémantique non logique et logique sur les concepts d'entrées et sorties du service. Le score numérique de l'appariement basé sur le calcul de similarité logique des concepts qui est combiné avec les valeurs structurels du plus court chemin (distance) entre les concepts ainsi que de leur profondeur (moy. De la profondeur de l'ontologie) pour



## 3.2. DÉCOUVERTE DE SERVICES WEB SÉMANTIQUES

le classement ultérieur. Opossum a été développé par Eran Toch (CMU, USA), Avigdor Gal, Dov Dori (Technion, Israël) et Iris Reinhartz-Berger (Université de Haïfa, Israël).

Le matchmaker OWLS-SLRlite [Meditkos & Bassiliades 2010] est similaire à celui de XSSD mais il calcule les scores de correspondance non-logiques sur les annotations des concepts d'E/S agrégées avec les distances entre ces concepts dans les ontologies de services respectifs. OWLS-SLR a été mis au point par Georgios Meditskos et Nick Bassiliades (U Thessalonique, Grèce) et a réalisé un relativement faible précision moyenne.

Le matchmaker SeMa2 [Masuch *et al.* 2012] (le successeur de JIAC-OWLSM) effectue une sélection de service basé sur le profil complet. Le calcul sémantique hybride se base sur les signatures sur des E / S en combinant la correspondance des relations de subsumption représentés par un score numérique fixe et simple avec le degré de similitude des noms de concept basé sur les chaînes de caractères. Le classement par pertinence de valeurs est déterminé par l'agrégation pondérée linéaire des scores des deux types de correspondance. L'appariement logique de base sur la comparaison structurelle des pré-conditions et effets en SWRL, à travers leurs relations de confinement d'instance. SeMa2 a été développé par Nils Masuch (TU Berlin, Allemagne) et atteint une précision moyenne raisonnablement élevé.

ISEM [Klusch & Kapahnke 2012] est un matchmaker sémantique hybride et adaptative qui améliore de manière significative OWL-MX3 en adoptant des filtres supplémentaires basé sur la non-logique et sur la cohérence-probante de la formation donnée défini lors de l'apprentissage en off-line.

Le matchmaker XSSD [Li 2013] effectue une découverte hybride de la signature sémantique basée sur le calcul logique des E/S en se basant sur la relation de subsumption et en complément il calcule la similitude des descriptions de tags de service. Le tri des services est effectué en fonction de leur pertinence et déterminé par le degré de correspondance basé sur la logique suivie du degré de similitude textuelle. XSSD a été développé par Jing Li et Dongjie Chu (U Beihang, Chine) et a réalisé une bonne précision moyenne.

La variante adaptative du marieur OWLS-iMatcher, les OWLS-iMatcher2, apprend en off-line et à l'avance les similarités textuelles a appliquée à des paires de signatures de services sémantiques dont chacun est représenté comme un vecteur de mot-clé pondéré. OWLS-iMatcher2 effectue également une sélection hybride sémantique dans le sens où il calcule les similitudes de texte naturel sur la base des dédoublements des annotations de concepts respectives dans les ontologies partagées.

### 3.2.3 Discussion

**L'inconvénient majeur** des approches d'appariement non logiques est qu'elles relèvent toujours des problèmes linguistiques, terminologiques ou statistiques induisant une fausse interprétation de la syntaxe ceci est dû par le manque de richesse sémantique des services. Mais face à l'évolution des moyens d'expression de la sémantique des services, certaines approches choisissent souvent de doper leurs techniques par quelques mécanismes logiques d'appariement et elles finissent, ainsi, par proposer des solutions hybrides offrant des meilleurs résultats dans les cas où la syntaxe ne suffit pas. Par contre, les approches logiques possèdent plusieurs inconvénients, la complexité élevée du raisonnement affaiblit les chances de scalabilité, en outre les raisonneurs logiques ont un faible rappel (beaucoup de faux négatifs), puisque la

subsumption ne couvre pas tous liens sémantiques, et en dernier nous constatons que la majorité des services web actuels ne sont pas annotés avec des langages logiques formels. D'autres part, nous constatons que la présence des descriptions informelles ou la sémantique implicite (la fréquence des termes) au niveau des services, peut être efficace dans la découverte, en plus la complexité du matching est moins élevée que celles des approches logiques. En général, ces techniques utilisent le matching de graphes, la linguistique, le data mining, les techniques de recherche d'information, et les mesures de similarité.

La majorité des approches de découverte citée en haut propose à l'utilisateur d'exprimer sa requête en utilisant un service requête exprimé en OWL-S. Les approches autour de l'état de l'art de la découverte et d'appariement de services ont démontré que la majorité des méthodes se basent sur l'appariement des services profils se référant à des éléments purement syntaxiques engendrant un temps relatif pour l'appariement du service requête avec les services profils disponibles sachant que le matchmaking du process délivre un nombre égal ou supérieur des services pertinents par rapport aux service profils (comme le montre la figure 3.3).

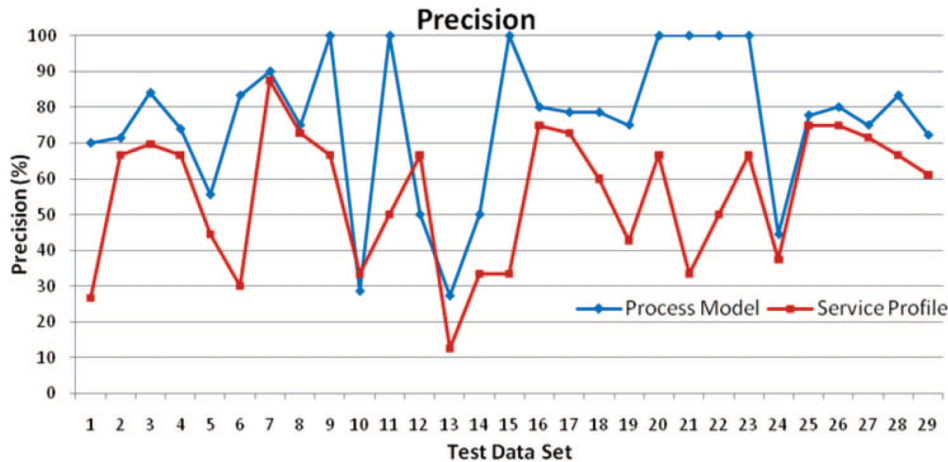


FIGURE 3.3 – Découverte profil vs Process [Stoilos *et al.* 2007]

Ainsi, le besoin pour l'utilisateur d'une description de sa requête totalement indépendante de tout langage de service est nécessaire. Les solutions exactes bien qu'efficaces, se sont montrées très coûteuses en termes de complexité et de temps d'exécution dans les systèmes à large échelle. Ainsi, il faut migrer vers d'autres techniques qui permettent un calcul rapide liés à la problématique de la composition des services web.

### 3.3 Composition des Services Web Sémantiques

La composition est définie comme "le processus de sélection, de combinaison et d'exécution de services en vue d'accomplir un objectif donné". La composition de services désigne une interaction entre deux ou plusieurs services en vue d'accomplir des objectifs déterminés. Un exemple type est la composition séquentielle entre deux services  $S_1(i_1, o_1)$  où  $i_1$  et  $o_1$  désignent respectivement l'entrée et la sortie de  $S_1$  et  $S_2(i_2, o_2)$  où  $i_2$  et  $o_2$  désignent respectivement l'entrée et la sortie de  $S_2$ . Elle consiste à invoquer  $S_1$  puis  $S_2$ , tel que  $S_2$  prend comme entrée

### 3.3. COMPOSITION DES SERVICES WEB SÉMANTIQUES

OWL-S Matchmaker	I/O	P/E	logique	non-logique	adaptative
iSeM (2012) [Klusch & Kapahnke 2012]	✓	✓	✓	✓	✓
OWLS-MX3 (2012) [Klusch <i>et al.</i> 2006]	✓		✓	✓	✓
SPARQLent (2010) [Sbodio 2012]	✓		✓		
SeMa2 (2012) [Masuch <i>et al.</i> 2012]	✓	✓	✓	✓	
OWLS-SLR (2010)	✓	✓		✓	
XSSD (2013)	✓		✓	✓	
OWLS-iMatcher (2008)	✓			✓	
OPOSSUM (2007) [Kuster <i>et al.</i> 2008]	✓		✓	✓	
FC-match (2005)	✓	✓	✓	✓	
WSC (2005)	✓		✓	✓	
CASD (2005)	✓		✓	✓	
EMMA (2012) [García <i>et al.</i> 2012]	✓			✓	

TABLE 3.1 – Les approches de la découverte sémantique (matchmaking)

$i_2$  la sortie  $o_1$  de  $S_1$ .  $S_1$  et  $S_2$  doivent coordonner leurs activités pour que  $S_2$  reçoive le message contenant  $o_1$ .

- Du point de vue de  $S_1$ , il doit s'exécuter, envoyer le résultat de son exécution, et attendre un accusé de réception. Du point de vue de  $S_2$ , il doit être en attente d'un message en entrée. Un fois le message reçu il doit envoyer un accusé de réception et s'exécuter avant de renvoyer sa sortie  $o_2$  si nécessaire. Ces deux points de vue représentent des descriptions comportementales des services  $S_1$  et  $S_2$
- Du point de vue de l'acteur qui exécute la composition, une séquence de deux étapes a lieu. La première étape correspond à l'invocation de  $S_1$ . Ce dernier reçoit une entrée  $i_1$  et s'exécute. Une fois son exécution terminée, la deuxième étape commence par l'exécution de  $S_2$ . L'exécution de  $S_2$  se termine par une sortie  $o_2$ . Cette vision est communément désignée dans la littérature par orchestration de services qui décrit l'ordre dans lequel les opérations ont lieu dans une composition de services. [Erl 2005] définissent l'orchestration comme un processus multi-étapes impliquant un certain nombre de transactions.
- Du point de vue d'un acteur externe,  $S_1$  reçoit un message, il s'exécute et envoie un message contenant  $o_1$  à  $S_2$ . Ce dernier envoie un message à un  $S_1$  pour accuser la réception de  $o_1$  et s'exécute à son tour. Cette vision est communément désignée dans la littérature par chorégraphie de services. La chorégraphie exprime une vue d'ensemble des services interagissant dans le cadre d'une composition de services. Selon [Erl 2005], la composition de services Web consiste à utiliser plusieurs services Web dans le but de créer de nouveaux services à valeur ajoutée. Cette combinaison de services Web, habituellement appelée composition, consiste à combiner plusieurs services Web afin de satisfaire une requête. Elle est un premier pas vers l'accessibilité des informations touristiques utiles à n'importe quel moment dans tous les domaines. La composition de services Web peut être vue comme une fonction de composition  $CWS$  telle que :  $WSC : I \times D \rightarrow O$  où  $I$  est l'ensemble des requêtes possibles de l'utilisateur,  $O$  l'ensemble des sorties qu'obtient l'utilisateur et  $D$  l'ensemble des

ensembles des descriptions de services Web.

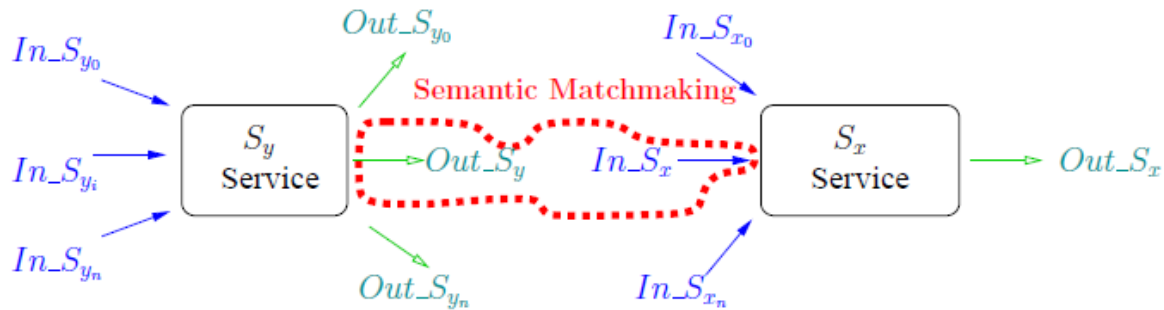


FIGURE 3.4 – Composition des Services Web Sémantiques

La composition requiert la description et l'organisation de l'interaction entre les services. Elle nécessite la gestion de plusieurs aspects comme les échanges de données entre les services, les pannes ou erreurs éventuelles, le contexte d'interaction, le degré d'automatisation des tâches, etc. Dans la littérature, une variété de spécifications, de langages et d'approches formelles ont étudié la composition. La composition automatique permet un développement plus rapide des applications à base de services. Elle consiste à préciser la requête d'un utilisateur sous forme d'objectifs à satisfaire. Un moteur de composition "intelligent" choisit la combinaison de services répondant à l'objectif décrit. Il génère la composition de service adéquate de manière transparente à l'utilisateur. Ce principe a interpellé plusieurs communautés de recherche travaillant dans le domaine de l'Intelligence Artificielle. Dans la suite nous classifions les approches de composition automatique en trois catégories : les approches sémantiques, les approches basées sur les automates et réseaux de Pétri & les approches basées sur les techniques de planification.

### 3.3.1 Approches basées sur les automates et réseaux de pétri

Les automates ou systèmes de transition d'états annotés offrent une sémantique précise et adaptée à la représentation de la composition. Les techniques de raisonnement applicables aux automates permettent de vérifier la structure de la composition et sa cohérence. Un automate est constitué d'un ensemble d'états, d'un ensemble de transitions annotées entre états et d'un ensemble d'actions. Les annotations des transitions contiennent des actions représentant le passage d'un état à un autre.

Un réseau Pétri est un ensemble de places et de transitions reliées par des arcs. L'aspect dynamique du réseau est décrit par un ensemble de règles. Chaque règle définit (i) les conditions à valider pour qu'une transaction ait lieu et (ii) les effets de cette transaction. Les réseaux de Pétri (Petri Nets) fournissent une représentation précise et sémantiquement riche de l'ordre d'exécution des activités. Ils permettent un raisonnement plus efficace sur la composition. Pratiquement, les réseaux de Pétri sont utilisés pour modéliser des systèmes concurrentiels.

Dans [Narayanan & McIlraith 2002], les auteurs proposent une traduction du Process model de OWL-S en un réseau de Pétri afin de raisonner sur la vérification automatique de

la composition. Ils proposent un outil qui génère automatiquement un réseau de Pétri pour une composition exprimée par un Process model de OWL-S. Cet outil vérifie les propriétés de la composition générée. Il juge la consistance du graphe modélisant le flux d'interaction. Il détermine, par exemple, si tous les états (nœuds) peuvent être atteints, si les contraintes décrites sont respectées, si des impasses existent, etc.

#### 3.3.2 Approches basées sur les techniques de planification

##### 3.3.2.1 Par calcul de situation

Dans [McIlraith & Son 2002], ils proposent une extension de Golog. Ce dernier est un langage de programmation logique basé sur la situation calculus. La technique présentée repose sur un ensemble de procédures génériques de haut niveau et sur la personnalisation des contraintes associées. Les procédures génériques représentent la requête utilisateur. Les contraintes à respecter sont écrites avec un langage de premier ordre. Chaque service est modélisé comme une action. Deux types d'actions sont possibles : des actions primitives et des actions complexes telles que ces dernières sont des compositions d'actions individuelles. Les auteurs proposent une approche qui permet à l'utilisateur de traduire manuellement les descriptions OWL-S des services impliqués en Prolog. Après cette traduction, le système utilise des règles d'inférences logiques pour automatiser le choix d'un service pour chaque action en respectant les contraintes associées.

Golog [McIlraith & Son 2002] et son successeur ConGolog [De Giacomo *et al.* 2000] sont des langages de programmation logique destinés aux domaines dynamiques (comme la planification) et orientés agents, dont les bases sont une version étendue du situation calculus [Levesque *et al.* 1998]. Des adaptations de Golog et ConGolog ont été proposées pour faire de la composition dynamique de services. Le principe de résolution est le suivant : les services disponibles, décrits en , sont traduits dans un formalisme permettant de les manipuler dans le domaine de la planification. Dans ce travail, deux traductions sont opérées :

1. en Golog pour manipuler les services à un niveau logique (avec les outils dédiés au situation calculus ) ;
2. en réseaux de Petri pour pouvoir utiliser les outils dédiés (vérification d'un réseau, simulation, etc.).

Dans [McIlraith & Son 2002], les auteurs présentent les fonctions suivantes :

- la simulation de l'évolution d'un service selon différentes conditions,
- la validation en observant ce qui est obtenu par rapport à ce qui est attendu,
- la vérification du maintien de certaines propriétés,
- la composition d'un service répondant à un but spécifique,
- l'analyse de performances évaluant la capacité de fournir un service dans le respect du temps, du niveau de services et d'utilisation de ressources.

La plupart de ces fonctions sont réalisées grâce à la traduction en réseau de Pétri. Ceci s'explique par les preuves que peut fournir un tel modèle et ses techniques d'analyse. Ainsi, la totalité des structures de composition de DAML-S sont traduites en réseau de Pétri. En ce qui nous concerne, seule la composition est en lien avec notre problématique. La traduction en Golog est utilisée pour la réaliser. La composition fournie est une séquence de services

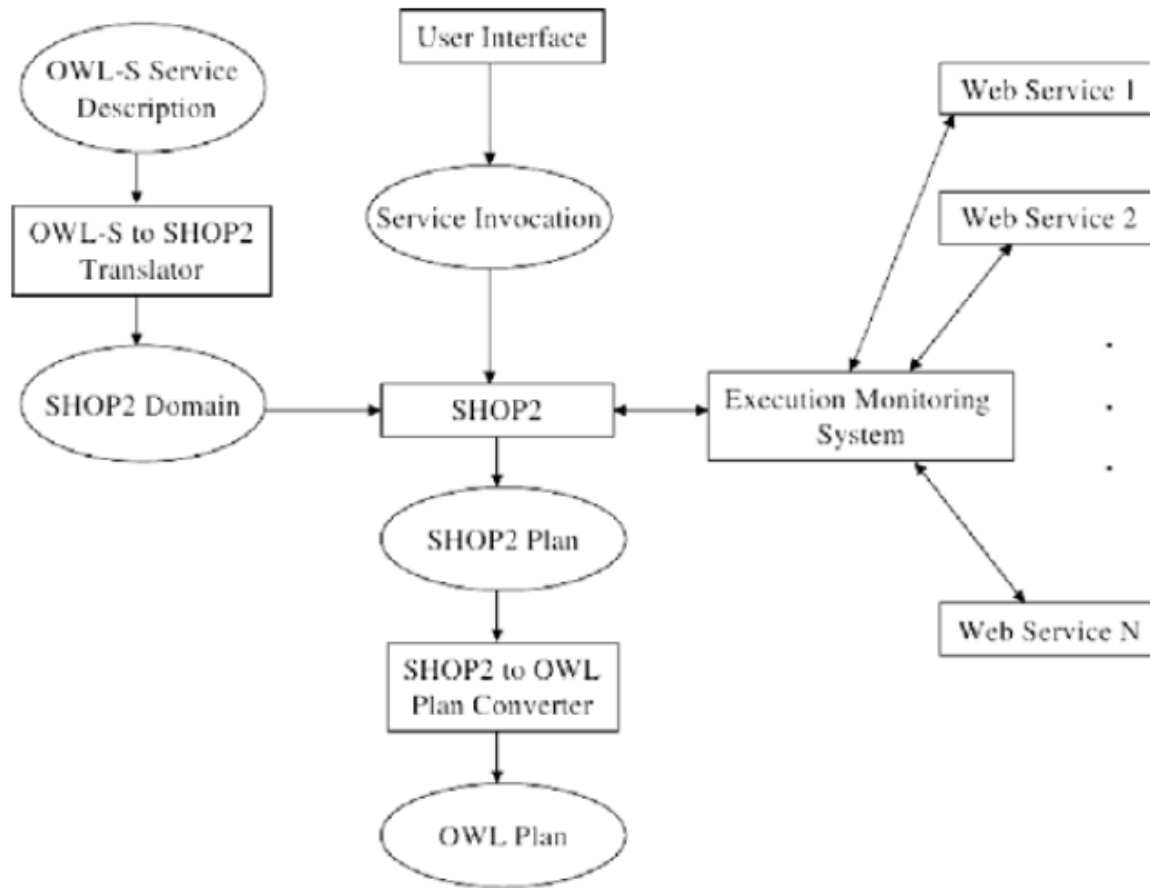


FIGURE 3.5 – Architecture de [McIlraith & Son 2002]

atomiques : c'est une limitation du point de vue de la flexibilité, puisque la structure de la composition fournie ne permet aucune alternative. Cette définition du problème est étendue dans [McIlraith & Son 2002], et les services disponibles sont décrits dans des procédures en ConGolog. Ces procédures permettent de répondre de façon générique aux objectifs de l'utilisateur : une procédure existe pour chacun des objectifs de l'utilisateur. ConGolog permet notamment d'exprimer la concurrence. L'utilisateur décrit formellement ses préférences et contraintes, qui sont ensuite utilisées pour la recherche d'une composition permettant de respecter ses contraintes.

### 3.3.2.2 Par décomposition hiérarchique de l'objectif

Nous présentons ici trois approches différentes qui utilisent une décomposition explicite de l'objectif. Cet objectif est décrit sous la forme d'un modèle hiérarchique du service attendu par l'utilisateur.

**Composition des Web services utilisant le planificateur SHOP2** L'approche [Wu *et al.* 2003] utilise la spécification pour la spécification des Web services. DAML-S dé-

### 3.3. COMPOSITION DES SERVICES WEB SÉMANTIQUES

finit un ensemble d'ontologies pour la description des propriétés et des capacités des Web services et permet de renforcer l'automatisation des différentes activités réalisées par les Web services y compris : la découverte, la composition, l'exécution, la surveillance, etc. Dans cette approche, les Web services sont modélisés comme des processus. On peut identifier trois types de processus : atomique, simple et composite. Le processus atomique est un modèle en une étape du point de vue client. Son exécution consiste en une simple invocation avec ses paramètres d'entrée. Le processus simple est aussi modélisé par une seule étape, mais contrairement à un processus atomique, il est possible d'accéder à l'intérieur de sa structure, ou de le remplacer par une expansion. Enfin, le processus composite représente une composition de Web services, *i.e.*, qu'elle est décomposable en d'autres processus atomiques, simples ou composites. Dans l'ontologie des processus, chaque processus dispose de plusieurs propriétés ; à savoir : inputs (optionnels), pré-conditions, outputs (conditionnels), effets (conditionnels). La décomposition des processus composites est spécifiée par le contrôle des constructeurs qui sont : Sequence, Unordered, Choice, If-Then-Else, Iterate, repeat-Until, repeat-While, Split et Split+join. Le but de la composition automatique dans ce processus est de trouver une collection d'instances de processus atomiques qui forme un chemin d'exécution pour certains processus composites de haut niveau. Ceci est assuré par l'utilisation du planificateur SHOP2 [Nau *et al.* 2003] qui implémente l'algorithme de planification HTN. La différence entre SHOP2 et les autres planificateurs consiste à ce que SHOP2 planifie les tâches dans le même ordre où elles seront ensuite exécutées. Ceci permet de savoir l'état courant du monde dans chaque étape du processus de planification.

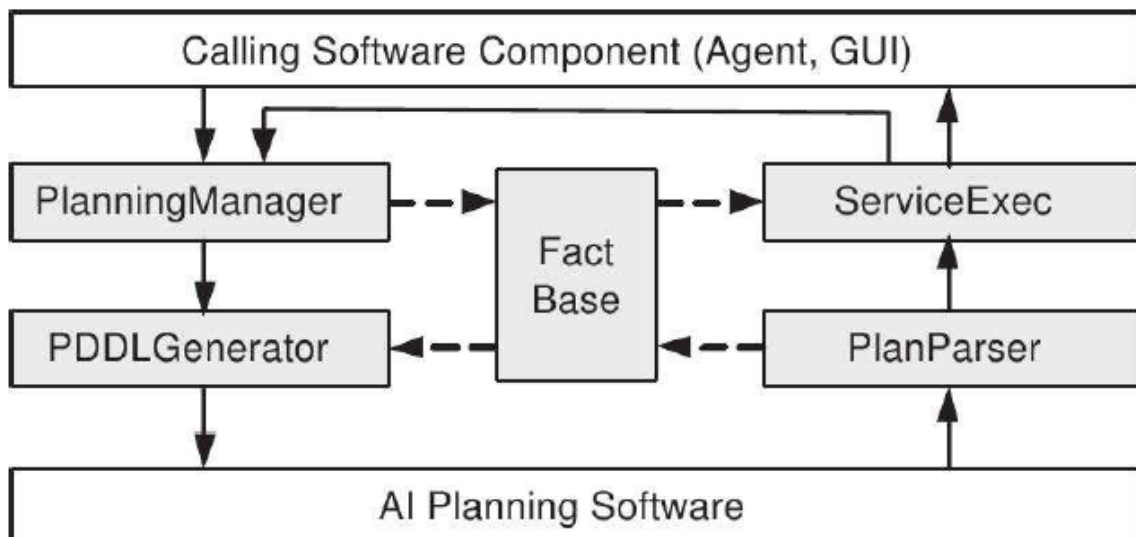


FIGURE 3.6 – L'architecture de WSPan, un système de planification de services, extraite de [Wu *et al.* 2003].

La technique permettant de traduire un modèle de collection de processus DAML-S en un domaine de planification SHOP2 repose sur les hypothèses suivantes :

- les processus atomiques peuvent avoir soit des effets qui définissent leurs connaissances, ou bien des sorties qui caractérisent les effets physiques de leur exécution. Par

conséquent, un processus atomique ayant seulement des sorties est modélisé par un service fournisseur d'information : `information providing`; et un processus atomique ayant seulement des effets est modélisés comme un Web service : `world-altering`;

- Il n'y a pas de processus composites avec des constructeurs de contrôle concurrent comme : **split** , **split-join** .

Une collection  $K$  de définitions de processus est codée dans un domaine SHOP2  $D$  comme suit :

- chaque processus atomique avec des effets dans  $K$  est codé comme un opérateur SHOP2 qui simule les effets du changement du monde du Web service ;
- chaque processus atomique avec des sorties dans  $K$  est codé comme un opérateur SHOP2 ayant des pré-conditions incluant un appel aux Web services fournisseur d'information ;
- chaque processus simple ou composite dans  $K$  est codé comme une ou plusieurs méthodes SHOP2. Ces méthodes spécifient comment décomposer une tâche HTN qui représente le processus simple ou composite.

Les détails de traduction des différents modèles de processus et le codage d'un problème de composition comme un problème de planification dans un domaine SHOP2 sont détaillés dans [Nau *et al.* 2003]. Cette approche a été implémentée et testée sur différentes instances de problèmes de composition de Web services. Dans les différents problèmes, SHOP2 est capable de trouver la meilleure solution possible. Cependant, la spécification des problèmes, les détails d'exécution ainsi que les résultats empiriques ne sont pas explicités [Wu *et al.* 2003].

### 3.3.2.3 Par PDDL

Le Planning Domain Definition Language (PDDL) est un langage de planification largement adopté par les communautés de l'Intelligence Artificielle. Dans [Mcdermott *et al.* 1998b], McDermott décrit les similarités entre l'approche de description et de composition OWL-S et le Planning Domain Definition Language (PDDL). Il présente une approche pour traduire les descriptions OWL-S en PDDL afin de modéliser la composition comme un problème de satisfaction d'un "planning". Il propose la notion de *value of an action* qui modélise l'exécution d'un service. Elle exprime des changements qui peuvent avoir lieu dans l'environnement.

CASCOM<sup>3</sup> (Context-aware business Application Service CO-ordination in Mobile computing environment) est un projet européen (2004-2007) pour l'implémentation et la validation des infrastructures de services pour les utilisateurs à travers les réseaux mobiles et fixes dans les environnements IP2P (intelligent agent-based peer-to-peer). La conduite de la vision CASCOM est que les services de l'entreprise peuvent se coordonner d'une manière flexible. Ils sont fournis aux utilisateurs via des agents intelligents dans un contexte d'environnement dynamique, ouvert, à grande échelle. D'où l'idée de développer des techniques pour la découverte et la composition automatique des Web services sans trop d'interaction avec l'utilisateur. L'architecture de CASCOM est de type multi-agents où un agent utilisateur (PA-Personal Agent) envoie une requête à l'agent de composition (SCPA Service Composition Planner Agent). Celui-ci obtient les descriptions des services nécessaires à la composition d'un troisième agent de découverte des services (SDA Service Discovery Agent). Même si l'ar-

---

3. <http://www.ist-cascom.org>



chitecture du CASCOM est une architecture multi-agents, le plan de composition est obtenu de manière centralisée par l'agent de composition SCPA. L'architecture interne de l'agent de planification est constituée de deux composantes principales :

- Le filtre permet de réduire l'ensemble des services qui seront utilisés pour la composition. Ce filtre est appliqué à l'ensemble des services retrouvés par l'agent de découverte des services SDA. Son objectif est de sélectionner un ensemble de services à utiliser pour la composition en vue d'accroître l'efficacité du processus de planification ;
- OWLS-XPlan est un planificateur de la composition des services qui prend en entrée un ensemble de services décrits en OWL-S, une description de l'état initial et l'état but, et retourne un plan. L'ensemble de la description du service est obtenu à partir du filtre. Les planificateurs qui implémentent les techniques de planification hiérarchique HTN sont efficaces dans le domaine pour lequel la connaissance est complète et détaillée sur les structures hiérarchiques d'action disponibles. Par contre, pour les domaines dans lesquels ce n'est pas le cas, *i.e.*, où il n'y a pas de méthodes ni de règles de décomposition fournies, un planificateur HTN serait incapable de trouver la solution. En revanche, les planificateurs basés sur la représentation STRIPS de l'action sont capables de trouver une solution fondée sur des actions atomiques telles qu'elles sont décrites dans les méthodes HTN, mais sans utiliser la structure de ce dernier. Les actions atomiques peuvent être combinées de multiples façons pour résoudre les problèmes de planification qui sont hiérarchiquement structurés selon des règles et des méthodes de décomposition. Cependant ils ne sont pas résolus exclusivement par des moyens de planification HTN. Pour cela, un planificateur hybride Xplan est développé. Xplan combine les avantages des deux approches par l'extension de l'efficacité de Graphplan basé sur le planificateur FastForward (FF), et les techniques HTN. Cela conduit à un degré élevé de souplesse par rapport à un planificateur HTN pur. En plus, l'utilisation des méthodes HTN améliore l'efficacité du planificateur FF.

La figure 3.7 montre un exemple de la façon dont XPlan de OWLS-Xplan [Klusch & Gerber 2005] utilise les techniques de planification par chaînage avant avec les techniques de décomposition HTN. En effet, XPlan utilise uniquement les éléments d'une méthode de décomposition qui sont nécessaires pour atteindre l'état but avec une séquence composite de services  $WS_1$  et  $WS_3$ . En revanche, le planificateur HTN décompose M totalement en  $WS_1$  suivi par  $WS_2$ , d'où le résultat qui ne contient pas  $WS_2$  et qui n'est d'aucune utilité pour atteindre l'objectif.

Afin d'utiliser Xplan pour la composition des services Web sémantique, celui-ci est complété par un outil de conversion de la description OWL-S des services à la description PDDL correspondante. La description PDDL est utilisée comme entrée de XPlan pour planifier la composition qui satisfait le but. Contrairement aux planificateurs HTN, Xplan trouve toujours une solution si elle existe dans l'espace d'action/état, bien que le problème soit NP-complet. Cette complexité est réduite de façon efficace en guidant la recherche par l'utilisation d'une heuristique d'approximation de la distance entre l'état initial et tous les états accessibles. Enfin, la génération du graphe de recherche est combinée avec une méthode de recherche hill-climbing pour élaguer l'arbre. L'information sur la qualité de l'action est utilisée par la recherche locale pour choisir parmi plusieurs choix qui sont également pondérés par les heuristiques. Cela se fait par le calcul de l'ensemble des actions utiles exécutables pour chaque

état de recherche tel que le but puisse être atteint. Une action utile d'un état de recherche  $q$  est une action qui satisfait au moins une proposition du but. S'il y a de nombreuses actions utiles, alors les actions de décomposition sont préférables parce qu'ils sont plus susceptibles de réussir dans le réseau de tâches comme partie du plan solution. Enfin, il est clair que dans toutes les techniques de planification, plus le nombre d'actions (services) mises à disposition du planificateur n'augmente, plus le temps de réponse n'augmente. Cette augmentation est surmontée par l'utilisation de filtres pour réduire le nombre de services passés au planificateur.

Idéalement, l'ensemble des services pris en compte pour créer le plan de composition devrait comprendre tous les services inscrits dans l'annuaire. Toutefois, cela peut être non pratique parce que le nombre de services augmente. Pour surmonter ce problème, il est nécessaire de réduire l'ensemble des services fournis au planificateur. Dans ce but, des filtres de services sont utilisés pour filtrer les services de l'annuaire qui sont moins pertinents pour le processus de planification.

Pour déterminer les services concrets qui passent le filtre au planificateur, trois fonctions sont considérées :

- *ThresholdFiltre* établit un seuil et filtre les services dont les classes ont un degré de pertinence inférieur à ce seuil ;
- *K-Filtre* retourne les  $k$  meilleurs services. Dans ce cas, le nombre de services qui passent le filtre est prédéfini ;
- *Filter* retourne un pourcentage de l'ensemble de services (sur la base de la pertinence de leurs classes correspondantes). Dans ce cas, le nombre de services pris en compte dans le processus de planification dépend de la taille de l'annuaire.

Hatzi et al. [Hatzi *et al.* 2013] propose une approche de composition appelé *Porsce* est une approche interactive puisqu'elle donne la possibilité à l'utilisateur d'insérer ses propres valeurs de métriques. Cette approche s'appuie sur *OWL-XPlan* qui transforme les *Process Model* (Processus atomique simple) en *PDDL*. Le fonctionnement général du système repose sur le calcul des mesures de similarité entre les concepts des ontologies qui ont pu être extrait à partir de la liste des concepts choisis initialement pour le processus de composition dynamique afin d'enrichir les *PDDL*. Son approche diffère des autres par l'utilisation non seulement d'un mais de deux planificateurs (*LPG-TD* [Gerevini *et al.* 2004] et *JPlan*) en plus autorise la re-planification à l'exécution. L'architecture adoptée, illustrée Figure 3.8, montre le fonctionnement général de *Porsce* (disponible au public). Cette vision du problème se fonde sur l'idée qu'un planificateur ne peut pas être suffisamment expressif pour répondre à toutes les exigences de la composition de services. Les planificateurs en question, fournit un plan séquentiel. Cette vision de la composition de services, avec re-planification, permet de surmonter l'indéterminisme de l'exécution de certains services (par exemple concernant les ressources disponibles, ou non, au moment de la planification).

**Approche de composition basée sur les liens sémantiques** Une approche pour la composition de Web services est développée dans [Lécué *et al.* 2008]. Dans cette approche, les services sont vus comme des fonctions ayant des paramètres i) d'entrée, et de sortie sémantiquement annotés par des concepts d'une ontologie de domaine et ii) des pré-conditions et effets conditionnels sur le monde. La composition des Web services est alors considérée comme une composition des liens sémantiques où les lois de cause à effet ont aussi un rôle

### 3.3. COMPOSITION DES SERVICES WEB SÉMANTIQUES

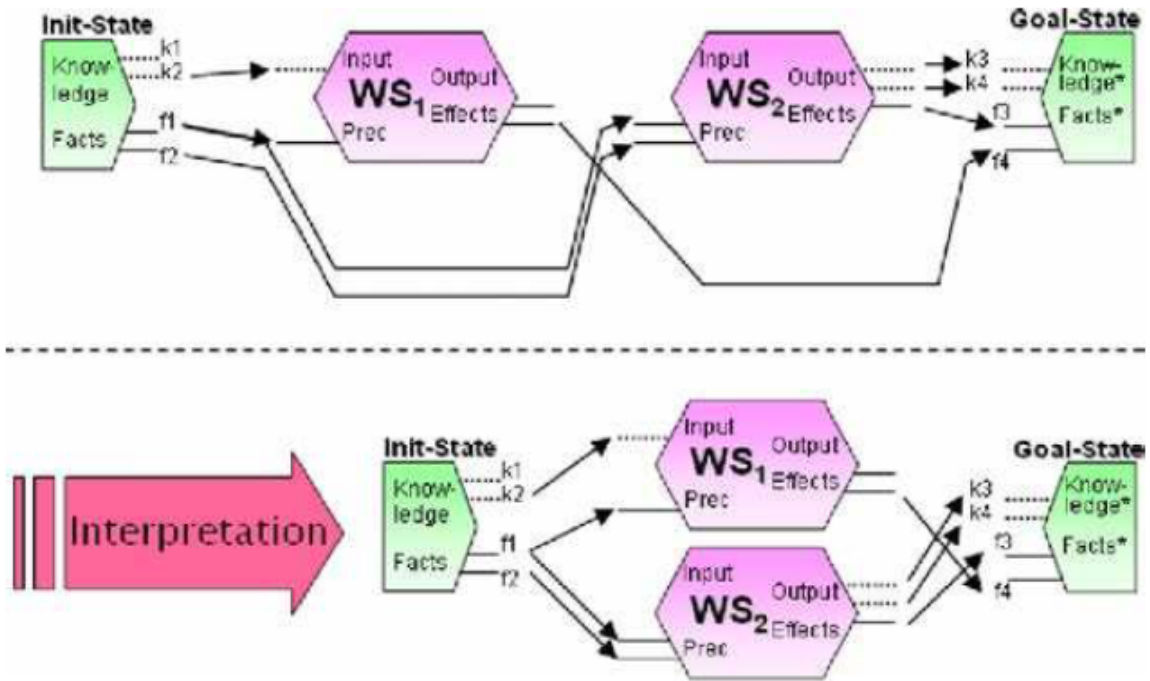


FIGURE 3.7 – Technique d’interprétation a posteriori du plan pour obtenir des compositions complexes, ici d’une séquence de services à un <Split + Join> du langage de description OWL-S .

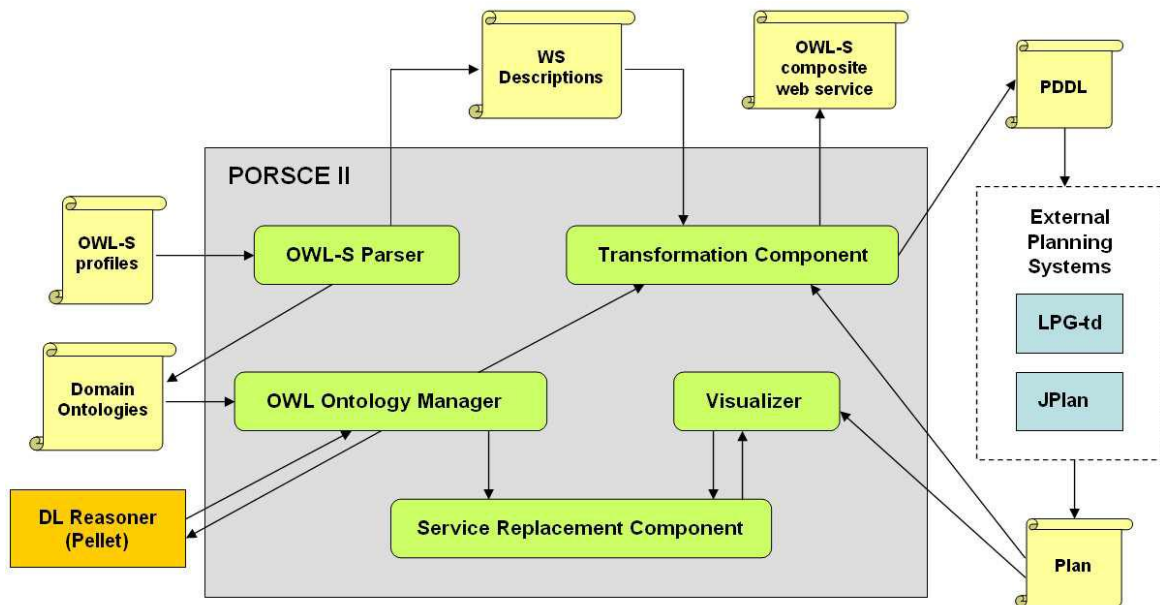


FIGURE 3.8 – Architecture de Porsce [Hatzi *et al.* 2013]

prépondérant. Dans cette approche, une matrice de liens sémantiques appelé Causal Link Matrix *CLM* est utilisée pour représenter les liens sémantiques entre les différents services

comme des correspondances output-input où les lois entre effets et pré-conditions de services ont aussi un rôle prépondérant. Ensuite, à partir de la requête définie par un état initial et un but, un algorithme récursif de régression (Ra4C) est utilisé pour extraire un plan qui répond à la requête.

Les paramètres (i.e. input and output) de Web services sont des concepts référents à des ontologies  $T$ . La correspondance (en anglais matchmaking) vise à trouver une similarité sémantique entre un paramètre de sortie  $Out_{s_1}$  et un paramètre d'entrée  $In_{s_2}$  de deux services  $s_1$  et  $s_2$ . Les liens causaux stockent cette similarité entre les paramètres de Web services  $sim_T(Out_{s_1}, In_{s_2})$ . Les relations de similarités sémantiques basiques utilisées dans le matchmaker sont : Exact et Subsume la valeur de similarité est 1 ; PlugIn, dans ce cas, la valeur de similarité est 2 ; Disjoint, la valeur de similarité est 0.

Dans la découverte des Web services, le matchmaking est effectué sur les mêmes catégories de paramètres, i.e., sur les paires d'entrée, ou les paires de sortie. A l'opposé, pour la composition de Web services, le problème est adressé par l'étude de matchmaking sur des paires distinctes de paramètres d'entrée et de sortie des différents Web services et  $s_x$  et  $s_y$ . Le résultat du problème de composition est un ensemble de Web services partiellement ordonnés pour former un workflow afin d'accomplir le but de la composition. L'ordre partiel est constitué des liens causaux, et des dépendances sémantiques entre certains services.

Quant à la matrice des liens causaux (CLM), elle vise à faciliter le processus de la composition des Web services et à améliorer ses performances en calculant tous les liens causaux entre un ensemble de Web services de façon simple, intuitive et flexible. Une fois que tous les services web en SW s ont été enchaînés selon le critère de liens de causalité, le problème de la composition de Web services est associé à un problème de planification en intelligence artificielle. Le problème de planification est formalisé par  $(SW_s, KB, \alpha)$ .  $SW_s$  se réfère à un ensemble de transitions d'état,  $KB$  est l'état initial, et  $\alpha$  Est le but. La méthode de composition de Web services consiste à trouver un plan qui produit les résultats  $\alpha$  selon une base de connaissances  $KB$ . Le processus de composition consiste en une recherche récursive de régression de l'ensemble des concepts de but  $\alpha$  à la base de connaissance initiale  $KB$  en utilisant la matrice de liens causaux  $CLM$ .  $CLM$  facilite la recherche parce que tous les services sont bien ordonnés via un modèle robuste et formel. L'idée principale est de contrôler et analyser la matrice de liens causaux  $CLM$  d'un domaine donné avec un moyen adéquat pour obtenir une composition de services et de leurs liens sémantiques qui saura satisfaire l'objectif de la composition. Dans le cas trivial, lorsque tous les buts dans  $\alpha$  ont des individus dans  $KB$ , le but  $\alpha$  est accompli par la condition initiale. Dans le cas plus complexe, le processus de composition doit récupérer, dans chaque étape, au moins un service  $s_x \in SW_s$  ayant un sous-but  $\alpha_i$  dans  $\alpha$  comme paramètre de sortie. En cas de succès, le processus est réitéré avec chaque paramètre d'entrée de  $s_x$  comme un nouveau but. Le processus de composition est exécuté récursivement jusqu'à ce que les concepts dans  $\alpha$  et les nouveaux buts (buts récursifs comme paramètres d'entrée de services impliqués dans la composition) aient des individus dans  $KB$ . Par conséquent, une disjonction de plans consistants est retournée. L'algorithme de composition retourne une disjonction de plans consistants contenant l'ensemble des plans valides. Enfin, il faut noter que l'algorithme utilisé dans cette approche est similaire à l'algorithme de recherche utilisant la stratégie de chaînage arrière. La différence principale est que dans cet algorithme, la matrice de liens causaux définissant l'ordre d'exécution des services

permet d'annuler l'étape de calcul des services exécutables à chaque itération, ce qui réduit le temps d'exécution de l'algorithme.

#### 3.3.3 D'autres approches

[Brogi *et al.* 2008, Brogi & Corfini 2008] modélise les Process Models en hypergraphe. Pour y parvenir, il effectue un pré-calcul sémantique qui se base sur un appariement logique des concepts structurels issus d'ontologies différentes en off-line. Ce pré-calcul est nécessaire pour déterminer l'intra-dépendance entre les Process Model. La construction de l'hypergraphe se fait sur la base de cet alignement (exact, plugin) en plus du fonctionnement interne de chaque Process Model dont les sommets constituent les concepts et les hyper-arêtes les liens sémantiques (exact et plugin) entre ces concepts et l'interdépendance des processus atomiques entre ces concepts. Afin de délivrer la composition final, l'auteur a développé un algorithme récursif afin d'effectuer une recherche dans l'hypergraphe partant des sorties de la requête à satisfaire correspondant aux processus atomiques répondant au même besoin et ceux en revenant en arrière jusqu'à ce qu'il n'y est plus de sorties à satisfaire. L'ensemble des processus atomiques final constitue la composition optimale.

[Paulraj *et al.* 2011, Paulraj *et al.* 2012] a développé un algorithme de découverte orienté composition ; a partir d'une requête service, l'ensemble des appariements sont calculés afin de chercher les Process Model pertinents. Pour chaque terme définis dans la requête service, l'algorithme parcourt le Process Model de la racine jusqu'à la feuille et ce à partir de la racine. Si un appariement sémantique est trouvé, le processus atomique est ajouté à la liste temporaire avec ces constructeurs s'il n'existe pas. Si le service ajouté manque d'entrée sémantiques, il lance une recherche jusqu'à ce qu'il trouve un processus atomique qui satisfasse le processus atomique initial. Enfin, il renvoie la liste ordonnée des processus atomiques qui génère la composition finale. Cependant, le principal inconvénient de cette approche est que le temps d'analyse du Process Model augmente avec sa taille.

#### 3.3.4 Discussion

La planification automatique est l'approche majoritairement adoptée pour la composition dynamique : la planification a pour objectif de structurer des actions, ici des services, sous la forme d'un plan qui permet, à l'exécution, d'atteindre un objectif exprimé. Nous notons l'existence de nombreux travaux [Sirin *et al.* 2004, Wu *et al.* 2003] sur la traduction de la description des services en domaine de planification, utilisant les langages adaptés à la planification (STRIPS [Fikes & Nilsson 1971], PDDL [Mcdermott *et al.* 1998b], HTN [Nau *et al.* 2003]). Cependant, aucun planificateur ne présente de résultat satisfaisant puisqu'une large partie des systèmes proposés s'appuie sur des planificateurs qui ne permettent pas de produire de plans flexibles.

Les questions concernant les critères d'ergonomie, ou encore l'intégration de l'utilisateur, sont la plupart du temps secondaires, ou ne sont pas soulevées du tout. Notre vision, centrée utilisateur, impose une contribution théorique et pratique. Ainsi, en planification le plan solution qui sera le plus apprécié sera celui contenant le moins d'actions et d'opérateurs possibles, la composition devra être riche en structures pour être flexible.

Nous constatons aussi que les approches de composition présentées, bien qu'elles permettent de décrire le schéma d'interaction voire de le générer automatiquement dans certains cas, ne réussissent pas à tenir compte de l'aspect dynamique de l'environnement des services. L'aspect dynamique se manifeste par des changements liés aux services et à leurs fournisseurs.

La majorité des approches (voir Figure 3.9) traite le problème de composition des Process Model sur la base d'un processus atomique et non un ensemble de processus atomiques à l'exception de [Brogi *et al.* 2008, Paulraj *et al.* 2011]. Ces deux approches orienté Process Model s'assure de la découverte des concepts sémantiques définis dans le process Model afin de déterminer les processus atomiques nécessaires à la composition. Principalement, ces deux approches présentent le problème d'évolutivité ils présentent deux algorithmes différents qui sont fonctionnel pour un référentiel relativement minime.

Aucune de ces approches ne détermine la stratégie à employer lorsque le nombre de services augmente exponentiellement ce qui en augmente la complexité. En outre, la plupart des approches de composition n'introduisent des techniques d'optimisation pour améliorer l'évolutivité et n'en identifient les fonctionnalités équivalentes ou dominantes qui pourraient apparaître lorsque de nombreux registres de services sont impliqués dans la composition.

### 3.3. COMPOSITION DES SERVICES WEB SÉMANTIQUES

	Automatique	Dynamique	Capacités Sémantique	Non déterministe	Scalabilité	Exactitude	Indépendance domaine	Langage pivot
Klusch et al. [Klusch & Gerber 2005]	✓	✓	Process Model				✓	PDDL + HTN
Paulraj et al. [Paulraj et al. 2011, Paulraj et al. 2012]	✓		Process Model	✓			✓	Aucun
Narayanan et al. [Narayanan & McIlraith 2002]	✓		Service profile	~			✓	SHOP2
Brogi et al. [Paulraj et al. 2011, Paulraj et al. 2012]	✓		Process Model		✓	✓	✓	Hypergraphe
Lecué et al. [Lécué et al. 2008]	✓	✓	Service profile + QoS Service profile + QoS	~		✓	✓	Golog calcul de situation
McIlraith et al. [McIlraith & Son 2002]			Process Model	~		✓	✓	automates et réseaux de pétri
Wu et al. [Wu et al. 2003]	✓		Process Model	~			✓	PDDL
Hatzi et al. [Hatzi et al. 2013]	✓	~	Process Model		✓	✓	✓	

TABLE 3.2 – Les approches de la composition sémantique (matchmaking)

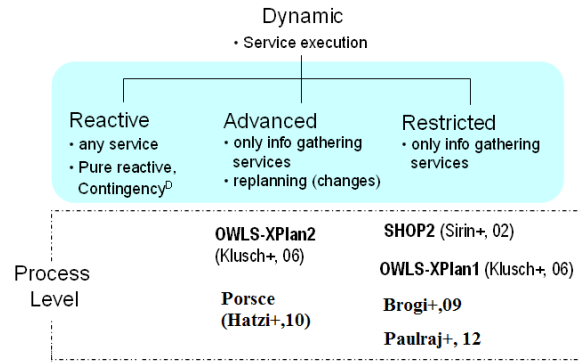


FIGURE 3.9 – Classification des approches pour la composition dynamique du Process Model [Klusch 2012]

### 3.4 Conclusion

Le Service Profile est écrit à la main et ne représente pas toutes les fonctionnalités du service, alors que le Service Model décrit toutes les IOPE et toutes les manières de le connecter avec d'autres Web services. Le Service Profile est dédié spécifiquement aux annuaires selon ses concepteurs, il doit servir à trouver facilement un service selon une requête. Comme la tâche de sélection doit être aisée et pour une meilleure composition sémantique et totalement automatique, la nécessité de se focaliser sur les fonctionnalités décrites sémantiquement celle incluse dans le Service Model. C'est pour cela, que notre thèse s'articule autour de la découverte des Service Model dû au fait de leur description explicite des fonctionnalités nécessaire à la sélection. Dans cette thèse, nous nous intéressons aux hétérogénéités sémantiques des données échangées entre les services Model engagés dans une composition.

Pour cela, nous proposons d'améliorer une approche originelle de découverte des Services Model. Notre principale contribution évolue autour de la notion de Données touristiques Liées pour enrichir l'échange des données entre services Web. Nous proposons un modèle orienté Données liées pour résoudre les hétérogénéités sémantiques entre Services Web Composés. L'efficacité de la recherche peut également être améliorée en utilisant des optimisations de recherche en fonction de la stratégie de recherche à suivre. Ces optimisations peuvent être appliquées à des actions disponibles pour chaque Etat par la taille des actions qui conduisent à des impasses, des actions qui sont équivalentes, ou des actions qui sont dominés (ne peut pas conduire à une meilleure solution).

Afin de résoudre les problèmes précédents, l'algorithme de composition doit répondre aux besoins suivants :

1. fournir des mécanismes de découverte pratiques à grains fins afin d'identifier les services pertinents à la composition.
2. Améliorer le temps de réponse de découverte de service afin de traiter rapidement les requêtes ;
3. Incorporer des techniques d'optimisations afin d'améliorer l'évolutivité de l'ensemble du processus de composition ;



4. trouver les compositions de services optimaux en minimisant différents critères tels que le nombre de services et la longueur de la composition en évitant les solutions complexes et ingérables.

Par la suite, nous détaillons les approches sémantiques dans le domaine touristique.



# 4

## E-tourisme & Package dynamique

### Sommaire

---

<b>4.1 Introduction : E-tourisme</b> . . . . .	<b>91</b>
<b>4.2 Le package dynamique: la nouvelle tendance de l'e-tourisme</b> . . . . .	<b>93</b>
<b>4.3 Enjeux</b> . . . . .	<b>95</b>
<b>4.4 Approches de Packages dynamiques Sémantiques</b> . . . . .	<b>95</b>
4.4.1 Approche touristique orienté web sémantique . . . . .	96
4.4.2 Approche touristique orienté Services Web Sémantiques . . . . .	102
<b>4.5 Discussion et synthèse</b> . . . . .	<b>105</b>
<b>4.6 Conclusion</b> . . . . .	<b>110</b>

---

### 4.1 Introduction : E-tourisme

---

Initialement, Internet a commencé par être un outil de communication entre les hommes et les machines. Au fil du temps, Internet a évolué d'une manière exponentielle, devenant accessible aux entreprises, chercheurs et aux particuliers. Seulement, de plus en plus d'entreprise utilise ce réseau afin de communiquer avec l'extérieur ou de relier leurs différents systèmes informatiques de manière à, transmettre des informations entre les différentes entités de l'entreprise. Par la suite, les entreprises se sont reliées entre elles pour faire du commerce, donnant naissance au B2B. Les nouvelles technologies de l'information et de la communication ont engendré de profondes mutations au sein de la société de consommation actuelle. Internet et les outils qui en découlent sont, aujourd'hui, ancrés dans les habitudes de consommation des utilisateurs, qui sont de plus en plus accros à cette technologie.

Aujourd'hui, cette technologie est le moteur de la croissance mondiale notamment grâce au e-commerce et aux échanges qui en résultent. La rencontre de ces deux secteurs d'activité, à savoir les nouvelles technologies et le tourisme a donné naissance à un secteur très porteur : le E-tourisme.

Comme tous les autres secteurs d'activité, l'industrie touristique est devenue mondialisée où l'évolution des produits et services touristiques est permanente. Actuellement, le tourisme est l'un des secteurs d'activité les plus dynamiques sur le web représentant des milliards en terme de chiffre d'affaires et véhicule des millions de visiteurs. Les entreprises du secteur

## CHAPITRE 4. E-TOURISME & PACKAGE DYNAMIQUE

---

touristique sont déjà habituées à évoluer rapidement pour répondre à la demande de consommateurs. En effet, en fonction des modes et des tendances, les compagnies aériennes comme les voyagistes ou les hôteliers doivent innover, créer de nouveaux services et surtout interagir entre eux. Les e-touristes, d'abord consommateurs puis "consommateurs" impliquant un besoin de se rassurer, de comparer et de s'informer. Pour cela, le Web leur offre la possibilité de comparer les prix rapidement, de s'informer via les divers sites commerciaux.

L'offre et les services disponibles sur le Web correspondent exactement à ce que recherchent des consommateurs actuels :

- une accessibilité illimitée 24H/24, 7 jours/7 ce qui correspond au besoin d'immédiateté et d'opportunité et qui permet de "mieux gérer son temps tourisme" <sup>1</sup>.
- des offres personnalisées liées aux profils et aux désirs des consommateurs ce qui correspond au besoin de solution " sur mesure ". Ceci est simplifié par des outils tels que les tris croisés par prix, les alertes emails ou les comparateurs de prix disponibles sur les sites de l'e-tourisme.
- un accès possible à partir de plusieurs lieux : entreprises, domicile ce qui correspond à un besoin de mobilité <sup>2</sup>.
- une navigation libre, des moteurs de recherche, des guides d'achats qui répondent à un besoin de posséder de l'information sur tout et d'être autonome.

Avec l'évolution des services offerts en ligne et la démocratisation de l'accès à Internet, les consommateurs sont devenus très exigeants et ont soif d'innovations technologiques. Le succès de la vente à la dernière minute est lié à ces nouvelles tendances de consommation. Le consommateur est devenu un **consommateur**. Aujourd'hui, il veut faire son choix en toute connaissance de cause et pouvoir comparer prix et produits afin d'acheter intelligemment.

Le consommateur recherche un service toujours plus personnalisé, souhaite avoir le choix dans ses options de séjour et exige de plus en plus de souplesse et de flexibilité dans son choix de voyage. Quoi de mieux alors qu'un prototype permettant à l'utilisateur de créer son voyage sur mesure ? C'est ce que propose cette thèse, un nouveau mode de recherche d'information, d'organisation et de réservation de séjours en ligne.

De nos jours, l'E-tourisme fournit une quantité riche d'information des destinations offertes à travers le monde, tel que : les maps, les plans, les itinéraires, les images, les sons ainsi que des services transactionnels. De ce fait, le Web actuel constitue la principale source d'information touristique des destinations pour les voyageurs, seulement les données qui y sont stockées sont peu structurées ou non structurées.

Avec l'avènement du Web Sémantique et du Web Service, l'e-tourisme doit en permanence s'adapter pour gagner en flexibilité. Le Web sémantique fournit le mécanisme adéquat pour la manipulation de ces données non structurées puisqu'il permet de développer des applications

---

1. Patrick Vicériat/ ITG Consultants, Innovations technologiques, E-tourisme, Pontarlier, 17 mai 2003

2. Patrick Vicériat, Rencontres Internet Midi Pyrénées, Tourisme et Nouvelles technologies, 17 octobre 2002

## 4.2. LE PACKAGE DYNAMIQUE : LA NOUVELLE TENDANCE DE L'E-TOURISME

intelligentes permettant aux utilisateurs et aux machines de communiquer efficacement. Actuellement, le web sémantique est utilisé dans divers domaines d'application tels que E-santé, E-tourisme, E-savoir, la E-gouvernance, E-marketing, E-business, etc. de ce fait, Le Web sémantique a eu un impact majeur sur l'industrie du tourisme.

Une technologie innovante est apparue sous le nom " Dynamic packaging " (Package Dynamique), un concept permettant aux utilisateurs individuels de configurer en ligne et automatiquement son voyage personnalisé en assemblant soi-même les différents produits voyage. Ce qui requiert une intégration sémantique des systèmes d'information touristique hétérogènes et distribués. L'interopérabilité jouera évidemment un rôle important et aurait besoin de définition de normes.

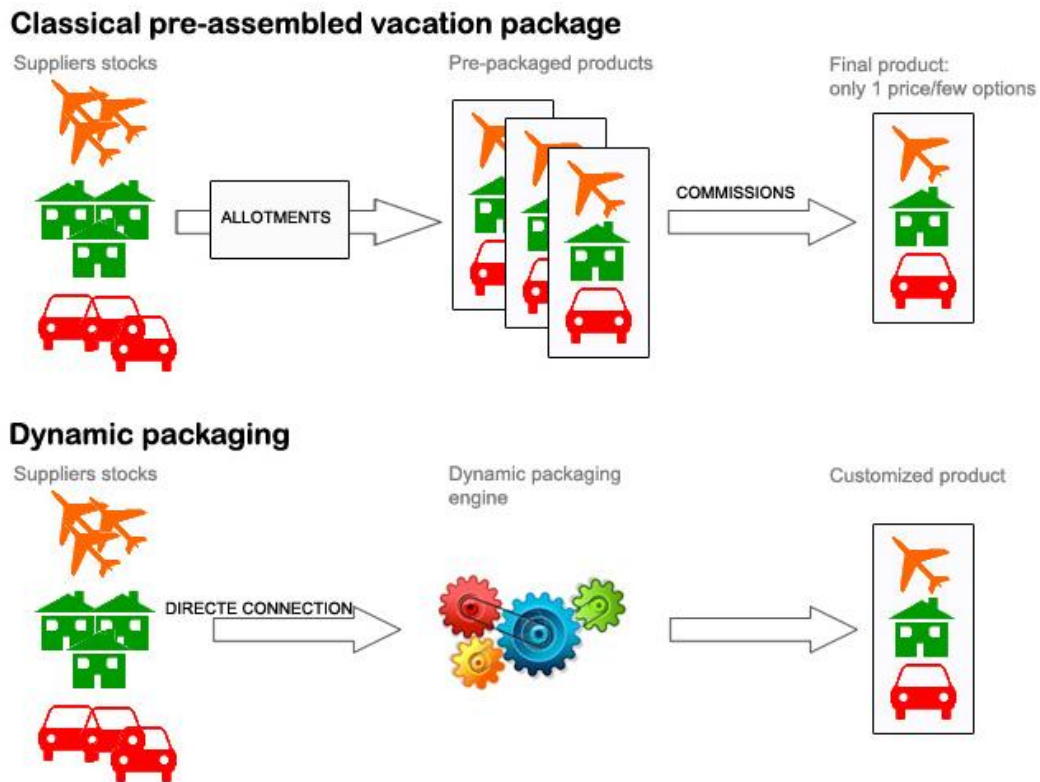


FIGURE 4.1 – Package dynamique vs pré-packaging [Schmeing *et al.* 2006, Ayazlar 2014]

Par la suite, je détaillerai le concept " Package Dynamique " et citerai quelques approches relatives à l'application des technologies sous-jacente du Web Sémantique au E-tourisme.

## 4.2 Le package dynamique : la nouvelle tendance de l'e-tourisme

En termes de voyage, les consommateurs ont été habitués aux forfaits de voyage proposés par les tour-opérateurs qui consistent à un package à prix tout compris et contenant au moins

## CHAPITRE 4. E-TOURISME & PACKAGE DYNAMIQUE

deux des éléments suivants :

- Un transport (vol ou train)
- 24h sur place ou l'hébergement pour une nuitée
- Une location de voiture
- N'importe quelle autre prestation touristique (excursion, activités culturelles, etc.) représentant une part significative dans le forfait

L'avantage principal des forfaits de voyage consiste à proposer un voyage tout-en-un à un prix négocié en amont, donc attractif. Cependant il manque de flexibilité pour le consommateur pour qui la durée, le plan de transport défini entre l'agent et le transporteur ainsi que les prestations annexes sont imposés.

Deux nouvelles tendances sont apparues ces dernières années : en premier lieu le besoin des consommateurs a évolué, ces derniers demandant plus de choix quant à la durée et les dates tout en ayant la possibilité de réserver sur Internet. En second lieu l'arrivée du low-cost dans l'aérien avec des tarifs compétitifs et des agences en ligne comme [booking.com](http://booking.com) ou [hotels.com](http://hotels.com) qui poussent le consommateur à comparer les offres et vouloir façonner lui-même son package en réservant les prestations de voyage séparément.

Pour répondre à cette demande de flexibilité et éviter à l'utilisateur de devoir réserver sur plusieurs sites Internet, les acteurs du voyage ont cherché des solutions leur permettant d'accéder dynamiquement à des sources vols, hôtels, loueurs de voitures, etc... Deux types de solution ont émergé : le **pré packaging** et le **package dynamique**.

	Forfait classique	Pré-packaging	Package dynamique
Distribution	Choix d'un forfait par le client	Choix d'un forfait par le client	Choix du vol et de d'hôtel par le client
	Limitation sur les dates de départ, les durées	Choix plus importants sur les dates de départ, les durées,...	Choix du vol et de l'hôtel par le client
	Traitement de la réservation auprès des fournisseurs	Réservation temps réel auprès du fournisseur	Réservation temps réel auprès du fournisseur
Production	Construction du forfait réalisée à l'avance	Construction dynamique en fonction de règles fixées à l'avance	Construction dynamique par le client
	Grille tarifaire saisie à l'avance	Tarification dynamique	Tarification dynamique
	Inventaires (vols, hôtels, transferts...) saisis à l'avance	Accès aux inventaires des fournisseurs (vols, hôtels,...)	Accès aux inventaires des fournisseurs (vols, hôtels,...)

FIGURE 4.2 – Les trois offres de forfaits disponibles pour l'utilisateur [sia partners 2013]

Avec le **pré-packaging**, le producteur fixe les règles de construction des produits à l'avance et la disponibilité, la tarification et la réservation des produits se fait en temps réel grâce aux connectivités vers les systèmes informatiques des fournisseurs. Pour le client, le produit ressemble à un forfait mais il est en réalité construit dynamiquement.

Avec le **package dynamique**, la construction du produit est réalisée par le client lui-même. Ce dernier va pouvoir choisir :

- Le vol qui correspond à ses critères : dates de départ et de retour, horaires, compagnie aérienne, ...
- L'hôtel qui correspond à son besoin, de l'auberge de jeunesse à l'hôtel 5 étoiles, suivant les critères qu'il impose : localisation, prix, services, ...
- Une éventuelle location de voiture qui sera disponible à son arrivée à l'aéroport

Cette tendance prend une part de plus en plus importante dans le marché du voyage et elle est devenue une offre indispensable pour les voyageurs.

---

### 4.3 Enjeux

Le package dynamique permet au consommateur de complexifier les critères de recherche et de composer un forfait de voyage unique avec des prestations de qualité variée. L'abondance d'information facilite cette tâche mais implique que le consommateur devient de plus en plus exigeant et cherche à comparer toutes les offres disponibles. Au niveau des distributeurs de voyage (agences en ligne, tour-opérateurs, compagnies aériennes, opérateur ferroviaires...) le package dynamique est une nécessité pour suivre les tendances du marché mais complexifie le travail car il impose de proposer un large choix de prestations tout en s'assurant de la qualité de ces dernières. Les enjeux technologiques sont également bien réels :

- Connectivités temps réel pour accéder aux sources vols, hôtels, voitures...
- Règles de gestion plus ou moins complexes pour la construction des produits
- Interfaces de vente utilisateur optimisées
- Maîtrise des temps de réponse alors que les produits sont construits dynamiquement par interrogation de sources externes
- Gestion de l'après-vente avec un dossier contenant des prestations venant de sources différentes
- ...

---

### 4.4 Approches de Packages dynamiques Sémantiques

Le Web constitue la principale source d'information touristique des destinations pour les voyageurs. Différents sites Web proposent des produits et des services touristiques issus des Systèmes d'information touristique (TIS) et d'organisation diverses. Les informations extraites de ces sources peuvent servir de tremplin pour le package dynamique, la planification personnalisée d'un voyage en temps réel, en réponse à la demande du consommateur ou l'agent de réservation. Sa principale motivation, repose sur le fait que les utilisateurs étaient découragés par la planification de leur voyage car il devait consulter manuellement différents Sites Web touristique indépendants, enregistrer ces informations personnelles à répétition, sans compter du temps d'attente de la confirmation de leur demande afin d'effectuer les

paiements par carte de crédit. Planifier un voyage était devenu une tâche complexe à gérer manquant de flexibilité et de fonctionnalité d'un côté, et occupait beaucoup de temps pour l'utilisateur d'un autre côté.

Avec le temps, les besoins des utilisateurs ont changé et exigent la possibilité de créer, de gérer et mettre à jour des itinéraires afin de construire des voyages sur mesure et en fonction de leur besoins réels. Ils souhaitent dorénavant combiner les différents composants de son voyage en fonction de ces préférences à un prix unique, tantôt sur les vols, la location de voiture, réservation de chambre d'hôtel, et trouver des activités de loisirs.

La principale fonctionnalité de package dynamique est de permettre aux consommateurs ou des agents de voyage de regrouper les composants de son voyage en permettant au client de spécifier un ensemble de préférences pour ses vacances et d'accéder dynamiquement à un ensemble de sources d'information touristique afin de trouver le ou les composants nécessaires en temps réel.

La sémantique est devenue un atout majeur pour l'industrie du tourisme. J'aborderai concrètement différentes approches qui démontrent comment l'utilisation des standards du Web (Service) Sémantique qui permet de publier des informations touristiques d'une façon plus riche, et d'en améliorer l'indexation par les principaux moteurs de recherche du web d'une part et de proposer des services enrichies d'autres parts.

### 4.4.1 Approche touristique orienté web sémantique

#### 4.4.1.1 SEED (SEmantic E-tourism Dynamic packaging)

Cardoso et son équipe [Cardoso & Schauer 2006, Cardoso 2006] sont les pionniers à développer une plate-forme pour la construction de packages dynamiques touristiques. Ainsi, Cardoso a défini une architecture faisant partie du projet SEED (SEmantic E-tourism Dynamic packaging) qui repose essentiellement sur l'utilisation des standards du Web sémantique, des Services Web et des processus métier comme le montre la figure 4.3.

L'architecture proposée par cette approche repose sur quatre couches distinctes :

L'intégration des sources d'informations touristiques : Le but de cette couche est de délivrer une base pour l'infrastructure du package dynamique qui nécessite l'intégration automatique des données afin de les interroger d'une manière uniforme à travers de multiples sites Web contenant des informations hétérogènes liés au tourisme.

Toutefois, ces données ne reposent pas sur un standard et engendre une diversité d'interprétation de ces données d'un site à un autre. Due à l'absence de normes pour exprimer les données touristiques, l'extraction de ses informations devient pertinente car les sites web proposent des données totalement hétérogènes comme par exemple les prix des produits et services sont exprimés dans de nombreuses devises (euros, dollars, livres sterling, etc.), les unités de temps exprimées totalement différemment ceux en heures, d'autres en minutes, d'autres en heures et minutes ... etc. L'échelle de l'unité de température n'est pas standard aussi. Elle peut être exprimée en degrés centigrades, ainsi qu'en degrés Celsius.

Afin d'unifier l'expression formelle de ces données, Cardoso a suggéré l'utilisation d'une ontologie de l'e-tourisme fournissant un vocabulaire commun des termes touristiques assurant l'interopérabilité.



#### 4.4. APPROCHES DE PACKAGES DYNAMIQUES SÉMANTIQUES

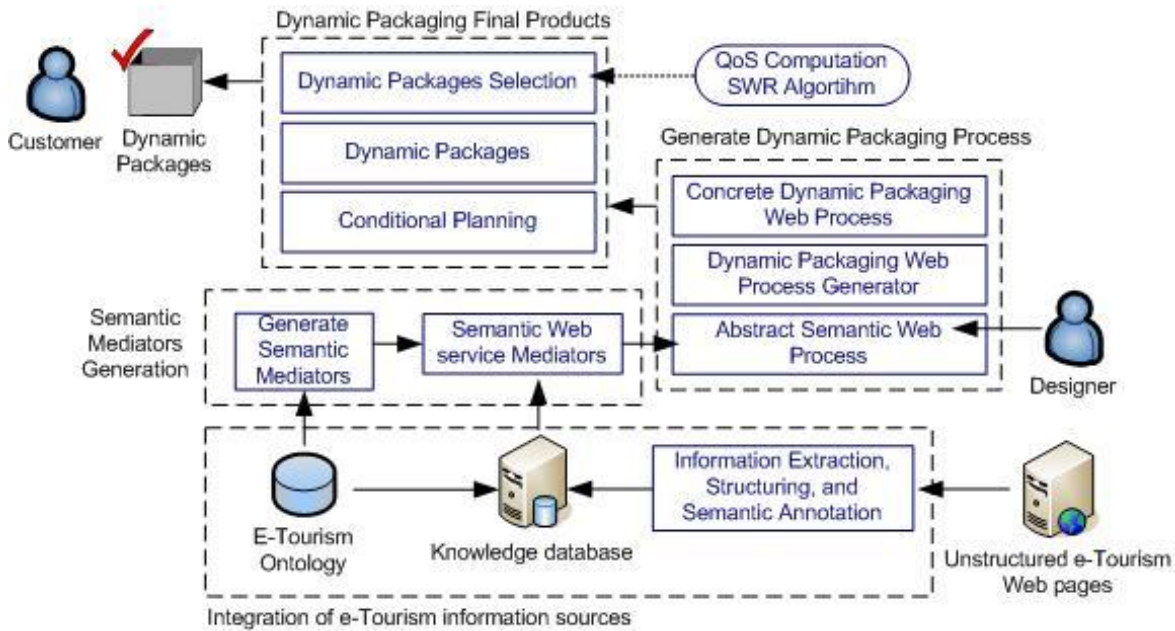


FIGURE 4.3 – Architecture du système Package dynamique du Projet SEED [Cardoso & Schauer 2006, Cardoso 2006]

Pour cela, il a développé une ontologie OWL sous l'éditeur Protégé, une taxonomie et ses axiomes. Cette ontologie couvre toutes les informations nécessaires à la modélisation de l'environnement touristique afin d'exprimer les produits et services touristiques. Cette ontologie doit être capable de répondre à quatre types de questions concernant les prédicats quoi, où, quand et comment, par exemples :

- Que peut faire un touriste tout en restant sur une île ?
- Où sont situés les endroits intéressants à voir et à visiter ?
- Le touriste peut-il visiter un endroit particulier ?
- Comment le touriste peut se rendre à sa destination, que peut-il y faire comme activité ?

L'ensemble des prestataires de l'information touristique possède déjà des sites web qui stockent la description des services et des produits touristiques. Pour cela, il propose simplement l'utilisation d'annotation sémantique pour résoudre les ambiguïtés des données présentes dans les sites Web touristique distincts en utilisant l'ontologie de l'e-tourisme.

L'annotation sémantique est le processus d'insertion des étiquettes au niveau des documents, dont le but est d'injecter de la sémantique au texte. L'ensemble de ces pages annotées sémantiquement sont ensuite stockées dans une base de connaissances.

Génération de médiateurs sémantiques : Les médiateurs sémantiques sont une vue virtuelle intégrant plusieurs sources d'information sémantiques (annotées) du e-Tourisme tout en fournissant un accès uniforme à ces diverses sources. Ces médiateurs sont générés automatiquement et liés entre eux formant ainsi une structure hiérarchique qui est

## CHAPITRE 4. E-TOURISME & PACKAGE DYNAMIQUE

dérivée directement de la taxonomie de l'ontologie de l'e-tourisme où chaque médiateur rassemble et intègre l'information sémantique de ses nœuds enfants dont les feuilles sont une collection d'information provenant de sources de données du e-tourisme qui ont été annotées sémantiquement. Pour mieux expliquer ce concept, La figurefig :Bdfdsf considère une taxonomie partielle extraite à partir de l'ontologie du e-tourisme.

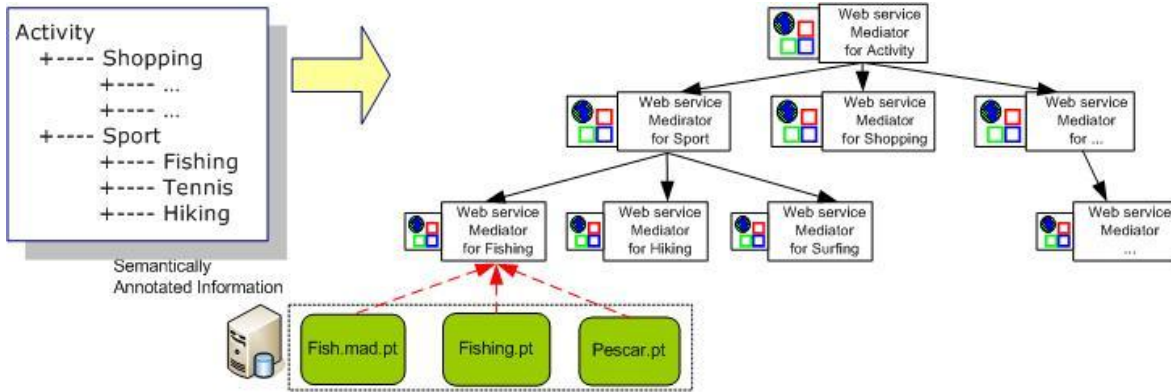


FIGURE 4.4 – Structure hiérarchique des médiateurs sémantiques [Cardoso & Schauer 2006, Cardoso 2006]

Génération des processus abstraits : L'objectif de cette phase est de construire un ensemble de processus métier qui sont capables de composer et combiner différents processus fournissant ainsi des solutions du package dynamique. Initialement, un processus Web abstrait spécifie le flux de contrôle et le flux de données d'une application, mais ne définit pas quels services Web seront utilisés lors de l'exécution. Ainsi faisant abstraction des descriptions de ressources, la portabilité et la réutilisation d'un processus permet la génération d'une nouvelle instance sur la base de différent processus. Comme par exemple, la Figure 4.5 montre un processus Web abstrait conçu pour construire un package dynamique qui inclut une activité de pêche le matin, puis séance shopping, les horaires d'un jeu de golf ou un film dans l'après-midi et un dîner pour la soirée.

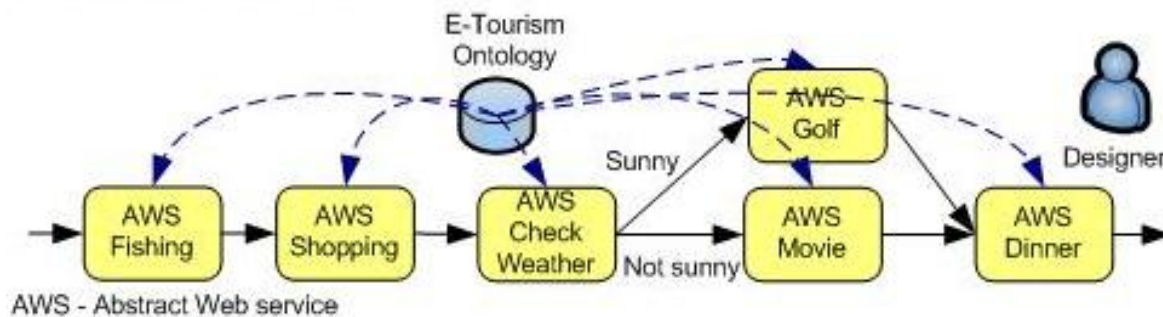


FIGURE 4.5 – Génération des processus abstraits [Cardoso & Schauer 2006, Cardoso 2006]

Génération des processus concrets : Les Processus concrets sont créés et générés automatiquement en utilisant un générateur approprié basé sur la disponibilité des services

Web. Chaque service défini dans le processus Web abstrait est transformé en un service exécutable tout en spécifiant les emplacements de la mise en œuvre de services Web. À ce stade, les processus concrets sont valides d'un point de vue fonctionnel, mais ils ne peuvent pas générer de package dynamique exécutable en raison de contraintes de temps ou de coûts.

Les Processus package dynamique finaux sont créés en utilisant la planification conditionnelle, le classement ainsi que la sélection. La planification conditionnelle est utilisée afin de permettre la génération de packages dynamiques correctes et valides en programmant par exemple l'ensemble des activités touristiques. Néanmoins, certains packages générés peuvent prendre plus de temps à s'exécuter par rapport à d'autres ou sont plus chers du point de vue touriste. Pour ces raisons, la phase suivante est responsable du classement et de la sélection des Package dynamique en fonction d'un ensemble de caractéristiques (qualité de service distincte (Quality of Service)) et répondant aux exigences de l'utilisateur. Une fois les processus évalués, ils sont présentés au touriste qui sélectionne le package dynamique le plus semblable à ses préférences.

### 4.4.1.2 Application de recommandation d'itinéraire personnalisé

L'application CRUZAR [Mànguez *et al.* 2010] met en place une application de recommandation d'itinéraire personnalisé en se basant sur les profils de l'utilisateur. Cette application diffère des autres qui reçoivent la même information, indépendamment de leurs intérêts. Cette application assure aux utilisateurs des informations personnalisées ainsi que des recommandations sur la base de leur profil. CRUZAR peut potentiellement générer un nombre infini d'itinéraires personnalisés mais il en offre la plus pertinente en adéquation au profil de chaque visiteur. Alors que certaines villes ont quelques itinéraires prédéfinis pour les profils les plus courants (tels que "Visiter Saragosse en seulement trois jours"), CRUZAR, est disponible sur le site web de la grande ville d'Espagne de Saragosse (Espagne qui possède une forte densité en terme de POI (Points d'Intérêt) et bénéficiant d'un agenda culturel assez riche et très dynamique. L'idée est de se baser sur des connaissances spécialisées (sous la forme de règles et des ontologies) et un référentiel complet de données pertinentes (instances) pour construire un itinéraire personnalisé pour chaque profil visiteur comme le montre la figure 4.7.

Cette approche collecte les données nécessaires à partir de bases de données relationnelles multilingues existantes et qui sont utilisés pour nourrir le contenu du site Web officiel de Saragosse. Ces données reposent sur plusieurs source d'informations : (1) La base de données CMS qui alimente le site Web du conseil municipal exposant des descriptions sur les monuments et les bâtiments historiques de la ville (plus de 2.000 des ressources), la liste des restaurants, les hébergements, les espaces verts, les zones commerciales et d'autres points d'intérêt pertinents. Cette application propose une mise à jour régulière des informations sur les événements culturels et les activités de loisirs à venir. Ce site Web affiche aussi des photographies de la région. (2) Un système d'information géographique IDEZar hébergé par l'Université de Saragosse utilise les services Web RESTfulres [Richardson & Ruby 2007] qui calcul le chemin le plus court entre deux points de la ville géo-référencées. Toutes ces sources de données sont transformées en données RDF [Carroll & Klyne 2004, Hayes 2004, Lassila & Swick 1999, Ma-

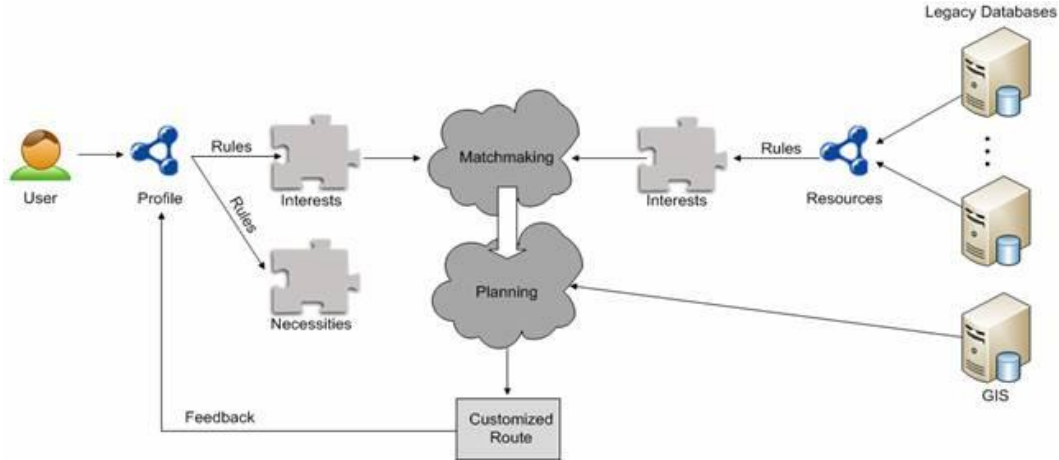


FIGURE 4.6 – Architecture de CRUZAR [Mànguez *et al.* 2010]

nola & Miller 2004, Beckett 2009b] en s'appuyant sur l'ontologie DOLCE<sup>3</sup> en utilisant des adaptateurs spécifiques appliqués régulièrement pour y insérer les données mises à jour. Cette ontologie sert à organiser ces données RDF [Carroll & Klyne 2004, Hayes 2004, Lassila & Swick 1999, Manola & Miller 2004, Beckett 2009b] et s'articule autour de trois types d'entités de domaine : 1) les ressources touristiques de Saragosse, principalement les événements et les POI, 2) l'extraction des préférences et le contexte du profil utilisateur, et 3) la configuration de l'itinéraire. Les événements et POI sont définis en fonction de leurs caractéristiques intrinsèques : la position géographique, le style ou la date artistique. Alors que les profils des visiteurs contiennent des informations sur leurs préférences : la date d'arrivée, la composition du groupe, les activités préférées, etc.

Afin de faire correspondre les informations locales avec les préférences, un vocabulaire commun est nécessaire dont le concept central est les "intérêts". Les préférences des visiteurs sont converties en un ensemble d'"intérêts" sous forme de règles de production qui sont susceptibles d'être satisfaits par les POI et les événements. Ces règles sont relativement simples et compréhensibles par les experts du domaine et sont exécutées à l'aide du moteur de règles Jena<sup>4</sup>.

Tous les POI de Saragosse sont dynamiquement classés reflétant leur "intérêt subjectif", selon le profil de chaque visiteur. À la fin du processus de correspondance, un score numérique est attribué à tous les POI afin de quantifier le niveau anticipé de l'intérêt de l'utilisateur. Initialement tous les POI ont un score fixé par les experts du domaine ("intérêt objectif").

Le processus de correspondance sémantique est exécuté individuellement pour chaque POI, et sa sortie est un score calculé pour la ressource ("intérêt subjectif"). La valeur de ce score dépend de combien les intérêts du visiteur (dérivé de leur profil) sont remplis par chaque POI qui sont triés par leur intérêt subjectif.

CRUZAR implémente un algorithme de matchmaking entre les entités décrites en RDF [Carroll & Klyne 2004, Hayes 2004, Lassila & Swick 1999, Manola & Miller 2004, Beckett 2009b],

3. <http://www.loa.istc.cnr.it/old/DOLCE.html>

4. <https://jena.apache.org/>

#### 4.4. APPROCHES DE PACKAGES DYNAMIQUES SÉMANTIQUES

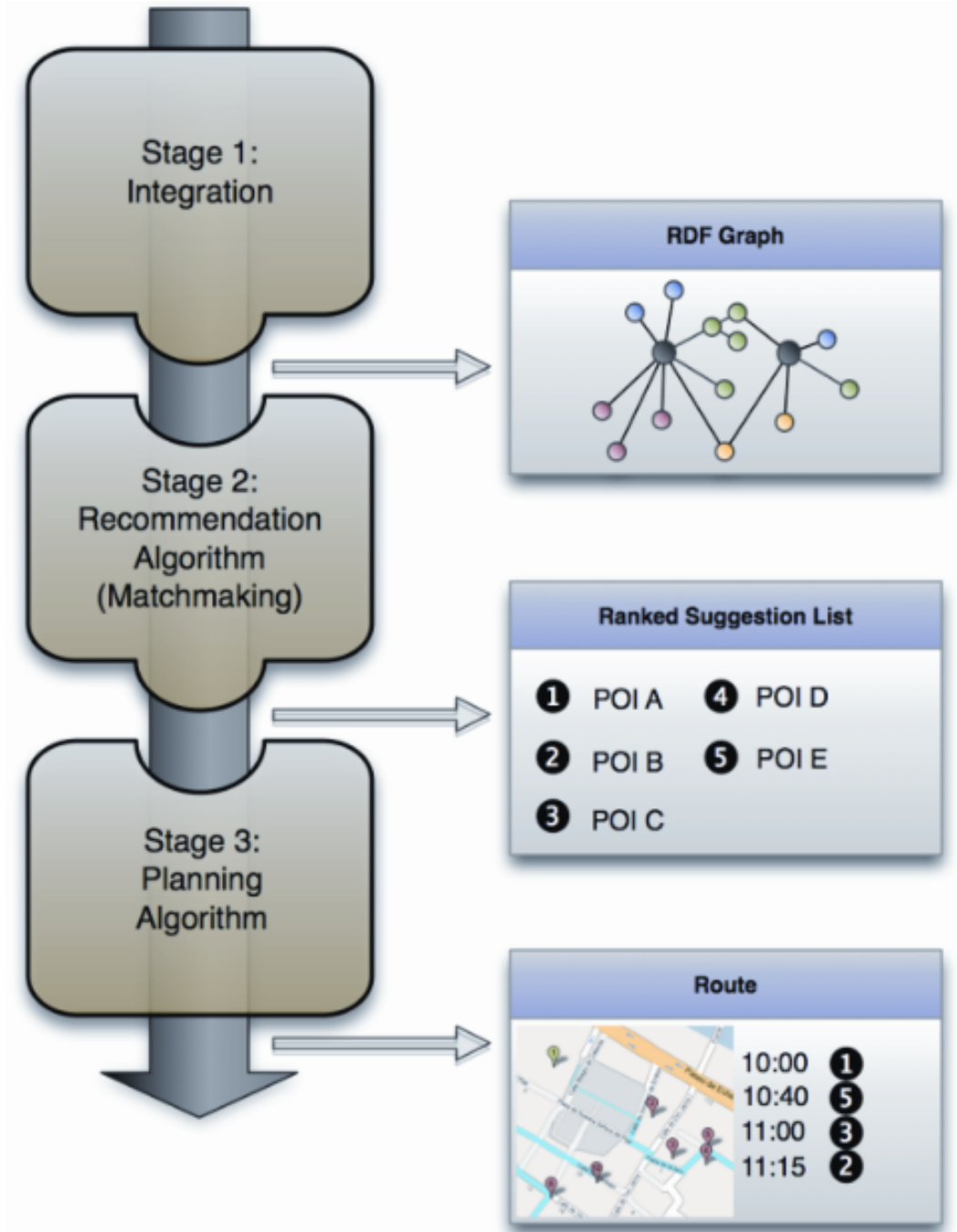


FIGURE 4.7 – Algorithme détaillé de CRUZAR [Mànguez *et al.* 2010]

et fournit les résultats à un algorithme de planification. Cet algorithme de planification est exécuté afin de créer l'itinéraire en équilibrant la distance avec le POI sélectionné tout en autorisant les visiteurs à modifier certaines options afin d'influencer cet équilibre. Le résultat est un itinéraire touristique personnalisé à travers la ville, qui est choisi en fonction de chaque

profil de l'utilisateur et offre un service innovant pour les visiteurs leur permettant de planifier leur voyage à l'avance et ceux en exploitant les connaissances d'experts.

Cette application fournit une interface accessible, riche en information qui comprend : la séquence des POI sélectionnés, un calendrier provisoire pour chaque jour, une carte (un map) mettant en évidence les points d'intérêt, des suggestions d'autres endroits intéressants à proximité de la route, et deux ensembles de restaurants recommandés un pour le déjeuner et l'autre pour dîner à proximité du dernier POI de l'itinéraire. Les activités complémentaires, tels que les événements (concerts, manifestations sportives, etc.) et des magasins, sont également suggérés. L'utilisateur peut interagir avec le trajet généré en modifiant les options proposées.

CRUZAR est une application sémantique créée pour la ville de Saragosse, qui est déjà déployée au niveau des serveurs Web du conseil municipal. Cette application intègre et organise les données provenant de différentes sources afin de capturer toutes les informations sur les itinéraires générés en se basant sur les contraintes.

### 4.4.2 Approche touristique orienté Services Web Sémantiques

#### 4.4.2.1 S-OGSA

L'approche S-OGSA [Corcho *et al.* 2006] est une architecture se référant à la notion de grille sémantique qui étend la technologie OGSA (Open Grid Services de l'Architecture) [Grimshaw *et al.* 2009] avec la sémantique. S-OGSA intègre la notion de grille, du Web sémantique et Services Web Sémantiques ensemble. Pour cela, elle démontre un scénario touristique illustrant plusieurs cas d'utilisation dans la figure 4.8 tout en capturant les exigences de fonctionnalité pour les packages dynamique.

Cas d'utilisation 1 : Une agence de Voyage reçoit la requête d'un touriste traçant son plan de voyage désiré (étape 1), qui la soumet à une organisation virtuelle qui a la capacité d'accéder aux services touristiques et ressources multi-institutionnelle (matériel). Celle-ci la transmet à un agent (étape 2) qui coordonne d'autres agents en décomposant la requête principale en sous-requêtes (étape 3) Pour chaque sous-requête assigné à un agent, des calculs complexes vont être effectués en fonction des SIG et des données sur le trafic routier, les tarifs des vols et d'hébergement qui sont fournis par GDS (Global Distribution System, Système de réservation centralisé) et CRM (Central Réservation Gestion). Enfin, l'agent principal collecte et rassemble les résultats de ses partenaires et retourne le plan à l'agence de voyage (étape 4) qui la restitue au touriste (étape 5)

Cas d'utilisation 2 : Le touriste soumet directement sa requête à l'organisation virtuelle lui permettant ainsi d'effectuer des recherches en fonction des services requis d'une façon autonome. Pour cela, le système a besoin de retracer et d'analyser les préférences et besoins de l'utilisateur, puis de les comparer avec les fonctionnalités des services existants. En figure 4.8, le système découvre et renvoie l'offre de dernière minute après la mise en enchères de ses services en ligne (étape 6, 7).

Cas d'utilisation 3 : Afin de promouvoir les produits et/ou services, certains fournisseurs de services fournissent un marketing multimédia, permettant aux touristes de consulter une quantité de photos d'archives et live streaming vidéo en ligne (étape 8).

Cas d'utilisation 4 : Les services sont en mesure d'accéder en toute transparence à d'autres

#### 4.4. APPROCHES DE PACKAGES DYNAMIQUES SÉMANTIQUES

services et ressources appartenant à différents prestataires de services. Par exemple, un service de pointe fournisseur préfère utiliser le service de gestion des revenus (de l'étape 9), qui est capable de interagir avec d'autres systèmes de vente en ligne (par exemple GDS (Global Distribution System, Système de réservation centralisé)) afin de maximiser les profits de l'entreprise en différenciant les prix des différents produits en ligne.

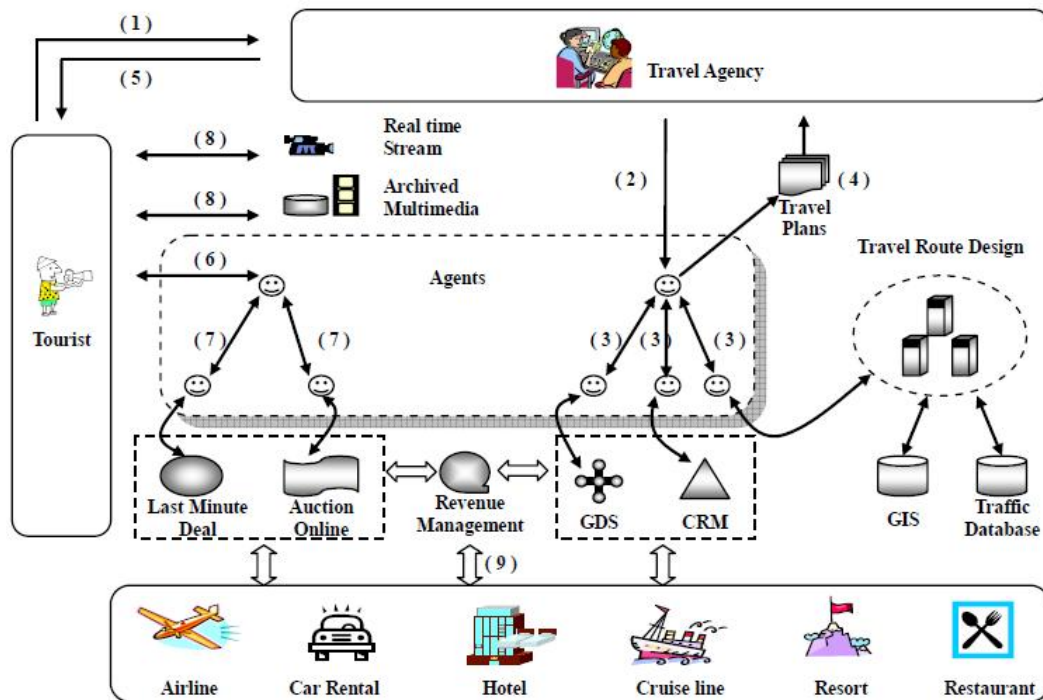


FIGURE 4.8 – Les Cas d'utilisation démontrant les exigences de cette architecture [Corcho et al. 2006]

Pour répondre aux exigences ci-dessus, le Web sémantique, les Services Web, la grille et l'agent ont été adoptés dans cette approche :

- Cette approche réutilise deux ontologies, l'ontologie S-OGSA en l'étendant et réutilise une partie de l'ontologie Globus<sup>5</sup>. Elle contient principalement les concepts ontologiques relatifs aux sur l'OMC (World Organisation du tourisme), un Thesaurus, qui est une normalisation internationale de la terminologie du tourisme, les rôles ontologiques qui contiennent des descriptions des rôles des utilisateurs (tourisme, agence de Voyage et fournisseur de services) , les ressources ontologiques qui indiquent la capacité du matériel, les logiciels et des moyens de communication qui prennent en charge les services, par exemple, CPU la performance et la bande passante réseau et finalement l'ontologie service qui définit un service uniforme fournissant les interfaces et fonctions conformes à la spécification de l'OTA, qui comprend les services aériens, croisière, destination, package dynamique, etc.

5. <http://www.globus.org>

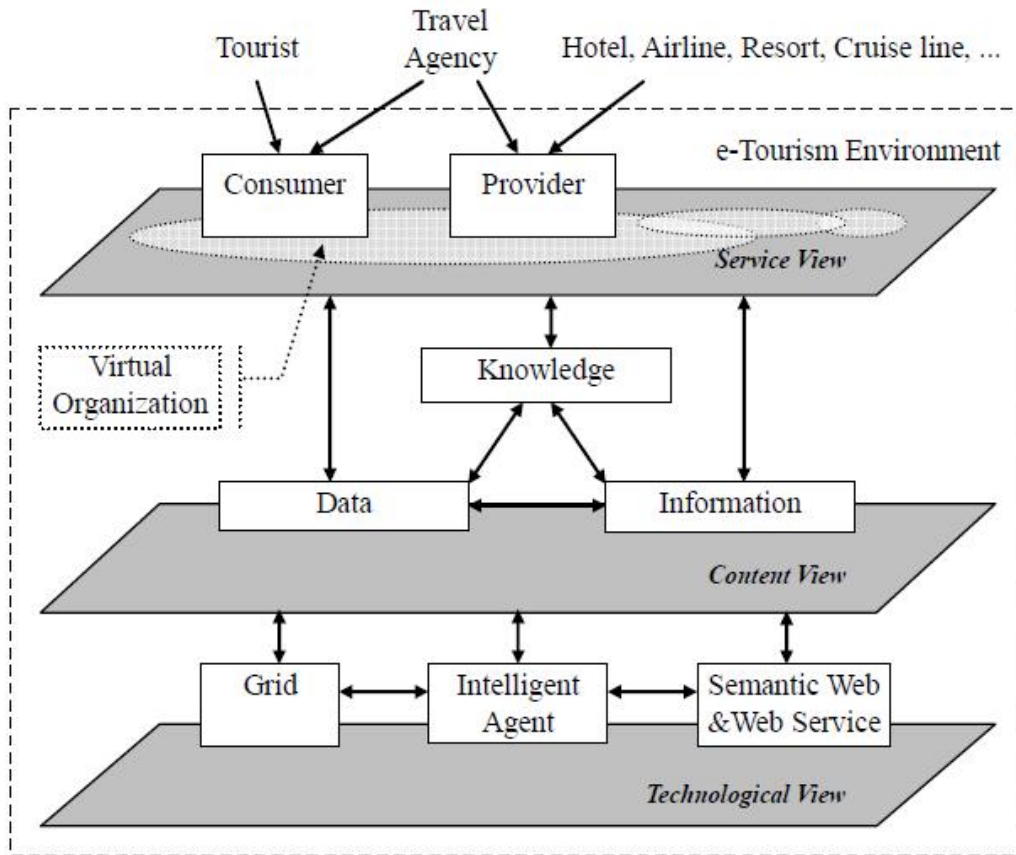


FIGURE 4.9 – L'architecture S-OGSA [Corcho *et al.* 2006]

- Dans ce cadre, la réconciliation sémantique est résolue principalement par un service qui encapsule les détails de la traduction des fournisseurs de service en collaborant avec les ontologies de service afin de gérer la capacité de stockage à travers une ontologie cartographique pour concevoir des légendes de cartes.
- La découverte dynamique de service et de ressources est essentielle pour l'e-tourisme. Cette approche repose sur l'utilisation de deux spécifications : myGrid [Stevens *et al.* 2004] qui met en œuvre la découverte de services en rajoutant une information sémantique aux entités du modèle WSDL présentes dans le registre UDDI et S-MDS (semantic monitoring and Discovery System) qui réalise la découverte des ressources grâce à l'extension de la grille Globus MDS ; La découverte de services dans le réseau est non seulement fondée sur les exigences de fonctionnalités mais caractérisée aussi par la disponibilité et les performances de ressources sous-jacentes.
- L'organisation virtuelle fournit un moyen efficace pour accéder aux services et les ressources entre les prestataires. Toutefois, l'autorisation des utilisateurs ne peut être pré-déterminée statiquement jusqu'à l'exécution. L'approche propose une politique de contrôle d'accès basée sur des rôles spécifiques qui se conforment à la OGSA-AuthZ qui est en mesure de déterminer l'admissibilité des utilisateurs dynamiquement. Sous cette



politique, les rôles des utilisateurs peuvent être déduits en fonction de leurs propriétés lors de l'exécution.

- La nécessité d'utiliser des agents intelligents afin d'effectuer des actions autonomes et accomplir des tâches complexes. Il y a les agents représentant les touristes ainsi que les agences de voyage qui translettent les requêtes, assurent leur suivi et fournissent les résultats. D'autres agents sont utilisés pour gérer les préférences de l'utilisateur, le besoin de tracer ce type de requête et de l'analyser en permanence afin de rafraîchir le profil dynamique de l'utilisateur en fonction de ses intérêts par rapport à un seuil prédéfini. Quant aux entreprises, l'agent du flux de travail doit être doté d'un moteur de raisonnement basé sur les règles déclaratives leur permettant d'inférer de nouvelles connaissances. L'ensemble des ces agents font partie d'un moteur de workflow assurant la collaboration personnalisable entre eux.

### 4.4.2.2 Plateforme sémantique touristique

[Bilbao *et al.* 2010] est une plateforme touristique offrant une approche basée sur la composition de SERVICES WEB SÉMANTIQUES [Jorge 2006, McIlraith *et al.* 2001] exprimée en SAWSDL [Kopecký *et al.* 2007] pour la conception de Packages dynamiques. Elle est basée sur deux modules, le premier module traite les requêtes des touristes en traduisant son profil et le contexte en concepts ontologiques via une ontologie du domaine. Ces concepts ontologiques font partie d'un vocabulaire commun qui contient une classification des produits, des services et des ressources du domaine du tourisme statique et dynamique. Ce vocabulaire est partagé entre les partenaires commerciaux. Cette traduction pour une interprétation correcte de messages, est nécessaire afin de faciliter la découverte et l'invocation des produits et services offerts par des tiers.

Une fois, la requête traduite, elle est transmise sous forme de paquet à un module qui va faire une transformation sémantique à savoir un abaissement (Lowering), cela signifie que les concepts sémantiques de l'ontologie ont besoin d'être transformés en paramètres WSDL [Chinnici *et al.* 2007]. Les résultats de la transformation d'abaissement sont les paramètres d'entrée et l'adresse du fichier WSDL [Chinnici *et al.* 2007] du service Web qui est soumis à la plate-forme ESB. Ces services Web vont exécuter les processus d'affaires définis dans la plate-forme ESB (processus d'orchestration) et une fois réalisée, le résultat contiendra une proposition du package dynamique. La réponse est un message SOAP qui doit être transformé en concepts sémantiques.

## 4.5 Discussion et synthèse

---

Cependant il existe des initiatives stratégiques mais qui se réfèrent assez souvent sur la valorisation unique des ressources touristiques (ressources naturelles, comme les montagnes, les plages et les ressources culturelles, de divertissement) d'une région spécifique n'offrant pas ainsi de services personnalisés qui comprennent les activités et l'information désiré, on peut citer notamment :

- Le projet Harmoniser, HARMOTEN, Harmonet<sup>6</sup> qui ont travaillé sur l'harmonisation

---

6. <http://euromuse.harmonet.org/web/guest/23>

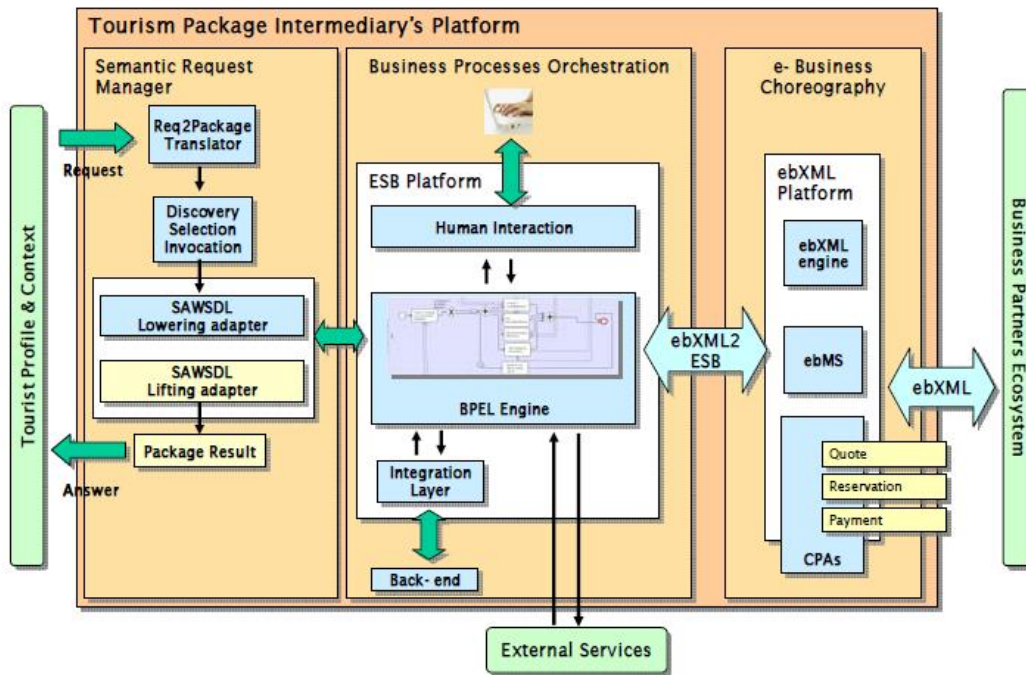


FIGURE 4.10 – Plateforme intermédiaire orienté package dynamique [Bilbao *et al.* 2010]

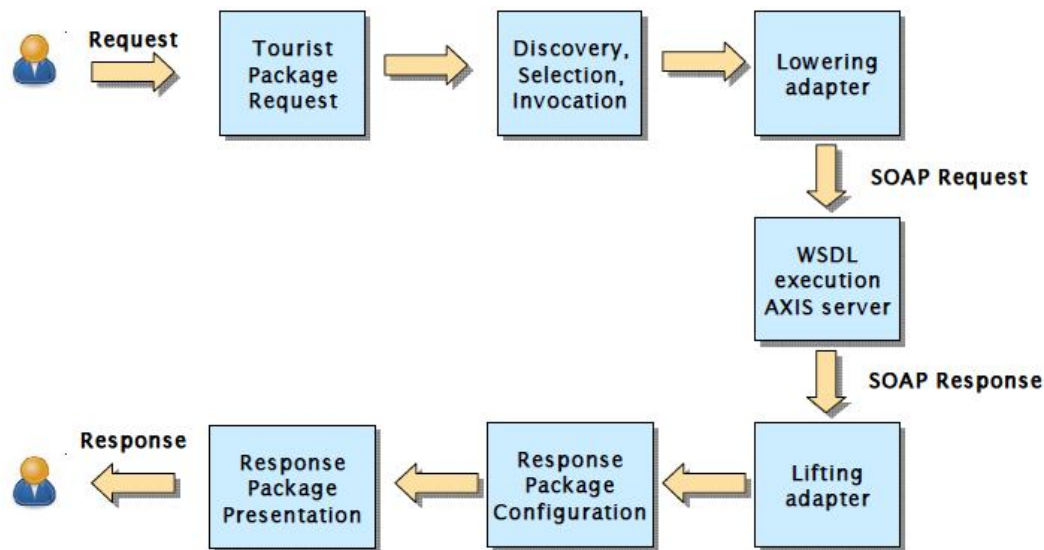


FIGURE 4.11 – Adaptateur Lowering et lifting [Bilbao *et al.* 2010]

des échanges de données dans le domaine touristique.

- Le Projet SATINE (<http://www.srdc.metu.edu.tr/webpage/projects/satine/>) qui a réalisé un cadre d'interopérabilité sémantique sécurisé pour exploiter des plateformes de services Web en conjonction avec les réseaux Peer-to-Peer dans l'industrie du tourisme.

- L’initiative DERI qui a travaillé sur le développement d’un portail sémantique du e-Tourisme reliant les clients et les agents de voyages virtuels de n’importe où et à tout moment en fonction des besoins des utilisateurs.
- Il existe différentes efforts de normalisation dans le tourisme, par exemple par l’Open Travel Alliance (OTA) ou IFITT.

Cependant, avant de détailler les différentes approches sémantique abordant ce domaine, nous démontrons à travers la table 4.1 la prolifération et l’existence [Barta *et al.* 2009, Siricharoen 2010] d’une panoplie d’ontologies touristiques qui ont été développés ces dernières années ainsi que de leur classification 4.13 en fonction du domaine de couverture (du général au spécifique).

Feature	Multilingual	Content	Public	Language
Mondeca Tourism Ontology	No	1000 concepts	No	OWL
Hi-Touch footnote <a href="http://www.mondeca.com/">http://www.mondeca.com/</a> Ontology	No	N.A.	No	OWL
Harmonise footnote <a href="http://www.harmon-ten.org/">http://www.harmon-ten.org/</a> Ontology	No	200 concepts	Yes	RDFS
QALL-ME <sup>7</sup> ontology	No	122 concepts, 55 properties and 52 relations	Yes	OWL DL
DERI e-Tourism Ontology	No	N.A.	Yes	OWL
EON <sup>8</sup> Travelling Ontology	No	N.A.	Yes	OWL
cDott Ontology (2009)	No	N.A.	No	OWL
TAGA <sup>9</sup> Ontology	No	N.A.	Yes	OWL
GETESS Ontology	Yes	1.043 concepts, 201 relations	No	OWL
Cruzar Ontology (2007)	No	34 concepts, 60 properties	No	OWL DL
ebSemantics <sup>10</sup> Ontology (2008)	No	N.A.	Yes	OWL
Hontology	Yes	282 concepts, 39 properties	Yes <sup>11</sup>	OWL

TABLE 4.1 – Les Ontologies touristiques existantes [Barta *et al.* 2009, Siricharoen 2010]

Quant aux approches cités ci-dessus, La première approche **cardoso** [Cardoso & Schauer 2006, Cardoso 2006] propose d’annoter des pages web dédiées au tourisme afin de construire le package dynamique en intégrant des médiateurs sémantiques au dessus de la

## CHAPITRE 4. E-TOURISME & PACKAGE DYNAMIQUE

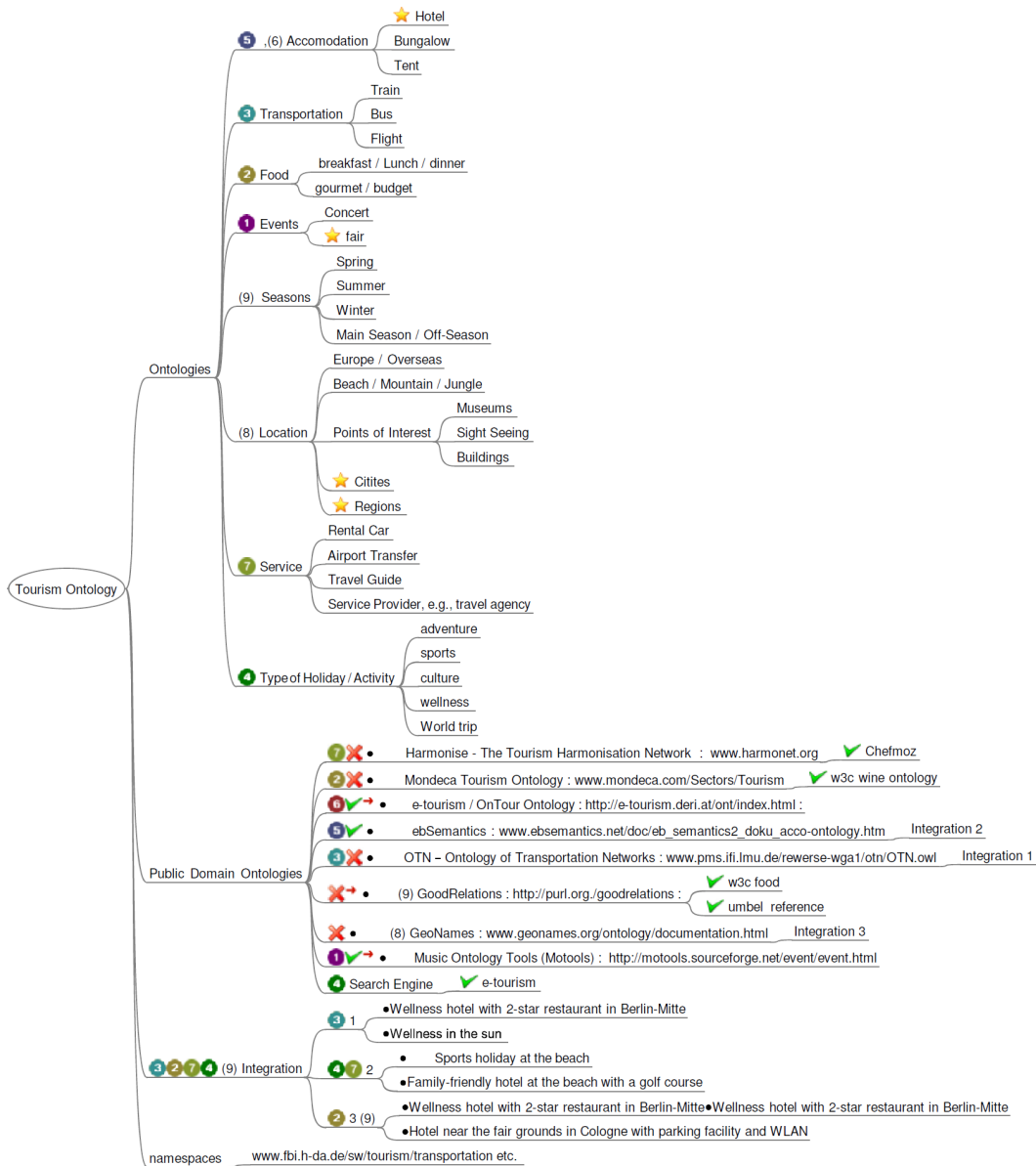


FIGURE 4.12 – Le domaine de l'ontologie touristique

première mais n'aborde aucunement le rafraichissement (les mises à jour) de ces données. Ce dernier point peut altérer le médiateur sémantique. Mais le principal inconvénient de cette approche est qu'elle ne prend pas en charge la redondance des médiateurs ce qui augmenterait largement la taxonomie et les possibilités de construire les Packages dynamiques. Cependant, cette approche n'aborde pas la réaction du système face aux pannes. En conclusion, cette approche a besoin de mûrir pour gérer l'évolutivité, robustesse et fiabilité.

Par contre, l'application **Cruzar** [Mànguez *et al.* 2010] est une application sémantique de recommandation d'itinéraire sur la base des points d'intérêts mais ne prend pas en considération les services de réservation nécessaire à la validation du voyage.

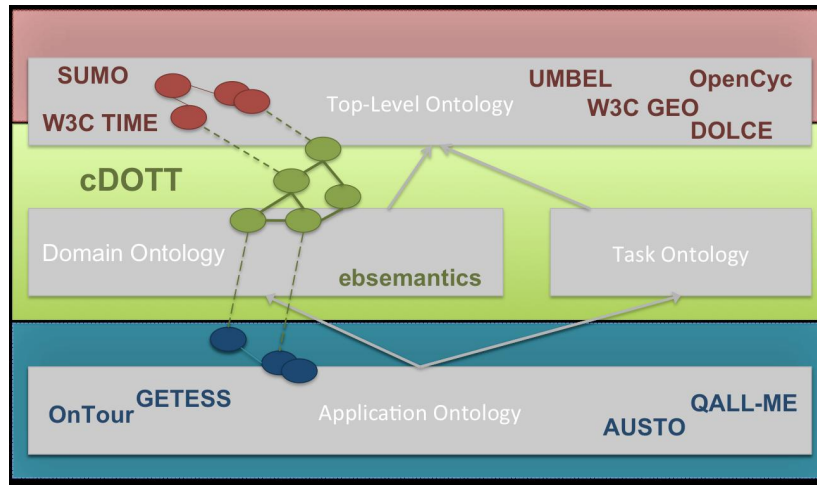


FIGURE 4.13 – Classification des types d’ontologie touristiques [Barta *et al.* 2009, Siricharoen 2010]

L’approche [Corcho *et al.* 2006] utilise la technologie de grille sémantique qui fournit une puissante plateforme. Cette dernière repose sur différentes sources de données touristiques qui sont gérées par des agents intelligents. Un agent principal décompose la requête principale en sous-requêtes qui gèrent la phase de découverte des services des ressources matérielles mais n’autorise à aucun moment la composition de ces services, hors rare sont les services qui puissent répondre à la totalité des exigences des utilisateurs.

Finalement, **Bilbao** [Bilbao *et al.* 2010] est une approche de composition de web services basée sur WSDL [Chinnici *et al.* 2007] basé sur l’abaissement sémantique qui permet de construire les packages dynamiques sur une base non sémantique. Cette approche présente l’inconvénient majeur du à l’abaissement sémantique, l’appariement d’une donnée syntaxique vers un mot sémantique et vice versa est-il pertinent ?

L’ensemble de ces approches abordent la notion de Package Dynamique combinée avec les technologies du web sémantique mais ne décrit pas la réaction du système face aux services et données redondantes disponibles. Comment choisir la solution optimale ? Comment gérer ce flux de données massifs présents sur le Web ?

De nos jours, Le problème auquel les consommateurs sont confrontés concerne le volume d’information et les services disponible qui peut conduire à une difficulté lors du processus décisionnel. L’ensemble de ces approches abordent la notion de Package Dynamique combinée avec les technologies du web sémantique mais ne décrit pas la réaction du système face aux volumes des services et données redondantes disponibles sur le web. Très peu de ces approches abordent l’aspect dynamique, Comment composer les SERVICES WEB SÉMANTIQUES [Jorge 2006, McIlraith *et al.* 2001] transactionnels afin de répondre aux besoins de l’utilisateur ? Comment choisir la solution optimale ? Comment gérer ce flux de données touristiques massifs présents sur le Web ? L’ensemble des approches existantes reposent sur l’utilisation d’une seule et unique ontologie. Cette dernière est-elle suffisante pour couvrir tout les besoins essentiels au développement des Packages dynamique ? L’approche Cardoso

	adaptatif	dynamique	autonome	robuste	ontologie	évolutif
Bilbao et al. (2010) [Bilbao et al. 2010]	✓	✓			~	
Cruzar(2010) [Mànguez et al. 2010]	✓		✓		unique	
S-OGSA (2006) [Corcho et al. 2006]	~	✓	✓	✓	unique	~
Cardoso et al. (2006) [Cardoso & Schauer 2006, Cardoso 2006]	✓	✓	✓		unique	

TABLE 4.2 – Les approches sémantiques touristiques

est la plus plausible et celle qui réponds aux mieux aux contraintes touristiques cependant il ne décrit pas la réaction du système du à l'inflation du volume d'information. D'où, La nécessité est de mettre à disposition un canevas intelligent spécialisé sur le e-tourisme et conçus pour développer des Dynamic Packaging intelligents en temps réel et répondant ainsi au caractéristiques, besoins et attentes du consommateur et ceux afin de personnaliser l'offre d'information et d'accroître la satisfaction des consommateurs et également de raccourcir le temps qu'ils passent à naviguer sur Internet.

## 4.6 Conclusion

Dans ce chapitre, nous avons présenté certains aspects importants du E-tourisme et notamment le concept du Dynamic Packaging. Nous avons exploré les approches existantes et traités un bilan de ces approches afin de nous positionner. Aucune des approches citées auparavant ne propose un formalisme d'interrogation à l'utilisateur pour exprimer ses exigences et besoins en passant par les Données Liées. Aujourd'hui, avec la prolifération du paradigme Données liées, nous assistons à une explosion dans le nombre de services web déployés. Cette augmentation dans le nombre de services candidats nous laisse devant un grand défi : comment choisir les meilleurs candidats qui garantissent une composition optimale ? La problématique à laquelle nous essayons de répondre est : comment tirer profit des données liées pour faciliter la création de services composés ? En vue des éléments présentés dans la section 2.2.5, et de la problématique posée ci-dessus, nous proposons une approche, concrétisée par un canevas nommé SQUIREL [Merazi & Malki 2015, Merazi 2011] , pour la composition de services basée sur les connaissances extraites des jeux de données touristiques. L'idée est de développer un système intelligent sémantique qui soit autonome lors de la collecte de l'information touristique, proposant une intégration transparente de ces données et de fournir une d'application qui utilise la sémantique de données issues des Données Liées et des Services Web Sémantique.

Deuxième partie

**Contributions**





# 5

## Contributions Scientifiques

### Sommaire

---

<b>5.1 Introduction</b>	<b>113</b>
<b>5.2 L'architecture du prototype SQUIREL</b>	<b>115</b>
5.2.1 Problématique	115
5.2.2 Proposition	115
<b>5.3 Couche d'Interrogation Sémantique</b>	<b>118</b>
5.3.1 Interrogation des jeux de Données sémantiques touristiques	122
5.3.2 Interrogation des Services Web Sémantiques	123
<b>5.4 Couche jeu de Données Touristiques</b>	<b>124</b>
<b>5.5 Couche Services Web Sémantique Inter-reliés</b>	<b>129</b>
5.5.1 Modélisation formelle de OWL-S	130
5.5.2 Positionnement de notre approche	133
<b>5.6 Conclusion</b>	<b>152</b>

---

### 5.1 Introduction

---

L'orientation service occupe de plus en plus une place importante dans la structuration des systèmes complexes. La conception et le développement d'applications évoluent progressivement d'un modèle traditionnel vers un modèle plus dynamique orienté services ou la réutilisation et l'adaptabilité jouent un rôle important dans un Web adaptatif. Dans notre travail, nous nommons clients les entités qui ont des besoins en termes de services, tels que les concepteurs de systèmes, et fournisseurs, les entités qui offrent les services, les utilisateurs souhaitant planifier un voyage personnalisé. Selon [Erl 2005], un ensemble de facteurs clés fait converger les clients vers une architecture orientée services. Nous en décrivons trois qui, de notre point de vue, sont essentiels : la réutilisation, la découverte, la composition, l'adaptabilité, et la substitution.

**La réutilisation :** Les unités logiques des architectures des systèmes sont divisées en services dans l'intention de promouvoir leur réutilisation. Nous appelons ces services des services élémentaires. Selon [Nickull *et al.* 2005], cette réutilisation nécessite que chaque service utilisé dans un système basé sur une soit, au préalable, décrit par son fournisseur. L'avantage de la réutilisation est un gain de temps et d'investissement lors de la conception d'une nouvelle application.

**La découverte :** Les services sont conçus afin d'être sélectionnés via des mécanismes de recherche. La découverte est permise par la description préalable des services et leur publication au sein d'un registre. La découverte est un point clé nécessaire à la réutilisation des services élémentaires dans une ARCHITECTURE ORIENTÉE SERVICE [Erl 2005].

**La composition :** Des collections de services peuvent être coordonnées et assemblées afin de former une composition de services. Cette possibilité de construire de nouveaux systèmes à partir de services existants constitue un des avantages de l'ARCHITECTURE ORIENTÉE SERVICE [Erl 2005].

Cependant, ce problème de composition automatique de Web services est par nature très difficile. En effet les données sont instables. Le Web étant dynamique, faire de la composition automatique reviendrait à un problème de planification en environnement dynamique. De plus, Le problème de création d'un nouveau Web service composite est en principe égal au problème classique de programmation automatique généralisée. Ce problème est insolvable de manière notoire en général par aucune technique connue'. Il n'y a pas de raison de croire que la version Web service de ce problème soit moins résistante à une solution générale..

Afin de tenir compte de notre contexte d'utilisation, notre approche de découverte orienté composition de services composites doit être :

- **adaptatif :** l'algorithme devra pouvoir générer de manière automatique les services web « adaptées » aux préférences et contraintes des utilisateurs. Il devra pouvoir repérer par la nature de la relation à extraire entre services web et ceci à chaque nouvelle utilisation ;
- **autonome :** l'intervention d'experts dans le processus de génération du workflow doit être limitée. L'utilisation de démarches s'inspirant des techniques d'intelligence artificielle semble donc opportune ;
- **robuste :** l'algorithme doit être en mesure de faire face au volume important de ressources à traiter sur la toile. Étant donné que les services web sémantique reposent sur différentes ontologies afin de décrire leurs fonctionnalités et que les sources de données utilisent différentes ontologies et vocabulaire afin d'enrichir son contenu.

L'avantage principal de l'architecture orientée services en général et Web en particulier, est que ce paradigme pourrait répondre aux besoins en termes de flexibilité et d'adaptabilité rapide exprimés par les utilisateurs. Une application à base de services est flexible puisqu'un service défaillant peut être remplacé par un autre sans modifier l'ensemble de l'application. De plus, elle est adaptable du fait que le service sélectionné est celui choisi comme étant le meilleur dans un contexte donné. De ce fait le système d'information touristique intelligent a besoin de mûrir en proposant des méthodes pour atteindre l'évolutivité, robustesse et fiabilité.

Dans ce qui suit, nous essayons de clarifier et préciser le canevas SQUIREL [Merazi & Malki 2015, Merazi 2011] présentant la découverte et la sélection orienté composition automatique et dynamique des services Web sémantique pour la construction de package dynamique valide et conforme à la requête de l'utilisateur.

## 5.2 L'architecture du prototype SQUIREL

---

### 5.2.1 Problématique

L'e-tourisme représente déjà la première activité commerciale sur le Web. La majorité des consommateurs ont utilisé Internet pour préparer leur voyage, au point que la vente en ligne a dépassé en valeur la vente traditionnelle en agence. Côté client, on assiste à la naissance d'un nouveau consommateur, en plus grande recherche de qualité et de sécurité, en demande accrue de personnalisation. Un consommateur plus opportuniste, avec un besoin marqué d'inédit et d'originalité, mais aussi d'authenticité. L'élaboration de produits " sur-mesure ", innovants et compétitifs ne sera possible que si les territoires deviennent " intelligents ", appelé " Package Dynamique".

Les Caractéristiques de l'E-Tourisme énonce de nombreux défis. L'un d'eux est le besoin d'interopérabilité entre les systèmes d'information, permettant la transparence des échanges d'informations entre les organismes de tourisme, des services de réservation en ligne.

De nos jours, Le problème auquel les consommateurs sont confrontés concerne le volume d'information et les ressources disponible en terme de service ou information qu'ils soient pertinents ou non avant ou pendant leur voyage et cela peut conduire à une difficulté lors du processus décisionnel.

### 5.2.2 Proposition

La nécessité est de mettre à disposition un canevas intelligent spécialisé sur le e-tourisme et conçu pour développer des Dynamic Packaging intelligents en temps réel et répondant ainsi au caractéristiques, besoins et attentes du consommateur et ceux afin de personnaliser l'offre d'information et d'accroître la satisfaction des consommateurs et également de raccourcir le temps qu'ils passent à surfer sur Internet. Pour cela, Nous pouvons faire appel aux données liées ouvertes et plus précisément les données orienté e-commerce, les ontologies GoodRelations permettant de valoriser les ressources d'un territoire donné en présentant une information riche, précise et pertinente, tout en facilitant l'accessibilité et la compréhension de l'offre du côté des consommateurs. Une telle solution doit permettre aux offreurs / prestataires (territoires, opérateurs ou marchands) de mettre en avant leurs offres en temps réels, et aux consommateurs de disposer d'une application spécialisé permettant la conception de séjours sur mesure à partir d'une information filtrée, organisée, spécialisée et contextualisée leur permettant :

1. une interface d'interrogation sémantique unifié pour la recherche d'information et de services web sémantiques.
2. L'extraction et modélisation de connaissances touristiques et
3. Un système de composition automatique et dynamique de services.

Pour cela, nous proposons une approche de découverte orientée composition de Services Web Sémantiques des processus composites OWL-S qui repose sur une architecture à trois niveaux permettant d'associer un système d'interrogation avec les systèmes de liage de données à travers la mise en correspondance des paramètres des services web sémantiques assurant

leur coordination. L'avantage de notre approche est double, l'utilisateur pourra ainsi interroger les deux couches en utilisant une requête sémantique exprimé en SPARQL divisé en deux parties l'une informatif et l'autre active. La première permet d'interroger les données instanciées à travers les données touristiques liées assurant la phase recherche d'information concernant les produits et offres touristiques et la deuxième permet d'interroger les Services Web Sémantiques en vue de leur combinaison. Ainsi, pour agréger une grande quantité de contenu et services qualifié et pertinent deux solutions se dessinent : les web services et/ou le web sémantique. Cette thèse s'inscrit dans le cadre de travaux relatifs au Web services Sémantique, dans la perspective de la complémentarité et de la coévolution de deux aspects du Web, l'aspect service et sémantique. Le développement du Web au cours de ces dernières années a fait émerger un énorme graphe de données structurées, sémantiques résultant en partie de l'activité des utilisateurs, le DONNÉES LIÉES [Heath & Bizer 2011, Berners-Lee 2009, Bizer *et al.* 2009, Berners-Lee 2006] .

Afin d'améliorer l'expérience de recherche et de réservation de l'utilisateur, nous nous intéressons particulièrement à l'utilisation de ce graphe afin de faciliter l'accès à l'information touristique présente sur le Web, et ce de manière utile, informative et enrichissante pour l'utilisateur d'une part et d'autre part d'assurer d'une manière flexible et dynamique la composition des services web sémantiques transactionnels relatifs à la réservation.

Le canevas SQUIREL [Merazi & Malki 2015, Merazi 2011] est un prototype permettant de développer automatiquement des packages dynamiques et repose essentiellement sur trois couches complémentaires :

D'abord le niveau interrogation sémantique supporte un langage prenant en considération les préférences de l'utilisateur CPSPARQL, un langage étendu du SPARQL ; Une deuxième couche constitue l'ensemble de données liées au domaine de l'e-tourisme permettant de valoriser les offres des prestataires, les GoodRelations. Ensuite le niveau réservation, constitue la réalisation de la composition qui requiert la description et l'organisation de l'interaction entre les services. Elle nécessite la gestion de plusieurs aspects comme les échanges de données entre les services, les pannes ou erreurs éventuelles, le contexte d'interaction, le degré d'automatisation des tâches, etc.

Nous proposons le canevas SQUIREL, une architecture multi-couches disposant chacune d'une fonctionnalité bien précise comme mentionné sur la Figure 5.1 :

L'extraction des données et Services Web sémantiques peut être filtré sur la base de leur catégorisation NAICS (North American Industry Classification System code for your company <http://www.census.gov/eos/www/naics/>) orienté tourisme.

- Transportation
  - Air transportation
    - 4811 Scheduled air transport
    - 4812 Non-scheduled air transport
  - All other transportation industries
    - 4821 Rail transportation
    - 4831 Deep sea, coastal and great lakes water transportation
    - 4832 Inland water transportation
    - 4851 Urban transit systems
    - 4852 Interurban and rural bus transportation

## 5.2. L'ARCHITECTURE DU PROTOTYPE SQUIREL

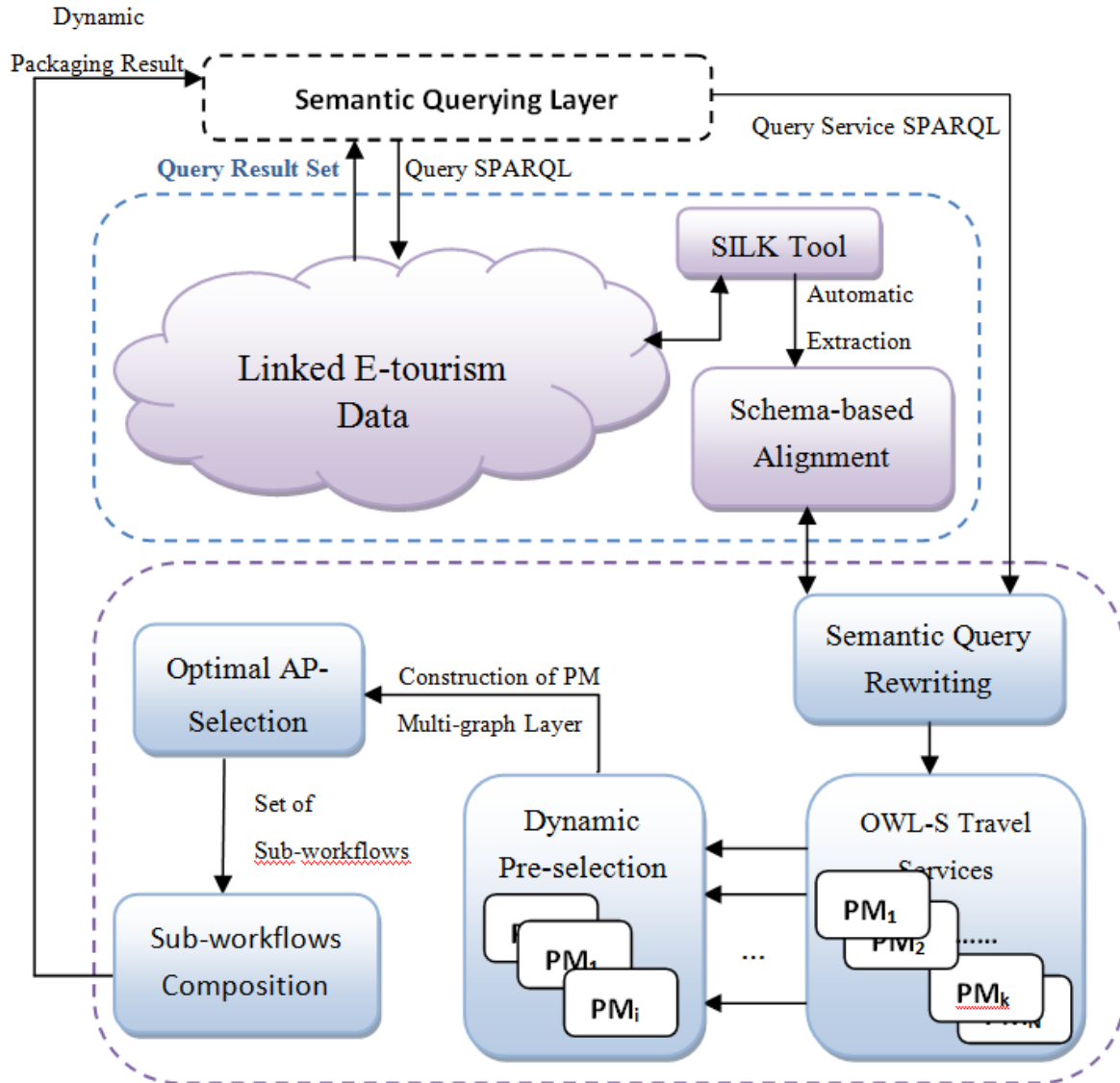


FIGURE 5.1 – SQUIREL : l'architecture touristique proposée

- 4853 Taxi and limousine service
- 4854 School and employee bus transportation
- 4855 Charter bus industry
- 4859 Other transit and ground passenger transportation
- 4871 Scenic and sightseeing transportation, land
- 4872 Scenic and sightseeing transportation, water
- 4879 Scenic and sightseeing transportation, other
- 5A05101 Automotive equipment rental and leasing
- Accommodation
  - 7211 Traveller accommodation
  - 721A1 RV (recreational vehicle) parks and recreational camps

- Food and beverage services
  - 5615 7220 Food services and drinking places
- Travel services
  - 5615 Travel arrangement and reservation services
- Recreation and entertainment
  - 51213 Motion picture and video exhibition
  - 7110 Performing arts, spectator sports and related industries
  - 7121 Heritage institutions
  - 713A1 Amusement and recreation industries
  - 7131 Amusement parks and arcades
  - 7132 Gambling industries
  - 7139 Other amusement and recreation industries

Afin de mieux comprendre notre approche, nous proposons l'exemple suivant qui sera traité à chaque étape où l'utilisateur souhaiterait réserver une chambre d'hôtel dans la ville où se déroule le congrès et éventuellement consulter les conditions météorologique de cette ville { **s :Eventreservation, s :lodgingreservation, wea :weatherconditiong** }, les principales données que fournit l'utilisateur est l'ensemble des entrées suivantes : le nom de l'événement, ces coordonnées personnels ainsi des données sur sa carte de crédit et le nom de l'hôtel { **tio :event, foaf :person, cc :creditcard, acc :hotel** }.

Pour cela, l'utilisateur dispose de quarts Process Model, le premier concerne le service événement qui dispose de quarts processus atomiques permettant à l'utilisateur de sélectionner l'événement désiré et de pouvoir s'y enregistrer tout en fournissant deux moyens de paiement au choix (bancaire ou par carte de crédit). Le deuxième service concerne la réservation d'un moyen d'hébergement regroupant une séquence de deux processus atomiques, le premier assure la sélection d'un moyen d'hébergement et le deuxième assure un paiement par carte de crédit uniquement. Le troisième service concerne la réservation d'une chambre d'hôtel qui est similaire au précédent, par le biais de trois processus atomiques, le premier assure la recherche d'une liste d'hôtels pour une ville donnée puis effectue une réservation. Le quatrième processus permet de consulter les conditions météorologiques de n'importe quel endroit dans le monde et offre deux processus atomiques au choix, en fonction des données insères, soit en fournissant le nom de la ville ainsi que du pays ou bien en fournissant la géolocalisation directement.

Ces services répondent exactement à la requête de l'utilisateur. Nous remarquerons que le service événement fournit des données pour les trois restants et les deux services d'hébergements répondent entièrement à la demande de l'utilisateur, comment choisir l'un deux ?

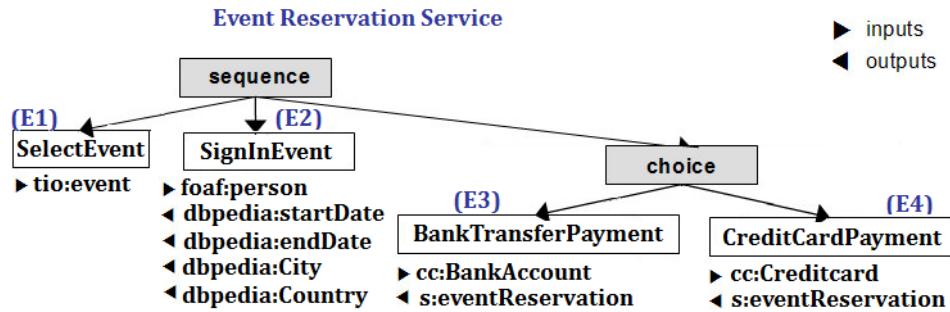
### 5.3 Couche d'Interrogation Sémantique

---

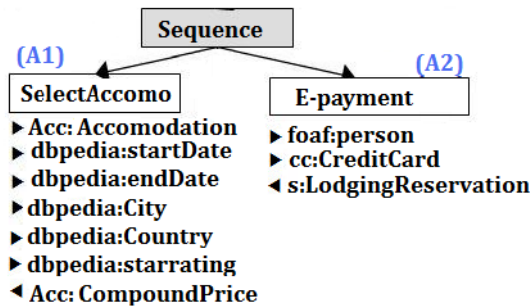
L'objectif de cette couche est de développer un prototype « customer-centric » capables de résoudre les problèmes d'interopérabilité et d'accès aux stocks en temps réel, de répondre de manière ultra personnalisée aux requêtes des clients, et d'agréger, d'organiser, de réserver et de présenter de manière pertinente toujours plus de contenus informatifs pour enrichir les produits touristiques et les services de réservation.

De nos jours, Le problème auquel les consommateurs sont confrontés concerne le volume

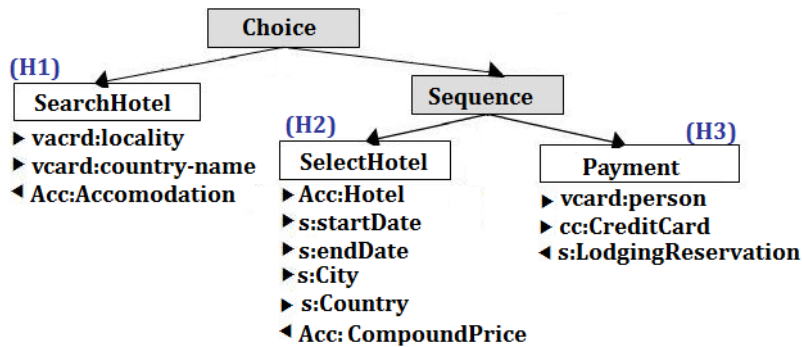
### 5.3. COUCHE D'INTERROGATION SÉMANTIQUE



**Accomodation Reservation Service**



**Hotel Reservation Service**



**Weather Service**

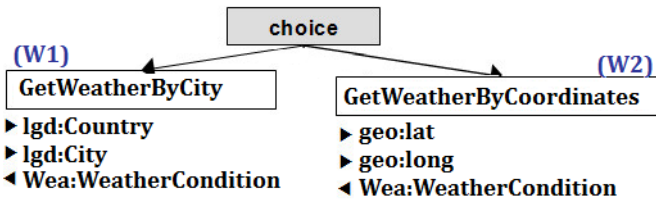


FIGURE 5.2 – Cas d'étude durant cette partie

d'information et les ressources disponible en terme de service ou information qu'ils soient pertinents ou non avant ou pendant leur voyage et cela peut conduire à une difficulté lors du processus décisionnel. Vu la quantité de triplets RDF importantes et le nombre de Service Web disponibles à travers le web. La nécessité de définir une seule interface servant de point d'accès pour l'interrogation de ces deux paradigmes devient incontournable. Pour des raisons pratiques et afin de faciliter l'accès à l'information et les Process Model, nous proposons d'utiliser une extension du langage sémantique SPARQL qui offre la même fonctionnalité à savoir l'interrogation sémantique (1) des données touristiques (base de connaissance) RDF représentant les produits et offres des prestataires afin d'obtenir les informations précises et concises et répondant à sa requête et (2) d'invoquer d'autre part les Services Models pertinents susceptible de faire partie du workflow ou plan optimal et final.

Cette couche orienté utilisateur assure l'interrogation des deux couches en dessous. Pour cela, on exploitera une extension de SPARQL permettant ainsi d'exprimer les besoins, les exigences et les préférences de l'utilisateur de façon non ambiguë et claire.

Pour cela, nous allons redéfinir ce qu'est SPARQL. est un langage qui s'approche de la syntaxe Turtle de RDF et s'inspire fortement de celle d'SQL ce qui la rend facile à appréhender et surtout facile à utiliser en connaissant la structure du graphe qui est à interroger. peut être utilisé pour exprimer des requêtes auprès de plusieurs sources de données, que les données soient physiquement stockées sous forme de graphe RDF [Carroll & Klyne 2004, Hayes 2004, Lassila & Swick 1999, Manola & Miller 2004, Beckett 2009b] ou non. souffre cependant de différentes limites. Par exemple, il ne propose pas pour le moment les fonctions d'agrégat, ni la possibilité d'ajouter des données dans un graphe, SPARQL étant uniquement dédié à des requêtes en lecture seule.

Dans son instanciation stricte, il ne propose que des requêtes en fonction de la sémantique de RDF et n'interprète donc pas les vocabulaires exprimés en RDFS ou OWL. Néanmoins, la plupart des langages de requêtes qui sont basées sur la sémantique de RDF, comme SPARQL, n'ont pas la capacité d'exprimer et d'extraire des chemins, ce qui est nécessaire pour de nombreuses applications. Par exemple, si l'on veut vérifier s'il existe un itinéraire d'une ville à l'autre. Dans cette approche, les requêtes sont des expressions régulières, et une réponse est une paire de sommets reliés par au moins un chemin du graphe dont la concaténation des étiquettes des arcs forme un mot qui appartient au langage engendré par l'expression régulière.

Plusieurs extensions de SPARQL ont été proposées pour interroger les données RDF en fonction de vocabulaires RDFS et d'ontologies OWL et permettent d'ajouter de nouvelles fonctionnalités. Citons pas exemple la recherche par chemins et plus uniquement par triplets (sparqler [Kochut & Janik 2007]) ou l'approximation de requêtes (ispsparl [Kiefer *et al.* 2007a]), des fonctionnalités proches étant implémentées dans le moteur SPARQL CORESE [Corby *et al.* 2004].

Dans l'approche [Alkhateeb 2005], l'auteur a proposé deux extensions de SPARQL ajoutant plus d'expressivité, la première est caractérisé par l'expression de chemins de longueurs variables (aléatoire) sur les prédicats dans un graphe RDF, qui permet d'utiliser des expressions régulières comme prédicats dans un triplet RDF par exemple, «Est-ce qu'il existe un voyage de la ville Oran à la ville Constantine?», appelé Path SPARQL (PSPARQL [Alkhateeb 2005, Alkhateeb 2008]). Comme fait auparavant dans les bases de données, chaque



### 5.3. COUCHE D'INTERROGATION SÉMANTIQUE

expression régulière peut encoder chemins LAR réglementaires dans un graphe RDF tels que la concaténation des arcs étiquettes dans chaque trajet former un mot qui appartient à la langue générée par l'expression régulière.

Les graphes PRDF [Alkhateeb *et al.* 2005, Alkhateeb *et al.* 2008, Alkhateeb 2008] servent à définir une extension de SPARQL, le langage de base PPARQL, remplaçant les motifs de graphes RDF utilisés dans SPARQL par des graphes PRDF, *i.e.*, des motifs de graphes avec des expressions régulières.

**Définition 5.3.1.** Soit  $\Sigma$  un alphabet. Un langage sur  $\Sigma$  est un sous-ensemble de  $\Sigma^*$  dont les éléments sont de  $\Sigma$  appelé mots. Un mot non vide  $\langle a_1, \dots, a_k \rangle$  dénoté  $a_1 \cdot \dots \cdot a_k$ . Si  $A = a_1 \cdot \dots \cdot a_k$  et  $B = b_1 \cdot \dots \cdot b_q$  sont deux mots issus de  $\Sigma$ , alors  $A \cdot B$  est un mot de  $\Sigma$  définie par  $A \cdot B = a_1 \cdot \dots \cdot a_k \cdot b_1 \cdot \dots \cdot b_q$ .

**Définition 5.3.2** (Motif d'expression régulière). Soit  $\Sigma$  un alphabet,  $X$  un ensemble de variables, l'ensemble  $\mathcal{R}(\Sigma, X)$  des motifs d'expression régulières est défini intuitivement :

- $\forall a \in \Sigma, a \in \mathcal{R}(\Sigma, X)$  et  $!a \in \mathcal{R}(\Sigma, X)$  ;
- $\forall x \in X, x \in \mathcal{R}(\Sigma, X)$  ;
- $\epsilon \in \mathcal{R}(\Sigma, X)$  ;
- Si  $A \in \mathcal{R}(\Sigma, X)$  et  $B \in \mathcal{R}(\Sigma, X)$  alors  $A|B, A \cdot B, A^*, A^+ \in \mathcal{R}(\Sigma, X)$ .

**Définition 5.3.3** (un graphe PRDF). un triplet PRDF est un élément de  $\mathcal{U} \cup \mathcal{B} \times \mathcal{R}(\mathcal{U}, \mathcal{B}) \times \mathcal{T}$ . Un graphe PRDF est un ensemble de triplets PRDF.

**Définition 5.3.4** (motif de graphe PPARQL). Un motif de graphe PPARQL est défini comme suit :

- Chaque graphe PRDF est un motif de graphe PPARQL ;
- Si  $P_1$  et  $P_2$  sont deux motifs de graphes PPARQL et  $K$  est une contrainte SPARQL, alors  $(P_1 \text{ AND } P_2), (P_1 \text{ UNION } P_2), (P_1 \text{ OPT } P_2),$  and  $(P_1 \text{ FILTER } K)$  sont des motifs de graphes PPARQL.

Syntax Form	Matches
$\text{- elt}$	Inverse path (object to subject).
$(\text{elt})$	A group path $\text{elt}$ , brackets control precedence.
$\text{elt1} \cdot \text{elt2}$	A sequence path of $\text{elt1}$ , followed by $\text{elt2}$
$\text{elt1} \mid \text{elt2}$	A alternative path of $\text{elt1}$ , or $\text{elt2}$ (all possibilities are tried)
$\text{elt}^*$	A path of zero or more occurrences of $\text{elt}$ .
$\text{elt}^+$	A path of one or more occurrences of $\text{elt}$ .
$\text{elt}?$	A path of zero or one $\text{elt}$ .

FIGURE 5.3 – La syntaxe du Motif d'expression régulière PPARQL

**Définition 5.3.5** (Répondre à une requête PSPARQL). Soit  $SELECT \vec{B} FROM u WHERE P$  une requête PSPARQL tel que  $P$  est un motif de graphe CPRDF, et  $G$  un graphe RDF identifié par une URI  $u$ , L'ensemble des réponses à cette requête est fournis ainsi :

$$\mathcal{A}^*(\vec{B}, G, P) = \{\sigma|_{\vec{B}}|G \models_{PRDF} \sigma(P)\}$$

Cependant, PSPARQL [Alkhateeb et al. 2005, Alkhateeb et al. 2008, Alkhateeb 2008] ne permet pas de spécifier des propriétés sur les nœuds qui appartiennent à un chemin défini par une expression régulière. Il est donc impossible d'enrichir une requête sémantique par exemple, "En outre, l'une des arrête doit fournir un accès sans fil au sein la chambre d'hôtel". Pour pallier à ce problème, [Alkhateeb 2008] propose l'extension de PSPARQL, appelé Constrained Path SPARQL (CPSPARQL [Alkhateeb et al. 2008, Alkhateeb et al. 2005]). Cette extension permet l'expression des contraintes sur les sommets des chemins traversés et renforce le processus de recherche pour trouver des chemins qui satisfont les modèles de graphes impliquant des expressions de chemin . Afin d'ajouter plus d'expressivité à Plus précisément, [Alkhateeb et al. 2008] a définit CPRDF (Pour Constrained Path RDF) qui étend la syntaxe et la sémantique de PRDF pour gérer les contraintes sur les sommets dans les chemins traversés.

[Alkhateeb et al. 2008]a définit d'abord une sorte d'expressions régulières constraints qui étend les habituels avec des contraintes (contraintes permettant de réduire l'espace de recherche en sélectionnant tout en faisant des nœuds de processus d'appariement des contraintes satisfaisants). Ensuite, il utilise des expressions régulières constraints d'étendre les graphes RDF (à savoir les modèles de base graphique de SPARQL) d'avoir des graphiques CPRDF (pour Constrained Chemins RDF). Les Motifs de graphe CPSPARQL sont construits sur le dessus de CPRDF de la même manière que SPARQL est construit au-dessus de RDF.

L'utilisateur final doit pouvoir accéder aux connaissances, et ce sans être confronté aux formalismes employés par les bases de connaissances avec lesquelles il interagit. Cette couche orienté utilisateur fournit une interrogation efficace des données touristiques du web sémantique d'une part et des services Web Sémantiques d'autre part. Par conséquent, dans ce travail, le modèle de processus ontologie est utilisée à la fois pour la découverte de services et de la composition et est expliqué en utilisant une étude de cas. Cette étude de cas décrit notre approche pour la réutilisation et l'intégration de l'information présente dans deux différents entreprises.

### 5.3.1 Interrogation des jeux de Données sémantiques touristiques

Pour interroger le jeu de données, l'utilisateur doit se conformer aux vocabulaires utilisés et définies dans le jeu de données touristiques définis dans la deuxième couche, voici un exemple d'une requête exprimée en CPSPARQL, qui cherche s'il existe un itinéraire direct ou indirect allant de la ville d'Oran vers une autre ville dont la capitale est l'Espagne qui n'utilise que les avions comme moyen de transport avec des chemins de longueur variable, qui est exprimé par l'expression régulière suivante :

`(ex:from-%const1%.ex:to)+`

[Requête  $Q_1$ ]

```

SELECT ?City
WHERE { CONSTRAINT const ]ALL ?Trip]: { ?Trip rdf:type ex:Plane. }
      ex:Oran (ex:from-%const1%.ex:to)+ ?City.
      ?City ex:cityIn ex:Spain.
}
```

Listing 5.1 – Exemple de requête  $Q_1$  basé sur les expression régulière en CPSPARQL

#### 5.3.2 Interrogation des Services Web Sémantiques

Partant du point où l'ensemble des descriptions des Services Models sont définies en termes de concepts ontologiques référencés dans le jeu de données touristique qui repose sur l'utilisation de plusieurs vocabulaires/ontologies. Initialement, pour le pré-traitement proposé, la requête de l'utilisateur est formulée en SPARQL qui est ensuite analysée afin d'en extraire les descriptions sémantiques (la signature) désirés. Le répertoire est filtré afin que seuls les services qui utilisent ces concepts ou ceux liés à ces concepts sont sélectionnés pour devenir pré-candidat au processus de découverte ultérieure. La phase de filtrage proposé repose sur la définition de deux filtres totalement différents et exprimés en requêtes SPARQL, à savoir le  $Q_{ALL}$  et le  $Q_{SOME}$ .

Afin de simplifier les définitions des deux filtres, nous notons,  $C_{S_i}$  la liste du sous-ensemble des concepts du service  $S_i \subset S$  et  $C_Q$  est le sous-ensemble de concepts définit dans la requête  $Q$ . ( $C_{S_i} \subset O$   $C_Q \subset O$ )

- Le premier filtre renvoie uniquement les services dont les descriptions sémantiques englobent l'ensemble des concepts sémantiques mentionnés par la requête utilisateur, c.à.d. que les services doivent satisfaire totalement chaque terme (concept) définit dans la requête par l'utilisateur.  $Q_{ALL}(S, Q) = \{S_i \in S : C_Q \subseteq C_{S_i}\}$ . L'exemple suivant cherche à identifier le web service prenant en considération l'hôtel, évènement, les coordonnées personnelles de l'utilisateur qui renvoie les conditions météorologique de la ville ou se déroule la conférence ainsi que la confirmation de sa réservation pour la chambre d'hôtel et pour la conférence.

```

SELECT ?a
WHERE {
?a proc:hasInput. proc:parameterType foaf:Person.
?a proc:hasInput. proc:parameterType cc:CreditCard.
?a proc:hasInput. proc:parameterType acc:Hotel.
?a proc:hasInput. proc:parameterType tio:Event.
?a proc:hasOutput. proc:parameterType s:EventReservation.
?a proc:hasOutput. proc:parameterType s:LodgingReservation.
?a proc:hasOutput. proc:parameterType w:WeatherCondition.
}
```

Listing 5.2 – Exemple de filtre  $Q_{ALL}$  en SPARQL

- Le deuxième filtre sélectionne les services dont la description sémantique qui se réfère à certains ou quelques (au moins un) concepts sémantiques définis dans la requête de l'utilisateur, c.à.d. que ces services peuvent répondre ou satisfaire partiellement les besoins et/ou préférences de l'utilisateur dans une certaine mesure et ce malgré les informations manquantes  $Q_{SOME}(S, Q) == \{S_i \in S : C_Q \cap C_{S_i} \neq \emptyset\}$ . voici la reformulation de l'exemple précédent en utilisant le filtre  $Q_{SOME}$ .

```
SELECT ?a
WHERE {
  {?a proc:hasInput . proc:parameterType foaf:Person.}
UNION
  {?b proc:hasInput. proc:parameterType cc:CreditCard.}
UNION
  {?c proc:hasInput . proc:parameterType acc:Hotel.}
UNION
  {?d proc:hasInput. proc:parameterType tio:Event.}
UNION
  {?e proc:hasOutput. proc:parameterType s:eventReservation.}
UNION
  {?f proc:hasOutput. proc:parameterType s:LodgingReservation.}
UNION
  {?g proc:hasOutput. proc:parameterType w:WeatherCondition.}
}
```

Listing 5.3 – Exemple de filtre  $Q_{SOME}$

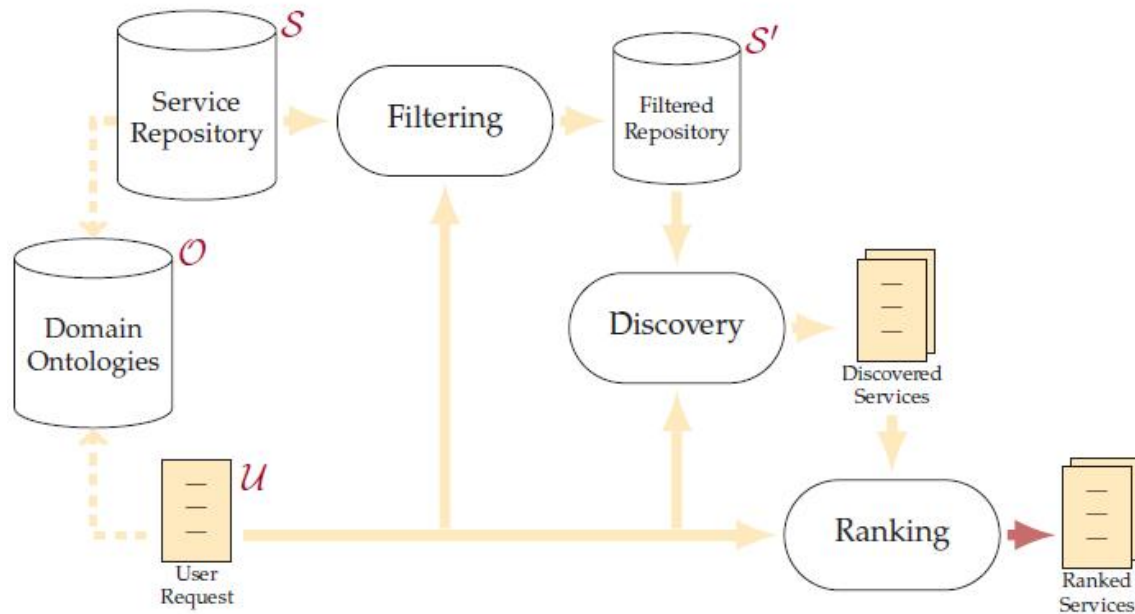
Cependant, cette approche n'a pas la prétention de fournir un nouveau mécanisme de découverte sémantique mais de l'améliorer en introduisant une étape de pré-traitement, le filtrage basé sur le langage d'interrogation sémantique exprimé en SPARQL ( $Q_{ALL}(S, Q) \subseteq Q_{SOME}(S, Q) \subseteq S$ ). Ce dernier raffine la phase de découverte des Processus composite (process Model OWL-S) en un sous-ensemble de services pertinents améliorant considérablement le matchmaking de services. Cette approche utilise les requêtes sémantiques qui présentent une complexité linéaire sur la taille de l'ensemble du patron de données, de ce fait elle n'augmente pas le temps d'exécution à l'égard du processus de matchmaking.

Cette approche implémente deux filtres purement logiques mais ne prend pas en considération les relations sémantiques non-logiques entre les concepts définissant ainsi l'intra-dépendance entre service (Voir Section 5.5.2.1)

## 5.4 Couche jeu de Données Touristiques

---

L'utilité et la pertinence des données se trouvent dans l'information qui peut en être extraite. Etant donné le taux élevé de publication des données sémantiques (auto-descriptives et hétérogènes) et leur complexité accrue, l'enjeu est la manipulation efficaces de données touristiques. Pour cela, Nous pouvons faire appel aux données orienté e-commerce, et plus spécialement les GoodRelations permettant de valoriser les ressources d'un territoire donné en présentant une information riche, précise et pertinente, tout en facilitant l'accessibilité et

FIGURE 5.4 – Le prototype EMMA [Garcàa *et al.* 2012]

la compréhension de l'offre du côté des consommateurs. Une telle solution doit permettre aux offreurs / prestataires (territoires, opérateurs ou marchands) de mettre en avant leurs offres en temps réels, et aux consommateurs de disposer d'une application spécialisée permettant la conception de séjours sur mesure à partir d'une information filtrée, organisée, spécialisée et contextualisée leur permettant :

L'information disponible est, cependant, souvent incomplète, erronée et enfermée dans une multitude de sites web. Notre objectif est d'assurer l'utilisation d'une complétude des données assez riche. En outre, des milliers de données touristiques sont créés chaque jour, ce qui nécessite un temps de recherche considérable cote expérience utilisateur. Afin d'assurer à l'utilisateur une recherche de d'information efficace, nous employons dans cette couche la technologie des DONNÉES LIÉES [Heath & Bizer 2011, Berners-Lee 2009, Bizer *et al.* 2009, Berners-Lee 2006] . Ce dernier est constitué autour d'un noeud centrale qu'est DBpedia et il est interconnecté avec d'autres sources de données intégrant différentes ontologies /vocabulaire et couvrant diverses domaines dont l'un d'entre eux le tourisme. Les Données Liées reposent sur le standard RDF. L'un des avantages les plus importants à notre problématique est qu'il facilite l'intégration des données de sources variées et permettant une évolution des schémas de données au cours du temps sans que cela ne nécessite une modification au niveau des clients.

L'objectif principal est d'intégrer l'information touristique de différentes pages web et présenter d'une manière significative. Le nombre de jeux de données touristiques est en forte développement, actuellement on peut répertorier 102 <http://datahub.io/dataset?q=tourism>, consulté le 02/09/2015 jeux de données [Cresci *et al.* 2014, Sabou *et al.* 2013] y sont dispo-

nibles. On va on explorer quelques un :

- El Viajero's tourism dataset<sup>1</sup> contient 9262184 triplets et intègre différents contenus issus des articles et des plates-formes numériques appartenant au Groupe Prisa numérique à partir des news et des blogs se portant sur le Voyage. Elle repose sur l'Open Provenance Modèle qui capture la provenance de chaque ressource, et puis rassemble les informations nécessaires à partir de 20.000 guides de voyage, photos, vidéos et messages appartenant à ces plates-formes. Ce jeu de données partage 3,093 intra-liens avec DBpedia et 55224 intra-liens avec geolinkeddata.
- Tourpedia [Cresci *et al.* 2014]<sup>2</sup> : TourPedia<sup>2</sup> a été développé au sein du projet Ope-ner ((Open Polarity Enhanced Name Entity Recognition)) qui est le Wikipedia du Tourisme et exposé comme un nud de données liée orienté goodRelations [Ashraf *et al.* 2011, Hepp 2008]<sup>3</sup>, qui fournit l'accès aux lieux. . Il contient des informations sur l'hébergement, les restaurants, les points d'intérêt et les attractions de différents endroits en Europe couvrant huit lieux : Amsterdam, Barcelone, Berlin, Dubaï, Londres, Paris, Rome et la Toscane. L'ensemble des données extraites proviennent de quatre médias sociaux : Facebook, Foursquare, Google Places et booking. TourPedia fournit deux ensembles de données principales : Lieux et Avis où chaque endroit contient des informations utiles telles que le nom, l'adresse et son URI à Facebook, Foursquare, GooglePlaces et booking afin de consulter les avis. TourPedia fournit deux méthodes pour accéder aux données : via une API Web et un moteur SPARQL. TourPedia donne un aperçu de la répartition des places dans les villes disponibles (Amsterdam, Barcelone, Berlin, Dubaï, Londres, Paris, Rome, Toscane) pour le chaque catégorie de lieu (hébergement, attraction, point d'intérêt et restaurant). Ce jeux de données partage 492,888 triplets (voir Figure 5.5)

Pour des raisons de lisibilité, nous adoptons dans la suite de ce travail le mécanisme de préfixe permettant d'alléger la notation de ces URI.  $T$  est l'ensemble des ontologies touristiques inclus dans le jeu de données touristique liées (tourpedia enrichi).  $T = \{O_1 \cup O_2 \cup \dots \cup O_m\}$ . où T correspond à l'union des ontologies prédéfinies dans notre jeu de données.  $T = \{w \cup gr \cup foaf \cup vcard \cup dc \cup dcterms \cup accs \cup tiodbpedia\}$

```
Prefix :          Base URI
w: <http://knoesis.wright.edu/ssw/ont/weather.owl#>
gr: <http://purl.org/goodrelations/v1#>
foaf: <http://xmlns.com/foaf/0.1/>
vcard: <http://www.w3.org/2006/vcard/ns#>
dc: <http://purl.org/dc/elements/1.1/>
dcterms: <http://purl.org/dc/terms/>
xsd: <http://www.w3.org/2001/XMLSchema#>
acc:< http://purl.org/acco/ns#>
```

---

1. <http://webenemasuno.linkeddata.es/>

2. <http://tour-pedia.org>

3. Ce vocabulaire intègre les marques RDF des informations concernant les produits proposés par la compagnie, par exemple le prix, la disponibilité et la description, la localisation, le nombre d'article disponibles, les offres quotidiennes, horaires d'ouverture, etc.. Ce dernier a une incidence sur le contenu affiché par le navigateur - par exemple Google.

## 5.4. COUCHE JEU DE DONNÉES TOURISTIQUES

TABLE 5.1 – les éléments conceptuels de l'ontologie GoodRelations

Feature	Description	Reference
Business Entity	present the company Web site (mailing address and contact details)	gr :BusinessEntity
Products and Offerings	describe the actual products or services that are being offered, and the offering, the actual business function (sell, repair, dispose, etc.) and other commercial properties.	gr :ProductOrServiceModel
Eligible Customers and Regions	The types of customers for which the given Offering is valid.	gr :eligibleCustomerTypes
Price Specifications	information on the prices fixed by vendors	gr :hasPriceSpecification
Delivery Options and Delivery Charge Specifications	shipment charge	gr :availableDeliveryMethods
Payment Options and Payment Charge Specifications	procedure for transferring the monetary amount for a purchase	gr :acceptedPaymentMethods
Warranty Promises	offer includes a bundle of services in case of defects or malfunction	gr :hasWarrantyPromise
Bundles	placeholder instance for unknown instances of a mass-produces commodity	gr :ProductOrServicesSomeInstancesPlaceholder
Services and Value Ranges	numerical interval that represents the range of a certain quantitative Product or Service Property in terms of the lower and upper bounds for one particular Product Or Service	gr :QuantitativeValueFloat gr :TypeAndQuantityNode
Shop Locations and Opening Hours	availability of a particular Offering at or from and defines the opening hours for a given DayOfWeek	gr :availableAtOrFrom gr :hasOpeningHoursSpecification

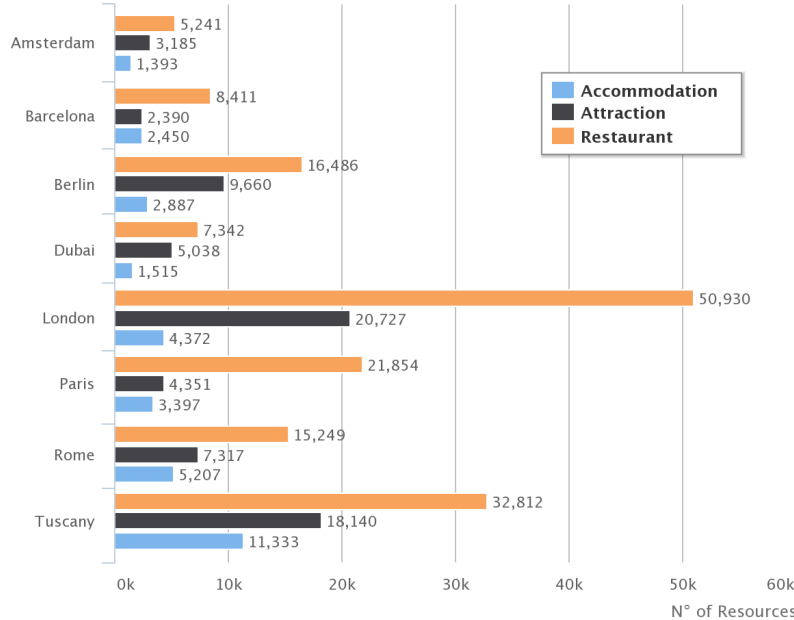


FIGURE 5.5 – Données de Tourpedia

```
s: <http://schema.org/#>
tio: < <http://purl.org/tio/ns#>
dbpedia: <http://dbpedia.org/>
owl: <http://www.w3.org/2002/07/owl#>
rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
proc: <http://www.daml.org/services/owl-s/1.1/Process.owl#>
```

Listing 5.4 – La liste des ontologies formant le jeu de Données liées touristique

L'alignement des instances est une tâche d'une importance capitale dans le web sémantique. Il vise à créer des liens exprimant une relation de similarité owl:sameAs entre deux instances. Notre objectif est d'utiliser un outil d'alignement des entités touristiques du jeu de données Tourpedia enrichie par d'autres vocabulaires variés et largement utilisés (Dbpedia, GeoNames, Schema.org, LinkedGeoData, Foursquare, etc.). Afin de pallier à cette diversité des modèles de représentations, il y a un besoin d'utilisation d'un outil d'alignement. Parmi les outils d'alignement, on trouve SILK [Jentzsch *et al.* 2010] qui se base sur un langage déclaratif de spécification des liens (Silk-LSL) et présente une méthode de liage efficace pour des jeux de données simples ainsi l'outil retournera la relation de similarité par une métrique de similarité sémantique. On peut citer quelques mappings concepts qui sont utiles pour le processus de découverte :

```
vcard:Person          owl:equivalentClass  foaf:Person;
dbpedia:event         owl:equivalentClass  tio:Event;
s:Hotel               owl:equivalentClass  Acc:Hotel;
s:Hotel               owl:equivalentClass  dbpedia:Hotel;
```



## 5.5. COUCHE SERVICES WEB SÉMANTIQUE INTER-RELIÉS

```
dbpedia:startDate    owl:equivalentClass    s:startDate;
dbpedia:endDate      owl:equivalentClass    s:endDate;
dbpedia:City         owl:equivalentClass    s:City;
dbpedia:Country      owl:equivalentClass    s:Country;
dbpedia:City         owl:equivalentClass    lgd:City;
dbpedia:Country      owl:equivalentClass    lgd:Country;
acc:Hotel            rdfs:subclassof        acc:accommodation;
.....
```

Listing 5.5 – Exemple d’alignement de schéma retourné par un outil de liage des Données Liées Ouvertes

### 5.5 Couche Services Web Sémantique Inter-reliés

Rappelons que l’ontologie OWL-S différencie trois types de processus dont les processus atomiques (atomic processes) qui n’ont pas de sous-processus. Un processus atomique est "une description d’un service qui attend un message en entrée et retourne un autre message". Il est décrit en termes d’entrées, sorties, pré-conditions et effets. Du point de vue de l’utilisateur, un processus atomique est une "boite noire" accomplissant une fonctionnalité. Le comportement interne d’un processus atomique n’est pas visible à l’utilisateur. Un processus atomique modélisant une fonctionnalité F doit fournir un grounding avec une description WSDL réalisant la fonctionnalité F. (ii) Les processus composites (composite processes) sont constitués d’un ensemble de processus atomiques et de contrôles de séquence. Un processus composite maintient un état. Chaque message reçu d’un client constitue une itération d’une phase dans ce processus. (iii) Les processus simples (simple processes) ne sont pas exécutables. La composition nécessite que l’objectif ait été anticipé qui peut être instanciée à travers les DONNÉES LIÉES [Heath & Bizer 2011, Berners-Lee 2009, Bizer *et al.* 2009, Berners-Lee 2006]. Notre vision, centrée utilisateur, impose une contribution théorique et pratique. Ainsi, alors qu’en planification le plan solution qui sera le plus apprécié sera celui contenant le moins de processus atomiques possibles, la composition devra être riche en structures pour être flexible, et laisser un maximum de degrés de liberté à l’utilisateur : les algorithmes de planification ne peuvent pas être les mêmes.

Le point d’entrée à notre canevas SQUIREL est une requête utilisateur  $R_Q = (I_Q, O_Q)$ . Une requête est essentiellement un ensemble de concepts d’entrée, qui représentent l’ensemble initial d’entrées disponibles  $I_Q$ , et un ensemble de concepts de sortie, qui sont les sorties que le service composite devrait retourner  $O_Q$ . L’ensemble des termes utilisées pour décrire les Process Models ou bien la requête sont exprimée en se référant aux concepts ontologiques issus de l’ontologie global  $T$ . Cette dernière étant l’union de l’ensemble des ontologies utilisée pour construire les jeux de données touristiques. Les Entrées et sorties de la requête sémantique sont utilisés pour découvrir les multiples Process Models concernés afin de composer leur fonctionnalité (sous-ensemble de processus atomiques) en faisant correspondre leurs entrées et sorties ensemble.

**Définition 5.5.1.** Soit  $PMs$  les Process Model transactionnelle (dont la catégorie de classification correspond à celle des services de voyage NAICS : 5615). Un  $PM$  est défini comme un

quatre-uplet  $(AP, E, I_S, O_S)$  où les processus atomiques ( $AP$ ) est l'ensemble des opérations indivisibles,  $I_{PM}$  est l'ensemble des paramètres d'entrées requis pour invoquer le  $PM$ ,  $O_{PM}$  est l'ensemble des paramètres sorties retournés par le  $PM$  après l'exécution du processus ( $I_S, O_S \subseteq O_S \subseteq T$ ) et  $E$  contient le flux de contrôle entre les processus qui est défini dans le processus composite.

Les Process Model doivent être sérialisées en turtle pour qu'il puisse être interrogeable. Etant donné l'entrée (respectivement la sortie) d'un Process Model  $PM$  qui correspond à un message reçu (livré), correspondant à la balise  $\langle \text{processus} : \text{hasInput} \rangle$  (respectivement  $\langle \text{processus} : \text{hasOutput} \rangle$ ) du code OWL-S. La formalisation des fonctionnalités du Process Model OWL-S dont les paramètres d'entrée  $I_i$  et sorties  $O_j$  en utilisant la syntaxe turtle est de la forme :  $\langle PM, pr : \text{hasInput}, I_i \rangle \cdots \langle PM, pr : \text{hasOutput}, O_j \rangle$ .

En se basant sur la définition formelle du problème de composition de service, nous proposons une approche basé sur les graphes [Diestel 1997] afin d'assurer une composition de processus composite automatique. La troisième couche montre l'aperçu de notre approche avec les différentes étapes.

### 5.5.1 Modélisation formelle de OWL-S

Nous passons à la modélisation interne des Process Models non moins ambitieux ni moins complexe. Le projet des Web services sémantiques est d'ajouter de la sémantique aux Web services, comme nous l'avons précédemment mentionné. Des Web services sémantiques seraient intuitivement des programmes dont les effets sur leur environnement seraient connus et dont les données manipulées possèderaient aussi une sémantique.

Afin de modéliser, la représentation interne du Process Model OWL-S nous permettant de réduire le temps d'exécution, nous faisons appel aux théories de graphes et matrices. Mais avant cela revenant sur ce qu'est OWL-S. L'OWL-S est un langage permettant de décrire les services Web de façon non ambiguë et interprétable par des programmes. Ce langage est basé sur le langage d'ontologie du Web (OWL). OWL-S est aussi une ontologie OWL particulière. OWL-S s'applique à accomplir la composition automatique des services Web. L'accomplissement d'une tâche complexe implique la sélection, la composition automatique des services Web. Par exemple, l'utilisateur peut vouloir faire tout le nécessaire pour son voyage à une conférence. Il veut acheter un billet mais aussi il veut réserver un hôtel proche du lieu de la conférence. Il peut également ajouter des contraintes de coût minimal. Afin qu'un agent puisse exécuter ce type de tâche, OWL-S fournit une spécification des pré-requis et des conséquences de l'exécution de chaque service individuel ainsi qu'un langage pour décrire les services composés et le flux de données. Pour atteindre ces objectifs, OWL-S définit une ontologie supérieure pour la description, l'invocation et la composition des services Web. Celle-ci est présentée dans ce qui suit L'ontologie supérieure d'OWL-S. La structuration de l'ontologie supérieure de OWL-S est motivée par la nécessité de fournir trois types d'information essentiels pour un service, à savoir :

- **Comment utilisons-nous le service :** Cette information est fournie dans le Service Model qui est utilisé pour, entre autres, composer les services. OWL-S modélise les services en tant que processus et celui-ci est défini par ses entrées/sorties. Trois types de

processus existent : les processus atomiques (AtomicProcess), simples ( SimpleProcess ) et composites (CompositeProcess). Un processus atomique représente le niveau le plus fin pour un processus et correspond à une action que le service peut effectuer en une seule interaction avec le client. Les processus composites sont décomposables en d'autres processus (composés ou non); leur décomposition peut être spécifiée en utilisant un ensemble de structures de contrôles tels que : Sequence, Split, If-Then-Else etc... Un processus atomique est utilisé pour fournir une vue d'un processus atomique ou une représentation simplifiée d'un processus composite.

Le Service Profile fournit des informations sur le domaine d'application du service Web. Chaque service Web est rangé dans une taxonomie facilitant la recherche du service Web pour les clients potentiels. Le Service Profile fournit également des éléments sur les conditions d'exécution du service telles que les pré-conditions et les effets du service Web au sens de OWL-S, ainsi que la description des entrées/sorties. L'ensemble de ces conditions d'exécution est proche de l'ensemble des pré-conditions et post-conditions de la planification. De plus, ces informations sont liées au Service Model. **Ce dernier décrit les fonctionnements des services Web en les décomposant en Processus atomique, Processus simple ou Processus composite.** Les Processus atomiques correspondent à des actions qu'un service Web peut effectuer. Les Processus atomiques ne comportent pas de sous-processus et n'exécutent qu'une seule étape consistant à interroger un service Web. Ils prennent un message d'entrée puis font quelque chose et renvoient un message. Pour chaque Processus atomique, le service Web correspondant doit être défini dans le ServiceGrounding. Les Processus simple ne peuvent pas être appelés directement comme les Processus atomique et ne sont pas associés à des services Web dans le ServiceGrounding. En revanche, ils sont vus comme des procédures ne comportant qu'une seule étape et pouvant être réalisées par un Processus atomique ou un Processus composite. Ils sont en fait une vue alternative des deux autres procédures. Les structures de contrôle proposées par OWL-S sont :

- Séquence : la séquence est une liste de processus qui doivent être exécutés dans l'ordre ;
- Split : appelle des sous-processus de façon concurrente ;
- Split+join : consiste à lancer différents processus de façon concurrente mais de les synchroniser en attendant que toutes les sous-procédure du split+join soient achevées pour les joindre ;
- Any-order : définit l'exécution d'un ensemble de tâches dans un ordre quelconque ;
- Choice : le choix consiste à choisir entre plusieurs sous-procédure équivalentes ;
- If-Then-Else : correspond au if-then-else habituel, c'est-à-dire si la condition du if est réalisée, alors la sous-procédure du then est exécutée, sinon la sous-procédure du else est exécutée ;
- Iterate : boucle continuellement avant d'être interrompu par whileCondition ou until-Condition ;
- Repeat-While et Repeat-Until : sont des boucles conditionnelles qui répètent une sous-procédure tant que (Repeat-While) la condition de la boucle n'est pas satisfaite ou jusqu'à ce que (Repeat-Until) la condition de la boucle soit réalisée. Enfin le ServiceGrounding définit les entrées et les sorties du service Web ainsi que les informations nécessaires pour appeler le service Web. Il permet de lier les AtomicProcess avec des

services Web existants.

Les composants principaux du modèle de processus (Process Model) sont l'ontologie du processus (Process Ontology) et l'ontologie de contrôle du processus (Process Control Ontology).

- **L'ontologie du processus** (Process Ontology). Cette ontologie décrit un service en termes de paramètres d'entrée et de sortie, de pré-conditions, des actions du service et dans le cas approprié, des sous-processus. Cette ontologie peut être utilisée afin de supporter l'invocation et la composition automatique de services Web.
- **L'ontologie de contrôle du processus** (Process Control Ontology). Cette ontologie de contrôle du processus décrit chaque processus comme étant un état, en prenant en compte son activation, son exécution et sa terminaison.

Le but est de stocker le flux de contrôle reliant les processus atomiques les uns aux autres. Pour cela, on va parser chaque Service Web touristique présents dans le registre UDDI afin d'extraire sa matrice d'adjacence en décomposant le Process Model. Ces matrices d'adjacence sont pré-calculés et stockés dans le même registre afin de pouvoir les invoquer à tout moment pur une éventuelle composition dynamique.

Sur cette base, la matrice d'adjacence  $APM$  est une matrice booléenne carré qui représente l'intra-dépendance des processus atomiques  $AP$  et ce en analysant le comportement complet du Process Model  $PM$ ,  $APM = [AP_{ij}]_{n \times n}$  où  $N$  désigne le nombre de processus atomiques explicités dans le process Model  $PM$  où ses éléments  $AP_{ij}$  ne peut être que "0" ou "1". Si deux sommets sont interconnectés par le constructeur de séquençement de  $AP_i$  à  $AP_j$  *i.e.*,  $seq(AP_i, AP_j)$  est définis dans  $E(PM)$ , alors  $APM[AP_i, AP_j] = 1$  sinon  $APM[AP_i, AP_j] = 0$ .

La matrice d'adjacence  $APM$  généré et stocker dans le registre UDDI est une matrice non diagonale dont tout les coefficients diagonaux sont nuls impliquant ainsi l'indépendance de chaque Processus atomique sur sur lui-même et formant une dépendance acyclique. **Enfin les Processus composites sont des procédures complexes regroupant plusieurs étapes pour être réalisées et définis l'intra-dépendances entre le processus atomiques  $AP$ . Elles sont décomposables en Processus atomique, Processus simple et Processus composites..** Cette décomposition peut être spécifiée à l'aide de structures de contrôle telle que if-then-else ou les séquence

On s'intéresse principalement à l'interconnexion définie entre l'ensemble des processus atomiques présents dans chaque processus. Pour cela nous avons élaboré quelques règles nous permettant d'extraire l'intra-dépendance des processus atomiques pour chaque Process model en termes de structure de contrôle de séquence plus simple, concrètement, dans ce cas, on parse le Process Model de la racine des processus composites, puis on procède récursivement jusqu'à atteindre les processus atomiques, le résultat est stocké dans une matrice carré.

- $Seq(S_1, S_2)$ , L'ordre d'exécution des deux processus atomiques est séquentiel *i.e.*,  $S_1$  doit s'exécuter suivi de  $S_2$  impliquant que  $S_2$  est le successeur de  $S_1 \Rightarrow APM[S_1, S_2] = 1$
- $Seq(S_1, \alpha(S_2, S_3)) = Seq(S_1, S_2) \wedge Seq(S_1, S_3)$  Un processus atomique  $S_1$  est suivi d'un processus composite qui définit deux processus atomiques conditionnels (If-then-Else) ou concurrentiels (Split), ce qui revient à avoir deux flux de séquence entre ( $S_1$  et  $S_2$ ) et/ou ( $S_1$  et  $S_3$ )  $\Rightarrow PM[S_1, S_2] = 1 \wedge APM[S_1, S_3] = 1$

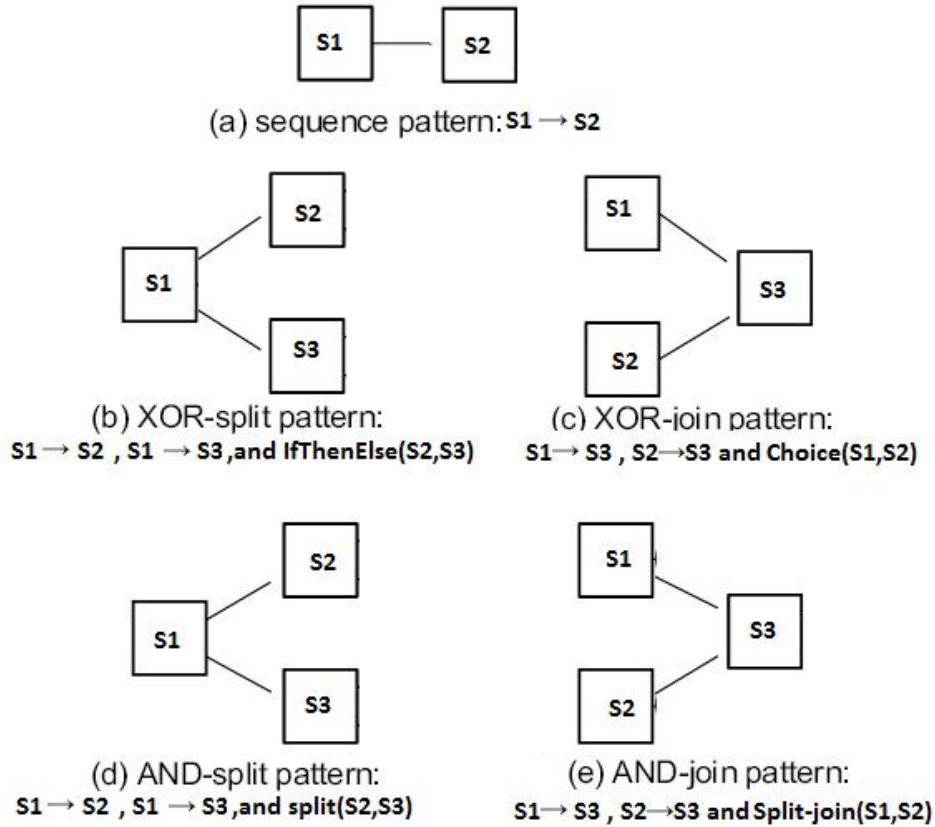


FIGURE 5.6 – Présentation des constructeurs du process model en constructeur de séquence-ment

où  $\alpha = \text{IfthenElse} \parallel \text{Split and } \text{IfThenElse}(S_2, S_3)$

- $\text{Seq}(\alpha(S_1, S_2), S_3) = \text{Seq}(S_1, S_2) \wedge \text{Seq}(S_1, S_3)$  Un processus composite qui définit deux processus atomiques aux choix (Choice) ou concurrentiels (Split-join), est suivi d'un processus atomique ce qui revient à avoir deux flux de séquence entre ( $S_1$  et  $S_3$ ) et/ou ( $S_2$  et  $S_3$ )  $\Rightarrow \text{APM}[S_1, S_3] = 1 \wedge \text{APM}[S_2, S_3] = 1$  où  $\alpha = \text{Choice} \parallel \text{Split-join}$ .

Considérons le service `EventReservationService` présenté en haut, il définit quatre processus atomiques inter-reliés entre eux par un séquençement de trois processus `SelectEvent` (E1), `SignInEvent` (E2) et le processus composite de choix entre `BankTransferPayement` (E3) and `creditcardpayement` (E4). Le service ainsi illustre une séquence de nœuds partant de E1 vers E2 à partir duquel on peut basculer soit vers E3 ou E4. En conclusion, la matrice d'adjacence inclus  $E(\text{PM}) = \{\text{Seq}(E1, E2), \text{Seq}(E2, E3), \text{Seq}(E2, E4)\}$ . Basé sur la description énoncée ci-dessus, nous présentons la matrice d'adjacence du service événement dans la figure 5.5.1 :

### 5.5.2 Positionnement de notre approche

La composition s'analyse à partir des critères suivantes :

	$E_1$	$E_2$	$E_3$	$E_4$
$E_1$	0	1	0	0
$E_2$	0	0	1	1
$E_3$	0	0	0	0
$E_4$	0	0	0	0

FIGURE 5.7 – Exemple de matrice d’adjacence du service événement

- Le plan final est fini : un nombre fini de processus atomiques satisfait la requête et sa dynamique ;
- Le plan final est complètement observable : chaque processus atomique est entièrement décrit et connu ;
- La plan final est déterministe : pour chaque requête  $e$  et pour chaque plan  $a$ , l’application de  $e$  sur  $a$  mène vers un unique résultat ; le plan est séquentiel et concurrentiel : un plan solution en tant que composition de processus atomiques est un ensemble totalement ordonné de ces processus atomiques ;
- l’objectif est déclaratif : un ensemble de propriétés de la composition souhaité décrit l’objectif de planification lorsque des propriétés sur la composition du plan pourraient être exprimées ;

Les descriptions sémantiques actuellement disponibles, en termes de SERVICES WEB SÉMANTIQUES [Jorge 2006, McIlraith *et al.* 2001] tels que OWL-S ou WSMO, présentent une grande complexité de leur description sémantique ainsi que de leur traitement engendrant des problèmes d’évolutivité lors de la découverte sémantique. Etant basées sur différents formalismes logiques, plusieurs travaux de recherches se sont concentrés sur l’amélioration et l’optimisation du mécanisme d’appariement en indexant les descriptions sémantiques ou en les mettant en cache. Dans cette partie, nous abordons une nouvelle approche de découverte automatique de Web Services Sémantiques exprimé en OWL-S améliorant considérablement l’espace de recherche en se basant sur une approche existante EMMA (Enhanced Match-Making Enhanced Addon). Cette dernière optimise la phase de découverte en mettant en œuvre une phase de filtrage qui permet de réduire le nombre de SERVICES WEB SÉMANTIQUES [Jorge 2006, McIlraith *et al.* 2001] candidats lors de la phase de découverte à ceux qui sont pertinents par rapport à la requête de l’utilisateur. Ainsi, les services non pertinents sont a priori rejetés, parce qu’ils ne satisfont pas l’ensemble des exigences et des préférences formulés par l’utilisateur. Par conséquent, prétraiter le répertoire des SWS, permet ainsi de réduire et d’économiser les ressources calcul et améliorer les performances de la découverte sémantique. Pour cela, on considère uniquement le sous-ensemble de services qui se réfèrent spécifiquement aux concepts cités dans la requête et non l’ensemble des services candidats contenus dans le répertoire.

Initialement, pour le pré-traitement proposé, la requête de l’utilisateur est formulée en SPARQL qui est ensuite analysée afin d’en extraire les descriptions sémantiques (la signature) désirés. Le répertoire est filtré afin que seuls les services qui utilisent ces concepts ou ceux

liés à ces concepts sont sélectionnés pour devenir pré-candidat au processus de découverte ultérieure.

La phase de filtrage proposé repose sur la définition de deux filtres totalement différents et exprimés en requêtes SPARQL, à savoir le  $Q_{ALL}$  et le  $Q_{SOME}$ . Le premier filtre renvoie uniquement les services dont les descriptions sémantiques englobent l'ensemble des concepts sémantiques mentionnés par la requête utilisateur, c.à.d. que les services doivent satisfaire totalement chaque terme (concept) définit dans la requête par l'utilisateur. La deuxième filtre sélectionne les services dont la description sémantique qui se réfère à certains ou quelques (au moins un) concepts sémantiques définit dans la requête de l'utilisateur, c.à.d. que ces services peuvent répondre ou satisfaire partiellement les besoins et / ou préférences de l'utilisateur dans une certaine mesure et ceux malgré les informations manquantes. Cependant, cette approche n'a pas la prétention de fournir un nouveau mécanisme de découverte sémantique mais de l'améliorer en introduisant une étape de prétraitement, le filtrage est basé sur le langage d'interrogation sémantique exprimé en SPARQL. Ce dernier raffine la phase de découverte des Processus composite (process Model OWL-S) en un sous-ensemble de services pertinents améliorant considérablement le matchmaking de services. C'est une approche qui utilise les requêtes sémantiques et présente une complexité linéaire sur la taille de l'ensemble du patron de données, de ce fait elle n'augmente pas le temps d'exécution à l'égard du processus de matchmaking. Dans la littérature, c'est la première approche à aborder le filtrage du répertoire de service sémantique puisqu'aucune des approches cités dans le chapitre 3 n'aborde cette hypothèse. Cependant cette approche est applicable en amont sur tout scénario de découverte de SWS existante cités dans le chapitre 3 .

### 5.5.2.1 Amélioration de la découverte automatique des Services Models

#### Motivation

Les motivations pour développer ou tendre vers ces SERVICES WEB SÉMANTIQUES [Jorge 2006, McIlraith *et al.* 2001] sont évidemment de faciliter les phases automatiques de découverte, sélection et composition de Web services. Cependant, malgré cette large adoption des Services Web Sémantique, de nombreux obstacles empêchent leur réconciliation sémantique lors de la composition du fait que l'interprétation consistante des données échangées entre services Web composés est gênée par des différences d'interprétation sémantiques. En effet, si leur sémantique est connue, alors chercher et composer des Web services pourra être fait automatiquement en donnant la sémantique cible. De plus, pour les mêmes raisons que pour le Web sémantique, il faut organiser les Web services entre eux. Une première solution pour les trier est que des machines puissent connaître leur sémantique. En effet, si la sémantique était connue, il suffirait d'indiquer sous forme par exemple de requête les objectifs que devrait remplir le Web service afin de pouvoir, par exemple, le sélectionner. L'utilité et la pertinence des données se trouvent dans l'information qui peut en être extraite.

Le Process Model décrit les fonctionnalités sémantiquement qui sont incluse dans le Service Model. C'est pour cela, que notre thèse s'articule autour de la découverte des Service Model dû au fait de leur description explicite des fonctionnalités nécessaire à la sélection. Dans cette partie, nous nous intéressons aux hétérogénéités sémantiques des données échangées entre les Process Models engagés dans une composition. Pour cela, nous proposons d'améliorer

rer une approche originelle de découverte des Services Models. Notre principale contribution évolue autour de la notion de l'implication de la nouvelle ère les Données touristiques Liées pour enrichir l'échange des données entre les services Web. Nous proposons un modèle orienté Données liées ouvertes pour résoudre les hétérogénéités sémantiques entre les Services Web.

Afin de générer les différentes possibilités de composition, la capacité d'analyser la compatibilité entre différentes fonctionnalités sémantiques est indispensable. Le processus de match-making sémantique, est chargé d'évaluer ce niveau de compatibilité sémantique entre les concepts issus d'un ensemble d'ontologies et définis dans les jeux de données sémantiques. Ces derniers sont en mesure délivrer l'alignement de schémas qui détermine les relations sémantiques entre les concepts de type exacte et plugin nécessaire au moteur de composition. Les descriptions sémantiques actuellement disponibles, en termes de SWS tels que OWL-S ou WSMO, présentent une grande complexité de leur description sémantique ainsi que de leur traitement engendrant des problèmes d'évolutivité lors de la découverte sémantique.

**Proposition** Notre approche de composition ne réutilise pas un workflow prédéfini, mais génère dynamiquement un plan de composition reflétant la requête utilisateur et formulé en temps réel fournissant ainsi un accès rapide aux Process Model augmentant ainsi la probabilité d'identifier ceux qui sont potentiels en utilisant un langage de requête sémantique SPARQL.

Nous proposons une approche de découverte plus facile en adaptation et en réutilisation, avec d'avantage d'expressivité et d'explicitation sémantique fonctionnelle et technique provenant du domaine des services Web. La majorité des approches de découverte s'articule autour du service profil, cependant le service Model fournit une description sémantique assez explicite de sa fonctionnalité permettant ainsi une découverte efficace des services les plus pertinents pour une meilleure composition dynamique. Dans la littérature, il existe des approches permettant d'interroger les Services Web Sémantiques en utilisant le standard des langages d'interrogation sémantiques SPARQL. Partant du point où l'ensemble des descriptions des Services Models sont définies en termes de concepts ontologiques référencés dans le jeu de données touristique. Ce jeu de données repose sur l'utilisation de plusieurs vocabulaires/ontologies.

Cette problématique est notamment étudiée dans plusieurs approches pratiques visant à utiliser des technologies du Web sémantique pour contribuer à la coordination des SERVICES WEB SÉMANTIQUES [Jorge 2006,McIlraith *et al.* 2001] . Une spécificité de ce contexte, assez peu abordé dans la littérature existante, est sans doute le besoin d'inciter les découvertes des SERVICES WEB SÉMANTIQUES [Jorge 2006,McIlraith *et al.* 2001] inattendues et fortuites à travers la richesse du DONNÉES LIÉES [Heath & Bizer 2011,Berners-Lee 2009,Bizer *et al.* 2009, Berners-Lee 2006] , une richesse qui repose notamment sur la diversité de types de liens et de ressources. C'est sur cette richesse que nous avons fondé le développement d'une approche de découverte très souple. Le premier axe de notre travail concerne la découverte de SERVICES WEB SÉMANTIQUES [Jorge 2006,McIlraith *et al.* 2001] pertinents et inattendus, notamment utiles pour optimiser le plan final en élargissant le champ de recherche par lesquelles une requête sémantique est initialement décrite, en y incluant des concepts similaires permettant ainsi de solliciter les compétences de d'autres services. L'ensemble des Services Models sont transcrit en Turtle-RDF afin qu'il soit interrogeable lors de la phase de découverte. Ce dernier est enrichi par les interrelations de données découvertes lors du processus de liages des jeux



de données.

**Définition 5.5.2.** *Le processus de découverte des Process Models DPS est un triplet nommé  $DPS = (T, R_Q, PMs)$  où  $T$  est l'ensemble des ontologies touristiques appartenant aux jeux de données touristiques,  $T = \{T_1 \cup T_2 \cup \dots \cup T_m\}$ , une requête utilisateur exprimé en CPSPARQL  $R_Q = (I_Q, O_Q)$  qui spécifient un ensemble de concepts ontologiques décrivant les entrées fournies  $I_Q$  et les sorties souhaitées  $O_Q$  ( $I_Q, O_Q \subseteq T_Q \subseteq T$ ) L'ensemble de ces Process Models décrivent uniquement les services transactionnel dont la catégorisation NAICS est relatif au code 5615 (Travel services category).*

Supposant qu'on a deux Process Models  $PM_1$  et  $PM_2$ . Le processus de découverte repose sur la qualité du matching entre les Process Models, appelée communément l'interdépendance des Services. C'est un mécanisme qui exploite le flux de donnée en termes de sorties d'un service ( $PM_1$ ) en rapport à l'entrée d'un autre service ( $PM_2$ ). Ce matching repose sur différents degrés de correspondance. Les plus généralement envisagés dans la littérature [Paolucci et al. 2002] sont :

exacte ( $\equiv$ ) : Une sortie sémantique  $O_1 \in O_{PM_1}$  d'un service  $PM_1$  correspond exactement à une entrée sémantique  $I_2 \in I_{PM_2}$  d'un service  $PM_2$  Ssi les deux concepts avec un degré de sont équivalent.

Plugin ( $\sqsubseteq$ ) : Une sortie sémantique  $O_1 \in O_{PM_1}$  d'un service  $PM_1$  correspond à un degré Plugin à une entrée sémantique  $I_2 \in I_{PM_2}$  d'un service  $PM_2$  Ssi  $O_1$  est un sous-concept de  $I_2$ .

subsumes ( $\sqsupseteq$ ) :  $I_2 \in I_{PM_2}$  d'un service  $PM_2$  correspond à un degré subsumes à une entrée sémantique  $I_2 \in I_{PM_2}$  d'un service  $PM_2$  Ssi  $O_1$  est un super-concept de  $iO_1$

Fail ( $\perp$ ) : Lorsque aucun des matchs précédents sont trouvé, les deux concepts sont incompatibles et le match a un degré de Fail ( $O_1 \perp I_2$ ).

L'objectif est de générer les compositions sémantiquement complets. Afin d'assurer une meilleure performance et efficacité de la découverte générant un flux de données pour la composition de Service Model compatibles à, on se concentre essentiellement sur les deux degrés de correspondance **Exacte (owl :equivalentclass)** et **Plugin (rdfs :subclassof)** car la troisième métrique donne lieu à une composition sémantique plus faible est potentiellement moins adaptée.

**Définition 5.5.3** (Relation d'alignement de concept généralisé). *Etant donné deux concepts  $A$  et  $B$ , défini dans deux ontologies séparées  $O_1$  et  $O_2$  (préfixés respectivement  $O1, O2$ ) ( $A \in O_1$  et  $B \in O_2$ ), la Relation d'alignement de entre deux concepts correspond à la : (1) Correspondance exacte ( $\equiv$ ) ou (2) Correspondance Plugin ( $\sqsubseteq$ ) défini comme suit :*

$$(A \preceq B) \iff (A \equiv B \vee A < B)$$

*et est formalisée dans PPARQL par l'expression régulière qui recherche un chemin de concepts entre  $A$  et  $B$  qui sont reliés par une séquence de concepts équivalents ou de sous-concept comme suit :*

`O1:A + (owl: equivalentClass | rdfs: subclassOf) O2:B;`

où

1.  $A \equiv B$ ,  $A$  est sémantiquement équivalent à  $B$ , et

2.  $A \prec B$ ,  $A$  est un sous-Concept de  $B$  Ssi :

- $(A \equiv C)$  et  $(C \sqsubseteq B)$  où  $(C \in O_2)$
- $(A \sqsubseteq D)$  et  $(D \equiv B)$  où  $(D \in O_1)$

Nous définissons deux types d'appariement de Process Model (partielle et totale) qui reposent sur la définition de la correspondance compatible qui sera utilisé pour découvrir les Process models ( $PM$ ) candidats nécessaire à la phase de composition.

**Définition 5.5.4.** [Correspondance compatible] Soit deux concepts  $A, B \in T$ , la correspondance compatible noté  $A \preceq B$  est défini par la Correspondance exacte  $(A \equiv B) \vee$  la correspondance de plug-in  $(A \prec B)$ .

Notons que la correspondance exacte est commutative par contre la correspondance de plug-in ne l'est pas. A partir de la, correspondance compatible, nous définissons une fonction **match** qui couvre un appariement sur un ensemble de concepts et est définie comme suit :

**Définition 5.5.5.** [La fonction match]

Étant donné deux ensembles de concepts  $A, B \subseteq T$ , la fonction match renvoie les concepts de  $A_1$  appariés à  $B_2$  tel que  $A_1, B_2 \in T$ , nous définissons la fonction  $match : T \times T \rightarrow T$  tels que  $match(A, B) = \{B_1 \in B \mid (A_1 \preceq B_1), A_1 \in A\}$ .

Afin de générer les solutions de composition, la capacité d'analyser la compatibilité entre différents fonctionnalités sémantiques est indispensable. Ce processus de matchmaking sémantique, est chargé d'évaluer le niveau de compatibilité sémantique entre les concepts issus d'un ensemble d'ontologies. Nous définissons les concepts de correspondance totale et partielle à travers la fonction précédente, comme suit :

**Définition 5.5.6.** [correspondance totale et partielle]

Etant donné deux ensembles de concepts sémantiques touristiques  $A, B \subseteq T$ , une correspondance totale entre  $A$  et  $B$  est défini par  $match(A, B) = B$  ( $\forall B_1 \in B \mid A_1 \preceq B_1, \forall A_1 \in A$ ).

Typiquement, un  $PM$  est considéré pertinent à la requête  $Q_R$ , Ssi  $(match(I_R, I_{PM}) = I_{PM} \wedge match(O_{PM}, O_R) = O_{PM})$ , i.e., il existe une correspondance totale entre les entrées fournies  $I_R$  et ceux du service  $I_{PM}$ , et entre les sorties de service  $O_{PM}$  et les résultats attendus  $O_R$ .

Alors qu'une correspondance partielle existe Ssi  $match(A, B) \subset B$  ( $\exists B_1 \in B \mid left A_1 \preceq B_1, \exists A_1 \in A$ ) Ssi  $match(I_R, I_{PM}) \subset I_{PM} \wedge match(O_{PM}, O_R) \subset O_{PM}$

Initialement, pour le pré-traitement proposé, la requête de l'utilisateur est formulée en SPARQL qui est ensuite analysée afin d'en extraire les descriptions sémantiques (la signature) désirés.

**Définition 5.5.7** (Service web Composite). Soit  $R_Q$  la requête utilisateur, le service web composite  $SWC = \{I_{SWC}, O_{SWC}, \{P = L_{PM}, \leq\}\}$  est un service composite pour une requête  $Q_R$ , où  $P$  est un ordre partiel sur l'ensemble des Process Model  $PM_i \in L_{PM}$  du  $SWC$ .

Une composition est valable pour toute requête  $R_Q$  Ssi, pour tout tri topologique  $L_{PM} = PM_1, PM_2, \dots, PM_N$  de  $P$ , où  $PM_j = \{I_{PM_j}, O_{PM_j}\}$ ,  $\forall j \in [1, N]$ , l'expression suivante est confirmé :  $(\text{match}(I_Q, I_{PM_1}) \subseteq I_{PM_1}) \wedge (\text{match}((I_R \cup O_{PM_1}), I_{PM_2}) \subseteq I_{PM_2}) \wedge \dots \wedge (\text{match}((I_R \cup \dots \cup O_{PM_1}), O_R) = O_R)$ .

Par contre cette approche n'est pas raisonnable, la composition nécessite le besoin de localiser les services pertinents en se basant sur la correspondance partielle des entrées / sorties. Plutôt que d'aborder le problème de la découverte sur la description complète des entrée / sortie, nous avons divisé ce problème en deux problèmes de découverte plus fines qui sont plus pertinents pour la composition de service : découverte d'entrées et de découverte de sorties.

**Définition 5.5.8** (découverte d'entrées). *Étant donné un ensemble de concepts  $C \subseteq T$ , le problème de découverte d'entrée sémantique peut être définie comme la recherche d'un ensemble de services pertinents  $PM = \{PM_1, \dots, PM_n\}$  où  $I_{PM_i}, O_{PM_i} \in PM_i$  tel que  $PM_i \in PM$ ,  $\text{match}(C, I_{PM_i}) \subseteq I_{PM_i}$  définissant les services consommant certaines ou l'ensemble des entrées  $C$  (correspondance partielle ou totale).*

**Définition 5.5.9** (découverte de sorties). *Étant donné un ensemble de concepts  $C \subseteq T$ , le problème de découverte des sorties sémantique peut être définie comme la recherche d'un ensemble de services pertinents  $PM = \{PM_1, \dots, PM_n\}$  où  $\{I_{PM_i}, O_{PM_i}\} \in PM_i$  tel que  $PM_i \in PM$ ,  $\text{match}(C, O_{PM_i}) \subseteq C$  définissant les services qui produisent certaines ou l'ensemble des sorties  $C$  (correspondance partielle ou totale).*

Basé sur ces définitions, nous introduisons la notion de pertinence d'entrée et de sortie :

**Définition 5.5.10.** *Un Process Model  $PM = \{I_{PM}, O_{PM}\}$ , où  $I_{PM}, O_{PM}$ ,  $I_{PM}$  est l'entrée pertinente pour un ensemble de concepts  $C \subseteq T$  Ssi  $\text{match}(C, I_{PM}) \neq \emptyset$ , alors que le Process Model  $PM$  à une sortie pertinente pour output pour un ensemble de concepts  $C \subseteq T$  Ssi  $\text{match}(O_{PM}, C) \neq \emptyset$ .*

**Définition 5.5.11.** *Si  $C \subseteq T$  est l'ensemble des concepts d'entrées disponibles, et un Process Model  $PM = \{I_{PM}, O_{PM}\}$  est invocable par  $C$  Ssi  $\exists \text{match}(C, I_{PM}) = I_{PM}$  une correspondance totale entre les entrées disponibles et les entrées de service*

Afin de surmonter la limitation de l'approche [García et al. 2012], nous proposons la réécriture de la requête initiale  $R_Q$  en une nouvelle requête  $R'_Q$  en préservant la sémantique de cette dernière, une approche qui combine deux points : l'intra-dépendance des services et liens sémantiques pré-calculés servant de mise en correspondance dans la phase de découverte. Les relations entre les entrées et les sorties des services sont pré-calculés et fournit par les données touristique liées par la deuxième couche de notre canevas. Pour cela on exploite un outil automatique de génération de liens à partir de deux sources de données où le degré de correspondance sémantique entre entités (qui peuvent représentés les entrées et sorties entre services) est calculé en utilisant différentes métriques (logiques, non logique).

**Définition 5.5.12.** *[L'intra-dépendance de PM] Soit  $PM_1, PM_2$  deux Process Model, et  $O_{PM_1}, I_{PM_2}$  l'ensemble de sorties sémantiques correspondant à  $PM_1$  et l'ensemble d'entrées correspondant à  $PM_2$ . Il ya une intra-dépendance fonctionnelle entre  $PM_1$  et  $PM_2$ , Ssi une sortie  $O_i \in O_{PM_1}$  qui correspond à une entrée  $I_j \in I_{PM_2}$  tel que :  $O_i \preceq I_j$*

Les outils automatiques de liages de données délivrent les interrelations des données au format RDF qu'on insérera automatiquement dans le jeu de données des Services Models. Cette approche combinée, dans laquelle les arcs du graphe peuvent être étiquetés par d'expressions régulières supportant les chemins représentant l'intra-dépendance d'une part et le matching exacte et plugin d'autre part combinant ainsi de multiples cas, représenté dans la figure 5.8 :

- La requête suivante exprime l'intra-dépendance directe :

```
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns\#>
prefix proc: <http://www.daml.org/services/owl-s/1.1/Process.owl\#>
select *
where {
    ?sj ( proc:hasOutput.proc:parameterType .
        -(process:hasInput.proc:parameterType )) ?si .
}
```

Listing 5.6 – Exemple de requête Q exprimant l'intra-dépendance directe

- La requête suivante exprime l'intra-dépendance indirecte par l'utilisation des expressions régulières qui recherchent tout les paires de services qui sont reliés par un chemin exprimant une séquence de nœuds partant de la sortie du service  $S_j$  parvenant jusqu'à l'entrée d'un autre service  $S_i$  indépendamment de la longueur du chemin<sup>4</sup> (au moins 1) en utilisant l'opérateur de répétition ainsi, représenté dans la figure 5.8 :

```
select *
where {
    ?sj + ( proc:hasOutput.proc:parameterType .
        -(process:hasInput .proc:parameterType )) ?si .
}
```

Listing 5.7 – Exemple de requête Q exprimant l'intra-dépendance indirecte

- Les outils automatiques de liages de données délivrent les interrelations des données au format RDF qu'on insérera automatiquement dans le jeu de données des Process Models. La requête suivante est une extension de la précédente seulement elle prend en compte les relations d'alignement sémantique généralisée où la sortie d'un service peut être équivalente ou un sous-concept à l'entrée de l'autre service en utilisant l'opérateur de répétition d'astérix et qui est représentés dans la Figure 5.8 :

```
select *
where {
    ? sj +(pr:hasOutput.*(owl:equivalentClass
        | rdfs:subClassOf).-pr:hasInput) ?si;
}
```

Listing 5.8 – Exemple de requête Q exprimant l'intra-dépendance directe étendue

---

4. Où "-" est l'opérateur inverse. Par exemple, étant donné le triplet RDF (s, p, o), nous pouvons déduire (o, p, s)

## 5.5. COUCHE SERVICES WEB SÉMANTIQUE INTER-RELIÉS

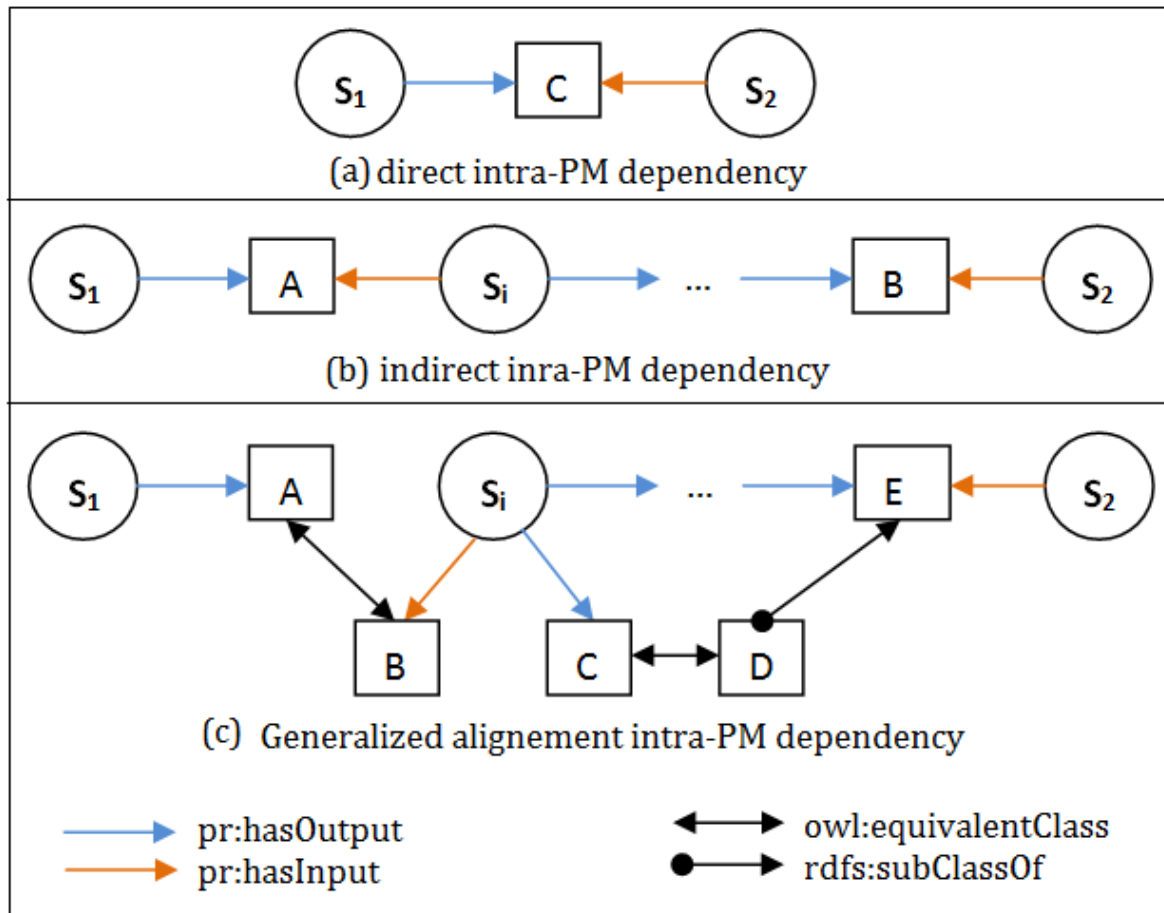


FIGURE 5.8 – Différentes intra-dépendances entre PM

Enfin réécrivons, la requête  $R_Q$  en  $R'_Q$  :

```

SELECT ?a
WHERE {
  {?a proc:hasInput . proc:parameterType foaf:Person.}
  UNION
  {?b proc:hasInput. proc:parameterType cc:CreditCard.}
  UNION
  {?c proc:hasInput . proc:parameterType acc:Hotel.}
  UNION
  {?d proc:hasInput. proc:parameterType tio:Event.}
  UNION
  {?e proc:hasOutput. proc:parameterType s:eventReservation.}
  UNION
  {?f proc:hasOutput. proc:parameterType s:LodgingReservation.}
  UNION

```

## CHAPITRE 5. CONTRIBUTIONS SCIENTIFIQUES

```
{?g proc:hasOutput. proc:parameterType w:WeatherCondition.}
}
```

Listing 5.9 – Exemple de requête  $R_Q$

Cette étape nous fournit la liste des Process Model pertinents à la composition et élimine ceux inutiles en inhibant la répétition afin d'éviter les cycles infinis. Sur cette base, la relaxation sémantique de la requête principale  $R_Q$  inclut l'alignement de schéma sur la base d'entrée et sorties sémantiques comme une disjonction des relations sémantiques d'équivalence / subsomption (`owl:equivalentClass` | `rdfs:subClassOf`) indépendamment du vocabulaire de l'ontologie utilisée. La nouvelle requête  $R_{Q'}$  est exprimée comme suit :

```
select distinct Sj? ?SI
Where {
  ?si pr:hasOutput ?O;

  ?si +(pr: hasOutput.* (owl: equivalentClass
                        | rdfs: subClassOf).-pr:? hasInput) ?sj;

  ?sj (pr: hasInput ) ?I;

  filter (((? O = O1) || (? O = O2) ...)
          && ((? I = I1) || (? I = I2) ...))
          && (?si!=?sj))
}
```

Listing 5.10 – La réécriture de requête  $R_{Q'}$

Poursuivons l'exemple précédent, ainsi  $R_{Q'}$  :

```
SELECT *
WHERE {

?si proc:hasInput. proc:parameterType ?In;
?si +(pr: hasOutput.* (owl: equivalentClass
                      | rdfs: subClassOf).-pr:? hasInput) ?sj;

?sj proc:hasOutput. proc:parameterType ?out;

Filter ( ((?In=acc:Hotel) || (?In=tio:Event) ||....)
        && ((?out=s:eventReservation.) ||
           (?out=s:LodgingReservation ) ....))}
```

Listing 5.11 – Exemple de requête Q de filtrage textuel SPARQL

Dans cette méthode le résultat de la composition retourné inclut l'ensemble des PM ( $L_{PM}$ ) considéré comme satisfaisant qui comportent tous les éléments souhaités par l'utilisateur.

Déterminer toutes les solutions possibles de la composition en se référant à l'utilisation d'expressions de chemins sur le flux de données sémantiques appartient à l'espace NP en

termes de complexité. En conséquence, un ensemble ordonné  $L_{PM}$  de pairs  $PM$  est retourné désignant respectivement les services initiaux, finaux et intermédiaires en inhibant la répétition par l'état ( $?Si != ?Sj$ ). La prochaine étape est de générer un multi-graphe sur la base de cette pré-sélection. Le graphe de la composition finalement généré sur la base des services pertinents et la requête enrichie par les alignements entre les I/O de l'information correspondant.

### 5.5.2.2 Sélection Automatique des Processus Atomiques pertinents à la composition

**Motivation** Le Web est un graphe dont les sommets sont les pages Web et les arcs sont les hyperliens entre les pages. De plus, une bonne partie des objets manipulés dans le Web sont représentés par des graphes : les documents XML, les services Web, les documents RDF, etc ; L'action Graphes et Web s'intéressent à l'exploitation et au développement de techniques de graphes pour la modélisation, la fouille et le développement d'applications Web. Le but de notre approche est de combiner les théories de graphes aux technologies du (Services) Web sémantiques. Dans ce contexte, sélectionner signifie très souvent composition optimal. La nécessité de classer les Process Model est primordial, l'algorithme suivant décrit une méthode incrémental qui trie ces Process Model équivalents ou subsumant pour en extraire uniquement les processus atomiques pertinents à la composition suivant un critère de similarité défini qui permet d'optimiser la taille du graphe initial Pour cela, nous avons abordé les théories de graphes afin de regrouper la liste des SERVICES WEB SÉMANTIQUES [Jorge 2006, McIlraith et al. 2001] pertinents en cluster de services et de raffiner ce dernier à des processus plus élémentaires, puis déduit de ce graphe, la composition optimal qu'il peut ensuite exécuter. Dans cette méthode le résultat de la composition sera considéré comme satisfaisant puisqu'il comportera tous les éléments souhaités par l'utilisateur lors de la phase de découverte. Finalement, le plan généré est retourné à l'utilisateur constituant ainsi son package dynamique.

**Proposition** Le processus de composition est déclenché à travers la requête utilisateur  $R_Q$  qui spécifie ces exigences en termes des concepts sémantiques attendus. L'objectif est de construire un graphe de l'ensemble des services concernés ainsi que des relations sémantiques entre eux en termes d'entrées et de sorties. Dans notre cas, la phase de découverte est responsable de récupérer les processus model pertinents  $L_{PM}$ .

Le principe est de construire un graphe avec un groupe de nœuds ayant certaines caractéristiques topologiques communes afin d'y établir des techniques d'optimisation. Apparemment, le problème est qu'il est difficile de repérer les "bons" Process Model quand il s'agit de composer automatiquement les Web services en vue d'un but à atteindre. Il faut dégager des critères opérationnels d'optimisation pour aider à la composition afin de diminuer l'espace résultat (le graphe de cluster de service). Ce graphe contient tous les Processus Models préalables à la composition qui pourraient directement ou indirectement être invoquée étant donné les entrées fournis. Le graphe est divisé en  $N$  couches, de sorte que chaque couche  $i$  a tous les services appariés par les sorties produites des couches précédentes où chaque couche  $L_i$  est définis comme suit :

$$L_i = \left\{ PM_i : PM_i \notin L_j (j < i) \wedge O_{PM_j} \subseteq I_{PM_i} \right\}$$

**Génération du graphe de cluster service (optimisation globale)** Cette ensemble ordonné partielle de services est essentiellement un graphe acyclique orienté Graphique (DAG), qui modélise l'ordre d'exécution implicite des Process Model entraînés par les correspondance sémantiques d'entrée / sortie, où les nœuds de la DAG sont les Process Model et les arcs sont les relations sémantiques valides conduisant à des compositions sémantiquement possibles. D'autre part, la représentation DAG saisit formellement la nature d'une composition où les Process Model peuvent être exécutées dans des ordres différents, à savoir, les ordres totaux (séquentiels) d'une composition qui conduisent au même résultat. En outre, puisque notre approche est centrée sur les flux de données, la représentation du DAG est un graphe simple (acyclique du fait qu'on a imposé une contrainte lors de la réécriture de la requête sémantique) qui capture le maximum de cas possibles plutôt que d'utiliser les cycles qui ne produisent pas de nouveaux types de données dans la composition.

Lorsque le système reçoit la liste des Processus Models pertinents, la phase actuelle consiste à générer un graphe dont les nœuds c'est des clusters de Process Models regroupant ceux qui sont équivalent et subsumant auxquels se rajoutent les relations sémantiques entre ces services compétents. La découpe en communautés (graph clustering) joue un rôle clef, non seulement en informatique, mais aussi par ses applications dans de nombreux domaines scientifiques, cette découpe peut être faite en rassemblant les nœuds similaires ayant un lien avec au moins la moitié des autres dans une même communauté ou cluster.

Un cluster de service est défini comme un ensemble de Services Models iso-fonctionnels mais avec des critères différentes (les processus qui y sont contenus, le nombre de processus atomiques contenu dans le Service Model, le nombre d'entrée / sortie). Le comportement de chaque service varie par rapport à un autre en fonction de sa modélisation interne, la qualité de service etc...

La stratégie adopté pour réduire la taille du graphe (en terme de fonctionnalité et non comportementales) est d'analyser l'équivalence et de la domination de certains services sur les autres en termes des fonctionnalités qu'ils offrent. Différents fournisseurs offrent des services similaires avec des interfaces qui se chevauchent. Dans les scénarios de ce genre, il est facile de se retrouver avec de grands graphes de composition qui rendent la tâche très difficile de trouver des compositions optimales dans un délai raisonnable. Une façon d'attaquer à ce problème est d'analyser la domination de la fonctionnalité entre les services afin de trouver ceux qui sont équivalents ou meilleurs que d'autres en termes d'entrée sorties qu'ils fournissent.

**Définition 5.5.13** (PM équivalent). *Un Process Model  $PM_i = \{I_{PM_i}, O_{PM_i}\}$  est équivalent en terme description fonctionnelle sémantique et non comportementale à  $PM_j = \{I_{PM_j}, O_{PM_j}\}$  ( $PM_i \equiv PM_j$ ) Ssi ( $I_{PM_i} \equiv I_{PM_j}$ ) et ( $O_{PM_i} \equiv O_{PM_j}$ ), dont les entrées sont équivalentes et les sorties aussi respectivement. i.e.,  $I_{PM_i} = \{I_{PM_i}^1, I_{PM_i}^2, \dots, I_{PM_i}^m\}$ ,  $I_{PM_j} = \{I_{PM_j}^1, I_{PM_j}^2, \dots, I_{PM_j}^m\}$ , Ssi  $\forall k \in \{1, \dots, m\}$ ,  $I_{PM_i}^k \preceq I_{PM_j}^k$  (de même pour les sorties)*

**Définition 5.5.14** (PM subsumant). *Un Process Model  $PM_i = \{I_{PM_i}, O_{PM_i}\}$  est susumant en terme description fonctionnelle sémantique et non comportementale à  $PM_j = \{I_{PM_j}, O_{PM_j}\}$  ( $PM_i \equiv PM_j$ ) si le premier domine le deuxième dans au moins un aspect (entrée-sortie-dominant ou dominant) et est au moins équivalent dans l'autre aspect. Ssi ( $I_{PM_i} \triangleright I_{PM_j} \wedge O_{PM_i} \triangleright O_{PM_j}$ )  $\vee$  ( $I_{PM_i} \equiv I_{PM_j} \wedge O_{PM_i} \triangleright O_{PM_j}$ )  $\vee$  ( $I_{PM_i} \triangleright I_{PM_j} \wedge O_{PM_i} \equiv O_{PM_j}$ ).*



## 5.5. COUCHE SERVICES WEB SÉMANTIQUE INTER-RELIÉS

*i.e.*,  $I_{PM_i} = \{I_{PM_i}^1, I_{PM_i}^2, \dots, I_{PM_i}^m\}$ ,  $I_{PM_j} = \{I_{PM_j}^1, I_{PM_j}^2, \dots, I_{PM_j}^m\}$ ,  $Ssi \exists k \in \{1, \dots, m\}$ ,  $I_{PM_i}^k \preceq I_{PM_j}^k$  noté ainsi  $PM_i \triangleright PM_j$

Ainsi, le graphe de composition de de cluster de services est essentiellement une couche graphe acyclique orienté (DAG),  $G = (V, E)$ , où :

- $V = \{PM_i\}_{i \in 1..k}$  est l'ensemble des sommets du graphe constituant le cluster de graphe, où  $L_{PM}$  est l'ensemble des Process Model pertinents tel que,  $PM_i \in L_{PM}$  et  $\forall i, j \in 1..k, (PM_i \equiv PM_j) \vee (PM_i \triangleright PM_j)$ .
- $E$  est l'ensemble des arêtes représentant une relation sémantique entre les concepts sortants d'un service  $PM_i$  vers les concepts entrants d'un autre service  $PM_j$ .

Comme expliqué précédemment, la génération du graphe de composition des process models pertinents se fait vers l'avant, couche par couche. Au cours de cette expansion vers l'avant du graphe, Le filtre étendu n'est pas intéressés par les services qui n'ont pas d'effets explicite sur la composition, où qui ne contribuent pas aux objectifs de sortie.

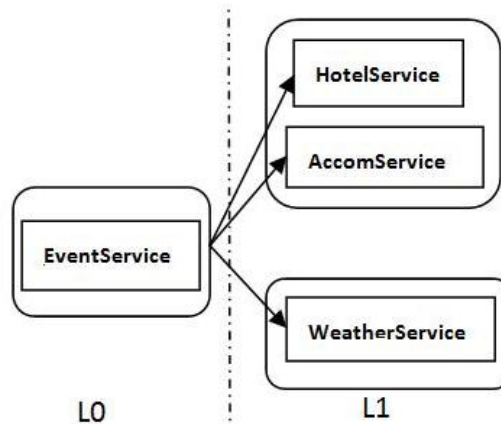


FIGURE 5.9 – Un multi-graphe représentant toutes les possibilités

La figure 5.9 démontre que le cluster de service de la première couche (EventReservationService) est inter-relié avec trois services regroupés en deux services, (1) le premier contient les deux services d'hébergement HotelReservationService et AccommodationReservationService, nous constatons que subsumant HotelReservationService est subsumant AccommodationReservationService car ce dernier est englobé dans le premier en termes d'entrées et de sorties ( $s : \text{startDate} / s : \text{endDate} / s : \text{City} / s : \text{Country}$  est équivalente à  $\text{dbpedia} : \text{startDate} / \text{dbpedia} : \text{endDate} / \text{dbpedia} : \text{City} / \text{dbpedia} : \text{Country}$ ) et possède le même paramètre de sortie ( $s : \text{lodgingReservation}$ ), Et (2) contient seulement le WeatherService.

Les optimisations précédentes ont pour but de réduire le graphe de composition généré, mais en conservant la même fonctionnalité. La prochaine étape consiste à effectuer une re-

cherche sur le graphique pour trouver la meilleure composition parmi toutes les compositions possibles qui répondent à la requête d'entrée / sortie sémantique en termes de processus atomiques.

**Recherche de la Composition optimale** Le fonctionnement du canevas peut être divisé en petites unités logiques plutôt que de prendre en considération tout le process model. Les optimisations précédentes ont pour but de réduire le graphe de composition généré, mais en conservant la même fonctionnalité. La prochaine étape consiste à effectuer une recherche sur le multigraphe afin de trouver la meilleure composition parmi toutes les compositions possibles qui répondent à la requête d'entrée / sortie sémantique.

Le multi-graphe généré précédemment sera utilisé afin de l'optimiser efficacement et ceux en ciblant la sélection uniquement des processus atomiques pertinents et utiles tout en préservant les exigences spécifiées dans la requête du consommateur. L'approche consiste à appliquer des optimisations locales au sein de chaque **cluster de service** candidat indépendamment des autres groupes de la même couche mais qui dépends des données issues de la couche précédente. Pour chaque groupe, un seul service peut être considéré en fonction de ses processus atomiques minimaux dont la satisfaction est maximale en termes d'entrées sémantiques. On propose une heuristique complexe qui englobe plusieurs critères tels que : le nombre de concepts non satisfaits, le nombre de concepts satisfaits, le nombre de concepts produits en sortie, la taille de la composition.

Une fois le graphe construit, on raffine ce dernier par rapport à ces processus atomiques, l'idée est de prendre en considération le Process Model dont le nombre minimal de processus atomiques est satisfait par le maximum de ses entrées 5.5.2.2.

$$\text{Sim}_{sem} = \sum_{s=1}^h \text{Sim}_{AP_s} / h \quad (5.1)$$

Ainsi,  $\text{Sim}_{sem}$  représente le degré d'appariement global pour chaque Process Model faisant parti d'un même Cluster de services du graphe, ce degré de similarité correspond à la sommation de l'ensemble des degrés d'appariements liées aux processus atomiques  $\text{Sim}_{AP_i}$  (on s'est inspiré du travail de Norton et al ; [Norton & Stadtmaller 2011]) utiles à la composition 5.5.2.2.

$$\text{Sim}_{AP_i} = \text{Sim}(AP_i.\text{input}_k^j, I_{av}) / |AP_i.\text{input}| \quad (5.2)$$

## 5.5. COUCHE SERVICES WEB SÉMANTIQUE INTER-RELIÉS

Nous proposons un algorithme de sélection raffinée des processus atomiques en appliquant des techniques d'optimisations locales. Nous proposons une heuristique complexe qui englobe plusieurs critères tels que : le nombre de concepts non satisfaits, le nombre de concepts satisfaits, et la taille de la composition. Cet algorithme est résumé par le pseudo-code répertorié ci-dessous.

```

1: AtomSelection(L, IQ, OQ)
2: Iav := IQ
3: Ireq := OQ
4: Comp := ∅
5: Sol := ∅
6: for i = 1 to GetNrLayer(L) do
7:   begin
8:     for j = 1 to GetNrCluster(L, i) do
9:       begin
10:        Service=GetService(i,j)
11:        for k = 1 to Service.size() do
12:          begin
13:            atom[k]=Service.GetAP(Iav, Ireq)
14:            if Service.size() = 1 then
15:              goto marker
16:              Sim[k] = Simsem(atom[k], Iav)
17:            end
18:            psol= ComputeMax(Sim)
19:          marker:
20:            Iav = Iav ∪ {atom[psol].Output}
21:            Ireq = {a | a ∈ {Ireq ∪ atom[psol].Input} ∧ ∄ b ∈ IAv : b ≼ a}
22:            Comp = Comp ∪ {atom[psol]}
23:          end
24:          Sol = Sol ∪ {ExtractSubMtx(APM[atom[psol]])}
25:        return Sol
26:      end

```

Le résultat ainsi renvoyé ne peut pas être vide, en raison de l'efficacité de la phase de pré-sélection. Notre algorithme prend comme paramètres d'entrée :

- Le multigraphe (L),
- I<sub>av</sub> l'ensemble des paramètres d'entrées disponibles (initialisé aux paramètres d'entrées fournies dans la requête I<sub>Q</sub> (ligne 2)),
- I<sub>req</sub> l'ensemble des objectifs à atteindre (initialisé aux paramètres de sorties désirés dans la requête O<sub>Q</sub> (ligne 3)),
- L'ensemble Comp, initialement vide et qui contiendra les processus atomiques sélectionnés lors des itérations (ligne 4)
- et l'ensemble Sol initialement vide et qui contiendra les sous-matrices extraites (ligne 4).

Après l'initialisation, l'algorithme de AtomSelection explore le multi-graphe L, en parcourant chaque nœuds (ligne 8) de chaque couche (ligne 6) et extrait le sous-ensemble de

processus atomiques pertinents (`Service.GetAP ()`) pour chaque Process Model présent dans le cluster de service actuel. Cette extraction doit satisfaire l'ensemble des entrées définies par le processus atomique et correspondant à  $I_{av}$  (ligne 11) en inhibant les processus atomiques inutilisés.

Etape suivante, l'algorithme calcule le degré de correspondance ( $Sim_{sem}$ ) des processus atomiques extraits pour chaque Process Model (ligne 16) et sélectionne les plus prometteurs dont la mesure de similarité est la plus élevée conformément à la formule 5.5.2.2 qui tri les Process Model en fonction de la taille des processus atomiques satisfaits (ligne 18). Ayons un Process Model contenant M processus atomiques, mais seulement H processus atomiques ( $H < M$ ) correspondant aux entrées disponibles.

La Similarité sémantique  $Sim_{AP}$  vérifie la compatibilité des entrées de la requête par rapport aux entrées des processus atomiques correspondant contenue dans le Process Model et éliminer les inutilisés.  $Sim_{AP}$  est un score de correspondance calculé sur un ensemble de paires  $(AP_k.j.input_i, I_{av})$  des paramètres d'entrée du  $i^e$  processus atomique contenue dans le cluster k de la couche j et les entrées disponibles.

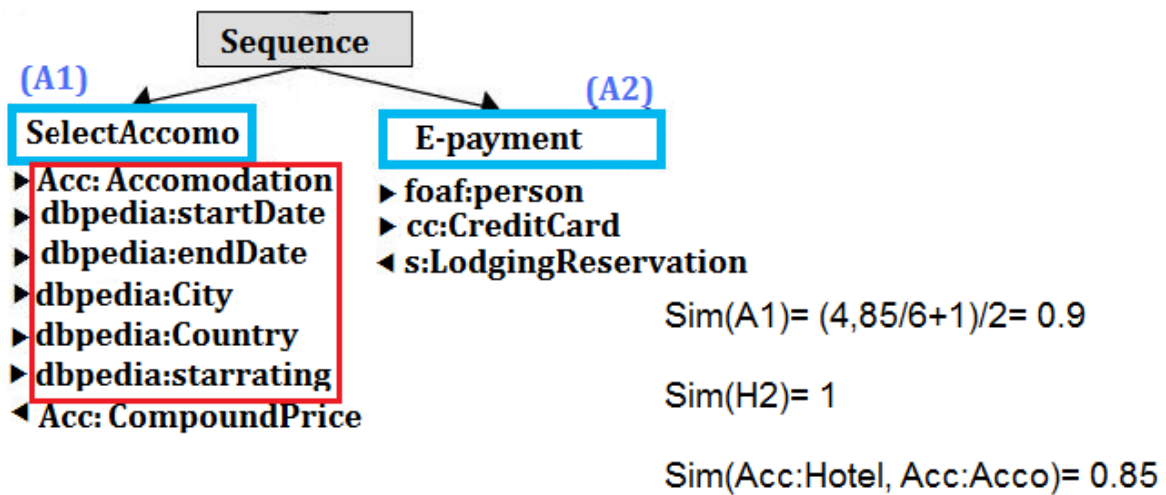
AtomSelection ajoute ce sous-ensemble de processus atomiques à la liste de composition *comp* (ligne 23), et par la suite d'en extraire la sous-matrice en fonction de ce sous-ensemble représentant un mini-workflow (ligne 21). Ensuite, l'algorithme rafraichit les entrées disponibles  $I_{av}$  en y ajoutant les sorties des processus atomiques sélectionnés (ligne 19), et met à jour également les entrées nécessaires  $I_{req}$  en ajoutant les entrées des processus atomiques tout en supprimant les concepts qui sont maintenant disponibles dans l'ensemble des entrées disponibles (lignes 20). Ensuite, AtomSelection poursuit itérativement les mêmes techniques sur chaque couche. Quand il n'y a pas de sorties générés, la sélection AtomSelection renvoie l'ensemble des sous-matrices sélectionnées (ligne 24), satisfaisant la requête du consommateur fonctionnelle. Sachant qu'une sous-matrice SPM d'un graphe  $APM = (AS, E)$  est un sous-graphe  $SPM = (AP', E')$  où  $AP' \subseteq AP$  et  $E' \subseteq E$ .

Poursuivons avec le même exemple, AtomSelection prend comme paramètres le graphique L, les entrées de la requête  $I_{av} = \{\text{tio} : \text{événement}, \text{FOAF} : \text{personne}, \text{cc} : \text{carte de crédit}, \text{acc} : \text{hôtel}\}$  et les sorties de la requête  $I_{req} = \{s : \text{Eventreservation}, s : \text{lodgingreservation}, w : \text{weathercondition}\}$ , tandis que la composition est un ensemble vide. Dans un premier temps, la première couche contient un service. AtomSelection va extraire directement les deux processus atomiques (E2 et E4) dont les entrées sont entièrement satisfaites par  $I_{av}$  du service EventService et également la sous-matrice correspondante. Dès lors, les deux ensembles  $I_{av}$ ,  $I_{req}$  sont rafraichis par les sorties et entrées de E2 et E4 tel que  $I_{av} = \{\text{tio} : \text{événement}, \text{FOAF} : \text{personne}, \text{cc} : \text{carte de crédit}, \text{acc} : \text{hôtel}, \text{dbpedia} : \text{stratdate}, \text{dbpedia} : \text{enddate}, \text{dbpedia} : \text{ville}, \text{dbpedia} : \text{pays}, s : \text{eventreservation}\}$ ,  $I_{req} = \{s : \text{lodgingreservation}, w : \text{weathercondition}\}$ . La seconde itération contient deux clusters de service, on applique l'algorithme d'optimisation pour chaque cluster de service, le premier dispose de deux services semblable, la nécessité de calculer le degré sémantique entre ces deux services est nécessaire afin d'en sélectionner la meilleure solution. Pour la HotelService, il extrait deux processus atomiques satisfaits (H2 et H3), en calculant le degré d'appariement qui est équivalent à 1 en raison du fait que toutes les entrées des processus atomiques sont satisfaisantes par les entrées disponibles. Pour le deuxième service, il extrait deux processus atomiques (A1 et A2) avec

## 5.5. COUCHE SERVICES WEB SÉMANTIQUE INTER-RELIÉS

un degré moindre d'appariement, car l'entrée « starranking » de A1 ne peut être satisfaite. D'où l'ajout du premier service dans la liste comp et extrait la sous-matrice correspondante. Quant à l'autre cluster de service, il ajoute W1 puis on rafraichit les deux ensembles  $I_{av} = \{tio : \text{événement}, FOAF : \text{personne}, cc : \text{carte de crédit}, acc : \text{hôtel}, dbpedia : \text{stratdate}, dbpedia : \text{enddate}, dbpedia : \text{ville}, dbpedia : \text{pays}, s : \text{eventreservation}, s : \text{lodgingreservation}, w : \text{weathercondition}\}$ ,  $I_{req} = \emptyset$ . Ainsi  $comp = \{E2, E4, H2, H3, W1\}$ . Sur ce la nécessité de les ordonner est primordiale.

### Accommodation Reservation Service



### Hotel Reservation Service

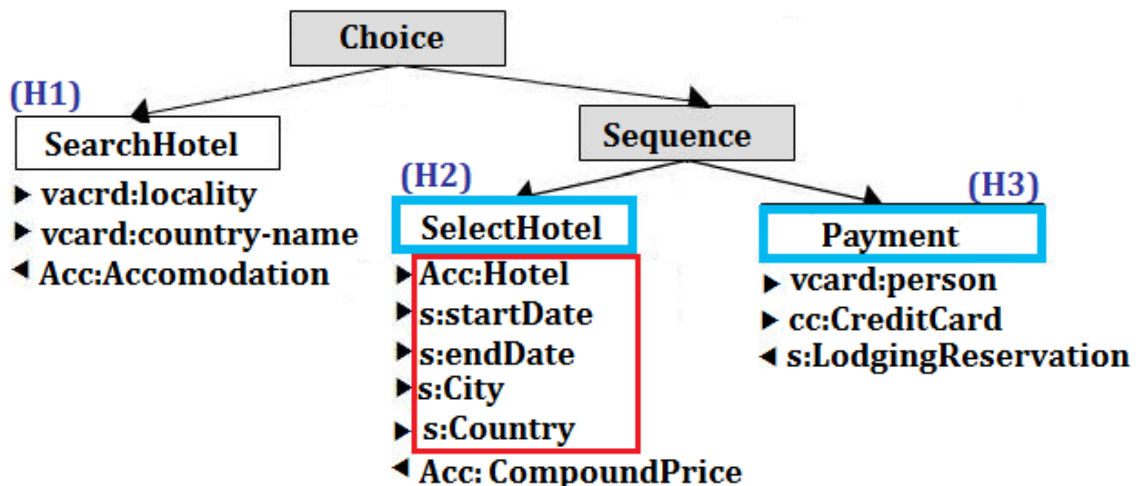


FIGURE 5.10 – sélection raffinée par Optimisation locale entre les deux processus de réservation

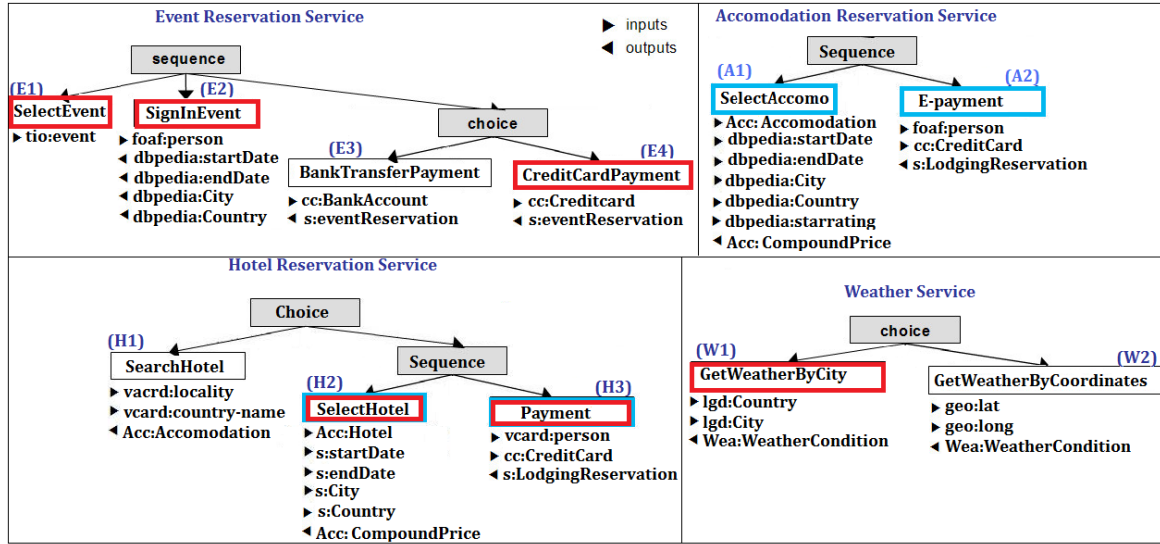


FIGURE 5.11 – Sélection finale des processus atomiques

### 5.5.2.3 Composition automatique des Processus atomiques

**Préliminaires** La phase finale de notre approche est la construction d'un workflow. Le workflow est une représentation d'une suite ordonnée de tâches effectuées par une personne, une entreprise etc, servant à décrire et exécuter un processus ou permettant de décrire et exécuter de manière automatique les étapes de traitements complexes et pertinents des données. D'une part, les business workflows sont des workflows utilisés dans de nombreux domaines comme les domaines commerciaux, financiers, ou encore pharmaceutiques. Ils ont pour but de décrire de manière organisationnelle les étapes d'un processus : par exemple définir les étapes de validation d'un document ou d'une commande de produit, avec des étapes de validation par certains acteurs de l'entreprise, paiements bancaires, génération de factures, transformation d'un document d'un format à un autre et à une heure précise. Notons que dans ce type de workflow, le résultat de l'exécution est connu à l'avance et le workflow ne sert donc qu'à traduire de manière explicite le processus qu'il implémente : on parle alors de chaîne de planification de travail. Dans un business workflow, les étapes (nécessitant parfois l'intervention humaine) sont reliées par des arcs qui représentent le flot de contrôle : on passe d'une étape à l'autre parce que des évènements déclenchent le passage. Dans notre cas, le but est de générer un workflow dynamique en temps réel dont les noeuds se sont les processus atomiques basé sur la fermeture transitive d'une matrice carré et acyclique .

**Motivation** Le problème de la composition sémantique prend en entrée une requête SPARQL utilisateur  $R_Q = \{I_Q, O_Q\}$ , ainsi le problème de la construction automatique de la composition des SERVICES WEB SÉMANTIQUES [Jorge 2006, McIlraith et al. 2001] consiste à identifier un Service Web Composite  $SWC = I_{swc}, O_{swc}, P = AP, \leq\}$  tel que  $match(I_Q, I_{swc}) = I_{swc}$  (le service composite est invocable avec les entrées disponibles) et  $match(O_{swc}, O_R) = O_R$  (le service composite récupère toutes les sorties demandées). Ce service se compose d'un ensemble partiellement ordonné  $P$  (une relation binaire " $\leq$ " sur un

sous-ensemble de process Model  $L_{AP}$ ).

**Contribution** Nous proposons une modélisation des transformations par des graphes pour la spécification des packages dynamiques et leurs adaptations. Une description des Packages dynamiques basées sur les graphes présente, outre l'aspect formel rigoureux, l'avantage d'offrir une description visuelle et facilement compréhensible qui ne nécessite aucun pré-requis concernant les langages formels. Tout nouveau besoin sera modélisé par l'ajout de nouvelles sous-matrices dans le mécanisme de transformation de graphes.

Cette partie introduit la notion de fermeture transitive de matrice d'adjacence afin de générer dynamiquement l'enchaînement ou l'ordre logique de l'ensemble des processus atomiques  $AP_i \subseteq PM_i$  sur la base du résultat retourné par l'étape précédente (liste de sous-matrices qui est générée du graphe de dépendance).

Etant donnée un ensemble de sous-matrice carré, le but est de les rassembler en une seule structure de données qu'est la matrice. Cette matrice globale contiendra les sous-workflow issu de l'étape précédente.

À la suite de la section précédente, un ensemble de sous-matrices représentant le comportement des sous-workflows doivent être fusionné pour former une nouvelle matrice carrée, la matrice composition de dépendance (CMD);

$$CMD = \bigcup_{i=1}^m SPM_i.$$

Par exemple, nous allons avoir deux sous-matrices  $SPM_1$  et  $SPM_2$  où les sommets de chaque sous-matrice sont définis comme ci-dessous  $V(SPM_1) = \{AP_{11}, AP_{12}, \dots, AP_{1k}\}$  et  $V(SPM_2) = \{AP_{21}, AP_{22}, \dots, AP_{2n}\}$ . La nouvelle matrice CMD généré fusionne  $SPM_1$  et  $SPM_2$  où :  $V(CMD) = \{AP_{11}, AP_{12}, \dots, AP_{1k}, AP_{21}, \dots, AP_{2n}\}$ .

$$MD = \begin{pmatrix} SPM_1 & 0 \\ 0 & SPM_2 \end{pmatrix}.$$

La fermeture transitive est utilisée dans plusieurs applications scientifiques (résolution de systèmes d'équations linéaires, inversion de matrices, calcul de valeurs propres, de déterminants, etc.). Dans cette partie, on s'inspire de la fermeture transitive d'une matrice carré qui est basé sur le calcul du produit de deux matrices d'une façon successive. Le but de la fermeture transitive est de générer le préordre<sup>5</sup> des processus atomiques formant ainsi un workflow.

Le *SWC* correspond à l'ordre partiel (Séquentiel et parallèle) des Processus Atomiques est le résultat de la fermeture transitive de *CMD* (*CMD\**). Ensuite, le *CMD* est rafraîchit en ajoutant les intra-dépendances identifiés lors de la première phase. La construction entièrement automatique des nouveaux résultats de service composite à partir de la fermeture transitive de *CMD* :

$CMD^+ = CMD^1 + CMD^2 + \dots + CMD^N$  où  $CMD^N$  est la matrice obtenue en multipliant *MDP* avec elle même (*N*) fois .  $CMD^+$  est une matrice avec le même ensemble de sommets et une arête entre deux sommets avec un chemin de longueur maximum (*N* - 1).

---

5. Un préordre est une relation binaire réflexive et transitive.

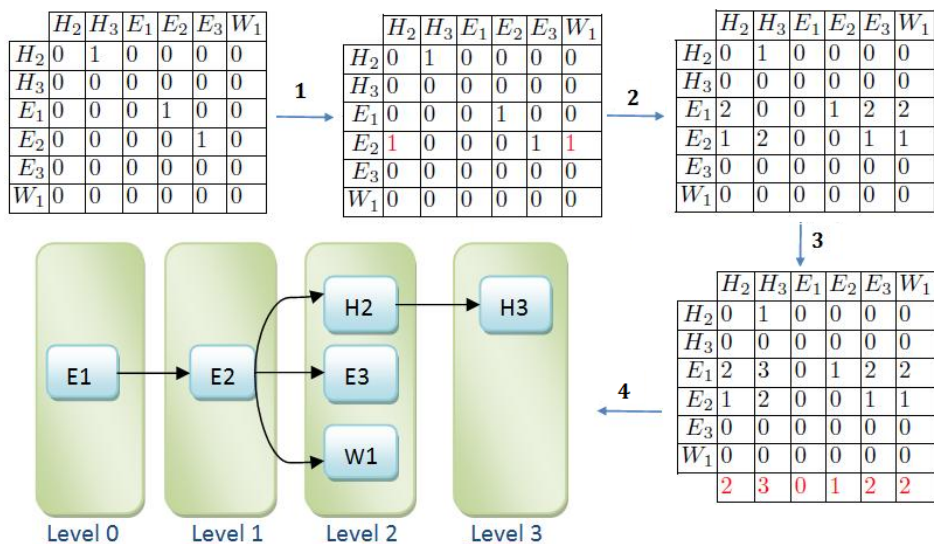


FIGURE 5.12 – La Fermeture Transitive de l'exemple touristique

Un chemin de longueur entre deux sommets  $AP_i$  et  $AP_j$  existe pour chaque arc  $AP_k$  tel que  $(AP_i, AP_k)$  et  $(AP_k, AP_j)$  sont définis dans E.

Le résultat de la fermeture transitive matrice contient un ensemble de chemins entre les nœuds qui définissent le niveau de connectivité requis pour chaque processus atomique dans le nouveau processus composite correspond à la sélection de la valeur maximale de chaque colonne de la matrice.

Le résultat de la fermeture transitive génère un multigraphe en couches où la valeur minimale des max-colonnes représente les nœuds de départs, et incrémentalement intègre les APs dans le flux de travail jusqu'aux nœuds finaux. En conséquence, un graphe de composition hiérarchique est généré automatiquement.

Afin de calculer rapidement la multiplication de matrice, nous avons utilisé l'algorithme de de Strassen [Thottethodi et al. 1998] fonctionnant en une exécution asymptotique de  $\Theta(n^{\log 7})$  ou  $\approx \Theta(n^{2,81})$ , ce qui de cet algorithme qui est asymptotiquement plus vite que la multiplication de la matrice traditionnelle, avec l'exécution  $\Theta(n^3)$ .

Les Web services peuvent être instables : pour de multiples raisons (panne, mauvais assemblage, non déterminisme, . . .), les services n'ont pas toujours le comportement spécifié (ce qui n'est pas le cas en planification classique). De plus, certains Web services contenus dans des registres de types UDDI peuvent être supprimés par les fournisseurs sans que l'annuaire soit à jour. Ainsi, notre algorithme offre la possibilité de remplacer un service par un autre dans une composition à travers le graphe de cluster.

## 5.6 Conclusion

Le canevas ainsi proposé possède les caractéristiques suivantes :

- *Modulaire* : les préoccupations concernant la conception de la composition sont séparées de celles concernant la spécification de ses propriétés fonctionnelles. Cette spécification



est faite indépendamment des propriétés fonctionnelles hétérogènes des services web composants.

- *Flexible* : il est possible de déterminer, au moment de la conception, quelles capacités de services sont sélectionnées pour une composition donnée. Cette caractéristique permet de relâcher la propriété d'atomicité pour la transaction et permet d'obtenir des compositions différentes selon la disponibilité de composants dans un moment d'exécution donné.

Le chapitre suivant aborde l'expérimentation de notre canevas.



# 6

## Implémentation & Expérimentations

### Sommaire

---

<b>6.1 Introduction</b>	<b>155</b>
<b>6.2 Implémentation &amp; Expérimentations</b>	<b>155</b>
6.2.1 Les outils technologiques utilisés	155
6.2.2 Les technologies sémantiques	156
<b>6.3 Description du prototype</b>	<b>156</b>
<b>6.4 Environnement de développement</b>	<b>159</b>
<b>6.5 Conclusion</b>	<b>162</b>

---

### 6.1 Introduction

---

Nous venons de détailler dans le chapitre précédent l'architecture proposée de découverte orienté composition. Dans le présent chapitre nous décrivons un prototype implémentant cette architecture ainsi que les résultats obtenus. Nous allons commencer par présenter les différents outils techniques liés à l'implémentation, ensuite nous présentons notre prototype baptisé " SQUIREL" ainsi que les expérimentations menées pour analyser notre approche.

### 6.2 Implémentation & Expérimentations

---

#### 6.2.1 Les outils technologiques utilisés

Le prototype a été développé sur un Intel Core2duo, avec une vitesse de 1.6 GHZ, doté d'une capacité mémoire de 2GB de RAM sous Windows Seven en utilisant des outils Open source graphiques et développés en JAVA. Nous détaillons, dans ce qui suit, chacun des outils et langages utilisés pour la manipulation des données ainsi que l'implémentation de l'interface utilisateur.

Pour le langage de programmation notre choix s'est porté sur le langage JAVA qui est un langage orienté objet simple ce qui réduit les risques d'incohérence; il est portable, il peut être utilisé sous Windows, sous Linux, sous Macintosh et sur d'autres plateformes sans aucune modification , enfin il possède une riche bibliothèque de classes comprenant des fonctions diverses telles que les fonctions standards, le système de gestion de fichiers , les

fonctions multimédia et beaucoup d'autres fonctionnalités ;

Pour le choix de l'environnement de développement, on a opté pour Eclipse car il possède de nombreux points forts qui sont à l'origine de son énorme succès puisqu'il offre une plateforme ouverte pour le développement d'applications et extensible grâce à un mécanisme de plugins. Il supporte plusieurs plates-formes d'exécution : Windows, Linux, Mac OS. C'est un environnement de développement très rapide à l'exécution grâce à l'utilisation de la bibliothèque SWT ;

### 6.2.2 Les technologies sémantiques

Ces outils sont représentés soit sous formes d'API, de Plug-ins :

L'API OWL-S <sup>1</sup> : c'est une API Java développé par Mindswap pour créer, lire, écrire, et exécuter des descriptions OWL-S des services atomiques ainsi que composites. Dans le cadre de notre système, nous avons utilisé plusieurs interfaces de cette API comme `OwlsService` et `OwlsProcess`, etc. Cette dernière nous permet de naviguer sur le Process Model afin d'extraire les processus composites et atomiques nécessaires à la représentation interne du flux de contrôle de chaque SERVICES WEB SÉMANTIQUES [Jorge 2006, McIlraith *et al.* 2001] transactionnel présent dans le registre d'une part, et d'autre part de transformer la fonctionnalité du Process Model en turtle afin de la sauvegarder dans le triple store. L'API `owl-s.jar` est ajoutée ainsi dans la Classpath du projet de notre système SQUIREL.

L'API Pellet Cette API permet le raisonnement sur les ontologies formalisées avec OWL, elle offre des mécanismes d'inférences basés sur les logiques de description. Cette API est utilisée dans le cas où aucun alignement n'est disponible à partir des jeux de données touristiques. On utilise ces mécanismes afin d'enrichir les triples stores des SERVICES WEB SÉMANTIQUES [Jorge 2006, McIlraith *et al.* 2001] par les alignements de concepts partant d'un seuil prédéfini à l'avance.

ARQ API est une API Java pour les applications du web sémantique. L'API a été définie sous formes des interfaces afin que le code de l'application puisse fonctionner avec différentes implémentations sans changement. Nous avons utilisé cette API afin de traiter les descriptions des ontologies de domaine. Nous l'avons utilisé aussi pour développer notre application web.

## 6.3 Description du prototype

---

L'architecture s'articule autour de plusieurs modules :

**Module d'Interrogation sémantique** : C'est un évaluateur pour répondre aux requêtes PSPARQL ou CPSPARQL. L'entrée à ce module est une requête sous une forme de texte rédigée dans le langage Turtle [Beckett 2006] où chaque RDF graphe est identifié par une URL dans la requête. L'évaluateur est fourni avec deux parseurs :

---

1. <http://on.cs.unibas.ch/owl-s-api/>

### 6.3. DESCRIPTION DU PROTOTYPE

un pour parser des graphes RDF écrites en Turtle, et l'autre pour parser des requêtes écrites selon la syntaxe de CPSPARQL, qui est compatible avec la syntaxe de SPARQL<sup>2</sup>. Ce module analyse la requête utilisateur afin de la rediriger vers le module concerné. Si la requête est informative, le module l'a redirigera vers le point d'accès du jeu de données touristiques sinon il l'enverra au module de Réécriture sémantique. La sortie de l'évaluateur est le résultat de la requête sous la forme de triplets en cas d'interrogation informative et de plan optimal en cas d'interrogation transactionnel.

Virtuoso SPARQL Query Editor

[About](#) | [Namespace Prefixes](#) | [Inference rules](#)

Default Data Set Name (Graph IRI)  
http://dbpedia.org

Query Text

```
select distinct ?Concept where {[[] a ?Concept]} LIMIT 100
```

(Security restrictions of this server do not allow you to retrieve remote RDF data, see [details](#).)

Results Format: HTML

Execution timeout: 30000 milliseconds (values less than 1000 are ignored)

Options:  Strict checking of void variables

(The result can only be sent back to browser, not saved on the server, see [details](#))

Run Query Reset

FIGURE 6.1 – Point d'accès SPARQL : virtuoso

**Module de Réécriture sémantique** : C'est un module qui consiste à transformer la requête initiale de l'utilisateur en une nouvelle requête tout en préservant sémantiquement les besoins de l'utilisateur. Cette réécriture consiste d'une part à filtrer les SERVICES WEB SÉMANTIQUES [Jorge 2006, McIlraith *et al.* 2001] en fonction de leurs description sémantiques et d'autres part d'intégrer directement ou indirectement les

2. <http://psparql.inrialpes.fr>

liens sémantiques issus des jeux de données afin d'enrichir la phase de découverte aux services les plus pertinents ; Le résultat retourné par cette requête relaxée est fourni au module de génération de graphe.

**Module de Transformation des Service OWL-S** : C'est un module qui consiste à transformer la partie fonctionnalité du Process Model de chaque SERVICES WEB SÉMANTIQUES [Jorge 2006,McIlraith *et al.* 2001] en format Turtle afin de la sauvegarder dans le triple store concerné. D'autre part, chaque Porcess Model doit avoir la représentation interne de son flux de contrôle sous forme de matrice d'adjacence à l'aide de l'API OWL-S. Ce module doit effectuer les deux transformations pour l'ensemble des SERVICES WEB SÉMANTIQUES [Jorge 2006,McIlraith *et al.* 2001] présents dans le registre UDDI et à chaque nouvelle entrée dans le registre afin de procéder à la meilleure composition possible.

**Module d'Acquisition de l'alignement de Schéma** : L'objectif de ce module est d'extraire les alignements de schémas entre concepts à partir du jeu de données utilisées. Pour chaque paire de source de données présentes dans le jeu de données est soumise à l'outil SILK afin d'évaluer les alignements sémantiques possibles. Ce module capture alors ces alignements afin de les transformer en Turtle et les renvoyer au triple store des SERVICES WEB SÉMANTIQUES [Jorge 2006,McIlraith *et al.* 2001] . Ce module effectue régulièrement une vérification pour d'éventuelles alignements à rajouter afin d'enrichir le triple store ;

**Module de Fermeture transitive de matrice** : Ce module sert à fusionner les sous-matrices ou matrices issues de l'algorithme d'optimisation du graphe de cluster de service et ceux indépendamment de leurs nombres en une nouvelle matrice carré. Une fois la matrice construite et enrichie par l'interdépendance entre les processus atomiques, on applique l'addition des multiplications successives de matrices en utilisant l'algorithme qui présente la moindre complexité. Une fois les calculs effectués, le but est de fournir le nouveau plan servant de package dynamique sous forme de niveaux ou chaque niveau correspond à la valeur maximale de chaque colonne. Finalement, en se basant sur l'algorithme de [Thottethodi *et al.* 1998], on pourra générer le nouveau plan de composition.

**Module de génération de graphe** : Ce module consiste à générer un multi-graphe de cluster de services. L'objectif de ce module est de réduire la taille du graphe en regroupant les Process Models similaires en cluster eux-mêmes réparties sur un multi-graphe où chaque niveau est une conséquence du niveau précédent, chaque Process model est un prédécesseur d'un autre. Les niveaux sont reliés par l'interdependance des Process Model. L'ultime but est de réduire la taille du cluster en un seul service pertinent en fonction des entrées et sorties fournies par l'utilisateur d'une part et les sorties délivrées des Process Models issus des niveaux précédents d'autre part. Ces deux points détermineront le meilleur service en fonction du nombre minimal de processus atomiques nécessaires à la composition mais dont la satisfaction est maximale en terme d'entrées et sorties. Dans ce module nous appliquons l'algorithme décrit dans la section 5.5.2.2.

## 6.4. ENVIRONNEMENT DE DÉVELOPPEMENT

Nombre de Process model	20	40	60	80	100
Nombre de processus atomique impliqués	13	19	33	35	38
Temps de découverte globale et locale	653.75	2764.64	6430.01	6218	6495
Temps de composition	0.243	0.356	0.987	545	687

TABLE 6.1 – Les mesures de l’expérimentation en millisecondes

### 6.4 Environnement de développement

Les mesures ont eu lieu pour les domaines de tailles différentes, à savoir les Process Model 20, 40, 60, 80 et 100 OWL-S. Toutes les expérimentations ont été faites à travers l’enrichissement du jeu de données des services par les alignements sémantiques de schéma. On a défini plusieurs requêtes dont les expressions régulières parcourraient le jeu de données services.

Pour la phase de découverte, on dispose de l’interrogation de deux aspects différents, soit la partie Process Model qui délivre des fonctionnalités pointant vers des concepts sémantiques référencés dans une ontologie ou bien la partie profil qui délivre des fonctionnalités purement syntaxiques. A travers le filtre défini par EMMA et enrichie par notre approche par des informations additionnelles assurent que les résultats ont montré que la précision de la découverte orienté Process Model est bien plus grande et concise que la découverte orienté Service profile. En outre, le nombre de résultats de la découverte orienté Process Model, est plus grand et pertinente par rapport au deuxième qui se focalise uniquement sur les mesures syntaxiques, ceci confirme la haute précision du premier algorithme pour une meilleure composition.

La Figure 6.2 montre les résultats obtenus pour chaque ensemble de requêtes et pour chaque ensemble de Process Model. Le temps de réponse est le temps moyen mesuré en nanosecondes afin de répondre à une requête Q en retournant le plan optimal. Les résultats obtenus montrent la faisabilité de notre approche, c’est une solution qui peut garantir une solution optimale pour une collection de test de 110 services formalisés en Process Model OWL-S ayant des tailles différentes et complexes. L’algorithme proposé permet de prendre en charge les liens sémantiques fonctionnelles (i.e. les entrées- sorties des services) mais pas les liens causaux (i.e. les pré-conditions et les post-conditions des services) ; Ces derniers restant une perspective pour la suite de notre travail. L’outil proposé est testé pour la collection de service dans le domaine de l’E-Tourisme mais qui peut se généraliser dans d’autres domaines éventuellement l’E-Santé.

Le principal problème auxquels j’ai été confronté est la non-disponibilité d’un benchmark bien plus étendu afin d’effectuer d’autres expérimentations. La majorité des approches traitent le problème de composition des Process Model contenant simplement un seul processus atomique annotant des concepts sur un ensemble réduit d’ontologies. Notre approche se distingue des autres existants par le fait qu’elle tire profit des avantages liés à l’utilisation des Données Liées et les technologies sous-jacentes nous permettant ainsi de

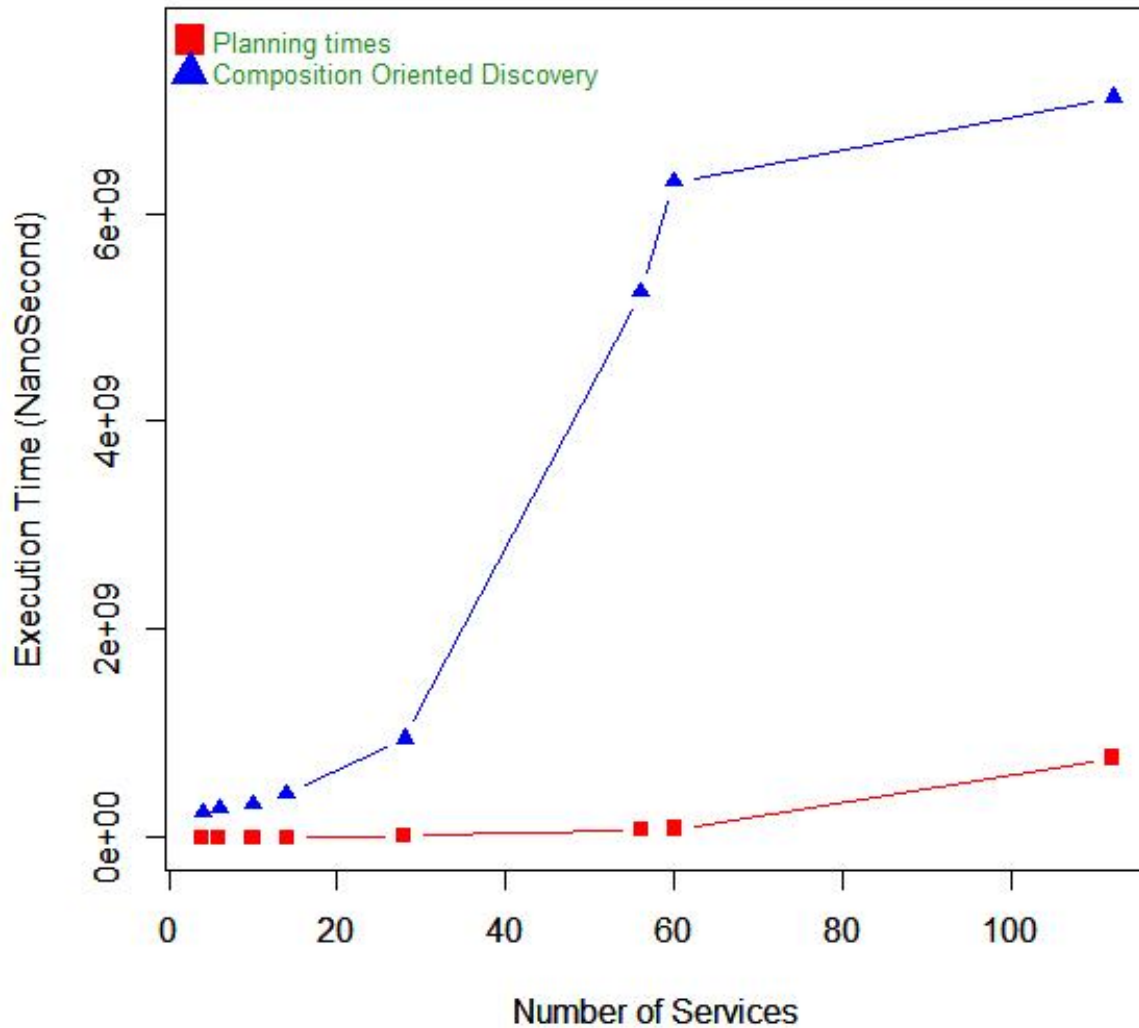


FIGURE 6.2 – Découverte orienté composition

réduire considérablement le temps de découverte à travers les liens sémantiques pré-calculés et rafraichit à chaque enrichissement de ces instances. L'avantage de notre approche et qu'elle ne dispose d'aucun temps de prétraitement ou de traduction comme l'est le cas pour la majorité des approches.

L'algorithme de la fermeture transitive (la planification) n'a pas montré de variation importante, car il dépend uniquement du nombre des processus atomiques candidats (voir Figure 6.3) et leur interdépendance et non sur le nombre de services Web disponibles. En ce qui concerne le temps de planification (fermeture transitive des matrices) montre une augmentation du temps de planification que le nombre de processus augment ; Cependant, l'algorithme de la fermeture transitive est rapide.

La partie la plus coûteuse du notre algorithme, est celle de la découverte (chainage avant),



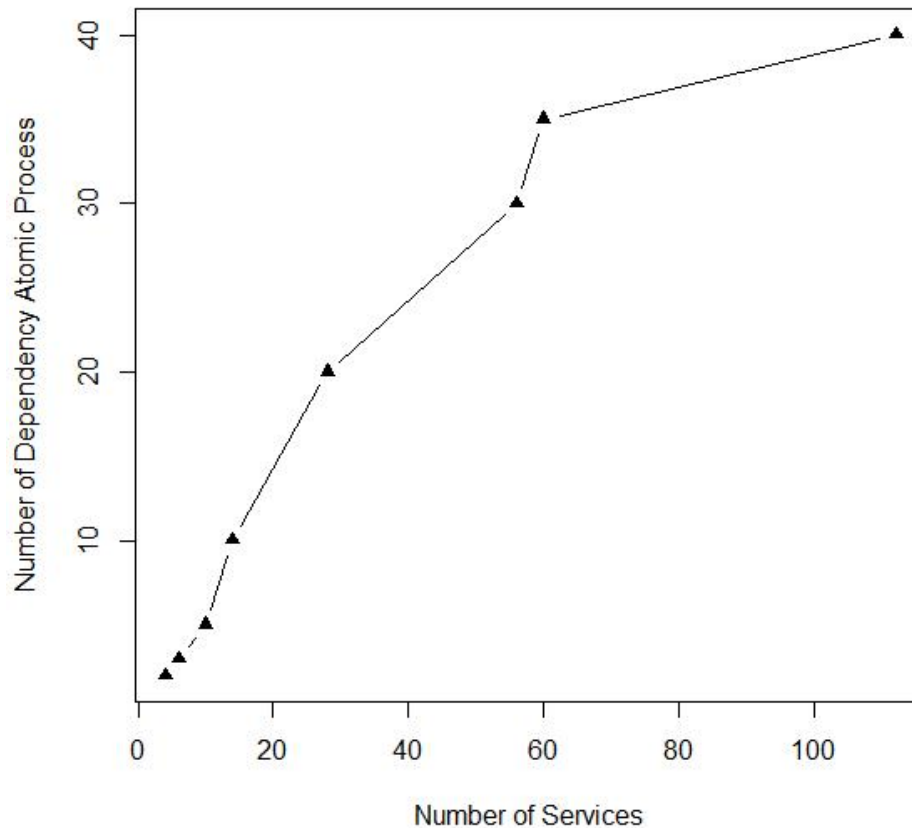


FIGURE 6.3 – Nombre de Processus atomiques

nous notons que le score de notre filtre étendu et hybride en termes de pertinence présente un taux plus élevé que ceux des filtres à base de techniques de recherche d'information ou de logique. De même, il est utile de noter que, la phase d'optimisation est peu coûteuse en termes de temps d'exécution du fait que l'ensemble des calculs de similarité sont pré-effectués et issues des jeux de Données de la deuxième couche. Cela indique que l'analyse des processus équivalent et subsumant de la fonctionnalité des services est une technique puissante pour réduire considérablement l'espace de recherche dans de grands problèmes d'échelle. Ces algorithmes vérifient la compatibilité sémantique du flux de données reliant les constituants de la composition. Plus précisément, les algorithmes s'assurent que le concept d'entrée de chaque service subsume ou équivalent au concept de sortie du service qui le précède. Le temps de découverte totale tend à augmenter dépendamment du nombre de services Web disponibles en raison de la plus grande complexité des triplets disponibles (les process Model peuvent avoir plusieurs entrées et sorties et les leur interrelation). La découverte est exponentielle, plus le nombre de service augmente, plus le temps de découverte est long.

On justifie le gain du temps d'exécution de notre approche par le biais d'un point d'accès SPARQL offrant ainsi un accès rapide aux triplets RDF représentant la fonctionnalité du Process Model. Quant au CPSPARQL, il offre la possibilité de sélectionner les Process

Model pertinents en effectuant une recherche précise en naviguant et parcourant tout les nœuds possibles.

### 6.5 Conclusion

---

L'objectif du développement du prototype est double. D'abord il s'agit d'instancier l'approche d'analyse et la méthodologie présentées dans ce travail. Les premiers résultats sont encourageants et ont démontré la justesse de l'approche. Ils ont permis de décèler de raffiner certains concepts dans le modèle formel. Le deuxième objectif est d'évaluer l'efficacité et la résistance à la montée en charge des techniques présentées. Ce second volet du travail est encore en friche et ouvre la perspective de nouvelles voies de recherche. En effet il reste à étudier et si possible résoudre des problèmes majeurs tels que l'adaptation de techniques de confinement de l'explosion combinatoire due à la complexité des modèles ou encore le problème de l'atteignabilité.

**Troisième partie**  
**Conclusion Générale**



# 7

## Bilan et Perspectives

### Sommaire

---

7.0.1	Bilan du travail réalisé . . . . .	165
7.0.2	Perspectives . . . . .	167

---

Etant donné les enjeux touristiques qui consistent à assurer des recherches informatives et actives plus intuitives. Tout au long de ce document nous avons présenté le travail dont est issu le prototype SQUIREL :

- En assurant une exploration qualitative des contenus touristiques.
- De renvoyer des résultats de recherche plus pertinents et satisfaisant les besoins des utilisateurs en utilisant un langage d'interrogation sémantique.
- De s'inspirer des techniques en Intelligence artificielle pour le web.
- En utilisant des ontologies touristiques et les technologies sous-jacentes notamment les données liées ouvertes et les web services sémantiques.

Nous présentons maintenant un bilan et nous dressons plusieurs perspectives, envisageables tant sur le plan théorique que sur le plan pratique.

### 7.0.1 Bilan du travail réalisé

Notre approche peut être vue comme une plate-forme qui tire parti de l'intelligence collective combinée avec la puissance des DONNÉES LIÉES [Heath & Bizer 2011, Berners-Lee 2009, Bizer *et al.* 2009, Berners-Lee 2006] . Ce principe sous-entend que le Web et les dispositifs qui lui sont connectés sont considérés comme une plate-forme globale de services réutilisables.

Aujourd'hui, le **consommateur recherche un service toujours plus personnalisé, souhaite avoir le choix dans ses options de séjour et exige de plus en plus de souplesse et de flexibilité dans son choix de voyage.** L'objectif est de développer un prototype " orienté utilisateur (customer-centric) " capables de résoudre les problèmes d'interopérabilité et d'accès aux stocks en temps réel, de répondre de manière ultra personnalisée aux requêtes des clients, et d'agrèger, d'organiser et de présenter de manière pertinente toujours plus de contenus informatifs pour enrichir les produits.

Ainsi, pour **agrèger une grande quantité** de contenu et services qualifié et pertinent deux solutions se dessinent : les Web services et/ou le Web sémantique. Cette thèse s'inscrit

dans le cadre de travaux relatifs aux SERVICES WEB SÉMANTIQUES [Jorge 2006, McIlraith *et al.* 2001], dans la perspective de la complémentarité et de la coévolution de deux aspects du Web, l'aspect service et sémantique. Le développement du Web au cours de ces dernières années a fait émerger un énorme graphe de données structurées, sémantiques résultant en partie de l'activité des utilisateurs, les Données liées Ouvertes (**Linked Open Data, LOD**). Afin d'améliorer l'expérience de recherche et de réservation de l'utilisateur, nous nous intéressons particulièrement à l'utilisation de ce graphe afin de faciliter l'accès à l'offre touristique (produits et services) présente sur le Web, et ce de manière utile, informative et enrichissante pour l'utilisateur d'une part et assurant d'une manière flexible et dynamique la composition des SERVICES WEB SÉMANTIQUES [Jorge 2006, McIlraith *et al.* 2001] transactionnels relatifs à la réservation d'autres part.

Cette problématique est notamment étudiée dans plusieurs approches pratiques visant à utiliser des technologies du Web sémantique pour contribuer à la coordination des SERVICES WEB SÉMANTIQUES [Jorge 2006, McIlraith *et al.* 2001]. Une spécificité de ce contexte, assez peu abordé dans la littérature existante, est sans doute le besoin d'inciter les découvertes des SERVICES WEB SÉMANTIQUES [Jorge 2006, McIlraith *et al.* 2001] inattendues et fortuites à travers la richesse des Données Liées Ouvertes (LOD), une richesse qui repose notamment sur la diversité de types de liens et de ressources. C'est sur cette richesse que nous avons fondé le développement d'une approche de découverte très souple. Le premier axe de notre travail concerne la découverte de SERVICES WEB SÉMANTIQUES [Jorge 2006, McIlraith *et al.* 2001] pertinents et inattendus, notamment utiles pour optimiser le plan final en élargissant le champ de recherche par lesquelles une requête sémantique est initialement décrite, en y incluant des concepts similaires permettant ainsi de solliciter les compétences d'autres services.

Dans nos travaux nous montrons comment les liens entre concepts dans le graphe Web peuvent être utilisés ensemble, dans le but de trouver de meilleurs SERVICES WEB SÉMANTIQUES [Jorge 2006, McIlraith *et al.* 2001] pour résoudre un problème donné et expliquons la représentation des services Web en fonction de graphe.

Le deuxième axe de notre travail concerne notre méthode de composition dynamique qui s'inspire des méthodes de planification. Pour répondre à une requête de l'utilisateur exprimée en SPARQL [Franconi & Tessaris 2005, Prud'hommeaux & Seaborne 2008, Arenas & Pérez 2011, Pérez *et al.* 2006](objet d'une traduction ultérieure en vue de l'intégration d'un enrichissement sémantique), on utilise la description des entrées et des sorties sémantiques des services Web pour construire un graphe de dépendance (à partir de la requête réécrite), qui symbolise les compositions (l'interdépendance services) possibles des Process Model où la phase de découverte génère l'ensemble des séquences d'actions équivalentes qui constituent les nœuds du graphe à savoir les clusters de services et les transitions sont les liens sémantiques. Les clusters de services regroupent les Process model iso-fonctionnels. Ensuite, un algorithme incrémental trie ces Process Model équivalents pour en extraire les processus atomiques pertinents à la composition suivant un critère de similarité défini qui permet d'optimiser la taille du graphe initial et raffine à des processus plus élémentaires, puis déduit de ce graphe, la composition final et optimal en faisant appel aux théories de matrices qu'il

---

peut ensuite exécuter sur ce graphe de données touristiques. Dans cette méthode le résultat de la composition sera considéré comme satisfaisant puisqu'il comportera tous les éléments souhaités par l'utilisateur et définis lors de la phase de découverte. Finalement, le plan généré est retourné à l'utilisateur constituant ainsi son package dynamique.

## 7.0.2 Perspectives

Nous identifions plusieurs pistes pour les travaux futurs :

Notre approche pourrait nettement être améliorée en intégrant des informations additionnelles lors de la rédaction de la requête sémantique en rajoutant d'une part les pré-conditions et effets liés à l'exécution des services et d'autres part les contraintes (propriétés non-fonctionnelles) (exprimé en CPSPARQL [Alkhateeb *et al.* 2008]) sur ces services en faisant appel aux paramètres non-fonctionnels tels que qualité de service, prix afin de raffiner le plan de la composition dynamique en termes de qualitatif.

La mise en place des principes des Données Liées a fait émergé une remarquable nouvelle ère qui est en mesure de répandre l'interopérabilité sémantique. L'objectif de [Aoyama & Kojima 2013, Ahmad & Dowaji 2013, Taheriyani *et al.* 2012, Rodriguez-Mier *et al.* 2015, Speiser & Harth 2011, Stadtmüller 2012], en outre, est de capitaliser sur le nuage des Données liées Ouvertes afin de passer à l'ère des Web de Services où les services liés seraient des citoyens des web de données ainsi devenant accessibles directement et facilement aux spécialistes DONNÉES LIÉES [Heath & Bizer 2011, Berners-Lee 2009, Bizer *et al.* 2009, Berners-Lee 2006] à travers des requêtes SPARQL [Franconi & Tessaris 2005, Prud'hommeaux & Seaborne 2008, Arenas & Pérez 2011, Pérez *et al.* 2006]. Leurs descriptions sémantiques en termes d'entrée et de sorties sont liées à la Communauté de données, et de cette manière, chaque service est un consommateur et producteur de données RDF [Carroll & Klyne 2004, Hayes 2004, Lassila & Swick 1999, Manola & Miller 2004, Beckett 2009b] où l'invocation de service contribue à la connaissance de ses consommateurs.

Cette volonté, d'ailleurs, vise à ce que les services puissent ainsi être facilement intégrés dans la composition de service qui généralisent les 'Mashups [Benslimane *et al.* 2008]' pour accueillir les effets secondaires, tout en restant des données RDF. Le but de Services Liées Ouverts (**Linked open Service, LOS**) est non seulement d'appliquer les principes des Données Liées à des Services, une approche déjà proposée dans [Speiser & Harth 2011], mais de proposer une liste d'autres principes spécifiques au service afin de les exposer directement sur les DONNÉES LIÉES [Heath & Bizer 2011, Berners-Lee 2009, Bizer *et al.* 2009, Berners-Lee 2006] (voir Figure 7.1) :

1. Décrire l'entrée et la sortie des services comme des motifs graphiques SPARQL.
2. Communiquer en RDF par négociation de contenu RESTful [Richardson & Ruby 2007].
3. La sortie devrait expliciter sa relation avec l'entrée.
4. Détailler la cartographie afin qu'elle soit ouvertement disponible permettant d'établir des requêtes SPARQL.

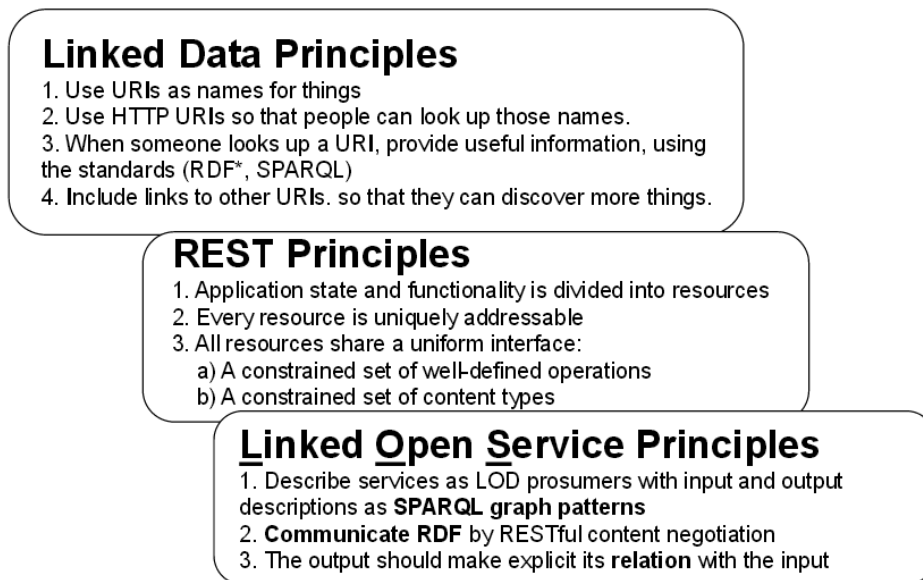


FIGURE 7.1 – Les principes des Services Liés Ouverts



- [Ahmad & Dowaji 2013] Hussien Ahmad et Salah Dowaji. *Linked-OWL : A new approach for dynamic linked data service workflow composition*. In Webology, 2013. :2013
- [Alkhateeb *et al.* 2005] Faisal Alkhateeb, Jean-François Baget et Jérôme Euzenat. *Complex Path Queries for RDF Graphs*. In Poster proc. 4th International Semantic Web Conference (ISWC), Galway (IE), 2005. 121, 122
- [Alkhateeb *et al.* 2008] Faisal Alkhateeb, Jean-François Baget et Jérôme Euzenat. *Constrained regular expressions in SPARQL*. In Hamid Arabnia et Ashu Solo, éditeurs, Proc. international conference on semantic web and web services (SWWS), Las Vegas (NV US), pages 91–99, 2008. 12, 121, 122, 167
- [Alkhateeb 2005] Faisal Alkhateeb. *Graphes à chemins : Graphes RDF/RDFS étiquetés par des expressions algébriques*. PhD thesis, Université Joseph Fourier, 2005. 120
- [Alkhateeb 2008] Faisal Alkhateeb. *Querying RDF(S) with regular expressions*. Thèse d’informatique, Université Joseph Fourier, Grenoble (FR), 2008. 120, 121, 122
- [Antoniou & Harmelen 2008] Grigoris Antoniou et Frank van Harmelen. *A semantic web primer*. The MIT press, 2 édition, 2008. ix, 1, 18, 19, 20, 22, 38, 47, 63
- [Aoyama & Kojima 2013] Mikiyo Aoyama et Hirotaka Kojima. *LISA : Linked Services Architecture Based on the Linked Data and Service Broker*. In 2013 IEEE 20th International Conference on Web Services, Santa Clara, CA, USA, June 28 - July 3, 2013, pages 617–618, 2013. :2013
- [Arenas & Pérez 2011] Marcelo Arenas et Jorge Pérez. *Querying Semantic Web Data with SPARQL*. In Proceedings of the Thirtieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS ’11, pages 305–316, New York, NY, USA, 2011. ACM. 12, 18, 37, 166, 167
- [Ashraf *et al.* 2011] Jamshaid Ashraf, Richard Cyganiak, Seán O’Riain et Maja Hadzic. *Open eBusiness Ontology Usage : Investigating Community Implementation of GoodRelations*. In WWW2011 Workshop on Linked Data on the Web, Hyderabad, India, March 29, 2011, 2011. 126
- [Ataa-Allah *et al.* 2008] Fadoua Ataa-Allah, Siham Boulaknadel, Abderrahim El Qadi et Driss Aboutajdine. *Évaluation de l’analyse sémantique latente et du modèle vectoriel standard appliqués à la langue arabe*. Technique et Science Informatiques, vol. 27, no. 7, pages 851–877, 2008. 66
- [Ayazlar 2014] Reyhan A. Ayazlar. *Dynamic Packaging Applications in Travel Agencies*. Procedia - Social and Behavioral Sciences, vol. 131, pages 326 – 331, 2014. 3rd World Conference on Educational Technology Researches 2013, {WCETR} 2013, 7-9 November 2013, Antalya, Turkey. x, 93
- [Baader *et al.* 2003] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi et Peter F. Patel-Schneider, éditeurs. *The Description Logic Handbook : Theory, Implementation, and Applications*. Cambridge University Press, New York, NY, USA, 2003. 21, 27, 28

## BIBLIOGRAPHIE

---

- [Baget & Mugnier 2002] Jean-François Baget et Marie-Laure Mugnier. *Extensions of Simple Conceptual Graphs : the Complexity of Rules and Constraints*. Journal of artificial intelligence research (JAIR), vol. 16, pages 425–465, 2002. 24, 26
- [Baget 2001] Jean-François Baget. *Représenter des connaissances et raisonner avec des hypergraphes : de la projection À la dérivation sous contraintes*. PhD thesis, Université de Montpellier II, 2001. 19
- [Baget 2005] Jean-François Baget. *RDF Entailment as a Graph Homomorphism*. In Proc. 4th International Semantic Web Conference (ISWC), Galway (IE), pages 82–96, 2005. 24
- [Bahloul 2006] Djida Bahloul. *Une approche hybride de gestion des connaissances basée sur les ontologies : application aux incidents informatiques*. Thèse de doctorat en informatique, Institut National des Sciences Appliquées de Lyon, Décembre 2006. 21, 28
- [Bao et al. 2009] Jie Bao, Elisa Kendall, Deborah McGuinness et Peter Patel-Schneider. *OWL 2 Web Ontology Language : quick reference guide*. Recommendation, W3C, 2009. <http://www.w3.org/TR/owl2-quick-reference/>. 31
- [Barros & Oberle 2012] Alistair Barros et Daniel Oberle. *Handbook of service description : Usdl and its methods*. Springer Publishing Company, Incorporated, 2012. :2012 :HSD :2341380
- [Barta et al. 2009] Robert Barta, Christina Feilmayr, Birgit Pröll, Christoph Grün et Hannes Werthner. *Covering the Semantic Space of Tourism : An Approach Based on Modularized Ontologies*. In Proceedings of the 1st Workshop on Context, Information and Ontologies, CIAO '09, pages 1 :1–1 :8, New York, NY, USA, 2009. ACM. vii, x, 107, 109
- [Beckett 2006] Dave Beckett. *Turtle - Terse RDF Triple Language*. Rapport technique, Hewlett-Packard, Bristol (UK), 2006. 26, 156
- [Beckett 2009a] Dave Beckett. *OWL 2 Web Ontology Language Document Overview*. Recommendation, W3C, Octobre 2009. 31
- [Beckett 2009b] Dave Beckett. *RDF/XML syntax specification (Revised)*. Recommendation, W3C, Février 2009. 18, 22, 31, 34, 57, 58, 99, 100, 120, 167
- [Benatallah et al. 2005] Boualem Benatallah, Mohand-Said Hacid, Alain Leger, Christophe Rey et Farouk Toumani. *On Automating Web Services Discovery*. The VLDB Journal, vol. 14, no. 1, pages 84–96, Mars 2005. :2005 :AWS :1053474.1053479
- [Benslimane et al. 2008] D. Benslimane, S. Dustdar et A. Sheth. *Services Mashups : The New Generation of Web Applications*. IEEE Internet Computing, vol. 12, no. 5, pages 13–15, 2008. 167
- [Berners-Lee et al. 1998] Tim Berners-Lee, Roy Fielding et L. Masinter. *Uniform Resource Identifiers (URI) : Generic Syntax*. RFC 2396, IETF, 1998. <http://www.ietf.org/rfc/rfc2396.txt>. 20
- [Berners-Lee et al. 2001] Tim Berners-Lee, James Hendler et Orin Lassila. *The semantic web*. Scientific American, vol. 284, no. 5, pages 34–43, 2001. 1, 17, 18, 19, 20, 38, 47, 63

- [Berners-Lee 1998] Tim Berners-Lee. Design issue : What the Semantic Web can represent. W3C, 1998. 1, 18, 19, 20, 38, 47, 63
- [Berners-Lee 2006] Tim Berners-Lee. Design issue : Linked data. W3C, 2006. 18, 34, 35, 116, 125, 129, 136, 165, 167
- [Berners-Lee 2009] Tim Berners-Lee. Linked-data design issues. Juin 2009. Published : W3C design issue document <http://www.w3.org/DesignIssue/LinkedData.html>. 18, 34, 35, 116, 125, 129, 136, 165, 167
- [Bernstein & Kiefer 2005] Abraham Bernstein et Christoph Kiefer. *iRDQL - Imprecise RDQL Queries Using Similarity Joins*. In K-CAP 2005 Workshop on : Ontology Management : Searching, Selection, Ranking, and Segmentation, OCT 2005. 68
- [Bilbao *et al.* 2010] Sonia Bilbao, Adelaida Lejarazu et Jesús Herrero. *Dynamic Packaging Semantic Platform for Tourism Intermediaries*. In Ulrike Gretzel, Rob Law et Matthias Fuchs, éditeurs, Information and Communication Technologies in Tourism 2010, pages 617–628. Springer Vienna, 2010. x, 7, 105, 106, 109, 110
- [Bizer *et al.* 2009] Christian Bizer, Tom Heath et Tim Berners-Lee. *Linked Data - The Story So Far*. International Journal on Semantic Web and Information Systems (IJSWIS), 2009. 18, 34, 35, 116, 125, 129, 136, 165, 167
- [Brachman & Levesque 1985] Ronald J. Brachman et Hector J. Levesque, éditeurs. Readings in knowledge representation. Morgan Kaufmann, San Mateo, CA, 1985. :kr
- [Bray *et al.* 2006] Tim Bray, Jean Paoli, C. Michael Sperberg-McQueen, Eve Maler et François Yergeau. *Extensible Markup Language (XML) 1.0 (Fourth Edition)*. Rapport technique, 2006. W3C Rec. 16 August 2006. 17, 23, 49, 57
- [Brickley & Guha 1999] Dan Brickley et Ramanathan Guha. *Resource description framework schema specification*. Proposed recommendation, W3C, 1999. <http://www.w3.org/TR/PR-rdf-schema>. 24
- [Brickley & Guha 2004] Dan Brickley et Ramanathan Guha. *RDF Vocabulary Description Language 1.0 : RDF Schema*. Recommendation, W3C, 2004. <http://www.w3.org/TR/2004/REC-rdf-schema-20040210/>. 24
- [Brogi & Corfini 2008] Antonio Brogi et Sara Corfini. *Ontology- and Behavior-Aware Discovery of Web Service Compositions*. Int. J. Cooperative Inf. Syst., vol. 17, no. 3, pages 319–347, 2008. 85
- [Brogi *et al.* 2008] Antonio Brogi, Sara Corfini et Razvan Popescu. *Semantics-based Composition-oriented Discovery of Web Services*. ACM Trans. Internet Technol., vol. 8, no. 4, pages 19 :1–19 :39, Octobre 2008. 7, 85, 86
- [Bruijn *et al.* 2005] Jos De Bruijn, Christoph Bussler, John Domingue, Dieter Fensel, Martin Hepp, Uwe Keller, Michael Kifer, Birgitta König-Ries, Jacek Kopecky, Rubén Lara, Holger Lausen, Eyal Oren, Axel Polleres, Dumitru Roman, James Scicluna et Michael Stollberg. *Web Service Modeling Ontology (WSMO)*, June 2005. W3C member submission. 8, 49, 53, 58
- [Burstein *et al.* 2004] Mark Burstein, Jerry Hobbs, Ora Lassila, Drew McDermott, Sheila McIlraith, Srinu Narayanan, Massimo Paolucci, Bijan Parsia, Terry Payne, Evren Sirin,

## BIBLIOGRAPHIE

---

- Naveen Srinivasan et Katia Sycara. OWL-S : Semantic Markup for Web Services. 22 novembre 2004 édition, Novembre 2004. Published : Website. :2004
- [Cardoso & Schauer 2006] Jorge Cardoso et Carola Schauer. *A Framework for Assessing Strategies and Technologies for Dynamic Packaging Applications in e-Tourism*. Journal of Information Technology and Tourism, vol. 9, no. 1, 2006. x, 7, 96, 97, 98, 107, 110
- [Cardoso 2006] Jorge Cardoso. *Combining the Semantic Web with Dynamic Packaging Systems*. In Proceedings of the 5th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases, AIKED'06, pages 133–138, Stevens Point, Wisconsin, USA, 2006. World Scientific and Engineering Academy and Society (WSEAS). x, 96, 97, 98, 107, 110
- [Carroll & Klyne 2004] Jeremy Carroll et Graham Klyne. *RDF Concepts and Abstract Syntax*. Recommendation, W3C, Février 2004. 18, 22, 31, 34, 57, 58, 99, 100, 120, 167
- [Cesarini & Thompson 2009] Francesco Cesarini et Simon Thompson. Erlang programming. O'Reilly Media, Inc., 1st édition, 2009. :2009 :EP :1717841
- [Charlet 2000] Jean Charlet. Ingénierie des connaissances : solutions récentes et nouveaux défis. Collection technique et scientifique des télécommunications, ISSN 0221-8579. Eyrolles, 2000. 21, 28
- [Cheatham & Hitzler 2014] Michelle Cheatham et Pascal Hitzler. *The properties of property alignment*. In Pavel Shvaiko, Jérôme Euzenat, Ming Mao, Ernesto Jiménez-Ruiz, Juanzi Li et Axel Ngonga, editeurs, Proceedings of the 9th International Workshop on Ontology Matching collocated with the 13th International Semantic Web Conference (ISWC 2014), Riva del Garda, Trentino, Italy, October 20, 2014., volume 1317 of *CEUR Workshop Proceedings*, pages 13–24. CEUR-WS.org, 2014. 38
- [Chein & Mugnier 2009] Michel Chein et Marie-Laure Mugnier. Graph-based knowledge representation. Springer, Heidelberg (DE), 2009. 22
- [Chinnici *et al.* 2007] R. Chinnici, J.-J. Moreau, A. Ryman et S. Weerawarana. *Web Services Description Language (WSDL) Version 2.0 Part 1 : Core Language*. Rapport technique, W3C Recommendation [en ligne], 2007. Disponible en ligne : <http://www.w3.org/TR/wsdl20> (Page consultée le 28 avril 2005). 105, 109
- [Clark 1999] James Clark. Xsl transformations (xslt) version 1.0. 1999. 31
- [Consortium 2004] World Wide Web Consortium. Xml schema specification. <http://www.w3.org/XML/Schema>, 2004. 49, 51
- [Corby *et al.* 2000] Olivier Corby, Rose Dieng et Cédric Hébert. *A Conceptual Graph Model for W3C Resource Description Framework*. In Proc. International Conference on Conceptual Structures, pages 468–482, 2000. 22
- [Corby *et al.* 2004] Olivier Corby, Rose Dieng-Kuntz et Catherine Faron-Zucker. *Querying the Semantic web with Corese Search Engine*. In Proc. 16th European Conference on Artificial Intelligence (ECAI), sub-conference (PAIS'2004), Valencia (Spain), pages 705–709, 2004. 31, 120

- [Corcho *et al.* 2006] Oscar Corcho, Pinar Alper, Ioannis Kotsiopoulos, Paolo Missier, Sean Bechhofer et Carole Goble. *An overview of S-OGSA : A Reference Semantic Grid Architecture*. Web Semantics : Science, Services and Agents on the World Wide Web, vol. 4, no. 2, pages 102 – 115, 2006. Semantic Grid àThe Convergence of Technologies. x, 7, 102, 103, 104, 109, 110
- [Cresci *et al.* 2014] Stefano Cresci, Andrea D’Errico, Davide Gazzà, Angelica Lo Duca, Andrea Marchetti et Maurizio Tesconi. *Towards a DBpedia of Tourism : the case of Tourpedia*. In Proceedings of the ISWC 2014 Posters & Demonstrations Track a track within the 13th International Semantic Web Conference, ISWC 2014, Riva del Garda, Italy, October 21, 2014., pages 129–132, 2014. 125, 126
- [De Giacomo *et al.* 2000] Giuseppe De Giacomo, Yves Lespérance et Hector J. Levesque. *ConGolog, a concurrent programming language based on the situation calculus*. Artif. Intell., vol. 121, no. 1-2, pages 109–169, 2000. 77
- [Dean & Schreiber 2004] Mike Dean et Guus Schreiber. *OWL Web Ontology Language : reference*. Recommendation, W3C, Février 2004. 27
- [Decker *et al.* 2000] Stefan Decker, Dieter Fensel, Frank van Harmelen, Ian Horrocks, Sergey Melnik, C. A. Michel Klein et Jeen Broekstra. *Knowledge Representation on the Web*. Description Logics, pages 89–97, 2000. 21
- [Diestel 1997] Reinhard Diestel. *Graph Theory*. Numeéro 173 de Graduate Texts in Mathematics. Springer, 1997. 130, 183
- [Doulkeridis *et al.* 2005] Christos Doulkeridis, Nikos Loutas et Michalis Vazirgiannis. *A System Architecture for Context-Aware Service Discovery*, 2005. 70
- [Erl 2005] T. Erl. *Service-Oriented Architecture : Concepts, Technology, and Design*. Rapport technique, Prentice Hall PTR, 2005. 760p. 47, 61, 62, 75, 113, 114
- [Euzenat *et al.* 2007] Jérôme Euzenat, François Scharffe et Antoine Zimmermann. *Expressive alignment language and implementation*. Deliverable D2.2.10, Knowledge Web Network of Excellence, 2007. 38
- [Euzenat 2001] Jérôme Euzenat. *An infrastructure for formally ensuring interoperability in a heterogeneous semantic web*. In Isabel F. Cruz, Stefan Decker, Jérôme Euzenat et Deborah L. McGuinness, editeurs, Proceedings of SWWS’01, The first Semantic Web Working Symposium, Stanford University, California, USA, July 30 - August 1, 2001, pages 345–360, 2001. 48
- [Euzenat 2002] Jérôme Euzenat. *Sémantique des représentations de connaissance*. 2002. Published : Notes de cours. 18, 28
- [Fikes & Nilsson 1971] Richard E. Fikes et Nils J. Nilsson. *STRIPS : A New Approach to the Application of Theorem Proving to Problem Solving*. Rapport technique 43R, AI Center, SRI International, 333 Ravenswood Ave, Menlo Park, CA 94025, May 1971. SRI Project 8259. :1971
- [Franconi & Tessaris 2005] Enrico Franconi et Sergio Tessaris. *The semantics of SPARQL*. Working draft, W3C, Novembre 2005. <http://www.inf.unibz.it/krdw/w3c/sparql/>. 12, 18, 31, 166, 167

## BIBLIOGRAPHIE

---

- [Garcàa *et al.* 2012] Josà Maràa Garcàa, David Ruiz et Antonio Ruiz-Cortàs. *Improving Semantic Web Services Discovery Using SPARQL-Based Repository Filtering*. Web Semantics : Science, Services and Agents on the World Wide Web, vol. 17, no. 0, 2012. x, 125
- [García *et al.* 2012] José María García, David Ruiz et Antonio Ruiz Cortés. *Improving semantic web services discovery using SPARQL-based repository filtering*. J. Web Sem., vol. 17, pages 12–24, 2012. 71, 75, 139
- [Gerevini *et al.* 2004] Alfonso Gerevini, Alessandro Saetti, Ivan Serina et Paolo Toninelli. *LPG-TD : a fully automated planner for PDDL2.2 domains*. In In Proc. of the 14th Int. Conference on Automated Planning and Scheduling (ICAPS-04) International Planning Competition abstracts, 2004. :2004
- [Glimm & Krötzsch 2010] Birte Glimm et Markus Krötzsch. *SPARQL beyond Subgraph Matching*. In Peter F. Patel-Schneider, Yue Pan, Pascal Hitzler, Peter Mika, Lei Zhang 0007, Jeff Z. Pan, Ian Horrocks et Birte Glimm, éditeurs, International Semantic Web Conference (1), volume 6496 of *Lecture Notes in Computer Science*, pages 241–256. Springer, 2010. 32
- [Grimshaw *et al.* 2009] Andrew Grimshaw, Mark Morgan, Duane Merrill, Hiro Kishimoto, Andreas Savva, David Snelling, Chris Smith et Dave Berry. *An Open Grid Services Architecture Primer*. Computer, vol. 42, no. 2, pages 27–34, 2009. 102
- [Grosz *et al.* 2003] Benjamin Grosz, Ian Horrocks, Raphael Volz et Stefan Decker. *Description logic programs : combining logic programs with description logic*. In Proc. 12th WorldWide Web conference, Budapest (HU), pages 48–57, 2003. 21, 27, 28
- [Gruber 1993] Thomas R. Gruber. *A Translation Approach to Portable Ontology Specifications*. Knowledge Acquisition, vol. 5, no. 2, pages 199–221, 1993. 18, 19, 27, 28, 38, 44
- [Guarino *et al.* 1994] Nicola Guarino, Massimiliano Carrara et Pierdaniele Giaretta. *An Ontology of Meta-Level Categories*. In Jon Doyle, Erik Sandewall et Pietro Torasso, éditeurs, KR'94 : Principles of Knowledge Representation and Reasoning, pages 270–280. Morgan Kaufmann, San Francisco, California, 1994. 27
- [Haarslev & Möller 2001] Volker Haarslev et Ralf Möller. *RACER System Description*. In Proceedings of the First International Joint Conference on Automated Reasoning, IJCAR '01, pages 701–706, London, UK, UK, 2001. Springer-Verlag. 28, 69
- [Haas & Brown 2004] H. Haas et A. Brown. *Web Services Glossary*. Rapport technique, W3C Working Group Note [en ligne], 2004. Disponible en ligne : <http://www.w3.org/TR/ws-gloss/> (Page consultée le 28 avril 2005). 1, 2, 3, 47, 49, 61, 63
- [Hacid *et al.* 2009] Mohand-Said Hacid, Freddy Lécué, Alain Léger, Christophe Rey et Farouk Toumani. *Les web services sémantiques, automate et intégration I. Introduction, éléments et scénarios, découverte de services web*. Technique et Science Informatiques, vol. 28, no. 2, pages 229–262, 2009. ix, 4, 9, 10
- [Harris & Shadbolt 2005] Stephen Harris et Nigel Shadbolt. *SPARQL query processing with conventional relational database systems*. In Web Information Systems Engineering (WISE'05 Workshops), pages 235–244, 2005. 31

- [Hatzi *et al.* 2013] Ourania Hatzi, Dimitris Vrakas, Nick Bassiliades, Dimosthenis Anagnostopoulos et Ioannis P. Vlahavas. *The PORSCE II framework : using AI planning for automated Semantic Web service composition*. Knowledge Eng. Review, vol. 28, no. 2, pages 137–156, 2013. x, 82, 83, 87
- [Hayes 2004] Patrick Hayes. *RDF Semantics*. Recommendation, W3C, Février 2004. 18, 22, 31, 34, 57, 58, 99, 100, 120, 167
- [Heath & Bizer 2011] Tom Heath et Christian Bizer. *Linked Data : Evolving the Web into a Global Data Space*. 2011. ix, 18, 34, 35, 39, 116, 125, 129, 136, 165, 167
- [Hendler 2008] J. Hendler. *Web 3.0 : Chicken Farms on the Semantic Web*. Computer, vol. 41, no. 1, pages 106–108, 2008. 1, 18, 19, 20, 38, 47, 63
- [Hepp 2008] Martin Hepp. *GoodRelations : An Ontology for Describing Products and Services Offers on the Web*. In Proceedings of the 16th International Conference on Knowledge Engineering : Practice and Patterns, EKAW '08, pages 329–346, Berlin, Heidelberg, 2008. Springer-Verlag. :2008
- [Hitzler *et al.* 2009] Pascal Hitzler, Markus Krötzsch et Sebastian Rudolph. *Foundations of semantic web technologies*. Chapman & Hall/CRC, 2009. ix, 19, 20
- [Hogan *et al.* 2010] Aidan Hogan, Axel Polleres, Jürgen Umbrich et Antoine Zimmermann. *Some entities are more equal than others : statistical methods to consolidate Linked Data*. 2010. 35
- [Horst 2005] Herman ter Horst. *Completeness, decidability and complexity of entailment for RDF Schema and a semantic extension involving the OWL vocabulary*. Journal of Web Semantics, vol. 3, no. 2, pages 79–115, 2005. 24
- [Jaeger *et al.* 2005] Michael C. Jaeger, Gregor Rojec-Goldmann, Christoph Liebethuth, Gero Mühl et Kurt Geihs. *Ranked Matching for Service Descriptions Using OWL-S*. In Kommunikation in Verteilten Systemen (KiVS), 14. ITG/GI-Fachtagung Kommunikation in Verteilten Systemen (KiVS 2005) Kaiserslautern, 28. Februar - 3. März 2005, pages 91–102, 2005. 70
- [Jentzsch *et al.* 2010] Anja Jentzsch, Robert Isele et Chris Bizer. *Silk - Generating RDF Links while publishing or consuming Linked Data*. In 9th International Semantic Web Conference (ISWC2010), Novembre 2010. ix, 40, 44, 45, 46, 128
- [Jones 2005] Steve Jones. *Toward an Acceptable Definition of Service*. IEEE Software, vol. 22, no. 3, pages 87–93, 2005. 47
- [Jorge 2006] Cardoso Jorge. *Approaches to developing semantic web services*. INTERNATIONAL JOURNAL OF COMPUTER SCIENCE, vol. 1, no. 1, 2006. ix, 1, 2, 7, 8, 48, 49, 59, 62, 63, 66, 105, 109, 134, 135, 136, 143, 150, 156, 157, 158, 166
- [Juric 2010] Matjaz B. Juric. *WSDL and BPEL Extensions for Event Driven Architecture*. Inf. Softw. Technol., vol. 52, no. 10, pages 1023–1043, Octobre 2010. :2010 :WBE :1836308.1836481
- [Kiefer *et al.* 2007a] Christoph Kiefer, Abraham Bernstein, Hong Joo Lee, Mark Klein et Markus Stocker. *Semantic Process Retrieval with iSPARQL*. In Proc. 4th European Semantic Web Conference (ESWC). Springer, 2007. 120

## BIBLIOGRAPHIE

---

- [Kiefer *et al.* 2007b] Christoph Kiefer, Abraham Bernstein et Markus Stocker. *The Fundamentals of iSPARQL : A Virtual Triple Approach for Similarity-based Semantic Web Tasks*. In Proceedings of the 6th International The Semantic Web and 2Nd Asian Conference on Asian Semantic Web Conference, ISWC'07/ASWC'07, pages 295–309, Berlin, Heidelberg, 2007. Springer-Verlag. :2007 :FIV :1785162.1785185
- [Klusch & Gerber 2005] Matthias Klusch et Andreas Gerber. *Semantic web service composition planning with OWLS-XPlan*. In In Proceedings of the 1st Int. AAI Fall Symposium on Agents and the Semantic Web, pages 55–62, 2005. 81, 87
- [Klusch & Kapahnke 2012] Matthias Klusch et Patrick Kapahnke. *The iSeM Matchmaker : A Flexible Approach For Adaptive Hybrid Semantic Service Selection*. Web Semantics : Science, Services and Agents on the World Wide Web, vol. 15, no. 3, 2012. 73, 75
- [Klusch *et al.* 2006] Matthias Klusch, Benedikt Fries et Katia Sycara. *Automated Semantic Web Service Discovery with OWLS-MX*. In Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS '06, pages 915–922, New York, NY, USA, 2006. ACM. :2006 :ASW :1160633.1160796
- [Klusch 2012] Matthias Klusch. *The S3 Contest : Performance Evaluation of Semantic Service Matchmakers*. In M. B. Blake, L. Cabral, B. König-Ries, U. Käster et D. Martin, éditeurs, Semantic Web Services à Advancement through Evaluation. Springer, 2012. ix, x, 67, 68, 88
- [Kochut & Janik 2007] Krys Kochut et Maciej Janik. *SPARQLer : Extended SPARQL for Semantic Association Discovery*. In Proc. 4th European Semantic Web Conference (ESWC'07), pages 145–159, 2007. :2007
- [Kopecký *et al.* 2007] Jacek Kopecký, Tomas Vitvar, Carine Bournez et Joel Farrell. *SAWSDL : Semantic Annotations for WSDL and XML Schema*. IEEE Internet Computing, vol. 11, no. 6, pages 60–67, Novembre 2007. :2007 :SSA :1304062.1304550
- [Kuster *et al.* 2008] Ulrich Kuster, Birgitta König-Ries et Andreas Krug. *OPOSSum - An Online Portal to Collect and Share Semantic Service Descriptions*. In Proceedings of the 5th European Semantic Web Conference (ESWC08), Poster Session, Tenerife, Canary Islands, Spain, June 2008. 72, 75
- [Lardinois 2013] Frederic Lardinois. *Google Improves Knowledge Graph With Comparisons And Filters, Brings Cards Cross-Platform Notifications To Mobile*. [http://techcrunch.com/2013/09/26/google-improves-knowledge-graph-with-comparisons-and-filters-brings-ca\\$ discretionary{-}{-}{-}\\$rds-to-mobile-search-adds-cross-platform-notifications](http://techcrunch.com/2013/09/26/google-improves-knowledge-graph-with-comparisons-and-filters-brings-ca$ discretionary{-}{-}{-}$rds-to-mobile-search-adds-cross-platform-notifications), 2013. ix, 20, 21
- [Lassila & Swick 1999] Ora Lassila et Ralph Swick. *Resource Description Framework (RDF) Model and syntax specification*. Recommendation, W3C, 1999. <http://www.w3.org/TR/REC-rdf-syntax>. 18, 22, 31, 34, 57, 58, 99, 100, 120, 167
- [Lécué *et al.* 2008] Freddy Lécué, Samir Salibi, Philippe Bron et Aurélien Moreau. *Semantic and Syntactic Data Flow in Web Service Composition*. In 2008 IEEE International Conference on Web Services (ICWS 2008), September 23-26, 2008, Beijing, China, pages 211–218, 2008. 82, 87



- [Levesque *et al.* 1998] Hector J. Levesque, Fiora Pirri et Raymond Reiter. *Foundations for the Situation Calculus*. Electron. Trans. Artif. Intell., vol. 2, pages 159–178, 1998. 77
- [Li & Horrocks 2003] Lei Li et Ian Horrocks. *Matchmaking Using an Instance Store : Some Preliminary Results*. In International Workshop on Description Logics (DL'03), 2003. Submitted. 69
- [Li 2013] Jing Li. *A Fast Semantic Web Services Matchmaker for OWL-S Services*. Journal of Networks, vol. 8, no. 5, 2013. 73
- [Maler *et al.* 2004] Eve Maler, Jean Paoli, C. M. Sperberg-McQueen, François Yergeau et Tim Bray. Extensible markup language (XML) 1.0 (third edition). Février 2004. <http://www.w3.org/TR/2004/REC-xml-20040204>. 17, 23, 49, 57, 58
- [Mandell & McIlraith 2003] Daniel J. Mandell et Sheila A. McIlraith. *Automating Web Service Discovery, Customization, and Semantic Translation with a Semantic Discovery Service*. In WWW (Posters), 2003. 70
- [Manola & Miller 2004] Frank Manola et Eric Miller. *RDF Primer*. Recommendation, W3C, 2004. <http://www.w3.org/TR/REC-rdf-syntax/>. 18, 22, 31, 34, 57, 58, 99, 100, 120, 167
- [Masuch *et al.* 2012] N. Masuch, B. Hirsch, M. Burkhardt, A. Healer et S. Albayrak. *SeMa2 : A Hybrid Semantic Service Matching Approach*. In Brian Blake, Liliana Cabral, Birgitta Kainig-Ries, Ulrich Kaister et David Martin, éditeurs, Semantic Web Services, pages 35–47. Springer Berlin Heidelberg, 2012. 73, 75
- [Mcdermott *et al.* 1998a] D. Mcdermott, M. Ghallab, A. Howe, C. Knoblock, A. Ram, M. Veloso, D. Weld et D. Wilkins. *PDDL - The Planning Domain Definition Language*. Rapport technique TR-98-003, Yale Center for Computational Vision and Control,, 1998. 8
- [Mcdermott *et al.* 1998b] D. Mcdermott, M. Ghallab, A. Howe, C. Knoblock, A. Ram, M. Veloso, D. Weld et D. Wilkins. *PDDL - The Planning Domain Definition Language*. Rapport technique TR-98-003, Yale Center for Computational Vision and Control,, 1998. :1998
- [McGuinness & Harmelen 2004] Deborah McGuinness et Frank van Harmelen. *OWL Web Ontology Language Overview*. Recommendation, W3C, 2004. <http://www.w3.org/TR/owl-features/>. 18, 27
- [McIlraith & Son 2002] Sheila A. McIlraith et T. C. Son. *Adapting Golog for Composition of Semantic Web Services*. In Proceedings of the Eighth International Conference on Knowledge Representation and Reasoning (KR2002), pages 482–493, Avril 2002. ix, 77, 78, 87
- [McIlraith *et al.* 2001] S. McIlraith, Cao Tran Son et H. Zeng. *Semantic Web Services*. IEEE Intelligent Systems, pages 46–53, Avril 2001. 1, 2, 7, 8, 48, 49, 62, 63, 66, 105, 109, 134, 135, 136, 143, 150, 156, 157, 158, 166
- [Meditkos & Bassiliades 2010] Georgios Meditskos et Nick Bassiliades. *Structural and Role-Oriented Web Service Discovery with Taxonomies in OWL-S*. IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 2, pages 278–290, 2010. 73

## BIBLIOGRAPHIE

---

- [Mendelzon *et al.* 1997] Alberto Mendelzon, George Mihaila et Tova Milo. *Querying the World Wide Web*. Int. J. on Digital Libraries, vol. 1, no. 1, pages 54–67, 1997. cite-seer.ist.psu.edu/mendelzon97querying.html. 31
- [Merazi & Malki 2015] Afaf Merazi et Mimoun Malki. *SQUIREL : Semantic Querying Interlinked OWL-S traveling Process Models*. International Journal of Information Technology and Computer Science(IJITCS), vol. 7, no. 12, pages 30–39, Novembre 2015. 11, 14, 110, 114, 116
- [Merazi 2011] Afaf Merazi. *JAKARA : Developing Smart Dynamic Packaging Based Linked E-tourism Data*. In Hamid R. Arabnia, Andy Marsh et Ashu M. G. Solo, éditeurs, The 2011 International Conference on Semantic Web and Web Services, SWWS'11, pages 142–148. CSREA Press, 2011. 11, 14, 110, 114, 116
- [Miller 1995] George A. Miller. *WordNet : A Lexical Database for English*. Commun. ACM, vol. 38, no. 11, pages 39–41, Novembre 1995. :1995 :WLD :219717.219748
- [Mitra & Lafon 2007] N. Mitra et Y. Lafon. *SOAP Version 1.2 Part 0 : Primer (Second Edition)*. Rapport technique, W3C Recommendation [en ligne], 2007. Disponible en ligne : <http://www.w3.org/TR/soap12-part0> (Page consultée le 28 avril 2005). 47
- [Mànguez *et al.* 2010] Ivàn Mànguez, Diego Berrueta et Luis Polo. *CRUZAR : An Application of Semantic Matchmaking to e-Tourism*. IGI Global, Hershey, PA, USA, 2010. :2010
- [Moreau *et al.* 2008] Erwan Moreau, François Yvon et Olivier Cappé. *Appariement d'entités nommées coréférentes : combinaisons de mesures de similarité par apprentissage supervisé*. In Actes de la Conférence sur le Traitement Automatique des Langues Naturelles (TALN), pages 488–497, Avignon, France, 2008. 40, 68
- [Motik *et al.* 2009a] Boris Motik, Bernardo Cuenca Grau, Ian Horrocks, Zhe Wu, Achille Fokoue et Carsten Lutz. *OWL 2 Web Ontology Language : profiles*. Recommendation, W3C, 2009. <http://www.w3.org/TR/owl2-profiles/>. 31
- [Motik *et al.* 2009b] Boris Motik, Peter Patel-Schneider et Bernardo Cuenca Grau. *OWL 2 Web Ontology Language : direct semantic*. Recommendation, W3C, 2009. <http://www.w3.org/TR/owl2-direct-semantics/>. 31
- [Munoz *et al.* 2007] Sergio Munoz, Jorge Pérez et Claudio Gutierrez. *Minimal deductive systems for RDF*. In Proc. 4th European Semantic Web Conference(ESWC), Innsbruck (AT), pages 53–67, 2007. <http://www.springerlink.com/content/g8w64n1264874118/>. 24
- [Munoz *et al.* 2009] Sergio Munoz, Jorge Pérez et Claudio Gutierrez. *Simple and efficient minimal RDFS*. Journal of web semantics, vol. 7, no. 3, pages 220–234, 2009. 18, 24, 27, 31
- [Narayanan & McIlraith 2002] Srinu Narayanan et Sheila A. McIlraith. *Simulation, Verification and Automated Composition of Web Services*. In Proceedings of the 11th International Conference on World Wide Web, WWW '02, pages 77–88, New York, NY, USA, 2002. ACM. :2002 :SVA :511446.511457

- [Nau *et al.* 2003] Dana Nau, Tsz-Chiu Au, Okhtay Ilghami, Ugur Kuter, J. William Murdock, Dan Wu et Fusun Yaman. *SHOP2 : An HTN Planning System*. J. Artif. Int. Res., vol. 20, no. 1, pages 379–404, Décembre 2003. :2003 :SHP :1622452.1622465
- [Nickull *et al.* 2005] D. Nickull, M. Connor, C. M. MacKenzie, B. Watson et M. Cowan. *Service Oriented Architecture and Specialized Messaging Patterns*. Rapport technique, Adobe Systems, 2005. White Paper [en ligne] : [http://www.adobe.com/enterprise/pdfs/Services\\_Oriented\\_Architecture\\_from\\_Adobe.pdf](http://www.adobe.com/enterprise/pdfs/Services_Oriented_Architecture_from_Adobe.pdf) (Page consult e le 28 avril 2008). 113
- [Nikolov *et al.* 2008] Andriy Nikolov, Victoria S. Uren, Enrico Motta et Anne N. De Roeck. *Handling Instance Coreferencing in the KnoFuss Architecture*. In Paolo Bouquet, Harry Halpin, Heiko Stoermer et Giovanni Tummarello, editeurs, IRSW, volume 422 of *CEUR Workshop Proceedings*. CEUR-WS.org, D ecembre 2008. 44, 46
- [Norton & Stadtmaller 2011] Barry Norton et Steffen Stadtmaller. *Scalable Discovery of Linked Services*. In Proceedings of the Fourth International Workshop on REsource Discovery, volume 737, Heraklion, Greece, 2011. CEUR-WS. 146
- [Paolucci *et al.* 2002] Massimo Paolucci, Takahiro Kawamura, Terry R. Payne et Katia P. Sycara. *Semantic Matching of Web Services Capabilities*. In Proceedings of the First International Semantic Web Conference on The Semantic Web, ISWC '02, pages 333–347, London, UK, UK, 2002. Springer-Verlag. :2002 :SMW :646996.711287
- [Papazoglou 2003] M. P. Papazoglou. *Service-Oriented Computing : Concepts, Characteristics and Directions*. In 4th International Conference on Web Information Systems Engineering (WISE 03), IEEE Computer Society, December 2003, pages 3–12, Washington, DC, USA, 2003. 47, 61, 63
- [Patel-Schneider *et al.* 2004] Peter Patel-Schneider, Patrick Hayes et Ian Horrocks. *OWL Web ontology language semantics and abstract syntax*. F evrier 2004. 18, 27
- [Paulraj *et al.* 2011] D. Paulraj, S. Swamynathan et M. Madhaiyan. *Process Model Ontology-Based Matchmaking of Semantic Web Services*. Int. J. Cooperative Inf. Syst., vol. 20, no. 4, pages 357–370, 2011. 85, 86, 87
- [Paulraj *et al.* 2012] D. Paulraj, S. Swamynathan et M. Madhaiyan. *Process model-based atomic service discovery and composition of composite semantic web services using web ontology language for services (OWL-S)*. Enterprise IS, vol. 6, no. 4, pages 445–471, 2012. 7, 85, 87
- [Peyton Jones *et al.* 1996] Simon Peyton Jones, Andrew Gordon et Sigbjorn Finne. *Concurrent Haskell*. In Proceedings of the 23rd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '96, pages 295–308, New York, NY, USA, 1996. ACM. :1996 :CH :237721.237794
- [Phocuswright 2015] Phocuswright. *Global Online Travel Overview Third Edition*. <http://www.phocuswright.com/Travel-Research/Market-Overview-Sizing/Global-Online-Travel-Overview-Third-Edition>, 2015. ix, 3, 5, 6
- [Pilioura *et al.* 2003] Thomi Pilioura, Aphrodite Tsalgatidou et Ros Batsakis. *Using WSDL/UDDI and DAML-S in Web Service Discovery*. In In Proceedings of the WWW 2003 Workshop on E-Services and the Semantic Web, 2003. 69

## BIBLIOGRAPHIE

---

- [Pérez *et al.* 2006] Jorge Pérez, Marcelo Arenas et Claudio Gutierrez. *Semantics and Complexity of SPARQL*. In Proceedings of the 5th International Semantic Web Conference, pages 30–43, Athens (GA US), 2006. 12, 18, 166, 167
- [Prud'hommeaux & Seaborne 2008] Eric Prud'hommeaux et Andy Seaborne. *SPARQL Query Language for RDF*. Recommendation, W3C, Janvier 2008. 12, 18, 31, 166, 167
- [Richardson & Ruby 2007] Leonard Richardson et Sam Ruby. Restful web services. O'Reilly, first édition, 2007. :2007 :RWS :1406352
- [Rodriguez-Mier *et al.* 2015] Pablo Rodriguez-Mier, Carlos Pedrinaci, Manuel Lama et Manuel Mucientes. *An Integrated Semantic Web Service Discovery and Composition Framework*. IEEE Transactions on Services Computing, page aceptado, 2015. 167
- [Russell & Norvig 2003] Stuart J. Russell et Peter Norvig. Artificial intelligence : A modern approach. Pearson Education, 2 édition, 2003. :2003 :AIM :773294
- [Sabou & Pan 2005] Marta Sabou et Jeff Z. Pan. *Towards Improving Web Service Repositories through Semantic Web Techniques*. In Proc. of the International Workshop on Semantic Web Enabled Software Engineering (SWESE), 2005. 70
- [Sabou *et al.* 2013] Marta Sabou, Irem Arsal et Adrian M. P. Brasoveanu. *TourMISLOD : A tourism linked data set*. Semantic Web, vol. 4, no. 3, pages 271–276, 2013. :2013
- [Sbodio 2012] MarcoLuca Sbodio. *SPARQLent : A SPARQL Based Intelligent Agent Performing Service Matchmaking*. In Brian Blake, Liliana Cabral, Birgitta Kainig-Ries, Ulrich Kaister et David Martin, editeurs, Semantic Web Services, pages 83–105. Springer Berlin Heidelberg, 2012. 71, 75
- [Schmeing *et al.* 2006] T. Schmeing, Cardoso J. et J. D. Fernandes. *eTourism : The Knowledge-based Packaging Model*. In Workshop on Dynamic and Intelligent Configuration of Tourism Services, University of Twente, The Netherlands, 2006. x, 93
- [Seaborne 2004] Andy Seaborne. *RDQL - A Query Language for RDF*. Member submission, W3C, 2004. 31, 68
- [sia partners 2013] sia partners. *Le package dynamique : la nouvelle tendance de l'e-tourisme - See more at :* <http://transport.sia-partners.com/20130927/le-package-dynamique-la-nouvelle-tendance-de-le-tourisme>, 2013. x, 94
- [Siricharoen 2010] Waralak Vongdoiwang Siricharoen. *Enhancing Semantic Web and Ontologies for e-Tourism*. Int. J. Intell. Inf. Database Syst., vol. 4, no. 4, pages 355–372, Septembre 2010. vii, x, 107, 109
- [Sirin *et al.* 2004] Evren Sirin, Bijan Parsia, Dan Wu, James Hendler et Dana Nau. *HTN Planning for Web Service Composition Using SHOP2*. Web Semant., vol. 1, no. 4, pages 377–396, Octobre 2004. 85
- [Sirin *et al.* 2007] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur et Yarden Katz. *Pellet : A Practical OWL-DL Reasoner*. Web Semant., vol. 5, pages 51–53, Juin 2007. :2007
- [Smith *et al.* 2004] Michael Smith, Chris Welty et Deborah McGuinness. *OWL Web ontology language guide*. Février 2004. 27

- [Speiser & Harth 2011] Sebastian Speiser et Andreas Harth. *Integrating Linked Data and Services with Linked Data Services*. In Proceedings of the 8th Extended Semantic Web Conference on The Semantic Web : Research and Applications - Volume Part I, ESWC'11, pages 170–184, Berlin, Heidelberg, 2011. Springer-Verlag. 167
- [Srinivasan *et al.* 2006] Naveen Srinivasan, Massimo Paolucci et Katia P. Sycara. *Semantic Web Service Discovery in the OWL-S IDE*. In 39th Hawaii International International Conference on Systems Science (HICSS-39 2006), CD-ROM / Abstracts Proceedings, 4-7 January 2006, Kauai, HI, USA. IEEE Computer Society, 2006. 69
- [Stadtmüller 2012] Steffen Stadtmüller. *Composition of Linked Data-Based RESTful Services*. In Philippe Cudré-Mauroux, Jeff Heflin, Evren Sirin, Tania Tudorache, Jérôme Euzenat, Manfred Hauswirth, Josiane Xavier Parreira, Jim Hendler, Guus Schreiber, Abraham Bernstein et Eva Blomqvist, éditeurs, International Semantic Web Conference (2), volume 7650 of *Lecture Notes in Computer Science*, pages 461–464. Springer, 2012. 167
- [Stevens *et al.* 2004] Robert Stevens, Robin Mcentire, Carole Goble, Mark Greenwood, Jun Zhao, Anil Wipat et Peter Li. *myGrid and the drug discovery process*. Drug Discovery Today : BIOSILICO, vol. 2, no. 4, pages 140–148, July 2004. 104
- [Stoilos *et al.* 2007] Giorgos Stoilos, Giorgos Stamou, Jeff Pan, Vassilis Tzouvaras et Ian Horrocks. *Reasoning with Very Expressive Fuzzy Description Logics*. Journal of artificial intelligent research, vol. 30, pages 273–320, 2007. ix, 74
- [Stollberg *et al.* 2005] Michael Stollberg, Uwe Keller et Dieter Fensel. *Partner and Service Discovery for Collaboration Establishment with Semantic Web Services*. In 2005 IEEE International Conference on Web Services (ICWS 2005), 11-15 July 2005, Orlando, FL, USA, pages 473–480, 2005. 70
- [Taheriyani *et al.* 2012] Mohsen Taheriyani, Craig A. Knoblock, Pedro Szekely et José Luis Ambite. *Rapidly Integrating Services into the Linked Data Cloud*. In Proceedings of the 11th International Conference on The Semantic Web - Volume Part I, ISWC'12, pages 559–574, Berlin, Heidelberg, 2012. Springer-Verlag. 167
- [Thottethodi *et al.* 1998] Mithuna Thottethodi, Siddhartha Chatterjee et Alvin R. Lebeck. *Tuning Strassen's Matrix Multiplication for Memory Efficiency*. In In Proceedings of SC98 (CD-ROM, 1998. 152, 158, 186
- [Trastour *et al.* 2001] D. Trastour, C. Bartolini et J. Gonzalez-Castillo. *A semantic web approach to service description for matchmaking of services*. In Proceedings of the International Semantic Web Working Symposium, SWWS, 2001. 66
- [Vázquez-Salceda *et al.* 2010] Javier Vázquez-Salceda, Wamberto Weber Vasconcelos, Julian A. Padget, Frank Dignum, Siobhán Clarke et M. Palau Roig. *ALIVE : an agent-based framework for dynamic and robust service-oriented applications*. In 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010), Toronto, Canada, May 10-14, 2010, Volume 1-3, pages 1637–1638, 2010. 67
- [Verborgh *et al.* 2014] Ruben Verborgh, Miel Vander Sande, Pieter Colpaert, Sam Coppens, Erik Mannens et Rik Van de Walle. *Web-Scale Querying through Linked Data Fragments*. In Proceedings of the 7th Workshop on Linked Data on the Web, Avril 2014. 37

## BIBLIOGRAPHIE

---

- [Walsh 2002] Aaron E. Walsh, editeur. Uddi, soap, and wsdl : The web services specification reference book. Prentice Hall Professional Technical Reference, 2002. :2002 :USW :582736
- [Wu *et al.* 2003] Dan Wu, Bijan Parsia, Evren Sirin, James A. Hendler et Dana S. Nau. *Automating DAML-S Web Services Composition Using SHOP2*. In Dieter Fensel, Katia P. Sycara et John Mylopoulos, editeurs, International Semantic Web Conference, volume 2870 of *Lecture Notes in Computer Science*, pages 195–210. Springer, 2003. ix, 78, 79, 80, 85, 87
- [Zimmermann & Euzenat 2006] Antoine Zimmermann et Jérôme Euzenat. *Three semantics for distributed systems and their relations with alignment composition*. In Proc. 5th conference on International semantic web conference (ISWC), Athens (GA US), pages 16–29, 2006. 38

# 8

## Généralités sur les graphes

### 8.0.1 Généralités sur les graphes

Notre approche aborde le problème de composition automatique en s'appuyant sur la théorie de graphes [Diestel 1997]. Cette dernière tente de visualiser concrètement le flux de contrôle du Process Model *i.e.*, l'intra-dépendance entre les processus atomiques en le modélisant. On en suit la solution par un algorithme approprié. Nous citons quelques exemples de situations qui peuvent être traités avec des graphes :

- Les problèmes d'ordonnancement, qui ont pour but la recherche d'un ordre optimal des tâches pour une réalisation complexe : il s'agit de trouver un ordre de réalisation des travaux, en minimisant le temps total et le coût total ;
- Les emplois du temps et la répartition des salles ;
- Les problèmes d'affectations (organiser des équipes de travail pour qu'elles soient le plus efficaces possibles) ;
- Les problèmes de maintenance (minimiser les stocks de pièces de rechange, ou les coûts dus à l'arrêt des machines) ;
- Les problèmes de compétition et de concurrence ;
- Les problèmes de classification de produits, ou d'individus.

Le premier problème connu d'utilisation d'un graphe pour résoudre un problème est celui des " 7 ponts de Königsberg ", résolu en 1735 par le mathématicien suisse Leonhard Euler (voir à ce sujet le livre page 256). Mais la théorie des graphes a réellement pris son départ pendant la seconde guerre mondiale, plus précisément en Angleterre en 1940, sous le nom d' " Operation Research ". L'État Major allié, qui devait accroître l'efficacité de ses opérations, en confia le travail au physicien Blackett. Il s'agissait de rechercher la meilleure rotation des équipages dans les avions<sup>1</sup>, l'implantation optimale des radars, plus tard l'organisation des convois transatlantiques. Avant d'aborder notre approche, petit rappel sur la théorie de graphes :

**Définition 8.0.1.** *Un graphe  $G = (S, F)$  est donné par : un ensemble  $S$  de sommets et un ensemble  $F \subset S \times S$  de flèches (ou arcs).*

- *Un chemin dans  $G$  est une suite  $s_0, s_1, s_1, s_2, \dots, s_{n-1}, s_n$  Variantes :*
- *On a aussi en général des étiquettes sur les flèches ou sur les sommets.*

---

1. Lors de la bataille d'Angleterre, il fallait gérer des centaines de pilotes venant de nombreux pays, parfois incapables de se comprendre, en assurant un maximum d'avions simultanément en vol, les escadrilles étant impérativement dirigées par des pilotes compatibles.

## CHAPITRE 8. GÉNÉRALITÉS SUR LES GRAPHES

- Graphes non orientés.
- Graphes représentés ( dans le plan par ex.)
- Multigraphes.

Les graphes se rencontrent comme :

1. Modèle de liaisons spatiales : réseaux de transport, électriques, ...
2. Représentations de liens logiques : graphes de dépendance.
3. Modèles de l'évolution temporelle : les sommets sont des états d'un système.
4. Schémas de programmes : les sommets sont les instructions.

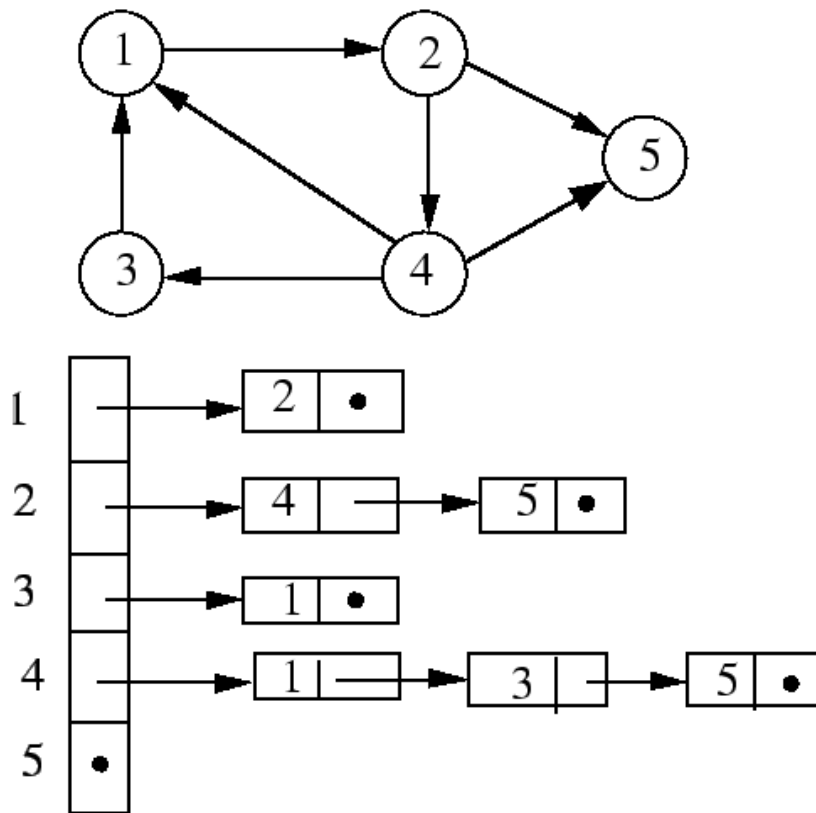


FIGURE 8.1 – Présentation de la matrice d'adjacence et sa liste de successeurs

1. Matrice d'adjacence :  $ij = \begin{cases} 1 & \text{si } (i, j) \in F \\ 0 & \text{sinon} \end{cases}$  Les éléments peuvent être des booléens ou des entiers. On peut représenter la matrice d'Adjacence sous forme d'une liste de successeurs : On donne pour chaque sommet la liste de ses successeurs :  $(s) = \{t \in S \mid (s, t) \in S\}$

Matrice d'incidence : On donne pour chaque flèche son origine et son extrémité :  $I_{if} = \begin{cases} 1 & \text{si } f = (i, \cdot) \\ -1 & \text{si } f = (\cdot, i) \\ 0 & \text{sinon} \end{cases}$



---

Par exemple :  $A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$

À partir de la matrice d'adjacence  $A$  du graphe  $G = (S, F)$ , On peut calculer la matrice d'adjacence  $A^*$  de  $G^*$  qui correspond à la *fermeture transitive* de  $G$ , i.e. le graphe  $G^* = (S, F^*)$  défini par  $(i, j) \in F^*$  ssi On suppose que  $A[i, j]$  vaut 1 s'il existe une chaîne de  $i$  à  $j$  passant uniquement par des sommets inférieurs ou égaux à  $k$ , et 0 dans le cas contraire.

$$A_{ij}^k = \{nb \text{ de chemins de longueur } k \text{ de } i \text{ à } j\}$$

Il existe donc une chaîne de  $i$  à  $j$  passant seulement par des sommets inférieurs ou égaux à  $k$  si et seulement s'il existe une chaîne de  $i$  à  $j$  ne passant que par des sommets inférieurs ou égaux à  $k-1$  ou alors s'il existe une chaîne de  $i$  à  $k$  passant par des sommets inférieurs ou égaux à  $k-1$  ET une chaîne de  $k$  à  $j$  passant par des sommets inférieurs ou égaux à  $k-1$ .  
Formule donnant la matrice d'adjacence  $A^*$  de  $G^*$  (en booléens) :

$$A^* = A + A^2 + \dots + A^{n-1}$$

puisque'un chemin de  $i$  à  $j$  peut être choisi de longueur au plus  $n - 1$ .  $\Rightarrow$  Algorithme naïf en  $\Theta(n^4)$ .

Par exemple : Si  $A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$  Alors  $A^* = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

Calculer la fermeture transitive d'un graphe en pré-traitement révèle souvent un gain important lors de traitement ultérieur. L'algorithme le plus répandu pour le calcul de la fermeture transitive d'un graphe est l'algorithme de Roy et Warshall. On rappelle que l'algorithme de Strassen permet la multiplication de matrices carrées de dimension  $N$  en  $\Theta n^{\log_2 7}$  opérations élémentaires.

$$\begin{aligned}m_1 &= (a_{12} - a_{22})(b_{21} + b_{22}) \\m_2 &= (a_{11} + a_{22})(b_{11} + b_{22}) \\m_3 &= (a_{11} - a_{21})(b_{11} + b_{12}) \\m_4 &= (a_{11} + a_{12})b_{22} \\m_5 &= a_{11}(b_{12} - b_{22}) \\m_6 &= a_{22}(b_{21} - b_{11}) \\m_7 &= (a_{21} + a_{22})b_{11} \\c_{11} &= m_1 + m_2 - m_4 + m_6 \\c_{12} &= m_4 + m_5 \\c_{21} &= m_6 + m_7 \\c_{22} &= m_2 - m_3 + m_5 - m_7\end{aligned}$$

Il existe plusieurs algorithmes de multiplication de matrices réelles : l'algorithme naïf de Warshall permet de construire la fermeture transitive d'un graphe orienté ou non orienté. Il doit son nom à Stephen Warshall (en). La construction de la fermeture transitive par l'algorithme de Warshall a une complexité en  $\Theta(n^3)$ . Cela dit, il peut être intéressant de construire la fermeture transitive d'un graphe une fois pour toutes, ainsi, on peut savoir si les sommets  $i$  et  $j$  appartiennent à la même composante connexe en un temps constant (réservé aux systèmes statiques).. Il existe des algorithmes plus efficaces, comme Strassen [Thottethodi *et al.* 1998] (en  $\Theta(n^{2.807355})$ ) ou Coppersmith et Winograd (en  $\Theta(n^{2.495364})$ ).

**RÉSUMÉ** Cette thèse s'inscrit dans le cadre de travaux relatifs aux Services Web Sémantiques, dans la perspective de la complémentarité et de la coévolution de deux aspects du Web, l'aspect **service** et **sémantique** dans le domaine touristique. Afin d'améliorer l'expérience de recherche et réservation de l'utilisateur, nous nous intéressons particulièrement à l'utilisation des données liées ouvertes afin de faciliter l'accès à l'offre de produits et services touristique présente sur le Web, et ce de manière utile, informative et enrichissante pour l'utilisateur d'une part et transactionnelle assurant d'une manière flexible et dynamique la composition des Services Web Sémantiques relatifs à la réservation de leurs séjours d'autre part. Dans nos travaux nous montrons comment les liens sémantiques entre concepts issus du graphe Web touristique peuvent être utilisés ensemble, dans le but de trouver de meilleurs Services Web Sémantiques afin de résoudre la requête sémantique (exprimé en SPARQL) fournit par l'utilisateur et expliquons notre méthode de composition dynamique à base de graphes de dépendance dont les nœuds du graphe à savoir les clusters de services (les Process model iso-fonctionnels, servant éventuellement dans une re-planification en cas de panne d'un composant dans le workflow généré) et les transitions sont les liens sémantiques représentant l'interdépendance entre services. Ensuite, un algorithme incrémental extrait les processus atomiques pertinents à la composition en triant les Process Model équivalents suivant un critère de similarité défini permettant ainsi d'optimiser la taille du graphe initial en termes de processus atomiques (opération indivisible) et de déduire la composition optimal et final en faisant appel aux théories de matrices constituant ainsi le package dynamique désiré. Les résultats des expérimentations montrent que celles-ci sont capables d'assurer le passage à l'échelle pour des problèmes bien plus complexes.

**MOTS-CLÉS :** Services web sémantiques, web sémantique, Architecture orientée service, découverte orienté composition, processus distribués, domaine touristique, package dynamique, Process model OWL-S, Données Liées ouvertes.

**ABSTRACT** This thesis lies within the scope of work relating to the Semantic web Services, from the point of view of the complementarity and the coevolution of two aspects of the Web, services and semantics in the tourist domain. In order to improve the search and reservation experience of the user, we are particularly interested in the use of the Linked Open Data in order to facilitate the access to the offer of tourist products and services present on the Web, and this in a useful way, informative and enriching for the user on one hand and compromise ensuring a flexible and dynamic composition of the Semantic Services Web relating to the reservation of their stays on the other hand. In our work we show how the semantic links between concepts resulting from the tourist graph Web can be used together for finding the better Semantic Web Services in order to solve the user's semantic request (expressed in SPARQL) and explain our method of dynamic composition based dependence graph where the nodes of the generated graph are related to the cluster of services (the Iso-functional Process models, possibly being useful in replanning phase of the breakdown of a component in the generated workflow) and the transitions are the semantic links representing the services'interdependence. Then, we propose an incremental algorithm that extracts the relevant atomic processes for the composition by sorting the equivalents Process Models according to a definite similarity criterion thus making it possible to optimize the size of the initial graph in terms of the number of their atomic processes (an indivisible operation) and to deduce the final and optimal composition plan using the matrices theories thus representing the desired dynamic package. The results of experiments show that they are able to ensure scalability for more complex problems.

**KEYWORDS :** Semantic web Services, Semantic Web, Service oriented architecture, discovery-oriented composition, distributed process, e-tourism domain, Dynamic Packaging, Process Model OWL-S, Linked Open Data.