



**HAL**  
open science

## Description, découverte et composition de services Web géospatiaux

Meriem Halilali

► **To cite this version:**

Meriem Halilali. Description, découverte et composition de services Web géospatiaux. Autre [cs.OH]. Université de Pau et des Pays de l'Adour, 2021. Français. NNT : 2021PAUU3035 . tel-03575967

**HAL Id: tel-03575967**

**<https://theses.hal.science/tel-03575967>**

Submitted on 15 Feb 2022

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

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



---

Université de Pau et des Pays de l'Adour - École Doctorale des Sciences Exactes et leurs Applications (ED SEA 211)

# Description, découverte et composition de services Web géospatiaux

## THÈSE

présentée et soutenue publiquement le 09 Novembre 2021 pour l'obtention du titre de

### Docteur en informatique

par

**Meriem Sabrina HALILALI épouse MEDAD**

#### Composition du jury

<i>Rapporteurs :</i>	Pr. Zaïa ALIMAZIGHI	Université des Sciences et de la Technologie Houari Boumedienne, Algérie
	Pr. Gilles GESQUIÈRE	Université Lumière Lyon 2, France
<i>Examineurs :</i>	Pr. Ernesto EXPOSITO	Université de Pau et des Pays de l'Adour, France
	Dr. Sana SELLAMI	Université d'Aix-Marseille, France
<i>Directeur :</i>	Pr. Mauro GAIO	Université de Pau et des Pays de l'Adour, France
<i>Co-Directeur :</i>	Dr. Eric GOUARDERES	Université de Pau et des Pays de l'Adour, France

---



# Remerciements

Je voudrais tout d'abord remercier mon directeur de thèse, monsieur Mauro Gaio, professeur à l'université de Pau et des Pays de l'Adour, et mon co-directeur de thèse monsieur Eric Gouarderes, maître de conférences à l'université de Pau et des Pays de l'Adour, pour toute leur aide, leur soutien et leurs conseils judicieux au cours de l'élaboration de cette thèse. Je remercie également, monsieur Florent Devin, docteur en informatique de la société Dataiku, pour son co-encadrement.

Je tiens à remercier monsieur Gilles Gesquière, Professeur à l'université Lumière Lyon 2, et madame Zaia Alimazighi, professeur à l'université des sciences et de la technologie Houari Boumediene (USTHB), pour l'honneur qu'ils m'ont fait en acceptant d'être rapporteurs de cette thèse, et pour le temps qu'ils ont consacré à mon travail.

Je tiens à remercier monsieur Ernesto Exposito, Professeur à l'université de Pau et des Pays de l'Adour, ainsi que madame Sana Sellami, maître de conférences à l'université d'Aix-Marseille, pour avoir accepté d'être dans mon jury de thèse en tant qu'examineurs. Je les remercie pour l'honneur qu'ils me font, pour leur participation scientifique ainsi que pour le temps qu'ils ont consacré à ma recherche.

Un grand merci aux membres du projet Choucas, aux membres du LMAP, aux membres du département d'informatique du collège STEE de l'UPPA, ainsi qu'au personnel de l'école doctorale SEA (ED 211) de l'UPPA pour leur accueil et leur support, tout au long de la réalisation de mes travaux. Je tiens particulièrement à remercier les personnes qui ont partagé mon quotidien durant ces années de thèse. Je pense aux amitiés tissées et à toutes ces personnes qui ont su être présentes en toutes circonstances.

Mes derniers remerciements vont à ma famille et à ma belle-famille, je ne saurais assez remercier mes très chers parents qui me manquent énormément avec les circonstances sanitaires qui nous ont séparés pendant plus de deux ans, sans eux, cette thèse n'aurait pas eu lieu, leur soutien moral et leurs encouragements sont pour moi les piliers fondateurs de ce que je suis et de ce que je fais. Si j'en suis là aujourd'hui, c'est grâce à vous ! Je vous suis infiniment reconnaissante de l'éducation et des valeurs que vous m'avez transmises. Un grand merci à Amine, mon époux qui a tout fait pour m'aider, qui m'a soutenue et supportée dans les moments les plus difficiles pour nous deux, qui n'a jamais cessé de m'encourager et qui a fait preuve de beaucoup de patience et d'amour envers moi. Je tiens à remercier également mon cher frère Chakib et ma petite jumelle Melissa pour leurs encouragements, leur soutien et les nombreuses blagues et fous rires partagés pour me remonter le moral dans les moments difficiles malgré la distance qui nous sépare.



# Résumé

Aujourd'hui, suite à l'avancement des technologies de services Web, un nombre croissant de *SWG* (*Service Web Géospatial*) conçus pour l'interopérabilité des informations géospatiales sur le Web ont vu le jour. L'*OGC* (*Open Geospatial Consortium*) fournit une liste de standards pour assurer un environnement orienté services. Le *WPS* (*Web Processing Service*) est un standard de l'*OGC* qui fournit un modèle de description syntaxique pour les services Web géospatiaux. La découverte de services Web permet d'identifier les services qui correspondent à une fonctionnalité souhaitée par un utilisateur en comparant la requête de l'utilisateur avec les éléments de description de services. En se basant sur une description syntaxique, la découverte et l'utilisation des *SWG* peut être limitée à de petits groupes d'utilisateurs. Généralement, elle se fait soit de façon manuelle sous certaines hypothèses restrictives en connaissant l'identifiant ou l'*URI* (*Uniform Resource Identifier*) des services, soit de manière syntaxique en interrogeant des annuaires ou des portails de services en comparant les mots clés de la requête avec les descriptions syntaxiques des services publiés.

Dans cette thèse, l'objectif est d'améliorer le processus de découverte de *SWG* en faisant face à trois défis de recherche. Le premier consiste à proposer une approche de description sémantique des *SWG* avec une prise en compte de la description des propriétés non fonctionnelles (*QoS* (*Quality of Service*)). Le deuxième consiste à proposer une approche de découverte sémantique des *SWG* en exploitant le modèle de description sémantique. Dans certains cas, il n'existe pas de service atomique qui puisse répondre aux besoins d'un utilisateur, par conséquent, nous sommes amenés à composer plusieurs services afin de satisfaire une requête. Ainsi, le troisième défi de recherche consiste à proposer une approche pour la composition automatique de *SWG*.

Pour faire face à ces différents défis, nous proposons un processus cohérent pour améliorer la découverte de *SWG* qui intègre les trois approches suivantes : une approche de description sémantique de *SWG* pour résoudre les problématiques liées aux descriptions syntaxiques de *SWG*. L'approche est basée sur l'ajout d'annotations sémantiques pour apporter une couche sémantique au modèle de description syntaxique de *SWG*. Ensuite, ces annotations sont exploitées pour améliorer le processus de découverte de *SWG*. L'approche de découverte sémantique de *SWG* est basée sur le calcul d'un score fonctionnel qui exploite une méthode d'appariement sémantique hybride et d'un score de *QoS*. En ce qui concerne le troisième défi, nous proposons une approche de composition automatique de *SWG* qui représente différents éléments de composition de *SWG* dans un programme logique *ASP* (*Answer Set Programming*). En effet, l'approche représente un graphe de dépendance de *SWG*, des règles et des contraintes de composition dans un programme logique, permettant à un solveur de rechercher des compositions de *SWG* en réponse à la requête utilisateur.

**Mots clés :** Description de services Web géospatiaux, Découverte de services Web géospatiaux, Composition de services Web géospatiaux.



# Abstract

Nowadays, with the advancement of web services technologies, a growing number of *SWG* designed for the interoperability of geospatial information on the web have emerged. The OGC provides a list of standards to ensure a service-oriented environment. The WPS is an OGC standard that provides a syntactic description model for geospatial Web services. Web service discovery identifies services that match a user's required functionality by comparing the user's request with the service description elements. Based on a syntactic description, the discovery and the usage of Web Services can be limited to small groups of users. Generally, it is performed either manually under some restrictive hypotheses by knowing the identifier or the URI of the services, or syntactically by querying directories or service portals by comparing the keywords of the request with the syntactic descriptions of the published services.

In this thesis, the objective is to improve the discovery process of *SWG* by dealing with three research challenges. The first one consists in proposing an approach for the semantic description of *SWG* considering the description of non-functional properties (*QoS*). The second one is to propose an approach for the semantic discovery of *SWG* by exploiting the semantic description model. In some cases, there is no atomic service that can meet the needs of a user, therefore we have to compose several services in order to satisfy a request. Thus, the third research challenge consists in proposing an approach for the automatic composition of *SWG*.

To overcome these different challenges, we propose a coherent process to improve the discovery of *SWG* that integrates the following three approaches : a semantic description approach for *SWG* to solve the problems related to the syntactic descriptions of *SWG*. The approach is based on the addition of semantic annotations to provide a semantic layer to the syntactic description model of *SWG*. Then, these annotations are exploited to improve the discovery process of *SWG*. The semantic discovery approach of *SWG* is based on the computation of a functional score that exploits a hybrid semantic matching method and a *QoS* score. Regarding the third challenge, we propose an automatic composition approach of *SWG* that represents different composition elements of *SWG* in a logical program. Indeed, the approach represents a dependency graph of *SWG*, rules and constraints of composition in a logic program, allowing a solver to search for compositions of *SWG* as an answer to the user request.

**Keywords :** Geospatial Web Services Description, Geospatial Web Services Discovery, Geospatial Web Services Composition.

# Table des matières

Table des matières	8
Table des figures	10
Liste des tableaux	11
<b>Introduction générale</b>	<b>17</b>
Contexte . . . . .	17
Problématique . . . . .	18
Objectifs et contributions . . . . .	19
Plan de thèse . . . . .	20
<b>I État de l'art</b>	<b>23</b>
<b>1 Principaux concepts et définitions</b>	<b>25</b>
1.1 Évolution vers les approches orientées services . . . . .	25
1.2 Modèle de service Web . . . . .	26
1.3 Pile de service Web . . . . .	27
1.4 Service Web géospatial . . . . .	30
1.5 Conclusion . . . . .	30
<b>2 Description, découverte et composition des services Web</b>	<b>31</b>
2.1 Standards et style architectural des services Web . . . . .	32
2.1.1 SOAP, WSDL, UDDI . . . . .	32
2.1.2 REST, WADL, hRESTS . . . . .	34
2.2 Standards des services Web géospatiaux . . . . .	35
2.2.1 Open Geospatial Consortium (OGC) et ISO/TC211 . . . . .	35
2.2.2 Standard <i>WPS</i> . . . . .	36
2.3 Approches sémantiques de description de services Web . . . . .	38
2.3.1 Approches basées sur des langages sémantiques . . . . .	38
2.3.2 Approches basées sur des annotations sémantiques . . . . .	40
2.4 Approches sémantiques de découverte des services Web . . . . .	41
2.4.1 Approches logiques . . . . .	42
2.4.2 Approches non logiques . . . . .	43
2.4.3 Approches hybrides (logiques et non logiques) . . . . .	43
2.5 Approches de composition de services Web . . . . .	44
2.5.1 Orchestration et chorégraphie de services Web . . . . .	45
2.5.2 Composition statique et dynamique . . . . .	46
2.5.3 Composition manuelle, semi-automatique et automatique . . . . .	47
2.5.4 Approches automatiques pour la composition de services Web . . . . .	48

2.6	Travaux de description, découverte et composition sémantiques dans le domaine géospatial . . . . .	51
2.7	Synthèse . . . . .	53
2.8	Conclusion . . . . .	58
<b>II</b>	<b>Contributions</b>	<b>59</b>
<b>3</b>	<b>Méthodologie</b>	<b>61</b>
3.1	Contexte et objectifs . . . . .	61
3.2	Méthodologie et principes de la solution . . . . .	62
3.2.1	Vision d'ensemble de l'approche de description sémantique des <i>SWG</i> . . . . .	63
3.2.2	Vision d'ensemble de l'approche de découverte sémantique de <i>SWG</i> . . . . .	64
3.2.3	Vision d'ensemble de l'approche de composition automatique de services <i>SWG</i> . . . . .	64
<b>4</b>	<b>Description sémantique de <i>SWG</i></b>	<b>67</b>
4.1	Terminologie, service et process . . . . .	68
4.2	Méta-modèle pour la description sémantique des services géospatiaux . . . . .	69
4.3	Ontologies de domaine : services et données géospatiales . . . . .	71
4.3.1	Ontologie de services géospatiaux . . . . .	72
4.3.2	Ontologie de données géospatiales . . . . .	73
4.4	Instanciation du méta-modèle par extension du modèle de description <i>WPS</i> 2.0 . . . . .	74
4.4.1	Principe . . . . .	74
4.4.2	Définition des rôles . . . . .	75
4.4.3	Exemple de description . . . . .	76
4.5	Conclusion . . . . .	78
<b>5</b>	<b>Découverte sémantique de <i>SWG</i></b>	<b>81</b>
5.1	Méthode d'appariement sémantique hybride . . . . .	82
5.1.1	Méthode non-logique . . . . .	82
5.1.2	Méthode Logique . . . . .	84
5.1.3	Méthode hybride . . . . .	85
5.2	Processus de découverte en trois étapes . . . . .	86
5.2.1	Calcul de score d'appariement fonctionnel : fonctionnalité . . . . .	87
5.2.2	Calcul de score d'appariement fonctionnel : <i>E/S (Entrées/Sorties)</i> . . . . .	88
5.2.3	Calcul de score non fonctionnel . . . . .	89
5.3	Conclusion . . . . .	90
<b>6</b>	<b>Composition automatique de <i>SWG</i></b>	<b>91</b>
6.1	Principe global de la solution . . . . .	92
6.2	Présentation des éléments de composition . . . . .	94
6.2.1	Graphe de dépendance sémantique . . . . .	94
6.2.2	Requête et contraintes utilisateur . . . . .	97
6.2.3	Règles de composition . . . . .	99
6.2.4	Sélection des compositions de services candidates . . . . .	101
6.3	Conclusion . . . . .	103
<b>7</b>	<b>Implémentation et expérimentations</b>	<b>105</b>
7.1	Mise en œuvre de l'approche de description de <i>SWG</i> . . . . .	106
7.1.1	Architecture fonctionnelle de la description de <i>SWG</i> . . . . .	106
7.1.2	Création d'un corpus de services <i>WPS</i> annotés . . . . .	106

7.2	Mise en oeuvre de l'approche de découverte sémantique de <i>SWG</i> . . . . .	108
7.2.1	Architecture fonctionnelle de la découverte de <i>SWG</i> . . . . .	108
7.2.2	Expérimentations sur l'approche de découverte . . . . .	108
7.3	Mise en oeuvre de l'approche de composition automatique de <i>SWG</i> . . . . .	116
7.3.1	Architecture fonctionnelle de la composition de <i>SWG</i> . . . . .	116
7.3.2	Cas d'étude et expérimentations sur l'approche de composition . . . . .	116
7.4	Conclusion . . . . .	120
	<b>Conclusion générale</b> . . . . .	<b>125</b>
	Bilan . . . . .	125
	Perspectives . . . . .	127
	<b>Annexe A</b> . . . . .	<b>129</b>
	<b>Bibliographie</b> . . . . .	<b>133</b>

## Table des figures

1.1	Avant SOA vs après SOA . . . . .	27
1.2	Interaction entre le client, le fournisseur et l'annuaire des services . . . . .	28
1.3	Pile de service Web et dimensions clés . . . . .	28
2.1	Enveloppe SOAP . . . . .	33
2.2	Modèle conceptuel de WSDL 2.0 . . . . .	33
2.3	Requête/réponse REST . . . . .	34
2.4	Modèle de description de Process WPS [104] . . . . .	36
2.5	Séquence d'interaction WPS Client-WPS Serveur . . . . .	37
2.6	Services Web sémantiques . . . . .	38
2.7	Niveau supérieur de l'ontologie du service <i>OWL-S (Ontology Web Language for Service)</i> . . . . .	39
2.8	Éléments de haut niveau de l'ontologie <i>WSMO (Web Service Modeling Ontology)</i> . . . . .	39
2.9	Processus d'annotation sémantique de services Web . . . . .	40
2.10	Exemple d'annotation SAWSDL d'une opération avec le concept "RequestPurchaseOrder" . . . . .	40
2.11	Exemple d'annotation SA-REST des entrées/sorties. . . . .	41
2.12	Composition de services Web . . . . .	45
2.13	Classification des méthodes de composition de services . . . . .	45
2.14	Orchestration de services . . . . .	46
2.15	Chorégraphie de services . . . . .	46
2.16	Taxonomie de service <i>SLR (Sea-Level Rise)</i> [10] . . . . .	53
3.1	Vue d'ensemble des approches . . . . .	63
4.1	Diagramme de classe UML décrivant le méta-modèle pour la description des services Web géospatiaux. . . . .	69

4.2	Diagramme de classe UML décrivant une abstraction de la description du process WPS [104]. . . . .	74
4.3	Extrait d'une description de Process WPS étendue . . . . .	76
4.4	Extrait de la description des propriétés non fonctionnelles . . . . .	78
5.1	Étapes de découverte . . . . .	83
5.2	Extrait d'une ontologie . . . . .	84
6.1	Vu d'ensemble de l'approche de composition de <i>SWG</i> . . . . .	93
6.2	Aperçu d'un extrait du graphe de dépendance sémantique . . . . .	95
6.3	Exemple d'une solution de composition montrant le score non fonctionnel, le score de dépendance, le score de service de début, et le score global de composition . . . . .	102
6.4	Exemple d'une solution de composition avec les valeurs de scores fonctionnels, non fonctionnels et global . . . . .	103
7.1	Architecture fonctionnelle de la description de <i>SWG</i> . . . . .	106
7.2	Architecture fonctionnelle de la découverte de <i>SWG</i> . . . . .	108
7.3	Extrait de l'ontologie de service pour l'exemple de requête R1 . . . . .	110
7.4	Extrait de l'ontologie de données pour l'exemple de requête R1 . . . . .	111
7.5	Extrait de l'ontologie de service pour l'exemple de requête R2 . . . . .	112
7.6	Extrait de l'ontologie de données pour l'exemple de requête R2 . . . . .	112
7.7	Taux moyen de rappel . . . . .	113
7.8	Taux moyen de précision . . . . .	113
7.9	Temps moyen de réponse . . . . .	113
7.10	Architecture fonctionnelle de la composition de <i>SWG</i> . . . . .	116
7.11	Temps moyen de réponse de l'environnement de composition. . . . .	119
7.12	Modèle conceptuel du paradigme <i>ASP</i> [69] . . . . .	129
7.13	Exemple d'une relation qui est considérée vraie sans condition (un fait). . . . .	130
7.14	Exemple d'une règle avec l'atome <i>a</i> vrai si les atomes <i>b</i> et <i>c</i> sont vrais. . . . .	130
7.15	Exemple d'une contrainte qui écarte tous les modèles avec l'atome <i>a</i> vrai. . . . .	130

## Liste des tableaux

2.1	Opérations WPS . . . . .	37
2.2	Catégories de filtres logiques . . . . .	42
2.3	Études des approches de description, découverte et composition de <i>SWG</i> . . . . .	57
4.1	Propriétés non fonctionnelles génériques . . . . .	71
4.2	Propriétés non fonctionnelles spécifiques au géocodage. . . . .	71
4.3	Rôles étendus pour la balise de métadonnées . . . . .	75
4.4	Exemple d'annotation de la fonctionnalité . . . . .	77
4.5	Exemple d'annotation de l'entrée d'un service . . . . .	77
4.6	Exemple d'annotation de la sortie d'un service . . . . .	77
4.7	Exemple d'annotation de propriétés non fonctionnelles . . . . .	77
5.1	Catégories de filtres logiques . . . . .	85

---

6.1	La planification pour la composition de services Web . . . . .	93
7.1	Requête <i>R1</i> . . . . .	110
7.2	Requête <i>R2</i> . . . . .	110
7.3	Résultats associés à la requête <i>R1</i> . . . . .	110
7.4	Résultats associés à la requête <i>R2</i> . . . . .	111
7.5	Résultats de sélection associés à la requête R1 . . . . .	115
7.6	Résultats de sélection associés à la requête R2 . . . . .	115
7.7	Requête R3 . . . . .	117
7.8	Résultats de sélection associés à la requête R3 . . . . .	118

# Glossaire

- ASP** Answer Set Programming. 5, 11, 20, 21, 50, 64, 91, 92, 94, 96, 97, 98, 99, 103, 105, 116, 117, 118, 120, 126, 129
- BPEL** Business Process Execution Language. 47, 127
- COBRA** Common Object Request Broker Architecture. 25
- CSSL** Composite Service Spécification Language. 49
- DoM** Degree of Match. 42, 43
- E/S** Entrées/Sorties. 9, 42, 43, 44, 52, 63, 64, 69, 70, 72, 73, 74, 77, 79, 81, 82, 84, 87, 88, 94, 106, 107, 108, 109, 110, 111, 113, 114, 117, 120, 125, 126
- EJB** Enterprise Java Beans. 25
- FTP** File Transfer Protocol. 29
- hRESTS** HTML Microformat for Describing RESTful Web Services. 32, 34, 35
- HTML** HyperText Markup Language. 35, 74
- HTN** Hierarchical Task Network. 49
- HTTP** Hypertext Transfer Protocol. 28, 29, 32, 34, 35, 44, 74, 75
- IA** Intelligence Artificielle. 45, 48, 50, 64, 91
- JMS** Java Message Service. 29
- JSON** JavaScript Object Notation. 34
- KVP** Keyword Value Pair. 35
- OGC** Open Geospatial Consortium. 5, 17, 18, 19, 35, 51, 55, 62, 67
- OWL** Ontology Web Language. 38
- OWL-S** Ontology Web Language for Service. 10, 18, 38, 39, 42, 43, 44, 49, 50, 52, 53, 54, 55, 57, 67
- P/E** Préconditions/Effets. 42, 43, 52
- PDDL** Planning Domain Definition Language. 49, 52, 57
- QoS** Quality of Service. 5, 7, 18, 19, 20, 30, 39, 51, 52, 55, 61, 62, 67, 70, 77, 81, 96, 103, 108, 109, 116, 126
- REST** REpresentational State Transfer. 32, 34, 44
- RPC** Remote Procedure Call. 32

- 
- SA-REST** Semantic Annotation of Web Resources. 40
- SAWADL** Semantic Annotation for Web Application Description Language. 41
- SAWSDL** Semantic Annotations Web Services Description Language. 40, 41, 42, 44, 52, 53, 54, 55, 57
- SIG** Système d'information géographique. 17, 19
- SLR** Sea-Level Rise. 10, 52, 53
- SMTP** Simple Mail Transfer Protocol. 29, 32
- SOA** Service Oriented Architecture. 25, 32
- SOAP** Simple Object Access Protocol. 29, 32
- SVM** Support Vector Machine. 44
- SWG** Service Web Géospatial. 5, 7, 9, 10, 11, 17, 18, 19, 20, 21, 32, 35, 51, 52, 55, 57, 58, 61, 62, 63, 64, 67, 68, 69, 70, 71, 72, 73, 74, 76, 78, 79, 81, 82, 84, 86, 88, 90, 91, 92, 93, 94, 96, 98, 100, 102, 103, 105, 106, 107, 108, 109, 111, 113, 114, 115, 116, 117, 119, 120, 121, 125, 126, 128
- UDDI** Universal Description, Discovery, and Integration. 32, 34
- UML** Unified Modeling Language. 49
- URI** Uniform Resource Identifier. 5, 34, 74, 75
- WADL** Web Application Description Language. 32, 34, 35, 41
- WFS** Web Feature Service. 35
- WMS** Web Mapping Service. 35
- WPS** Web Processing Service. 5, 8, 9, 17, 18, 19, 20, 31, 35, 36, 51, 53, 54, 55, 61, 62, 63, 64, 65, 67, 68, 73, 74, 75, 76, 77, 78, 79, 81, 89, 91, 92, 94, 103, 105, 108, 120, 121, 125, 126, 127
- WSDL** Web Service Description Language. 32, 38, 40, 53, 54
- WSML** Web Service Modeling Language. 39, 49, 52
- WSMO** Web Service Modeling Ontology. 10, 38, 39, 41, 52, 57
- XML** eXtensible Markup Language. 29, 32, 34, 35, 77, 107

# **Introduction générale**



# Introduction générale

## Sommaire

---

Contexte . . . . .	17
Problématique . . . . .	18
Objectifs et contributions . . . . .	19
Plan de thèse . . . . .	20

---

## Contexte

Le contexte général dans lequel s'inscrit cette thèse puise sa source dans un effort de recherche à la fois dans le domaine des services Web, du Web sémantique, et dans celui des systèmes d'information géographique *SIG* (*Système d'information géographique*). Ce choix s'explique notamment par le constat réalisé dans le cadre du projet Choucas [111] dans lequel nous sommes impliqués. Ce projet implique plusieurs équipes de recherches interdisciplinaires qui contribuent chacune à la création d'une géo-application pour l'amélioration de l'aide à la localisation des victimes en montagne. Le projet Choucas a pour objectif de proposer des méthodes et des outils permettant le traitement et le partage des données géographiques issues de sources hétérogènes en se basant sur un modèle d'abstraction orienté service. Par exemple, l'une des tâches en cours est de proposer un processus de mise en correspondance automatique des traces, des textes de description de randonnées et des ressources géographiques (répertoires toponymiques et bases de données géographiques)[42]. L'accomplissement de cette tâche implique le développement de fonctionnalités comme l'identification d'entités spatiales et le géocodage qui seront exposées sous forme de services Web.

Dans ce contexte, nous constatons qu'aujourd'hui au vu des dernières avancées dans le domaine de l'ingénierie Web, le partage de données et de traitements géospatiaux est progressivement remplacé par le partage de services géospatiaux et les *SIG* basés sur le Web ont migré vers un paradigme orienté services [98, 30, 21]. La notion de « service Web » désigne une application disponible sur Internet, publiée par un fournisseur de service, et accessible par les clients via des protocoles standards [24].

L'*OGC* fournit une liste de standards pour assurer un environnement orienté services. Les directives des standards pour la description des services, permettent la résolution des problèmes d'hétérogénéité syntaxique et structurelle [143]. Dans le cadre de cette thèse, le Web Processing Service (*WPS*) standard est considéré [104]. Ce standard est conçu pour normaliser le protocole d'exécution de services de traitements de données *SIG* sur le Web et pour normaliser le modèle de description de ces services pour fournir une description qui est destinée au catalogage et la découverte de ces services.

En se basant sur les modèles de description de services des standards actuels de l'*OGC*, la découverte et l'utilisation des *SWG* peuvent être limitées à de petits groupes

d'utilisateurs. La découverte de services est le processus qui permet d'identifier les services qui correspondent à une fonctionnalité souhaitée par un utilisateur. Cette découverte peut se faire de façon manuelle sous certaines hypothèses restrictives (en connaissant l'identifiant ou l'URL de service, ses fonctions et comment interagir correctement avec lui) ou en recherchant les services en formulant des requêtes en utilisant des mots clés. Mais dans la réalité, le nombre important et hétérogène de services disponibles, et la prise en compte des nouveaux défis à relever, comme l'intégration automatique de *SWG* dans des chaînes de services (par exemple, en utilisant un service de géocodage du fournisseur X et un service de planification d'itinéraire du fournisseur Y et leur mise à la disposition des utilisateurs instantanément), nécessite l'amélioration du processus de découverte des services. Ainsi, l'objectif est de découvrir, d'intégrer et de composer de nouveaux *SWG* automatiquement pour répondre à de nouveaux besoins et étendre ainsi les processus métiers.

## Problématique

Bien que les normes de l'*OGC* aient fait des progrès significatifs vers l'interopérabilité syntaxique des services et l'accès aux données, elles ne résolvent pas les problèmes d'hétérogénéité sémantique [88, 124]. Les normes ne fournissent pas de descriptions sémantiques interprétables par les machines. Ce manque de sémantique dans les descriptions des services est considéré comme une limitation primaire pour atteindre l'interopérabilité sémantique, c'est-à-dire l'échange d'informations de manière non ambiguë, avec une compréhension appropriée de la signification des fonctionnalités des services, et des données qu'ils fournissent. Ceci est considéré comme un obstacle pour la découverte sémantique et la composition automatique des *SWG*. Une autre question importante qui doit être relevée au niveau des normes de l'*OGC*, est l'absence de la représentation des propriétés non fonctionnelles, également connues sous le nom d'attributs de *QoS* [55, 50, 22]. Le manque de sémantique et l'absence de propriétés non fonctionnelles pénalisent la sélection et la recommandation de *SWG*, puisque les applications ne peuvent pas raisonner sur plusieurs descriptions de services et aider les utilisateurs à trouver le service qui répond le mieux à leurs exigences.

D'un autre côté, plusieurs études de la sémantique géospatiale ont mis en évidence différentes orientations de recherche basées sur plusieurs défis. Deux grands axes de recherche sur la sémantique géospatiale sont importants dans ce contexte : la modélisation et la découverte sémantique, l'intégration et la composition basées sur la sémantique [63].

Des solutions issues du monde académique mettent au point des modélisations basées sur le Web sémantique pour la description des services Web géospatiaux. Les solutions existantes ont tendance à utiliser les langages de description sémantique de services Web comme par exemple *OWL-S*. Cependant, le principal inconvénient de cette démarche est qu'elle nécessite le renoncement à l'utilisation du modèle de description syntaxique proposé par les standards comme *WPS* et la redéfinition des descriptions existantes dans d'autres langages ce qui risque d'être coûteux et difficile à mettre en oeuvre. D'autres travaux proposent des approches de découverte sémantique mais au meilleur de nos connaissances, ces approches de découverte se basent sur des techniques d'appariement sémantique naïf qui n'exploitent pas convenablement les sémantiques des services. De nombreuses études rapportent que la découverte des services géospatiaux dans des environnements hétérogènes est l'un des défis les plus immédiats et les plus contraignants [148, 27] indiquant que ce processus peut prendre jusqu'à 50% du temps de

l'utilisateur [106, 147]. La phase de découverte joue aussi un grand rôle pour faciliter le processus de composition automatique de services. À l'heure actuelle et à notre connaissance, il n'existe pas de système pour la composition automatique qui cible les services géospatiaux et plus spécifiquement les services *WPS* et qui soit suffisamment souple pour pouvoir composer les services *WPS* existants. De plus, peu de travaux dans le domaine géospatial utilisent dans leurs processus de découverte ou de composition de services les propriétés non fonctionnelles, de surcroît la non considération des attributs spécifiques au domaine, tels que la fréquence de mise-à-jour des bases de données toponymiques, la précision des coordonnées, etc.

Outre cela, la plupart des travaux actuels se concentrent sur une seule des problématiques soulevées ci-dessus. Cela a conduit à l'émergence de plusieurs sous-solutions avec des plus-values et des lacunes mais complètement disjointes les unes des autres et non compatibles avec les standards actuels. Enfin, cela constitue une autre problématique qui consiste à établir des liens entre les défis soulevés et à construire un environnement cohérent de description, découverte et composition de *SWG*.

## Objectifs et contributions

Afin d'améliorer le processus de découverte des services Web géospatiaux, il est indispensable de s'appuyer sur un modèle approprié de description syntaxique et sémantique. La description est une phase impérative qui permet de définir les fonctionnalités d'un service à travers ses propriétés fonctionnelles et de déterminer sa qualité à travers ses propriétés non fonctionnelles. L'utilisation des technologies du Web sémantique peut être considérée comme la prochaine étape dans l'évolution des standards actuels de l'*OGC* qui fournissent exclusivement une description syntaxique. Dans le Web sémantique, la sémantique de l'information est définie dans des descriptions formelles (méta-information), basées sur des structures de connaissances (ontologies) pouvant être traitées par des machines. Le Web sémantique s'appuie sur des théories développées dans plusieurs disciplines existantes telles que la représentation des connaissances et la logique mathématique. Ceci permet de raisonner sur les descriptions pour découvrir les *SWG* à travers plusieurs techniques d'appariement sémantique et de proposer des techniques de composition de *SWG*. La synergie potentielle entre les *SIG* et le Web sémantique constitue une base importante de cette recherche.

Afin de proposer une solution pour améliorer le processus de découverte de *SWG*, nous identifions les principaux objectifs de recherche (OR) suivants :

- OR1- Proposer une approche de description sémantique des propriétés fonctionnelles de *SWG* et prise en compte de la description des propriétés non fonctionnelles (*QoS*).
- OR2- Proposer une approche de découverte sémantique en exploitant la sémantique des informations décrites.
- OR3- Proposer une approche pour la composition automatique de *SWG* en exploitant les approches de description et découverte proposées pour pallier à l'absence de services atomiques qui puissent répondre aux besoins d'un utilisateur.

En d'autres termes, l'objectif général de cette thèse est d'améliorer le processus de découverte de *SWG* en proposant un cadre général pour la description, la découverte et la composition de *SWG* tout en préservant la compatibilité avec les standards actuels [52]. Le présent manuscrit présente les approches suivantes :

- **Approche de description sémantique de *SWG*** : notre première proposition vise à résoudre les problématiques liées aux descriptions syntaxiques de *SWG*. Il s’agit, dans un premier temps, de proposer une approche sémantique pour la description de *SWG* à base d’annotations sémantiques. L’approche est basée sur un méta-modèle qui représente une abstraction du nouveau modèle de description en exprimant les éléments pertinents d’une description des *SWG*. Ce méta-modèle inclut la description syntaxique et sémantique des propriétés fonctionnelles des services géospatiaux. Précisément, il utilise des annotations sémantiques et des ontologies de domaine que nous avons conçues pour apporter une couche sémantique à la couche de description syntaxique proposée par les standards actuels des *SWG*. De plus dans ce méta-modèle nous avons pris en compte la description des propriétés non fonctionnelles pour la représentation de la qualité des *SWG*. Dans un second temps, nous avons fait une proposition d’extension du modèle de description de service *WPS*. Cette dernière fera l’objet de l’instanciation du méta-modèle proposé.
- **Approche de découverte sémantique de *SWG*** : notre seconde proposition vise à résoudre les problématiques liées à la découverte des *SWG* atomiques. Nous proposons une solution pour la découverte de *SWG* sémantiques basée sur une approche d’appariement sémantique hybride. Nous définissons également une stratégie d’aide à la sélection de *SWG* basée sur des scores d’appariement sémantiques fonctionnels et des scores de qualité des *SWG* calculés à partir des valeurs de propriétés non fonctionnelles des *SWG*. L’idée est d’aider l’utilisateur dans la sélection des services les plus appropriés parmi ceux qui sont candidats. Cette approche a été mise en oeuvre et des expérimentations sur un ensemble de services *WPS* ont été réalisées.
- **Approche de composition automatique de *SWG*** : dans certains cas, un seul service ne peut pas répondre aux besoins d’un utilisateur, et par conséquent, nous sommes amenés à composer plusieurs services afin de satisfaire une requête. Notre troisième proposition présente une approche de composition de *SWG* basée sur la planification en utilisant le langage de programmation logique par contraintes *ASP*. Plus précisément, notre approche de composition automatique consiste à représenter les éléments et les critères de composition, dans un programme logique *ASP* et les solutions de compositions possibles seront recherchées par un solveur *ASP* qui consiste au calcul d’un ensemble de modèles stables *ASP*. La représentation de certains éléments nécessite des calculs de scores d’appariement sémantiques. Dans ce contexte nous exploitons la méthode d’appariement sémantique hybride proposée dans le cadre de notre contribution pour la découverte de *SWG*. Nous définissons également une stratégie de sélection de compositions de services candidates en calculant des scores globaux d’appariement sémantique, et des scores globaux de *QoS*. Cette approche a été mise en oeuvre et un cas d’études, ainsi que des premières expérimentations sur un ensemble de services *WPS* ont été réalisés.

## Plan de thèse

Outre la présente introduction qui expose le contexte, la problématique et les objectifs de cette thèse, le reste du mémoire est organisé en deux grandes parties présentées comme suit :

1. La première partie est constituée de l’état de l’art de la thèse. Le Chapitre 1 traite du contexte dans lequel s’inscrit le sujet de thèse à travers un certain nombre de

définitions et concepts de base liés au paradigme orienté service. Le Chapitre 2 expose un état de l'art visant à recenser et analyser plusieurs travaux existants liés à la description, la découverte et la composition des services Web en général et des services Web géospatiaux. Le but est de présenter les principaux modèles et d'identifier les manques et limites des solutions industrielles et académiques, en vue de dégager les enjeux de cette thèse et afin de préparer le lecteur à une bonne compréhension des contributions qui y sont présentées.

2. La seconde partie présente les contributions scientifiques de cette thèse. Elle est introduite par le Chapitre 3 donnant une vision d'ensemble des approches proposées. Le Chapitre 4 présente notre approche pour la description sémantique des *SWG* basée sur un méta-modèle, en couvrant les aspects fonctionnels et non fonctionnels. Le chapitre 5 décrit l'approche d'appariement sémantique proposée et le processus pour la découverte et la sélection de *SWG*. Le chapitre 6 présente notre approche pour la composition automatique de *SWG* basée sur la planification en *ASP*, ainsi que la stratégie de sélection des solutions de compositions candidates. Le chapitre 7 démontre la faisabilité des approches et présente les résultats d'expérimentations des méthodes proposées. Enfin, notre conclusion générale présente un bilan des approches proposées dans cette thèse et se termine par quelques perspectives.



**Première partie**

**État de l'art**



# Chapitre 1

## Principaux concepts et définitions

### Sommaire

---

1.1	Évolution vers les approches orientées services . . . . .	25
1.2	Modèle de service Web . . . . .	26
1.3	Pile de service Web . . . . .	27
1.4	Service Web géospatial . . . . .	30
1.5	Conclusion . . . . .	30

---

### Introduction

L'objectif de ce chapitre est de présenter le contexte de notre travail en définissant les différents concepts. Après la présentation de l'évolution vers les approches orientées services, nous présentons le modèle de service Web, ses définitions, ses caractéristiques et les concepts relatifs. Ensuite, nous introduisons la pile de service Web qui prend en considération différentes dimensions autour des services Web. Enfin nous présentons une définition de service Web géospatial.

### 1.1 Évolution vers les approches orientées services

En génie logiciel, diverses approches ont été proposées pour le développement de logiciels. L'objectif de chaque nouveau paradigme proposé était de pallier aux limitations des paradigmes précédents en réutilisant certains principes et en créant de nouveaux. C'est ainsi que les approches orientées objets [134], orientées composants [133] et dernièrement les approches orientées services ont vu le jour.

Les architectures à base de composants ont été proposées pour dissimuler la complexité d'intégration d'applications autonomes et hétérogènes [141] : *COBRA* (*Common Object Request Broker Architecture*), *EJB* (*Enterprise Java Beans*), et .NET. Cependant, le couplage entre les objets est relativement fort dans ces technologies à base de composants distribués puisque chaque architecture propose sa propre infrastructure. Par conséquent, des difficultés ont été soulevées pour la mise en œuvre de ces architectures dans le cadre d'une infrastructure ouverte telle que Internet. Pour pallier les limites liées à ces architectures, les efforts de conception ont donné lieu au concept d'architecture orientée service *SOA* (*Service Oriented Architecture*) [113]. On trouve plusieurs définitions des *SOA* dans la littérature parmi lesquelles nous retenons les suivantes :

**Définition 1.** Selon [136] le paradigme orienté services est un paradigme interdisciplinaire destiné pour construire des systèmes distribués en intégrant des composants qui sont indépendants de la plateforme technologique, des langages d'implémentation et des systèmes d'exploitation. L'idée de base est de délivrer les fonctionnalités d'une application encapsulées sous forme de services qui seront exploités directement par l'utilisateur final ou utilisés pour former d'autres services.

**Définition 2.** Selon [137] une architecture orientée service instaure un modèle architectural qui a pour objectif d'améliorer l'efficacité, l'agilité et la productivité d'une entreprise en considérant les services comme le principal moyen à travers lequel la logique d'une solution est représentée à l'appui de la réalisation des objectifs stratégiques.

Ces définitions reflètent un concept important des SOA qui est « les services ». Le concept de service est l'élément fondamental sur laquelle reposent les SOA. Un service est un composant logiciel qui possède les propriétés suivantes : d'après [41] et [114], un service est autonome au sens où il ne dépend d'aucun contexte ou état d'autres services ; c'est une boîte noire pour les consommateurs du service ; il répond à une ou plusieurs fonctionnalités ; il a besoin d'une ou plusieurs données (les entrées), et fournit une ou plusieurs données (les sorties). Par conséquent, l'intégration et la gestion des applications à base de services bénéficient d'un couplage faible d'une part, entre le fournisseur et le consommateur de service et d'autre part entre les différents services réunis dans une même application. Un consommateur d'un service n'a pas besoin d'avoir connaissance des détails techniques tels que la technologie d'implémentation ou la plateforme d'exécution d'un service. Ce faible couplage permet le développement d'applications de façon parallèle et indépendante, ce qui entraîne une réduction des coûts de développement et d'intégration des applications [34]. La figure 1.1<sup>1</sup> montre comment chaque fonctionnalité métier requise dépend de l'application qui l'utilise avant l'arrivée des SOA, et comment ces fonctionnalités sont transformées en services métier réutilisables qui peuvent être utilisés indépendamment par plusieurs applications sur différentes plateformes avec le paradigme des SOA.

## 1.2 Modèle de service Web

Il existe de multiples définitions d'un service Web.

**Définition 3.** Selon [24] la notion de service Web désigne un service (une application ou un composant logiciel) disponible sur Internet ou sur intranet, publié par un fournisseur de service, et accessible par les clients via des protocoles standards.

**Définition 4.** Selon [2] les services Web sont la nouvelle vague des applications Web. Ce sont des applications modulaires, autonomes et auto-descriptives qui peuvent être publiées, localisées et invoquées sur le Web.

Ces définitions reflètent le fonctionnement des services Web qui est principalement basé sur trois acteurs majeurs en interaction : le client<sup>2</sup>, le fournisseur et l'annuaire de services. Ces acteurs prennent part à un processus ou interactions dans les SOA, de la manière suivante (voir figure 1.2) :

1. Source : [http://fr.logimethods.ca/enterprise-soa-soe\\_fr.php/index.php](http://fr.logimethods.ca/enterprise-soa-soe_fr.php/index.php)

2. Dans la suite du manuscrit, nous utilisons indifféremment les termes utilisateur et client pour désigner le consommateur des services ou des compositions de services.

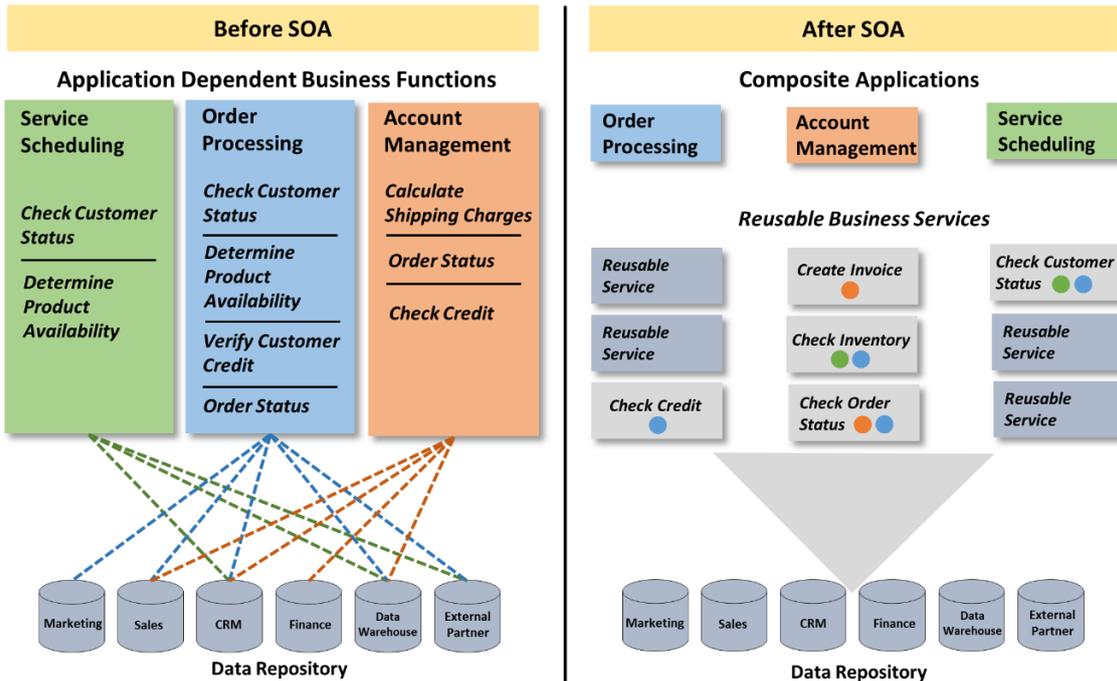


FIGURE 1.1 – Avant SOA vs après SOA

- *Le fournisseur de service* : un fournisseur commence par la création du service Web, puis la publication de la description de service, dans un annuaire de services Web. Cette description contient les fonctionnalités offertes par le service et la manière dont elles peuvent être invoquées. En particulier, les paramètres d'entrée requis pour le fonctionnement de chaque opération de service ainsi que les paramètres de sortie que produit l'opération de service suite à son exécution.
- *L'annuaire de service* : un annuaire de services est un registre des services disponibles. Il reçoit et enregistre les descriptions de services publiées par les fournisseurs. Il reçoit et répond aux recherches de services émises par les clients.
- *Le client* : un client peut découvrir dans l'annuaire, les services répondant à ses besoins. Après qu'une liste de description de services disponibles est retournée par l'annuaire, le client choisit dans la liste le service le plus adéquat et établit une connexion avec le fournisseur de service afin d'invoquer les opérations de service souhaitées. Notons que c'est bien les opérations de service qui sont invoquées et non pas les services. La notion d'invocation de service est un abus de langage.

### 1.3 Pile de service Web

Le W3C a proposé une pile de service Web [144], un type d'architecture qui vient compléter et étendre l'architecture de base ou de référence qui permet de décrire les interactions et relations entre les trois acteurs principaux comme celle représentée dans la figure 1.2. Cette pile de service prend en considération des processus plus complexes comme celui de la composition. Elle est constituée de plusieurs couches superposées, d'où son nom. Différentes représentations et extensions de cette pile ont été proposées dans la littérature [32, 151, 64]. Nous présentons dans la figure 1.3 une vision globale

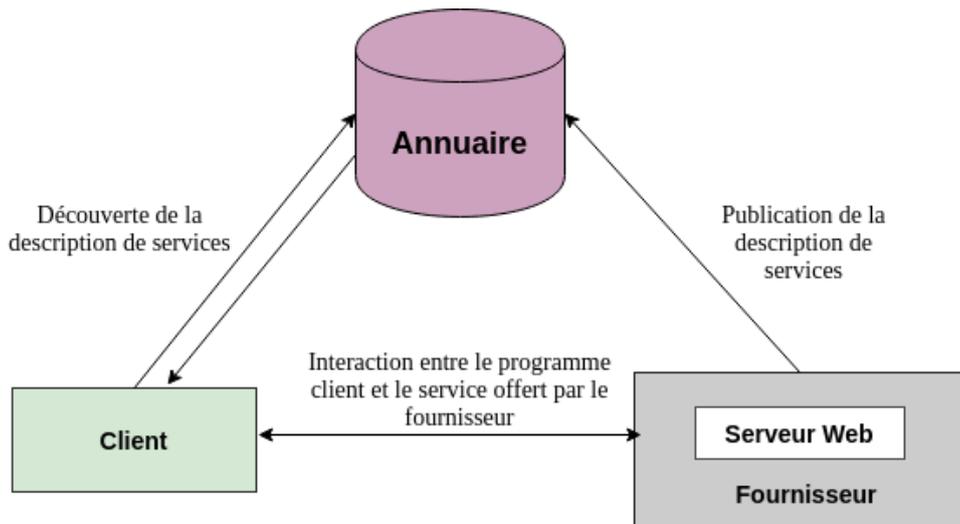


FIGURE 1.2 – Interaction entre le client, le fournisseur et l’annuaire des services

de ces couches et des dimensions clés qui constituent cette pile. Chaque couche de cette architecture s’appuie sur des langages, des protocoles et des modèles.

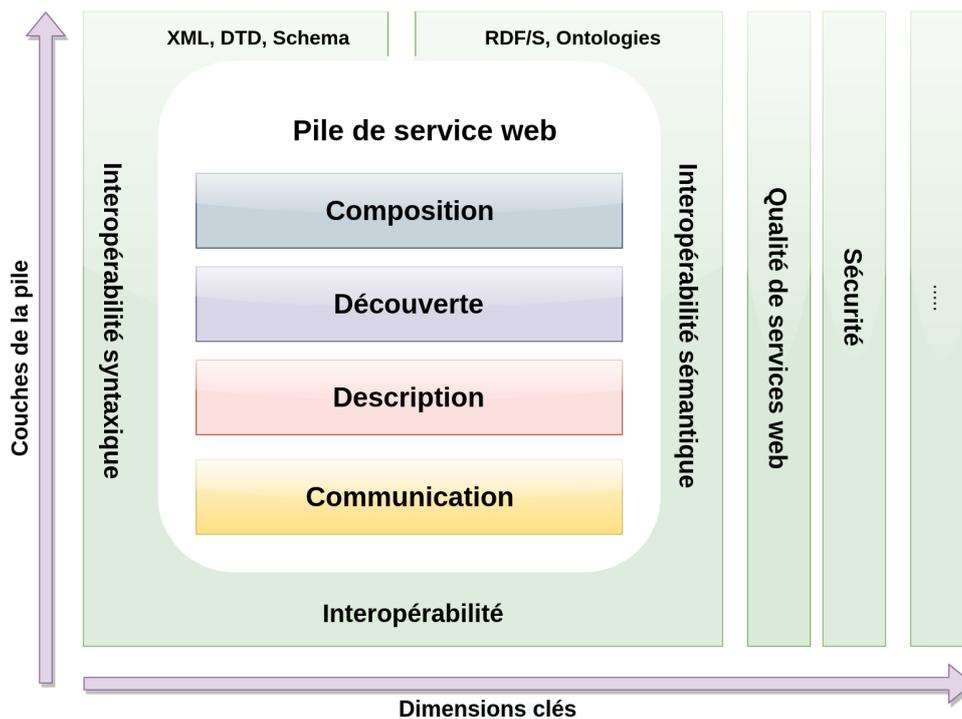


FIGURE 1.3 – Pile de service Web et dimensions clés

- **Couches de la pile** : les quatre principales couches sont la communication, la description, la découverte et la composition.
  - **Communication** : cette couche regroupe les protocoles réseau, et les protocoles d’échanges ou de messagerie, où les communications sous-jacentes sont établies . Un ensemble de différents protocoles réseau permet de réaliser l’accessibilité aux services Web et l’échange de messages. La large adoption du protocole *HTTP (Hypertext Transfer Protocol)* en fait le premier choix de protocole réseau standard pour les services Web disponibles sur Internet. D’autres

protocoles réseau pourraient également être pris en charge, tels que le *SMTP* (*Simple Mail Transfer Protocol*), *FTP* (*File Transfer Protocol*) ou *JMS* (*Java Message Service*). Le modèle de messagerie fonctionne avec une grande variété de protocoles de réseau. Par exemple, le protocole *SOAP* (*Simple Object Access Protocol*), qui est adopté comme un standard peut être combiné avec le protocole *HTTP*.

- **Description** : la couche de description (ou de représentation) sert à décrire les services Web. Elle spécifie leurs fonctionnalités, leurs opérations, leurs types de données d'entrées/sorties et les informations qui les lient en utilisant une interface de service. La découverte des services Web s'appuiera sur la description de ces derniers pour rechercher les services Web appropriés.
- **Découverte** : la couche de découverte sert à rechercher des services Web. Les fournisseurs de services peuvent stocker les descriptions de services dans un registre de services. Les demandeurs de services peuvent interroger le registre des services et rechercher les services sur la base des descriptions de services stockées.
- **Composition** : la couche de composition prend en charge des interactions plus complexes entre les services Web, ce qui supporte l'interopérabilité des services Web. Elle repose sur les fonctionnalités d'interaction de base fournies par les technologies des couches inférieures de la pile de service Web. Par exemple, elle a besoin de la découverte des services Web pour rechercher les services Web sur la base de leurs descriptions. Les services sont utilisés pour construire une chaîne de services, qui consiste en une séquence de services Web coordonnés pour développer des applications logicielles ou d'autres services plus riches en termes de fonctionnalités métiers.
- **Dimensions clés** : il existe plusieurs dimensions relatives au service Web, nous citons principalement l'interopérabilité, la qualité de services Web et la sécurité.
  - **Interopérabilité** : cette dimension fait référence à la mesure dans laquelle les services Web coopéreraient pour atteindre un objectif commun. Les services Web sont conçus pour regrouper des applications provenant d'environnements distribués et hétérogènes et assurer l'interopérabilité entre eux [128]. L'interopérabilité peut se produire à deux niveaux : syntaxique et sémantique. L'interopérabilité syntaxique concerne les caractéristiques syntaxiques des services Web. Ces caractéristiques syntaxiques comprennent, par exemple, le nombre de paramètres et les types de données de ces paramètres. Le langage *XML* (*eXtensible Markup Language*) par exemple, permet de faciliter l'interopérabilité syntaxique en codant les informations syntaxiques dans des documents *XML* [151]. L'interopérabilité sémantique est la question la plus difficile pour parvenir à une interopérabilité véritablement transparente. Elle traite des propriétés sémantiques des services Web. Parmi les exemples de caractéristiques sémantiques figurent le domaine d'intérêt d'un service Web et la fonctionnalité fournie par une opération. Le Web sémantique gagne du terrain en tant que solution

pour doter les services Web de sémantique [151]. Par exemple les ontologies sont utilisées pour des spécifications formelles et explicites de conceptualisations partagées [33]. La "conceptualisation" se réfère à une abstraction d'un domaine qui identifie les concepts pertinents dans ce domaine.

- **Qualité de services Web (QoS) :** la prolifération des services Web devrait introduire une concurrence entre un grand nombre de services Web qui offrent des fonctionnalités similaires. Le concept de *QoS* est considéré comme un élément clé pour distinguer les services Web concurrents [140]. La qualité de service englobe différents paramètres de qualité qui caractérisent le comportement d'un service Web dans la fourniture de ses fonctionnalités. Ces paramètres peuvent inclure plusieurs aspects de qualité des services Web afin de répondre aux exigences de différents domaines d'application.
- **Sécurité :** la sécurité est une question importante pour le déploiement des services Web. Les services Web permettent une interopérabilité au risque de laisser des intrus extérieurs s'attaquer aux applications et bases de données internes puisqu'ils ouvrent le réseau pour donner accès à ces ressources à des utilisateurs extérieurs [46]. La sécurité des services Web doit porter sur des aspects comme l'authentification, l'autorisation, la confidentialité et l'intégrité [151].

## 1.4 Service Web géospatial

**Définition 5.** *Selon [28] les services Web géospatiaux se distinguent des services Web généraux qui assurent l'interopérabilité entre les systèmes dans différents domaines, par le type de traitement (par exemple, le géocodage), ou la présence de données géographiques en entrée (par exemple, une bounding box), ou en sortie (par exemple, le plan d'une ville), ou une combinaison de ces éléments.*

**Définition 6.** *Selon [92] un service Web géospatial peut être considéré comme une application Web modulaire qui fournit des services sur des données, des informations ou des connaissances géospatiales.*

De plus, les organisations géospatiales proposent leurs propres formats et standards intégrant par exemple des modèles de descriptions de services qui leurs sont propres.

## 1.5 Conclusion

Dans cette première partie de l'état de l'art, nous avons présenté le contexte dans lequel s'inscrit le sujet de la thèse en introduisant les définitions, concepts de base du paradigme orienté service. Premièrement, nous avons présenté l'évolution vers les approches orientées services, et les définitions qui leurs sont liées. Ensuite, nous avons introduit le modèle de service et nous l'avons détaillé à travers une architecture qui décrit différentes couches et dimension clés des services Web. Enfin, nous avons présenté une définition des services Web géospatiaux se trouvant au cœur de nos travaux de thèse.

## Chapitre 2

# Description, découverte et composition des services Web

### Sommaire

---

2.1	Standards et style architectural des services Web . . . . .	32
2.1.1	SOAP, WSDL, UDDI . . . . .	32
2.1.2	REST, WADL, hRESTS . . . . .	34
2.2	Standards des services Web géospatiaux . . . . .	35
2.2.1	Open Geospatial Consortium (OGC) et ISO/TC211 . . . . .	35
2.2.2	Standard <i>WPS</i> . . . . .	36
2.3	Approches sémantiques de description de services Web . . . . .	38
2.3.1	Approches basées sur des langages sémantiques . . . . .	38
2.3.2	Approches basées sur des annotations sémantiques . . . . .	40
2.4	Approches sémantiques de découverte des services Web . . . . .	41
2.4.1	Approches logiques . . . . .	42
2.4.2	Approches non logiques . . . . .	43
2.4.3	Approches hybrides (logiques et non logiques) . . . . .	43
2.5	Approches de composition de services Web . . . . .	44
2.5.1	Orchestration et chorégraphie de services Web . . . . .	45
2.5.2	Composition statique et dynamique . . . . .	46
2.5.3	Composition manuelle, semi-automatique et automatique . . . . .	47
2.5.4	Approches automatiques pour la composition de services Web . . . . .	48
2.6	Travaux de description, découverte et composition sémantiques dans le domaine géospatial . . . . .	51
2.7	Synthèse . . . . .	53
2.8	Conclusion . . . . .	58

---

### Introduction

L'objectif de ce chapitre est de recenser et d'analyser plusieurs travaux liés à la description, la découverte et la composition de services Web généraux et de services Web géospatiaux.

Dans la Section 2.1, nous présentons les standards et style architectural des services Web, suivi par la section 2.2 qui présente les standards des services Web géospatiaux. Dans la section 2.3 nous nous intéressons aux approches de description sémantique de services Web. Ensuite, nous présentons les approches de découverte sémantique de services

Web dans la section 2.4. La section 2.5 aborde les approches de composition des services Web. Dans la section 2.6 nous nous penchons principalement sur les travaux récents qui proposent des solutions aux problématiques de description, découverte et composition de *SWG*. Dans la section 2.7 nous proposons une synthèse des différentes approches et travaux présentés dans ce chapitre. Enfin la section 2.8 présente une conclusion pour le chapitre.

## 2.1 Standards et style architectural des services Web

Les efforts de recherche et de développement récents autour des services Web ont conduit à un certain nombre de spécifications qui définissent aujourd'hui les architectures de référence des services Web. Dans cette sous-section nous allons décrire les principales technologies des services Web actuels incluant des normes (SOAP, WSDL, UDDI) et des styles d'architecture tels que *REST* (*REpresentational State Transfer*) ainsi que les modèles de description des services *REST*, notamment *WADL* (*Web Application Description Language*) et *hRESTS* (*HTML Microformat for Describing RESTful Web Services*).

### 2.1.1 SOAP, WSDL, UDDI

Pour garantir l'interopérabilité dans une architecture *SOA*, des propositions de standards ont été élaborées. Nous citons, notamment les standards émergents *SOAP*, *WSDL* (*Web Service Description Language*) et *UDDI* (*Universal Description, Discovery, and Integration*).

- **SOAP** : conçu en 1998, *SOAP* est une spécification de protocole pour la mise en œuvre de services Web [18]. Il repose sur la transmission de messages *SOAP* pour la messagerie et les appels de procédure à distance *RPC* (*Remote Procedure Call*), et exploite les protocoles de transport existants tels que *HTTP* et *SMTP* pour la communication au lieu de définir son propre protocole. Les messages *SOAP* échangés sont encapsulés dans des enveloppes *SOAP*, structurées dans des documents *XML*. L'enveloppe *SOAP* identifie le document *XML* comme un message *SOAP*, et contient un en-tête optionnel et un corps obligatoire, comme le montre la figure 2.1. L'en-tête contient des informations de métadonnées facultatives (par exemple, authentification, paramètres de livraison, détails de routage), tandis que le corps contient les données destinées à être traitées par le destinataire d'un message.
- **WSDL** : les services *SOAP* sont décrits en utilisant le standard WSDL (*Web Service Description Language*) [24, 23]. Le *WSDL* définit la description de l'interface du service d'une manière standard, indépendante de l'implémentation. Il fournit des détails sur la manière de localiser le service et décrit l'ensemble des opérations prises en charge. Il permet de séparer la description de la fonctionnalité abstraite d'un service des détails concrets sur la manière d'accéder à la fonctionnalité. La partie abstraite décrit principalement le service en termes de messages envoyés et reçus. Elle inclut des éléments "operation". Les opérations sont groupées dans un élément "interface" sans préciser le protocole d'échange de message. La partie concrète d'une description *WSDL* d'un service Web est la liaison d'une définition d'interface à un protocole concret. La figure 2.2 illustre la structure d'une description WSDL 2.0.

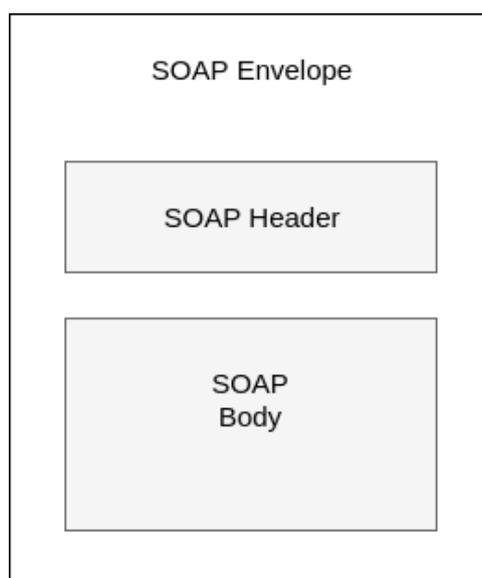


FIGURE 2.1 – Enveloppe SOAP

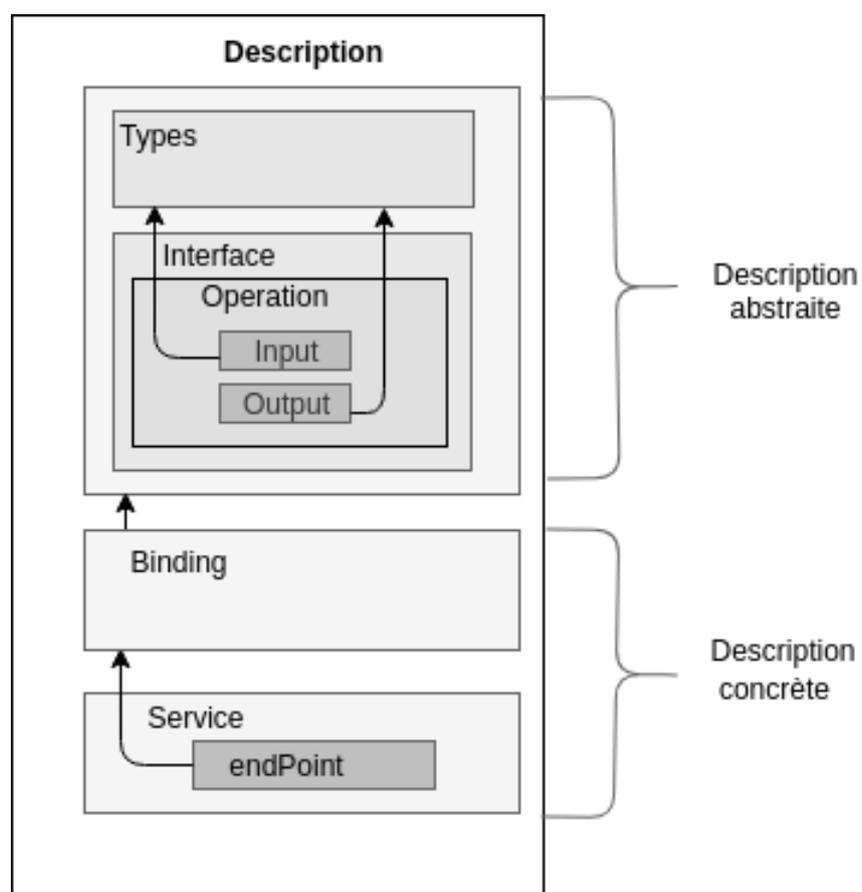


FIGURE 2.2 – Modèle conceptuel de WSDL 2.0

- **UDDI** : pour soutenir la découverte des services Web existants, les registres *UDDI* permettent de publier des interfaces de services Web, les rendant ainsi disponibles pour des clients externes [4]. En offrant aux utilisateurs un moyen unifié et systématique de trouver des fournisseurs de services grâce à un registre centralisé des services, les registres *UDDI* peuvent être interrogés pour localiser des services Web en fonction de leurs caractéristiques. Tout comme un annuaire téléphonique, les registres *UDDI* encodent les informations sur les services Web sous trois catégories : (i) les pages blanches contiennent le nom et les coordonnées de contact, (ii) les pages jaunes contiennent une catégorisation basée sur les types d'entreprises et de services, et (iii) les pages vertes contiennent des détails techniques sur le service.

### 2.1.2 REST, WADL, hRESTS

Roy Fielding, dans sa thèse "*Architectural Styles and the Design of Network-based Software Architectures*" [40] a proposé un style d'architecture nommée *REST*. Des descriptions de services Web *REST* ont été proposées. Nous citons, notamment *WADL* et *hRESTS*.

- **REST** : un style architectural qui s'articule autour d'une notion d'abstraction d'information qui est une ressource. Toute information qui peut être nommée peut être une ressource : une image, un document, un objet non virtuel (par exemple une personne), une collection d'autres ressources, etc. Les ressources sont identifiées par les *URI* et manipulées par le biais d'une interface uniforme. *REST* s'appuie sur *HTTP* comme protocole de transport, permettant aux clients de manipuler les ressources en utilisant les verbes *HTTP* standards. Les verbes *GET*, *POST*, *PUT* et *DELETE* sont utilisés respectivement pour récupérer, créer, modifier et supprimer une ressource (*URI member*) ou plusieurs (*URI collection*) ressources (voir figure 2.3). Les Services Web *REST* sont utilisés pour développer des architectures orientées ressources. Les applications qui mettent en oeuvre les architectures orientées ressources en respectant tous les critères de ce style architectural, sont nommées *RESTful*.

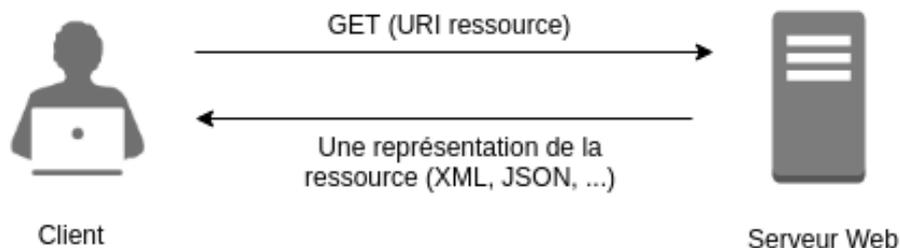


FIGURE 2.3 – Requête/réponse REST

Aucun format d'échange entre le client et le serveur n'est imposé. Les données peuvent être représentées en *XML*, *JSON* (*JavaScript Object Notation*), ou en d'autres formats. Cependant, le client a la possibilité d'indiquer le format dans lequel il souhaite communiquer, soit en paramètre supplémentaire dans l'URL ou en spécifiant le type de contenu échangé dans l'attribut *content-type* dans l'entête *HTTP*.

- **WADL** : pour permettre la description des services REST, W3C définit *WADL* qui est basé sur le langage XML comme une alternative plus simple au WSDL (conçu initialement pour les services SOAP) [51]. Un document WADL décrit l'ensemble des ressources et donne des détails sur la méthode d'accès (verbes HTTP).
- **hRESTS** : *hRESTS* [76] est un microformat basé HTML. C'est un mécanisme qui enrichit les documentations de services existants avec des annotations qui transportent des informations syntaxiques, dans le but de les réutiliser comme des descriptions formelles de ressources. Ceci permet de rendre la documentation *HTML (HyperText Markup Language)* d'une API Web exploitable par la machine.

## 2.2 Standards des services Web géospatiaux

### 2.2.1 Open Geospatial Consortium (OGC) et ISO/TC211

Dans le contexte des services Web et de la normalisation, l'*OGC* et l'*ISO/TC211* [7] sont les principaux acteurs de la normalisation des *SWG* et fixent ainsi les règles de publication, de description et d'invocation de *SWG*.

L'*OGC (Open Geospatial Consortium)* est une organisation internationale fondée en 1994 pour répondre aux problèmes d'interopérabilité des systèmes d'information géographique (SIG). Elle est également connue sous ses anciens noms : Open GIS Project et Open GIS Consortium. L'*OGC* se concentre sur l'élaboration de normes de mise en œuvre en adaptant ou en étendant les définitions et spécifications communes de services Web.

L'*ISO/TC211* se concentre sur le développement de normes théoriques et abstraites. Les normes *ISO/TC211* spécifient des méthodes, des outils et des services pour l'acquisition, le traitement, l'analyse, l'accès, la présentation et le transfert d'informations spatiales entre différents utilisateurs, systèmes et lieux [7].

Les standards de l'*OGC* et de l'*ISO/TC211* sont les plus importants, les plus largement adoptés et implémentés par les universités, les entreprises et les agences gouvernementales. L'*OGC* s'est consacré à la normalisation des interfaces et des modèles de description services Web géospatiaux pour permettre l'interopérabilité. L'*OWS Context (The OGC Web Services Context Document)* [19] a été créé pour permettre à un ensemble de ressources d'informations d'être transmis entre les applications principalement sous la forme d'un ensemble de services. Dans ce contexte, il y a eu une prise en considération des séries de spécifications qui ont été développées. Nous citons parmi elles les spécifications *WFS (Web Feature Service)*, *WMS (Web Mapping Service)* et *WPS* développées respectivement pour les services de données spatiales, les services de visualisation d'informations spatiales et pour les services de traitement d'informations spatiales. Les services Web de l'*OGC* partagent plusieurs caractéristiques communes comme l'adoption du protocole *HTTP* pour la communication (*HTTP/POST + XML* et *HTTP/GET + KVP (Keyword Value Pair)*) et différents pour d'autres, comme les modèles de description des services. Par exemple, *WPS* définit un modèle de processus (*process*) conçu pour fournir une description syntaxiquement interopérable des fonctions de traitement géospatial. Le modèle de *Process WPS* a été conçu de sorte qu'il peut être utilisé dans des catalogues pour la découverte de services Web géospatiaux.

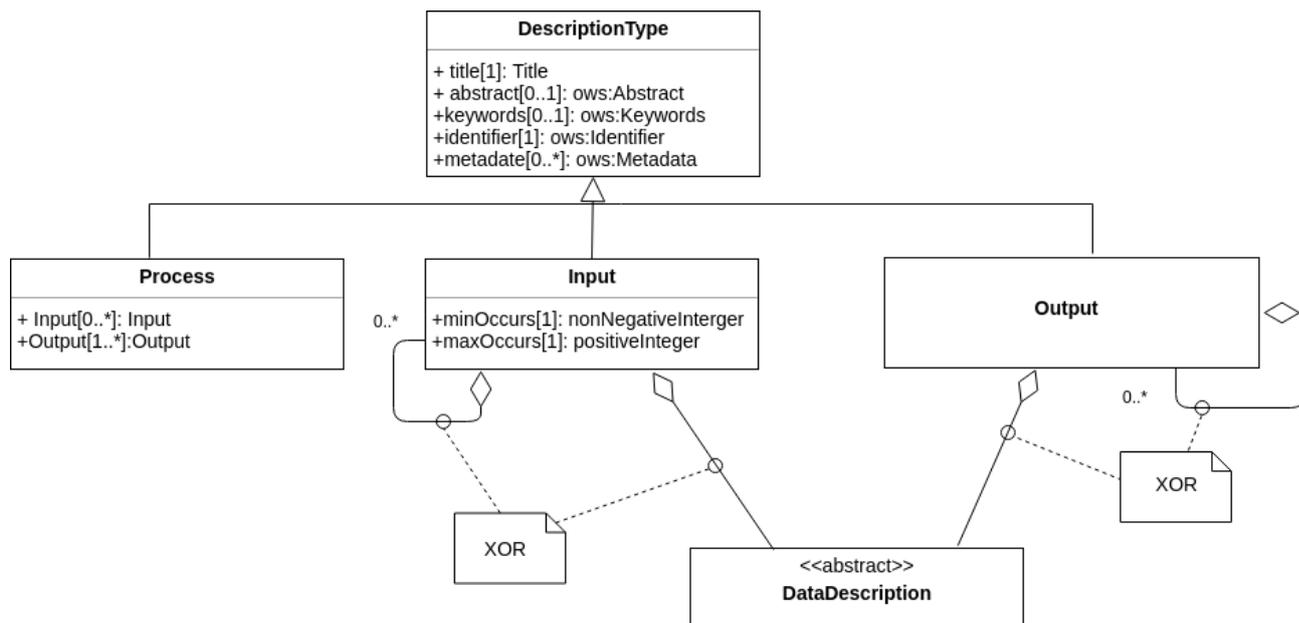


FIGURE 2.4 – Modèle de description de Process WPS [104]

Le standard WPS est très réputé, selon [35, 1, 39, 29] il s'impose comme la norme prometteuse pour le traitement géospatial sur le Web et a été mis en œuvre dans de nombreux projets.

## 2.2.2 Standard WPS

WPS est un standard qui d'une part définit pour les fournisseurs un modèle pour la publication des *Process* de traitements géospatiaux sur le Web (calcul d'itinéraire, géocodage, etc) et d'autre part définit pour le client un modèle pour demander la description et l'exécution d'un *Process* et comment la sortie du *Process* est gérée. La dernière version de ce standard est la version WPS 2.0 [104].

Dans ce contexte et selon la dernière version de WPS, une description de *Process* WPS est le modèle d'information qui spécifie l'interface d'un *Process*. Une description de *Process* est utilisée pour fournir une description lisible par la machine, du *Process*, et fournit également des informations de base sur les entrées et les sorties du *Process*. Elle est utilisée pour enregistrer le service dans un catalogue et comme modèle pour la découverte des services géospatiaux [104].

Dans la figure 2.4 la classe de description de *Process* est une extension de la classe *DescriptionType*. Elle hérite des attributs de cette classe qui seront utilisés pour définir le *Process* en spécifiant par exemple son identifiant, son titre ou des métadonnées associés pour fournir des informations supplémentaires ou plus de détails sur le *Process*. De plus la classe *Process* définit les entrées et sorties du *Process*.

Les structures de description des entrées et sorties du *Process* héritent également des attributs de la classe *DescriptionType*. Ces attributs sont utilisés pour exprimer l'identifiant, le titre ou des métadonnées associés qui fournissent des informations supplémentaires ou plus détaillées sur les entrées et les sorties du *Process*. Les entrées d'un *Process* ont une cardinalité afin de transmettre plusieurs valeurs avec le même identifiant, ou de déclarer les entrées de processus comme étant facultatives (cardinalité '0'). Concernant les sorties ils ont une cardinalité '1'.

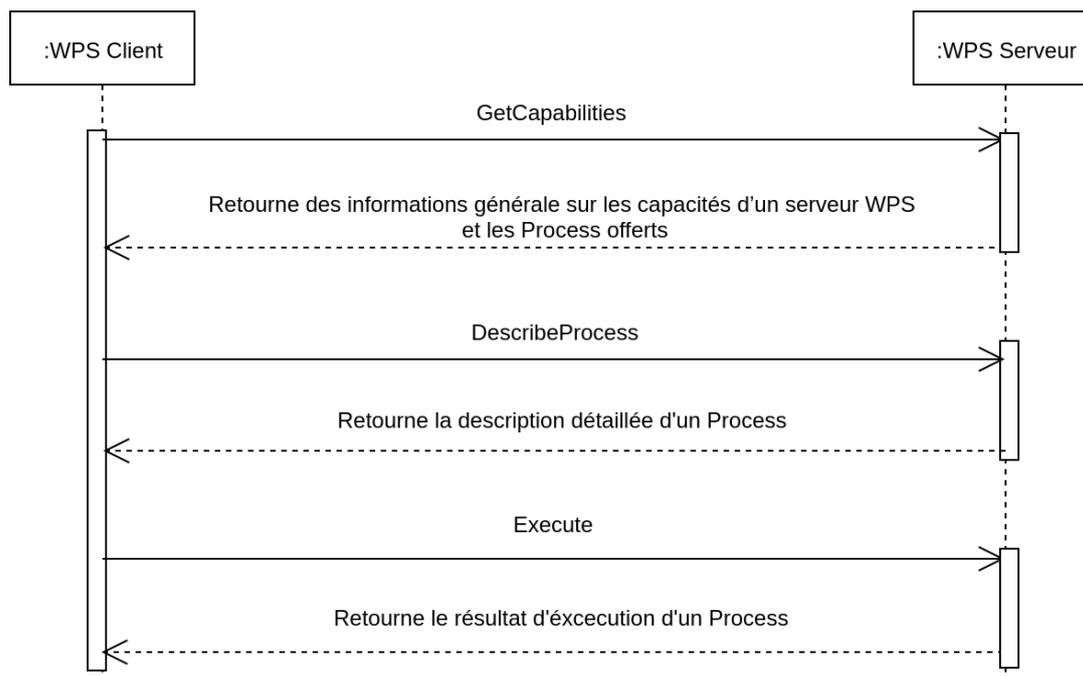


FIGURE 2.5 – Séquence d'interaction WPS Client-WPS Serveur

La communication entre serveur et client WPS (voir figure 2.5) est basée sur des règles pour paramétrer et exécuter un *Process*. Le standard WPS spécifie des opérations qui peuvent être invoquées par un client WPS et exécutées par un serveur WPS. Ci-dessous un tableau qui décrit plus en détails l'ensemble des opérations WPS :

TABLE 2.1 – Opérations WPS

Opération	Définition
<b>GetCapabilities</b>	Cette opération permet à un client de demander des informations générales sur les capacités d'un serveur WPS et les <i>Process</i> offerts.
<b>DescribeProcess</b>	Cette opération permet à un client de demander une description complète détaillée d'un <i>Process</i> offert par un serveur.
<b>Execute</b>	Cette opération permet à un client d'exécuter le <i>Process</i> en spécifiant l'identifiant du <i>Process</i> ainsi que les entrées et les sorties.
<b>GetStatus</b>	Cette opération permet à un client de demander les informations sur l'état d'une tâche de traitement en cours.
<b>GetResult</b>	Cette opération permet à un client de demander les résultats d'une tâche de traitement en cours.
<b>Dismiss</b>	Cette opération permet à un client d'annuler une tâche de traitement en cours.

Comme le montre la figure 2.5 pendant une séquence d'interaction WPS, un client doit d'abord émettre une demande *GetCapabilities* au serveur pour obtenir une liste des *Process* disponibles. Ensuite, il peut émettre une demande *DescribeProcess* pour obtenir une description détaillée d'un *Process* particulier offert, y compris les formats de données pris en charge.

Pour exécuter un *Process* avec les données d'entrée souhaitées, un client émettra une demande *Execute*.

Les opérations *GetStatus*, *GetResult* et *Dismiss* sont utilisées en conjonction avec l'exécution asynchrone. Si un serveur WPS n'offre qu'une exécution synchrone de *Process*, ces opérations ne sont pas mises en œuvre.

## 2.3 Approches sémantiques de description de services Web

Les approches sémantiques de description de services Web se situent à la convergence de deux domaines de recherche : les services Web et le Web sémantique (voir figure 2.6). Selon [17], le Web sémantique consiste à enrichir les ressources d'information sur le Web avec une sémantique. Cette sémantique permet aux systèmes informatiques de comprendre la signification de l'information. Dans ce contexte, la sémantique de l'information est définie dans des descriptions formelles basées sur des structures de connaissances (ontologies) pouvant être traitées par des machines. De ce fait, les machines peuvent communiquer à travers un contenu de pages Web décrit explicitement par un vocabulaire basé sur des ontologies de référence.

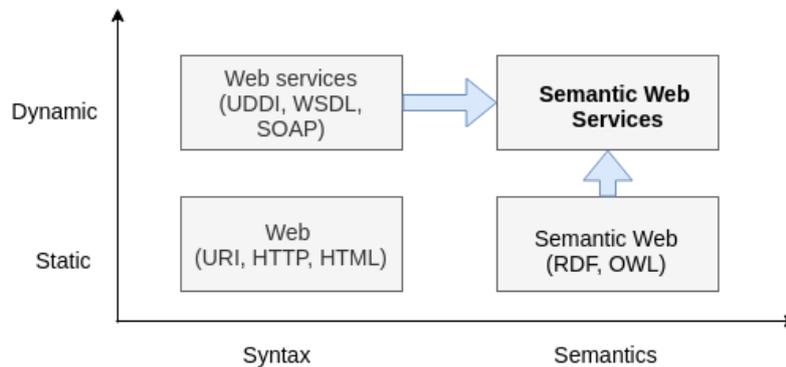


FIGURE 2.6 – Services Web sémantiques

La description sémantique d'un service peut être définie comme une explication informationnelle détaillée d'un service. Cette description sémantique permet que le service et ses paramètres aient une signification bien définie et puissent être compris par une machine. Elle est différente de la description classique de service Web comme celle proposée par le *WSDL* qui est une description syntaxique.

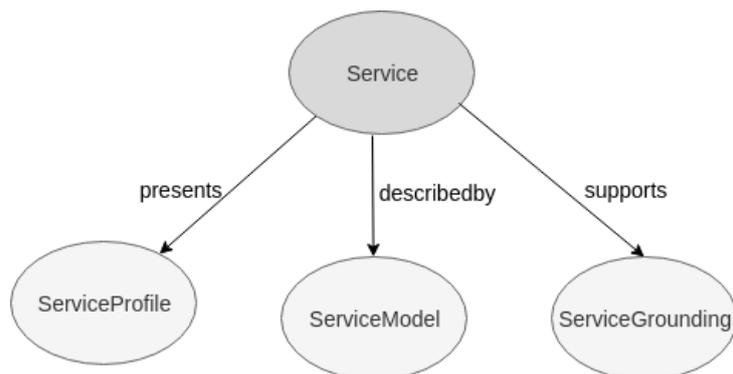
La mise en œuvre d'une description sémantique peut être réalisée selon deux types d'approches : les approches de description sémantique à base d'annotation et les approches basées sur les langages sémantiques. Nous décrivons dans ce qui suit les approches sémantiques de description de services Web.

### 2.3.1 Approches basées sur des langages sémantiques

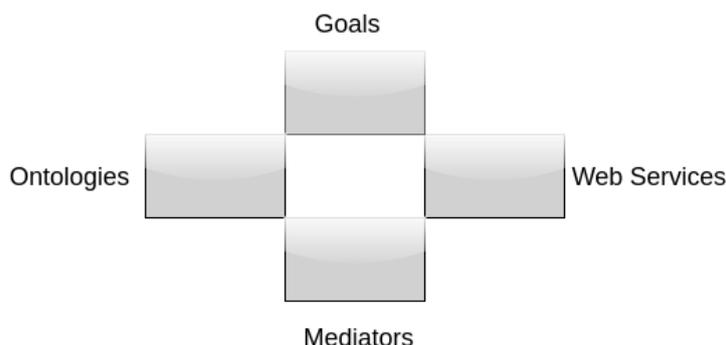
Les langages sémantiques de description de services Web, décrivent les services Web sur la base du langage de l'ontologie. Les deux langages de description sémantiques de services Web les plus cités sont *OWL-S* et *WSMO*.

- **OWL-S** : *Semantic Markup for Web Services* [96] est un langage permettant de décrire les services Web en se basant sur le langage d'ontologie du Web *OWL* (*Ontology*

*Web Language*). La description OWL-S est organisée en trois éléments principaux : (1) le profil (*Service Profile*), qui est utilisé pour décrire ce que le service va faire ; (2) le modèle de processus (*Process Model*), qui décrit la fonctionnalité et spécifie les façons avec lesquelles le client peut interagir avec les services. Cette perspective est capturée par la classe *ServiceModel* ; (3) les liaisons avec le service (*Service Grounding*), qui spécifie les détails sur la façon d'accéder aux services, les protocoles et les formats de messages, la sérialisation, le transport et l'adressage. L'objectif principal de *OWL-S* est de permettre aux utilisateurs de découvrir, de composer et d'exécuter automatiquement un service Web [138].

FIGURE 2.7 – Niveau supérieur de l'ontologie du service *OWL-S*

- **WSMO** : *WSML (Web Service Modeling Language)* est le langage utilisé pour décrire formellement tous les éléments *WSMO*. *WSMO* est une ontologie pour la description sémantique des services Web qui étendent le *WSMF (Web Service Modeling Framework)* [38]. Elle se compose de quatre éléments principaux (voir figure 2.8) : (1) Les ontologies (*Ontologies*), qui fournissent la terminologie utilisée par les autres éléments en termes de concepts, relations, fonctions, instances et axiomes ; (2) Les objectifs (*Goals*), qui définissent le problème, ou l'objectif des utilisateurs, qui doit être résolu par le service Web ; (3) Les services Web (*Web Services*), qui définissent les différents aspects des services Web notamment les fonctionnalités offertes par le service ; (4) Les médiateurs (*Mediators*), qui permettent de lier les différents éléments afin de permettre l'interopérabilité entre les composants hétérogènes. *WSMO* permet également la prise en charge des propriétés non fonctionnelles pour décrire les aspects non fonctionnels des services Web (*QoS*).

FIGURE 2.8 – Éléments de haut niveau de l'ontologie *WSMO*

Plusieurs travaux considèrent les descriptions avec des langages basés sur les ontolo-

gies comme OWL-S comme des modèles «lourds» qui nécessitent une restructuration de la description de services Web selon la structure du langage proposé [103, 105]. D'où le besoin d'une annotation plus légère permettant d'ajouter de la sémantique sans passer par un nouveau langage de description.

### 2.3.2 Approches basées sur des annotations sémantiques

L'annotation sémantique utilise des métadonnées dont la signification est formellement spécifiée dans une ontologie. L'annotation sémantique est donc une référence à une ontologie qui décrit les concepts et les relations entre les concepts d'un domaine [25]. Dans le contexte des services Web, l'annotation sémantique est une solution pour enrichir la description afin d'automatiser les processus de découverte et de composition. La figure 2.9 décrit le processus d'annotation sémantique de services Web.

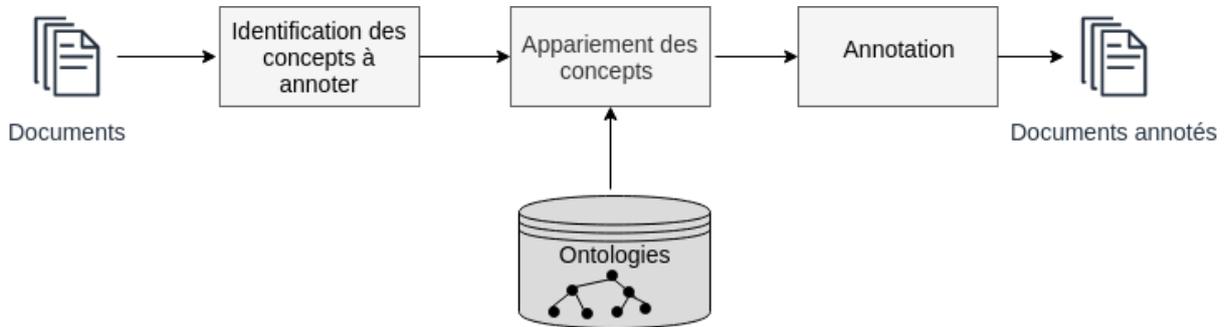


FIGURE 2.9 – Processus d'annotation sémantique de services Web

Les approches éminentes d'annotations sémantiques de services Web sont : *SAWSDL* (*Semantic Annotations Web Services Description Language*) [77], *SA-REST* (*Semantic Annotation of Web Resources*) [126] et hRESTS avec MicroWSMO [76].

- **SAWSDL** : *Semantic Annotations for WSDL and XML Schema* [77] est une recommandation W3C qui fournit un mécanisme permettant l'ajout d'annotations sémantiques au *WSDL*. Il est le successeur du *WSDL-S* [9] qui est un langage de description qui définit des concepts dans des ontologies externes qui permettent d'annoter sémantiquement des services Web *WSDL*. *SAWSDL* utilise principalement l'attribut "modelReference" pour annoter tous les éléments *WSDL* pour référencer un concept d'une ontologie. De plus *SAWSDL* proposent deux attributs "liftingSchemaMapping" et "loweringSchemaMapping" pour les mappages de schéma qui spécifient la transformation des données entre les messages et le modèle sémantique associé. La figure 2.10 présente un extrait d'annotation d'un fichier *SAWSDL*<sup>1</sup>.

```

...
<wsdl:operation name="order" pattern="http://www.w3.org/ns/wsdl/in-out"
  sawsdl:modelReference="http://www.w3.org/2002/ws/sawsdl/spec/ontology/purchaseorder#RequestPurchaseOrder">
  <wsdl:input element="OrderRequest"/>
  <wsdl:output element="OrderResponse"/>
</wsdl:operation>
...

```

FIGURE 2.10 – Exemple d'annotation *SAWSDL* d'une opération avec le concept "RequestPurchaseOrder"

- **SA-REST** : *Semantic Annotation of Web Resources* [126] ou également connue sous le nom de version RESTful de *SAWSDL* puisque elle a adopté l'idée de *SAWSDL*

1. <https://www.w3.org/TR/sawsdl/>

en liant les descriptions de service à des méta-modèles sémantiques par le biais d'annotations. La différence entre SAWSDL et SA-REST est que le SAWSDL ajoute leurs annotations à la description de service formelle dans WSDL, mais SA-REST ajoute leurs annotations aux descriptions HTML des services Web pour les rendre interprétables par la machine. L'annotation SA-REST est construite comme un triplet RDF. Elle permet d'annoter les entrées, les sorties, les opérations ainsi que le type de requête nécessaire pour invoquer le service. La figure 2.11 représente un extrait d'une description SA-REST.

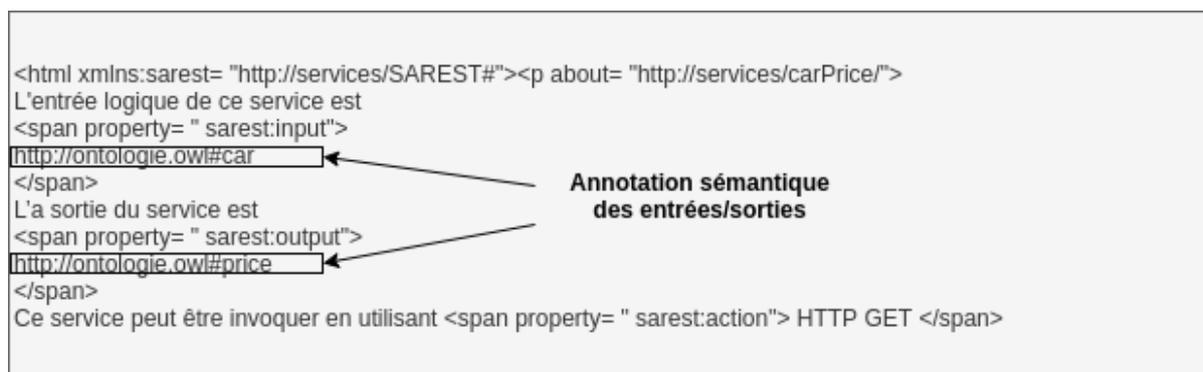


FIGURE 2.11 – Exemple d'annotation SA-REST des entrées/sorties.

- **hRESTS et MicroWSMO** : MicroWSMO est une extension de hRESTS [76] (un microformat basé HTML pour la description de services Web REST) pour l'ajout des annotations sémantiques. MicroWSMO a trois éléments principaux basé sur l'utilisation de trois attributs dont la sémantique est similaire à ceux de *SAWSDL*. Les attributs sont : "model", "lifting" et "lowering" qui joue le même rôle que les attributs "modelReference", "lifting-SchemaMapping" et "loweringSchemaMapping" définis dans *SAWSDL*.

Il existe également d'autres approches, pour la description sémantique de services Web proposées dans littérature, nous citons principalement, *SAWADL* (*Semantic Annotation for Web Application Description Language*) [94] qui est une extension de *WADL* qui adopte également l'idée de *SAWSDL* pour enrichir les descriptions *WADL* avec une couche sémantique. Nous citons également des approches basées sur des langages sémantiques telles que *WSMO-lite* [142] qui s'inspire de l'ontologie *WSMO*, se concentrant sur un petit sous-ensemble de celle-ci pour définir une extension limitée de *SAWSDL*. *WSMO-lite* représente les annotations *SAWSDL* avec des descriptions de service sémantiques concrètes. L'objectif de *WSMO-lite* est de proposer une description plus légère tout en prenant en charge de nouveaux défis dans l'intégration de services Web comme la prise en charge de propriétés non fonctionnelles [78].

## 2.4 Approches sémantiques de découverte des services Web

Les approches de découverte de services Web sémantiques dépendent, en particulier, de la méthode d'appariement sémantique utilisée, également connue sous le nom de "semantic service matchmaking". Le processus de découverte ou de matchmaking de services comprend (1) l'appariement sémantique d'une requête de service donnée avec chaque service enregistré dans un répertoire de services, et (2) le renvoi des résultats de découverte

(une liste de services pertinents). La découverte ou le matchmaking concerne uniquement ces deux étapes et ne gère pas l'interaction avec les services sélectionnés.

Les approches actuelles d'appariement sémantiques de services Web peuvent être classées en trois principales classes [74] : les approches logiques, les approches non logiques et les approches hybrides.

L'appariement sémantique basé sur la logique utilise les concepts d'ontologies et des règles logiques pour vérifier la compatibilité entre la requête et le service. L'appariement sémantique non logique exploite, par exemple, des méthodes d'appariement de graphes, de schémas, ou différentes mesures de similarité. L'appariement sémantique hybride utilise une combinaison de classes logiques et non logiques. L'idée est de remédier à certaines limites de chacune de ces deux classes grâce à différentes combinaisons hybrides qui réussissent là où chacune de ces deux approches échouent [75]. La grande majorité des travaux de découverte de services sémantiques actuellement disponibles exploitent l'une des approches de description sémantique préalablement présentées comme *OWL-S* et *SAWSDL*.

L'appariement sémantique fait correspondre la description de services et la description de la requête selon plusieurs éléments impliqués dans la description de ces derniers. Nous citons principalement, la correspondance qui inclut la comparaison des paramètres d'entrées et de sorties de services (*E/S*), la correspondance qui inclut la comparaison des préconditions et des effets (*P/E* (*Préconditions/Effets*)), la correspondance qui inclut les deux (*E/S* et les *P/E*). D'autres approches peuvent inclure en plus la comparaison d'autres éléments tels les structures de description des services et de la requête, etc.

### 2.4.1 Approches logiques

Cette catégorie d'approches utilise les concepts d'ontologies et des règles logiques pour vérifier la compatibilité entre la requête et les services en comparant soit les *E/S*, soit les *P/E*, soit les deux. Ils sont principalement basés sur des filtres d'appariement sémantiques appelés filtres *DoM* (*Degree of Match*). Les filtres sont déterminés en se basant sur les relations de subsomption entre des concepts au sein d'une ontologie. Les premiers filtres ont été définis sur les *E/S* dans les travaux de [112] et sont détaillés dans le tableau 2.2.

TABLE 2.2 – Catégories de filtres logiques

Nom	Définition	Formelle rep
<i>Exact</i>	Les sorties du service et de la requête sont équivalents.	$Output(R) \equiv Output(S)$
<i>Plugin</i>	Les sorties du service sont des généralisations des sorties de la requête.	$Output(R) \sqsubseteq Output(S)$
<i>Subsume</i>	Les sorties du service sont des spécifications des sorties de la requête.	$Output(R) \sqsupseteq Output(S)$
<i>Fail</i>	Aucun filtre ne peut être déterminé parmi ceux mentionnés ci-dessus.	$Output(R) \not\equiv Output(S)$

Dans cette approche, ils comparent les sorties de la requête avec les sorties des ser-

vices offerts et ils les classent selon les filtres d'appariements ( $|Exact| > |Plug - in| > |Subsume| > Fail$ ). Si deux services  $S1$ ,  $S2$  ont le même filtre d'appariement, alors ils comparent leurs entrées avec les entrées de la requête. L'approche est simple en termes d'implémentation mais elle peut avoir de faibles taux de rappel et de précision à cause de la naïveté des règles utilisées qui peuvent générer des faux positifs et des faux négatifs. [60] est un autre exemple de travaux de services Web sémantiques qui effectuent une correspondance sémantique basée sur la logique. La définition des filtres *DoM* diffèrent dans les différents travaux. Par exemple, dans certains cas, le filtre de correspondance "plug-in" peut être défini de manière inverse à la définition d'origine. D'autres approches ont pris en compte la correspondance des *E/S* et des *P/E* [132].

Selon [20] l'utilisation unique des approches logiques basée sur la signature complète des services et de la requête est considérée comme un appariement rigide notamment pour les classements de services qui dépendent des filtres d'appariements. Ceci peut tolérer un grand nombre de faux positifs au niveau du score de subsumption surtout si la chaîne de subsumption, reliant les deux concepts à comparer, est longue.

### 2.4.2 Approches non logiques

Après avoir présenté, les approches de découverte de services basées sur les mécanismes logiques, dans cette section nous présentons les approches de découverte non logiques.

L'appariement non-logique utilise des mécanismes pour la recherche d'information basés sur des mesures de similarités tels que cosinus [56], le coefficient de Jaccard [59], des mécanismes pour la correspondance de graphes structurés basés sur des similarités structurelles [65] ou des mécanismes de calculs de distance de concepts numérique sur des ontologies données tel que la mesure de similarité Wu et Palmer [149]

Dans certains travaux ils utilisent une combinaison de plusieurs mesures comme c'est le cas pour l'approche COV4SWS.KOM [80] qui applique des mesures de similarité du domaine de la relation sémantique, à savoir les métriques de Lin [89] et Resnik [122] et utilise une méthode d'agrégation des résultats en suivant la stratégie de la moyenne pondérée. Nous citons aussi les travaux de [117] qui assurent un appariement sémantique non-logique en combinant une méthode de similarité structurelle et textuelle.

### 2.4.3 Approches hybrides (logiques et non logiques)

Les approches hybrides combinent des méthodes logiques et non logiques pour l'appariement des services Web dans le but de remédier à certaines limites de chacune de ces deux classes grâce à différentes combinaisons hybrides en agrégeant les résultats d'appariement.

[60] présentent l'approche hybride OWLS-M où l'appariement porte principalement sur les paramètres d'*E/S* des services *OWL-S*. L'approche utilise à la fois le raisonnement logique en définissant des filtres logiques et des mécanismes non-logiques en utilisant des techniques de recherche d'information. Les résultats d'appariement sont ordonnés selon la pertinence des services après que l'utilisateur spécifie d'un côté le filtre d'appariement souhaité pour l'appariement logique et un seuil de la valeur de similarité syntaxique pour l'appariement non-logique. Différentes variantes de OWLS-M ont été proposées, jusqu'à OWLS-M4. Ces variantes conservent les mêmes filtres logiques mais diffèrent dans les approches non-logiques utilisées.

D'autres travaux ont succédé les travaux des approches OWLS-M, nous citons principalement OWLS-MX3 [71] qui applique tout comme les autres variantes OWLS-M, un

appariement logique et non logique. Une autre caractéristique de l'OWLS-MX3 est qu'il utilise une approche adaptative (un modèle *SVM* (*Support Vector Machine*)) pour agréger les résultats d'appariement.

*SeMa<sub>2</sub>* [97] présentent une approche hybride combinant des techniques d'appariement logique et non-logique pour les services *OWL-S*. En plus de la comparaison logique, *SeMa<sub>2</sub>* traite la comparaison syntaxique des noms de service et des descriptions de service.

Le SAWSDL-MX2 [75] présentent une approche hybride pour l'appariement des *E/S* des services SAWSDL. Il introduit également une couche d'apprentissage adaptative où le modèle *SVM* est constitué de valeurs pour les critères logiques, et non-logiques syntaxique et structurel.

iSeM [72] proposent une approche hybride. Il hérite de ses prédécesseurs -MX, les caractéristiques techniques notamment les techniques d'appariement logique et non logique, et l'approche *SVM* pour l'agrégation des résultats de toutes les mesures. iSeM 1.0 présente deux caractéristiques supplémentaires, à savoir qu'il est adapté pour *OWL-S* et *SAWSDL*.

[80] proposent une approche hybride pour les services SAWSDL en appliquant un appariement d'*E/S* qui inclue l'approche non-logique COV4SWS, et une approche logique LOG4SWS. LOG4SWS a été inspiré des filtres logiques de OWLS-MX3. LOG4SWS utilise les graphes bipartites pour trouver la similitude optimale entre la requête et les descriptions de service. De plus il exploite WordNet avec une stratégie de retour arrière dans le but de trouver des similarités entre les opérations ou les paramètres de services dans le cas où aucune relation de similarité n'est détectée.

[81] proposent XAM4SWS, une approche hybride basée sur LOG4SWS, qui supporte des services RESTful enrichis sémantiquement. En particulier, hRESTS et MicroWSMO. L'appariement utilise un filtrage logique combiné à une comparaison non logique structurelle des annotations sémantiques. XAM4SWS considère que la fonctionnalité offerte par le service est décrite principalement dans la partie opération, il est donc centré opération. La similarité entre deux opérations est calculée en fonction de la similarité entre les *E/S*. Les résultats sont agrégés au moyen d'une moyenne pondérée. De plus, les détails spécifiques à *REST*, tels que le type de requête *HTTP*, sont pris en compte pour le classement général des résultats.

Actuellement, les *Matchmakers*, les plus performants en termes de précision moyenne, pour la découverte de services Web est iSeM (0.922) pour les services décrit avec *SAWSDL*, et *SeMa<sub>2</sub>* (0.877) pour les services décrit avec *OWL-S*.

## 2.5 Approches de composition de services Web

La composition d'un service Web peut être définie comme la combinaison de plusieurs services Web pour atteindre un objectif. Le besoin de composition de services Web provient de la nécessité d'atteindre un objectif complexe qui ne peut pas être réalisé par un service autonome. Elle consiste à combiner plusieurs services pour satisfaire un besoin complexe de l'utilisateur dans le cas où il est constitué de plusieurs sous-but, chacun réalisé par un service différent comme le montre la figure 2.12. En ce sens, la composition du service consiste à combiner des services en utilisant des structures de contrôle pour des exécutions parallèles ou séquentielles à partir d'un ensemble de services Web existants.

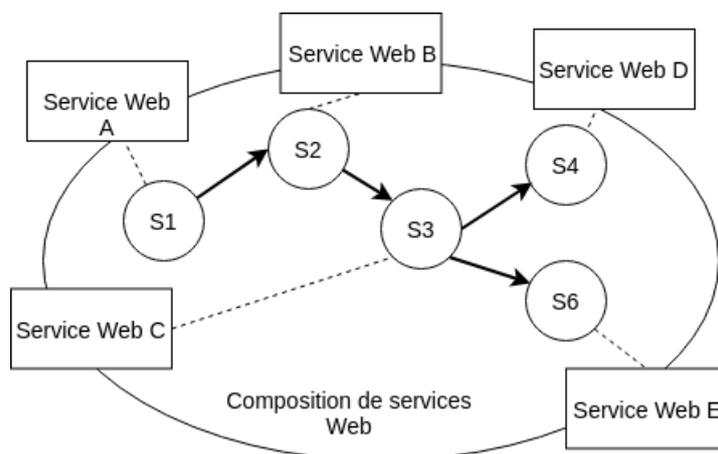


FIGURE 2.12 – Composition de services Web

Nous pouvons identifier plusieurs concepts et questions liés au problème de la composition des services Web, abordés dans la littérature [15, 118, 125, 78, 85]. Dans ce contexte, nous définissons les méthodes de composition existantes (voir figure 2.13). Selon le type de contrôle des services, nous distinguons l’orchestration et de la chorégraphie de services Web. Ensuite, selon la gestion des services nous distinguons la composition statique et dynamique. Enfin, selon le degré d’automatisation, nous abordons la composition manuelle, semi-automatisée et automatisée des services.

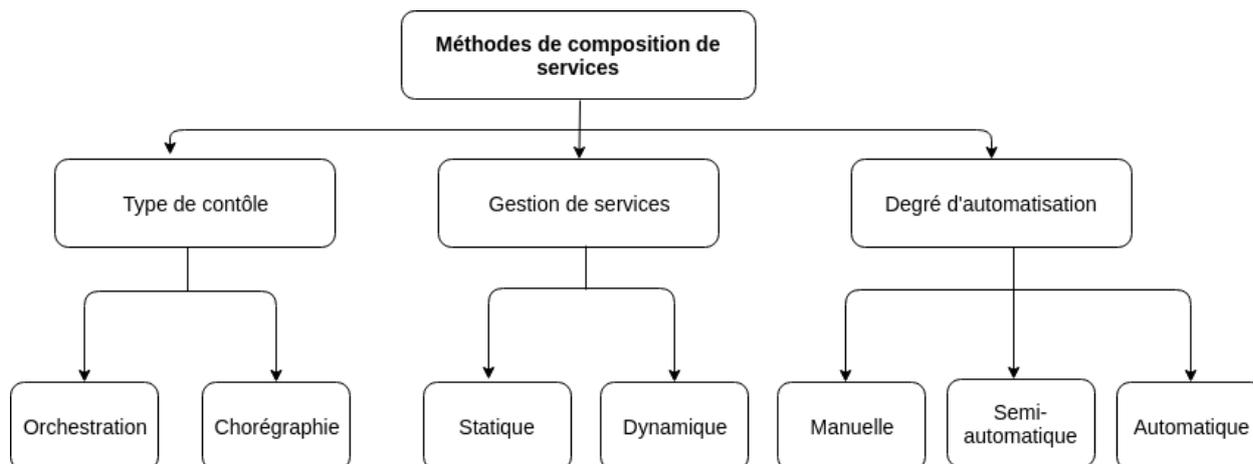


FIGURE 2.13 – Classification des méthodes de composition de services

Dans cette section, nous allons présenter les différents types de composition de services présents dans la littérature. Ensuite, nous allons passer en revue les approches automatiques pour la composition utilisant les techniques d’IA (*Intelligence Artificielle*).

### 2.5.1 Orchestration et chorégraphie de services Web

Selon le type de contrôle des services, la composition du service peut être réalisée en utilisant l’un des deux paradigmes suivants : l’orchestration de services et la chorégraphie de services [115].

L’orchestration de services représente un processus opérationnel centralisé unique (l’orchestrateur) qui coordonne l’interaction entre les différents services. L’orchestrateur est chargé d’invoquer et de combiner les services. Les relations entre tous les services participants sont décrites par un point final unique, c’est-à-dire le service composite.

L'orchestration comprend la gestion des transactions entre les différents services. L'orchestration utilise une approche centralisée pour la composition des services.

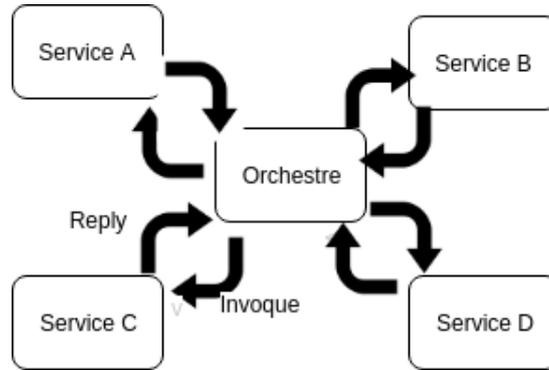


FIGURE 2.14 – Orchestration de services

La chorégraphie de service est une description globale des services participants, qui est définie par l'échange de messages, les règles d'interaction et les accords entre deux ou plusieurs points terminaux. Elle permet à chaque partie impliquée de décrire son rôle dans l'interaction. La chorégraphie utilise une approche décentralisée et collaborative pour la composition des services.

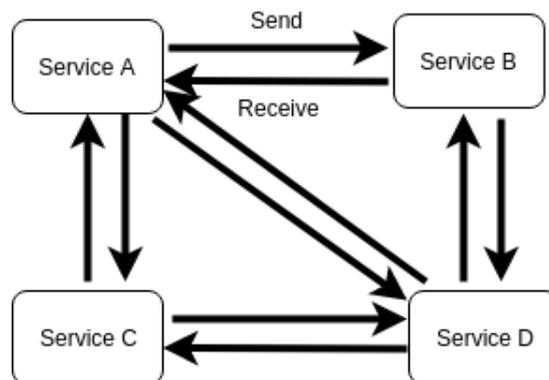


FIGURE 2.15 – Chorégraphie de services

En d'autres termes, une chorégraphie décrit les interactions de collaboration entre plusieurs services, alors que l'orchestration représente un contrôle centralisé du point de vue d'une partie. Cela signifie qu'une chorégraphie diffère d'une orchestration en ce qui concerne l'endroit où doit résider la logique qui contrôle les interactions entre les services impliqués.

Les figures 2.14 et 2.15 illustrent ces approches à un niveau supérieur.

### 2.5.2 Composition statique et dynamique

Selon la gestion des services, et plus précisément, en fonction du moment au cours duquel les services Web sont composés, la composition des services peut être classée selon les deux catégories : la composition statique et la composition dynamique [125].

Plus précisément, la principale différence entre ces deux catégories statique et dynamique dépend du temps pendant lequel un service Web concret est intégré dans la composition. Dans la première catégorie, la liste de services requis pour la composition sont choisis, liés, puis déployés au moment de la conception ; tandis que dans la deuxième

catégorie la liste des services est sélectionnée dynamiquement lors de la réception d'une requête. Selon [54], « *The service composition techniques are classified generally as Dynamic and Static composition that depend on timing. When Web services are composed, Static Composition happens through the design when it is planned and through the system architecture. Dynamic composition happened during runtime when the single provider does not provide the requested service.* »

Les solutions de composition statiques, sont généralement choisies lorsque les services impliqués dans la composition et les exigences de composition sont statiques, c'est-à-dire qu'ils ne changent pas ou changent rarement, et aussi lorsque les services sont disponibles en permanence. Ainsi, les solutions de compositions statiques ne peuvent pas être utilisées dans un environnement flexible où ils ne sont pas adaptables aux changements tels que l'intégration de nouveaux services par les fournisseurs ou la mise à jour des informations concernant les services disponibles.

Pour surmonter les problèmes qui se manifestent dans la composition statique, la composition de services dynamique intervient. Le but de la composition dynamique est de composer des services complexes à la volée, avec une prise en compte des mises à jour concernant les informations des services.

Dans la littérature, généralement les approches adoptées pour la composition dynamique des services consistent à développer des modèles avec des schémas de composition abstraits prédéfinis et à sélectionner les services lors de la phase d'exécution. Dans ces modèles, le schéma de composition abstrait comprend un certain nombre d'actions, qui sont généralement représentées comme des activités abstraites. L'ordre d'exécution des différentes activités y est indiqué. La phase de sélection consiste à choisir un service concret adapté à chaque activité et à l'intégrer dans le plan de composition.

En revanche, ces démarches ne permettent pas de faire face à certains cas où le schéma de composition ne peut être défini que partiellement ou ne peut être défini à l'avance. La gestion des crises est un exemple de cas où les crises surviennent souvent de manière inattendue, ce qui empêche d'identifier précisément les bonnes actions avant qu'une crise ne se produise et de disposer d'informations contextuelles à l'avance (par exemple, les victimes, les témoins, les conditions météorologiques, etc.).

### 2.5.3 Composition manuelle, semi-automatique et automatique

En fonction du niveau d'automatisation, les approches actuelles de composition des services Web peuvent être regroupées en trois catégories : manuelle, automatisée et semi-automatisée.

Dans la composition manuelle, un concepteur humain (administrateur ou fournisseur de services) procède à la composition des services en reliant manuellement les services Web. Les travaux de composition manuelle [135] requièrent la définition d'un flux de contrôle (un Workflow abstrait) et d'un flux de données manuellement, soit directement, soit par l'intermédiaire d'outils en utilisant un langage de service Web standard, tel que *BPEL (Business Process Execution Language)*<sup>2</sup>. Ensuite, il relie manuellement les services Web aux actions du processus abstrait. Ce type de composition, en plus de nécessiter une intervention humaine constante, prend du temps surtout si le nombre de services est important, et les résultats sont sujets à des erreurs, et il n'y a aucune garantie qu'il réponde à l'exigence de l'utilisateur.

---

2. *BPEL* : [https://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=wsbpe1](https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpe1)

L'automatisation du processus de composition, concerne principalement l'automatisation de création de schéma de composition spécifiant le flux de contrôle et de données entre les différentes actions et l'affectation de services concrets à ces actions ou directement entre les différents services concrets sans passer par un schéma abstrait. Le degré d'automatisation du processus de composition est une dimension importante de toute stratégie de composition de services.

La composition semi-automatisée est un type de composition où les utilisateurs sont assistés pendant le processus de composition. Dans ce type de composition l'automatisation peut concerner plusieurs aspects, par exemple, l'utilisation d'interfaces graphiques pour aider l'utilisateur à construire le schéma de composition sous la forme d'un Workflow, ou la suggestion de services concrets pour découvrir et sélectionner les services à composer. Cependant, le rôle de l'utilisateur reste central dans la définition du schéma de composition ou dans le choix des services concrets à utiliser. Ce type de composition permet de résoudre certains des problèmes liés aux méthodes de compositions manuelles, mais elles ne sont pas encore adaptées à une utilisation dans des environnements de services à large échelle.

Le type de composition automatique consiste en une composition où l'intervention de l'utilisateur n'est requise que pour la spécification des exigences sous la forme d'une demande de service. Ainsi le processus de composition est entièrement automatisé et transparent pour l'utilisateur. La sélection des services se fait selon les besoins de l'utilisateur et, dans certains cas, une prise en compte des contraintes, préférences de l'utilisateur et d'information contextuelle. Le processus de composition peut avoir lieu avant ou pendant l'exécution des services. Les méthodes de composition automatique sont particulièrement adaptées aux environnements volatiles, tels que les environnements intelligents ou les environnements dans un contexte mobile où la disponibilité des services est dynamique et les besoins des utilisateurs sont variables et personnalisés.

#### **2.5.4 Approches automatiques pour la composition de services Web**

Aujourd'hui, l'un des problèmes ouverts est la réalisation d'une composition de services automatisée qui reste un défi et présente plusieurs problèmes ouverts (hétérogénéité des descriptions, dynamisme de la composition des services, etc.) [12]. [125] définit le processus composition automatique comme une sorte de système d'intégration sémantique des services. En donnant un ensemble de descriptions de services et les exigences de l'utilisateur, une spécification de service composite peut être générée automatiquement. Les approches de composition automatisée de services exploitent généralement les technologies du Web sémantique [17]. La composition des services du Web sémantique tient compte de la signification et de l'objectif des services. Le but du service Web sémantique est de faire en sorte que les machines puissent automatiquement découvrir et composer les services pour atteindre un objectif en fonction des besoins des utilisateurs. La composition sémantique est réalisée sur la base de la description sémantique des services Web et des méthodes de découverte qui permettent de définir les différents liens entre les services Web [84].

Afin de parvenir à une composition automatisée des services, des travaux ont mis en relation cette démarche avec les techniques de résolution automatisée des problèmes disponibles dans l'IA ([36, 31, 119, 57, 120, 15, 118, 125, 78, 85]). Plus précisément, avec les techniques de planification de l'IA. Ainsi, le problème de composition est défini comme un problème de planification. L'analyse de la littérature montre les différentes recherches dans ce domaine. Nous détaillons dans ce qui suit les principales classes identifiées dans

la littérature :

- **Planification à base de règles** : la planification basée sur des règles est un type de planification fondée sur un ensemble de règles et de conditions dans la même philosophie que les déclarations *if-then-else*. [101] proposent une méthode de composition à base de règles. La méthode est décomposée en quatre phases. La première phase exploite le langage de spécification *CSSL (Composite Service Specification Language)* pour décrire la description à haut niveau. La seconde est la phase d'appariement qui utilise des règles de composabilité pour générer des plans conformes aux spécifications du service demandé. La troisième consiste en une phase de sélection. La dernière phase est une phase de génération d'un diagramme *UML (Unified Modeling Language)* associé à la composition.
- **Planification basée sur *PDDL (Planning Domain Definition Language)*** : *PDDL* [99], est une tentative pour standardiser les données d'entrée d'un planificateur. Le but de ce langage est l'adoption d'un formalisme commun pour la description des domaines de planification, ce qui a motivé la communauté de recherche sur les services Web à l'adapter au contexte de la composition des services. [36] proposent une solution pour la composition automatique de services Web, une traduction de *WSML* en *PDDL*. [150] proposent une solution pour la composition de services Web en traduisant les descriptions *OWL-S*, en *PDDL* et en considérant les propriétés fonctionnelles et non fonctionnelles.
- **Planification basée sur le réseau hiérarchique de tâches *HTN (Hierarchical Task Network)*** : la planification basée sur *HTN* est fondée sur des langages très expressifs permettant de spécifier les connaissances. *HTN* suit une méthode de planification par la décomposition des tâches. L'objectif est exprimé comme un ensemble de tâches à réaliser, c'est-à-dire les tâches de l'objectif, et auxquelles il est possible d'associer différents types de contraintes. Un couple (contraintes, tâches) est appelé un réseau de tâches. La recherche d'une solution consiste à décomposer les tâches de l'objectif en sous-tâches satisfaisant les contraintes jusqu'à ce qu'un ensemble de sous-tâches primitives soit trouvé : les tâches primitives peuvent être exécutées.  
Dans [129] le système de planification *SHOP2* [107] est exploité pour la composition automatique des services qui prend en compte la sémantique fonctionnelle des services. [91] est une approche qui est également basée sur la planification *HTN* permettant de composer automatiquement des services Web en incluant en plus des préférences de l'utilisateur. Le travail décrit principalement comment les préférences sont traduites dans un langage de planification *HTN*. Récemment les travaux [131, 16] ont proposé une méthode de composition automatique des services en se basant sur *HTN* et en considérant les propriétés fonctionnelles et non fonctionnelles.
- **Planification basée sur la logique** : la planification basée sur la logique est un type de planification qui consiste à la définition d'un ensemble de règles de composition dans la logique du premier ordre. Plus précisément, le plan de composition est construit sur la base de ces règles et un ensemble de contraintes.  
[100] présentent une méthode de planification basée sur la logique et plus précisément sur le calcul de situations. Le principe de cette méthode est la définition d'un

ensemble de situation où chaque situation est créée par une action effectuée sur un état. Les auteurs considèrent un service Web comme une action. La méthode proposée se base sur le langage Golog, un langage de programmation logique de haut niveau conçu à partir du langage formel de calcul de situation. [116] ont proposé une extension de Golog (conGolog) dans le but de faire correspondre les spécifications *OWL-S* au langage de calcul de situation. D'autres travaux [130] ont proposé une version modifiée de Golog pour prendre en compte les préférences des utilisateurs dans la planification basée sur Golog pour la recherche de compositions optimales. [110] présentent une méthode de composition automatique de services Web développée pour le projet Phylotastic dans le domaine de la biologie. La méthode se base sur l'utilisation du langage de programmation logique *ASP* pour la formulation de l'ontologie de domaine, et du problème de composition. Au début ils proposent la conception d'un Workflow abstrait manuellement, et procèdent par la suite à la sélection de services concrets automatiquement en prenant en compte les relations logiques d'exactitude et de subsumption. Nous notons que l'utilisation d'*ASP* pour la formulation de problèmes en *IA* comme la planification a été rendue populaire grâce au développement de plusieurs solveurs performants dédiés à l'*ASP* tels que DLV [86] et dernièrement Clasp [44].

- **Planification à base de graphe :** la planification basée sur des graphes a également été exploitée pour proposer des méthodes de composition de services Web. Une planification basée sur les graphes construit un graphe acyclique dirigé. Les nœuds représentent les services et les arcs les dépendances sémantiques.

Dans ce type de méthode, ils divisent le processus de composition en deux modules : le module de construction de graphe de planification et le module de recherche d'un schéma de composition.

La recherche de solution est généralement basée sur un espace constitué d'états, d'actions, de fonctions de transaction et de fonctions de coût. L'objectif de cette planification est de rechercher la meilleure solution pour atteindre l'état visé. Dans ce contexte, deux types de planification sont connues : la recherche en avant, qui commence par l'état initial et atteint un état cible, et la recherche en arrière, où l'action opposée est effectuée. Dans les deux cas, la solution consiste à trouver un ensemble d'actions qui relie l'état initial à l'état d'objectif [11]. Cependant, dans ce type de planification l'espace de recherche possible peut très vite augmenter. Afin de réduire l'espace de recherche, des solutions basées sur des fonctions heuristiques, ou des algorithmes spécifiques sont utilisées pour estimer l'utilité des chemins de recherche à choisir.

[146] présentent une méthode pour la composition automatique de services Web basée sur un modèle de graphe de planification. Ils présentent un algorithme pour la construction du graphe et la recherche de solutions possibles en se basant sur une heuristique.

[123] présentent une méthode pour la composition automatique de services Web basée sur un modèle de graphe de planification. Ils se basent sur le calcul de la correspondance sémantique entre les paramètres des services pour construire le graphe de planification et utilisent des techniques d'élagage avec retour arrière pour supprimer les services inutiles à la composition.

[79] proposent une méthode de planification pour la composition automatique de

services Web basée sur un modèle de graphe de planification en intégrant les informations contextuelles relatives aux services Web. Dans un premier temps, ils proposent d'enrichir les descriptions de services Web avec des informations contextuelles en utilisant une ontologie basée sur un modèle de contexte. Dans un deuxième temps ils construisent le graphe en se basant sur un processus de découverte sémantique et contextuelle des services Web. L'étape de recherche de solutions possible se base sur une des techniques d'élagage avec retour arrière. Enfin, une étape de classement en calculant des scores est prise en compte pour chaque solution candidate.

D'autres méthodes de composition automatique de services ont été proposées dans la littérature. Celles-ci peuvent soit être classées dans plusieurs classes parmi celles mentionnées ci-dessus, soit elles peuvent être classées dans de nouvelles classes spécifiques. Nous citons comme exemple, les méthodes basées sur les réseaux de petri [48]. Nous citons également, les approches basées sur le raisonnement par cas (et/ou) apprentissage. [145] abordent la problématique de la composition à travers un grand nombre de services dans un environnement dynamique. Les auteurs proposent un modèle de composition et d'adaptation qui intègre les connaissances de l'apprentissage du renforcement. Plus spécifiquement, un algorithme de composition de services basé sur le Q-learning est défini sur la base du modèle proposé.

### 2.6 Travaux de description, découverte et composition sémantiques dans le domaine géospatial

Le domaine géospatial, et plus particulièrement l'*OGC* et l'*ISO/TC211* intègre des standards pour la description syntaxique des *SWG* comme l'exemple de la norme *WPS* que nous avons détaillé dans la section 2.2.2. Toutefois, ces standards ne tiennent pas compte du fait que la conceptualisation des caractéristiques est nécessaire, ce qui entraîne une hétérogénéité sémantique lors de la découverte des *SWG* [93]. Ainsi, les méthodes de découverte dans les catalogues basés sur ces standards de descriptions, permettent aux utilisateurs soit de demander une description des services s'ils connaissent leurs identifiants, soit de rechercher les services en formulant des requêtes en utilisant des mots clés.

Selon [152], les *SWG* ont le potentiel de devenir intelligents, c'est-à-dire qu'ils ont le potentiel de faciliter leurs découverte et composition automatisées dans le but de générer les bonnes informations requises au bon moment. Pour atteindre cet objectif, il est essentiel de passer de l'interopérabilité syntaxique actuellement soutenue à l'interopérabilité sémantique.

Comme nous l'avons vu dans les précédentes sections, les Web services sémantiques ont été utilisés pour permettre la découverte et la composition de services et assurer une meilleure précision des résultats de recherche. Ainsi, les technologies du Web sémantique [17] sont prometteuses pour atteindre l'interopérabilité sémantique des *SWG*.

Une seconde problématique, concernant les standards actuels des *SWG*, est l'absence de description des propriétés non fonctionnelles, également connues sous le nom d'attributs *QoS* qui est essentielle dans la sélection et recommandation de *SWG* atomiques ou composites. Cette problématique a fait l'objet de plusieurs travaux de recherche [131, 16, 58]. Dans la littérature, nous trouvons divers attributs de *QoS* [66, 102], nous citons comme exemples la performance et la disponibilité. Une spécificité du domaine

géospatial, est qu'en plus de la prise en compte des attributs de *QoS* génériques, d'autres attributs de *QoS* spécifiques au domaine géospatial [88] peuvent être prises en compte, telles que la précision de coordonnées géographiques ou la fréquence de mise à jour des gazetiers utilisés.

Un certain nombre de travaux [68, 14, 62, 26, 13, 26, 36, 87, 37, 10] dans le domaine géospatial ont abordé les problématiques liées à l'exploitation des technologies du Web sémantique pour la description, la découverte (et/ou) la composition des services géospatiaux (et/ou) les problématiques liées à la prise en compte de la *QoS*. Nous présentons dans ce qui suit les études en liaison avec ces derniers, en soulignant les différents modèles et approches utilisés.

Concernant ces défis, [62] ont tenté d'identifier des moyens et concepts pour inciter à l'intégration des technologies du Web sémantique dans le domaine géospatial et particulièrement l'annotation des services géospatiaux. Les approches proposées restent à appliquer et à tester dans divers cadres pratiques. Également, [13] présentent l'environnement SWING qui comprend un support pour le cycle de vie des services Web sémantiques géospatiaux sous forme d'une étude de cas sur l'utilisation de la sémantique. Ils proposent l'utilisation du langage *WSML* pour la description de services mais ne précise pas de méthode de découverte. Ils proposent également une méthode pour la composition statique de services géospatiaux.

[26] proposent une approche de composition automatique basée sur une méthode de planification avec prise en compte d'exigences de qualité des données géospatiales. Plus précisément ils utilisent une méthode de chaînage arrière. Les services Web sont décrits sémantiquement en utilisant le langage *OWL-S*, la méthode d'appariement utilisée est un appariement exact entre les concepts d'*E/S*.

[36] présentent une approche pour la composition automatique de services Web géospatiaux. Cette solution s'appuie pour la description sur *WSMO* et l'approche de découverte n'a pas été spécifiée. Pour la solution de composition, ils proposent une solution basée sur la planification et plus précisément, le langage de planification *PDDL* [99], a été choisi pour décrire la définition du domaine et du problème de planification.

[87] présentent une solution pour la découverte de services géospatiaux. Ils utilisent *OWL-S* comme modèle de description. La méthode de découverte établit un appariement de la fonctionnalité suivi par un appariement des *E/S* et des *P/E*. La méthode d'appariement de la fonctionnalité consiste en une approche logique où tous les supers et sous classes d'une fonctionnalité requise sont retournées. Pour la représentation sémantique des fonctionnalités de services ils utilisent le modèle de taxonomie proposé par l'ISO 19119 [6]. Pour l'appariement des données l'approche n'a pas été détaillée.

[37] présentent une approche pour la composition automatique des *SWG* dans une étude de cas liée à la sélection de sites pour s'abriter après un tremblement de terre. Pour la description des services, ils ont utilisé l'approche d'annotation *SAWSDL*, pour la découverte ils utilisent un appariement strict exact et pour la composition ils utilisent l'algorithme  $A^*$  et une approche de planification multi-agents.

[10] présentent une approche de composition semi-automatique des *SWG* basée sur une taxonomie spécifique pour l'analyse des impacts de l'élévation du niveau de la mer (*SLR*). Ils l'ont ensuite mise en œuvre en utilisant le plugin PROPHETS du framework jABC [108]. La solution comprend la description de services dans un langage propre à jABC. La définition des taxonomies a pour but d'ajouter une couche sémantique à la description de *SWG* et ainsi annoter la description de service avec des concepts en utilisant les termes définis dans la taxonomie. Pour la découverte de services ils spécifient des contraintes et

appliquent un appariement strict exact entre les concepts spécifiés dans les contraintes et les concepts qui annotent les descriptions de services.

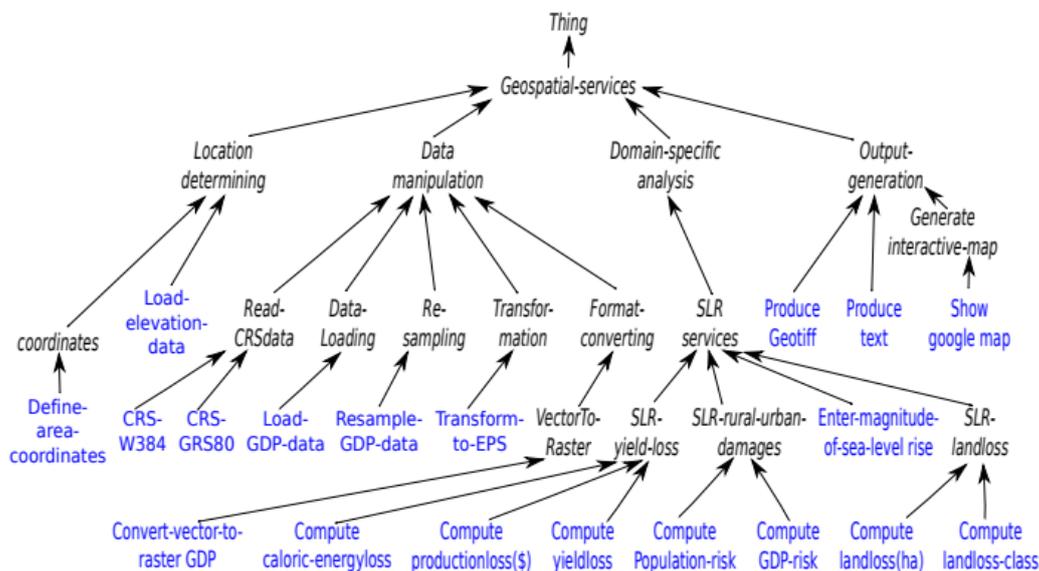


FIGURE 2.16 – Taxonomie de service *SLR* [10]

## 2.7 Synthèse

La description de services Web est une étape primordiale pour leur découverte. Dans un premier temps nous avons présenté les différents standards et styles architecturaux des services Web en général et des services Web géospatiaux en incluant les langages de description syntaxique de services Web les plus éminents (tels que WSDL, WADL ou *WPS*), qui jouent un grand rôle pour garantir une interopérabilité syntaxique. Cependant la découverte des services Web basée uniquement sur de tels descriptions consiste en une découverte syntaxique qui ne garantit pas la compréhension de la signification de l'information. De plus ce type de description ne permet pas la prise en considération des processus plus complexes comme celui de la composition.

La vision du Web sémantique vise à donner une signification à l'information sur le Web afin qu'elle puisse être compréhensible par la machine et l'homme. Ceci a incité les chercheurs des deux communautés du Web sémantique et des services Web à améliorer les descriptions de services Web en proposant des approches de description sémantiques de services Web. L'objectif est de les rendre plus compréhensibles par la machine et leur donner une signification bien définie afin de faciliter la découverte et la composition des services. Il existe deux types d'approches de description de services Web sémantiques. Les approches basées sur les langages sémantiques et les approches basées sur l'annotation sémantique des descriptions existantes. Les premières approches telles que *OWL-S* décrivent le service et la sémantique sous-jacente en se basant des langages sémantiques d'ontologie comme OWL. Les secondes approches ajoutent des annotations à des descriptions syntaxiques existantes telles que *SAWSDL* qui ajoute des annotations sémantique aux descriptions *WSDL*. Les approches à base d'annotation sont considérées comme des approches plus légères par rapports aux approches basées sur des langages. Les approches à base d'annotation n'exigent pas un grand changement des systèmes qui intègrent déjà

les modèles de description syntaxique existantes. Nous notons qu'à ce jour et pour le meilleur de notre connaissance, il n'existe pas de modèles de description sémantique spécifique aux standards de descriptions syntaxiques des services Web géospatiaux et plus précisément au standard *WPS*. Ainsi, les travaux académiques du domaine des services Web géospatiaux tendent à définir les services géospatiaux dans des langages tels que *OWL-S* ou *SAWSDL* en décrivant les services avec *WSDL*. Enfin, quels que soient leurs types, les principaux objectifs des approches de description sémantique de services Web, sont l'amélioration des processus de découverte et de composition automatique.

Dans ce contexte, des travaux ont proposé des outils pour la découverte des services Web sémantiques (*Matchmaker*) basés sur des approches d'appariement sémantique (*Matchmaking*). Les approches d'appariement sémantiques peuvent être classées en trois principales classes : les approches logiques, les approches non logique, et les approches hybrides. L'appariement sémantique basé sur la logique permet d'effectuer un raisonnement logique sur les descriptions de services. Le problème de l'utilisation unique d'une approche d'appariement logique est généralement sa tendance à générer des faux positifs puisque les résultats sont basés sur des filtres d'appariement avec des valeurs discrètes ou dans certains cas des faux négatifs quand ces filtres sont très rigides. L'utilisation unique d'une approche d'appariement non logique peut également générer des faux positifs ou des faux négatifs et cela dépend de la nature de l'approche utilisée puisque celles-ci sont très variées. Pour ces raisons, les *Matchmakers* les plus performants aujourd'hui ont tendance à proposer des approches hybrides pour remédier à certaines limites de chacune de ces deux classes grâce à différentes combinaisons hybrides qui réussissent là où chacune de ces deux approches échoue.

Dans le but de relever plus de défis dans le domaine des services Web, des approches de composition ont été proposées. Ces dernières sont distinguées selon le type de contrôle des services comme de l'orchestration ou de la chorégraphie de services Web, selon la gestion des services comme des compositions statiques ou dynamiques, et enfin selon le degré d'automatisation, comme des compositions manuelles, semi-automatiques ou automatiques. L'orchestration de services représente un processus centralisé unique tandis que la chorégraphie utilise une approche décentralisée et collaborative pour la composition des services. Les solutions de composition statiques, sont généralement choisies lorsque les services impliqués dans la composition et les exigences de composition sont statiques, c'est-à-dire qu'ils ne changent pas ou changent rarement, et aussi lorsque les services sont disponibles en permanence. Ainsi, les solutions de compositions statiques ne peuvent pas être utilisées dans un environnement flexible où ils ne sont pas adaptables aux changements tels que l'intégration de nouveaux services par les fournisseurs ou la mise-à-jour des informations concernant les services disponibles. Pour surmonter les problèmes qui se manifestent dans la composition statique, la composition de services dynamique intervient. Le but de la composition dynamique est de composer des services complexes à la volée, avec une prise en compte des mises-à-jour concernant les informations des services. La composition de services peut inclure un nombre important de services Web ce qui rend la tâche de composition des services de plus en plus difficile et peut prendre beaucoup de temps. Il convient de rappeler que pour construire une composition de service, il faut définir un schéma de composition qui spécifie le contrôle et le flux de données entre les différentes activités et attribue les services concrets aux activités de processus ou entre les différents services concrets directement. Le niveau d'automatisation dans la création du schéma de composition est une caractéristique importante d'une stratégie de composition de services. Concernant la composition automatique des services, l'état de l'art montre

certaines problématiques et limitations, notamment le besoin de flexibilité et le besoin d'adaptation aux contraintes utilisateurs. En effet, de nombreuses approches de composition existantes partent du principe que les modèles de composition sont prédéfinis. En outre, il n'est pas toujours possible de prédéfinir des schémas de processus vu que les besoins et contraintes utilisateurs peuvent changer. Il est donc nécessaire de générer de manière dynamique et automatique des schémas de composition adaptés aux contraintes de l'utilisateur.

Enfin, afin d'étudier l'avancée des travaux de recherche autour des *SWG* sémantiques concernant les différentes approches de description, de découverte et de composition spécifiquement dans le domaine géospatial, nous avons souligné dans un premier temps les problématiques des standards actuels des *SWG*, par la suite, nous avons passé en revue les travaux qui ont abordé ces problématiques. En se basant sur les modèles de description de services des standards actuels de l'*OGC*, la découverte et l'utilisation des *SWG* peuvent être limitées à de petits groupes d'utilisateurs. Cette découverte peut se faire soit de façon manuelle sous certaines hypothèses restrictives, soit recherchant les services en formulant des requêtes en utilisant des mots clés. Bien que les normes de l'*OGC* aient fait des progrès significatifs vers l'interopérabilité syntaxique des services et l'accès aux données, elles ne résolvent pas les problèmes d'hétérogénéité sémantique. Les normes ne fournissent pas de descriptions sémantiques interprétables par les machines. Ce manque de sémantique dans les descriptions des services est considéré comme une limitation primaire pour la découverte et la composition automatique de *SWG*. Une seconde problématique concernant les standards actuels des *SWG* services géospatiaux est l'absence de description des propriétés non fonctionnelles, également connues sous le nom d'attributs de *QoS* qui est essentiel dans la sélection et recommandation de *SWG* atomiques ou composites, notamment les propriétés non fonctionnelles spécifiques au domaine géospatial telles que la précision de coordonnées géographiques ou la fréquence de mise à jour des gazetier utilisés.

Des solutions issues du monde académique mettent au point des solutions relatives à ces problématiques. Dans le tableau 2.3, nous présentons l'évaluation sommaire des travaux existants les plus récents selon les critères suivants :

- Description sémantique, indique l'approche de description sémantique utilisée. Non spécifique au domaine géospatial pour les approches génériques utilisant les langages de description génériques, ou spécifique au domaine géospatial pour les approches qui utilisent des langages de description de *SWG*.
- Découverte sémantique, indique l'approche de découverte sémantique utilisée.
- Composition automatique, indique l'approche de composition automatique utilisée.
- Propriétés non fonctionnelles, indique la prise en compte de propriétés non fonctionnelles.

Les solutions existantes ont tendance à utiliser les approches de description sémantiques de services Web comme par exemple *OWL-S* ou *SAWSDL*. Cependant, le principal inconvénient de cette démarche est qu'elle nécessite le renoncement à l'utilisation du modèle de description syntaxique proposé par les standards comme *WPS* et la redéfinition des descriptions existantes dans d'autres langages ce qui risque d'être coûteux et difficile à mettre en œuvre. Une autre problématique est liée aux approches de découverte utilisées qui ont tendance à utiliser un appariement strict exact entre les éléments à appairer ou uniquement un appariement sémantique logique basé sur des filtres d'appariement. Ces méthodes restent naïves et n'exploitent pas convenablement les sémantiques des ser-

vices pour bien les exploiter dans le processus de découverte et de composition, de plus l'utilisation unique des filtres logiques peut tolérer un grand nombre de faux positifs au niveau du score de subsumption surtout si la chaîne de subsumption, reliant les deux concepts à comparer est longue. Concernant, les approches de composition utilisées, des efforts ont été fait pour automatiser les processus de composition en utilisant des approches de composition automatiques qui impliquent des structures de contrôle pour des exécutions séquentielles uniquement. Notamment, l'utilisation de l'algorithme A\* dans les travaux de [37] pour rendre le processus de recherche plus efficace par rapport aux travaux de [26] qui utilisent la méthode de chaînage arrière. Cependant, bien que les systèmes importent des descriptions sémantiques, les informations sémantiques fournies par les ontologies de domaine ne sont pas convenablement utilisées, le module de planification exige une correspondance exacte entre les éléments de description des services. De plus, peu de travaux dans le domaine géospatial utilisent dans leurs processus de découverte ou de composition de services les propriétés non fonctionnelles, et spécifiquement la non considération des attributs spécifiques au domaine. La prise en compte des propriétés non fonctionnelles a pour but d'aider l'utilisateur dans sa sélection et particulièrement quand plusieurs services ont des scores d'appariement fonctionnel similaires.

TABLE 2.3 – Études des approches de description, découverte et composition de *SWG*

Approche		[26] (2012)	[36] (2013)	[87] (2015)	[10] (2016)	[37] (2018)
Description sémantique	Non spécifique au domaine spatial	<i>OWL-S</i>	<i>WSMO</i>	<i>OWL-S</i>	jABC	<i>SAWSDL</i>
	Spécifique au domaine géospatial	-	-	-	-	-
Découverte sémantique		Appariement strict exact	Non spécifiée	Appariement logique avec deux filtres	Appariement strict exact	Appariement strict exact
Composition automatique		Automatique (chaînage arrière)	Automatique ( <i>PDDL</i> )	-	Semi-automatique (plugin PRO-PHETS)	Automatique (Algorithme A* et planification multi-agent)
	Propriétés non fonctionnelles au domaine géospatial	-	-	-	-	-
	Qualité de données géospatiales					

## 2.8 Conclusion

Dans ce chapitre, nous avons présenté plusieurs travaux existants dans la littérature, liés aux services Web en général et aux services Web géospatiaux en particulier. Dans un premier temps, nous avons étudié les standards des services Web, suivis par les approches de description, découverte et composition de services Web. Dans un deuxième temps, nous avons étudié des travaux récents ayant proposé des solutions pour la description, la découverte et la composition de *SWG*. Nous avons proposé une synthèse et un tableau qui résume les travaux étudiés et nous avons discuté leurs apports et manques, dans le but de préparer le lecteur à bien situer nos contributions.

**Deuxième partie**

**Contributions**



# Chapitre 3

## Méthodologie

### Sommaire

---

3.1	Contexte et objectifs . . . . .	61
3.2	Méthodologie et principes de la solution . . . . .	62
3.2.1	Vision d'ensemble de l'approche de description sémantique des <i>SWG</i>	63
3.2.2	Vision d'ensemble de l'approche de découverte sémantique de <i>SWG</i>	64
3.2.3	Vision d'ensemble de l'approche de composition automatique de services <i>SWG</i>	64

---

### Introduction

Ce chapitre introductif de la partie contributions a pour but de donner une vision d'ensemble des travaux présentés dans ce manuscrit. Dans un premier temps, nous présentons un bref rappel du contexte où s'inscrivent nos travaux et des objectifs à atteindre (section 3.1). Ensuite, nous donnons une vue d'ensemble de nos approches de description, de découverte et de composition de *SWG* (section 3.2).

### 3.1 Contexte et objectifs

Dans le contexte de l'intégration et l'exploitation des sources de données géospatiales hétérogènes et des ressources de traitements géospatiales, ces dernières sont mises à la disposition des utilisateurs via des services Web. L'absence de descriptions appropriées des *SWG* et de méthodes efficaces de découverte de ces *SWG* se traduit par une utilisation limitée, généralement uniquement par ceux qui ont une connaissance approfondie des ressources. Nos travaux visent à améliorer le processus de découverte de *SWG* qui présente certaines particularités, et limites qui sont :

- Les architectures géospatiales font la promotion de leurs propres formats et standards tel que le standard *WPS*, intégrant un modèle de description de services qui lui est propre.
- Les standards des *SWG* permettent l'interopérabilité syntaxique des *SWG*, en décrivant syntaxiquement que les propriétés fonctionnelles des *SWG* mais ne fournissent pas une description des propriétés non fonctionnelles (*QoS*).
- Les standards des *SWG* montrent une absence d'informations sémantiques nécessaires pour raisonner sur les services Web (par exemple, faire correspondre correctement les entrées et sorties).

- Les technologies de services Web sémantique actuelles sont indépendantes du domaine géospatial et ne prennent donc pas en charge les aspects spécifiques au domaine.
- Le domaine géospatial a son propre vocabulaire qui nécessite la conception d'ontologies spécifiques au domaine qui peuvent être utilisées pour décrire les SWG.
- L'absence d'informations sémantiques interprétables par la machine dans les langages de description de *SWG* et les schémas de classification utilisés contraignent les registres pour la prise en charge d'une découverte de service efficace, ou la proposition des solutions de composition automatisée.

Nos travaux visent à fournir un cadre général pour améliorer le processus de découverte de SWG tout en préservant la compatibilité avec les standards actuels de l'*OGC*, ainsi que l'évaluation des performances des solutions proposées. De ce point de vue, nous considérons trois objectifs de recherche (OR) :

- OR1- Proposer un modèle de description sémantique des propriétés fonctionnelles de *SWG* et prise en compte de la description des propriétés non fonctionnelles (*QoS*).
- OR2- Proposer une approche de découverte automatique en exploitant la sémantique des informations décrites.
- OR3- Proposer une approche pour la composition automatique de SWG en exploitant les approches de description et découverte proposées pour pallier à l'absence de services atomiques qui puissent répondre aux besoins d'un utilisateur.

Dans la section suivante, nous donnons une vision d'ensemble des solutions proposées, leurs objectifs ainsi que l'apport des techniques du Web sémantique utilisées.

## 3.2 Méthodologie et principes de la solution

En vue d'améliorer le processus de découverte de SWG, notre solution intègre, trois apports (voir figure 3.1). Le principal apport qui concerne la découverte sémantique de SWG nécessite une description sémantique de SWG. Ainsi, nous proposons un méta-modèle pour la description sémantique de SWG qui sera instancié comme extension du modèle *WPS*. Ensuite, cette description sémantique sera exploitée dans l'approche de découverte sémantique. Enfin, afin de proposer une solution alternative à l'utilisateur en cas d'absence de services atomiques satisfaisant la requête utilisateur dans la phase de découverte, nous proposons une approche de composition automatique de *SWG*.

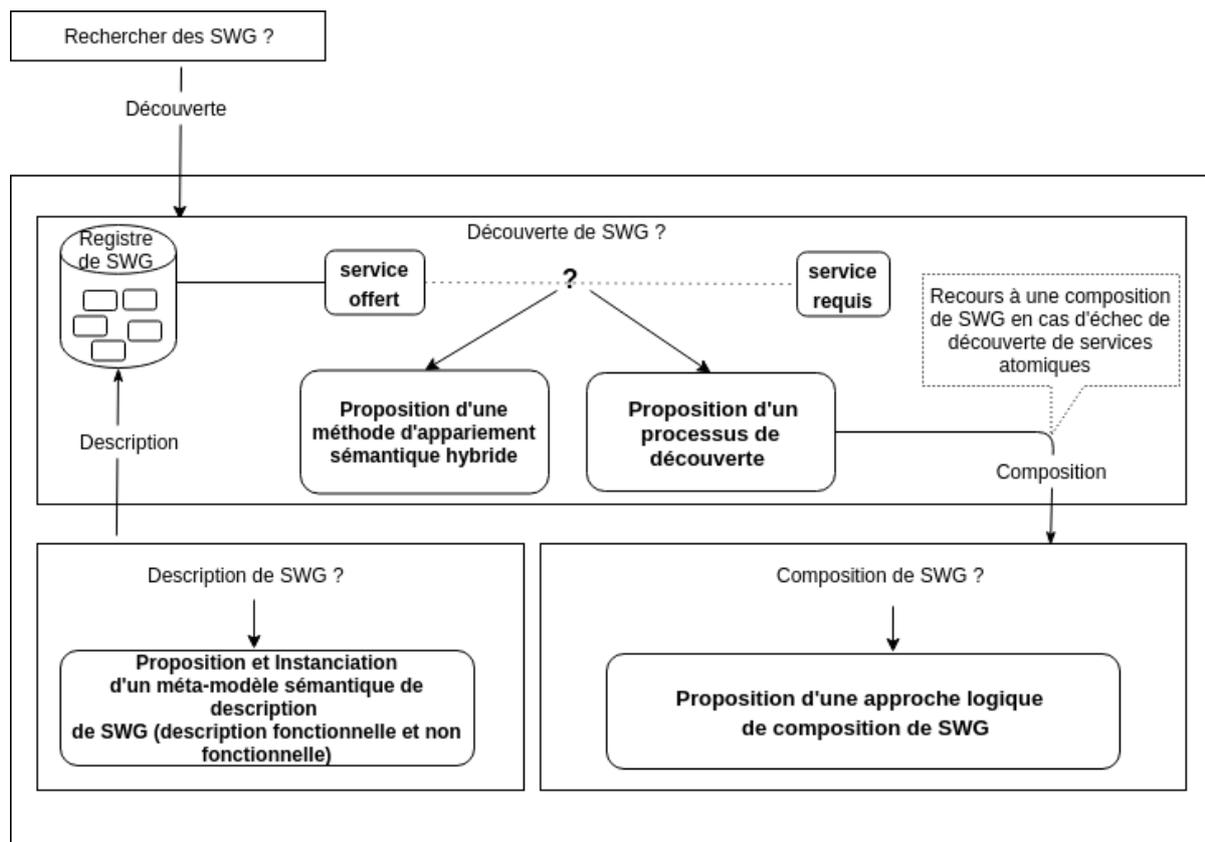


FIGURE 3.1 – Vue d'ensemble des approches

### 3.2.1 Vision d'ensemble de l'approche de description sémantique des SWG

Comme annoncé dans la partie description de *SWG* de la figure 3.1, l'apport concerne la proposition et l'instanciation d'un méta-modèle sémantique de description de *SWG*. Plus précisément, ceci concerne la (1) méta-modélisation, (2) la conception d'ontologies de domaine et (3) l'instanciation, comme suit :

**La méta-modélisation** : le but est de fournir un méta-modèle pour la description des *SWG* qui inclut la description des propriétés fonctionnelles des services géospatiaux d'un côté et la description de propriétés non fonctionnelles d'un autre. Précisément, nous utilisons les ontologies pour apporter une couche sémantique à la couche de description syntaxique proposée par les standards actuels des *SWG*, en se focalisant sur l'annotation sémantique de la fonctionnalité et des entrées/sorties des *SWG*. De plus dans ce méta-modèle nous avons pris en compte les propriétés non fonctionnelles pour la représentation de la qualité des *SWG*.

**La conception d'ontologies de domaine** : elle consiste à la conception de deux ontologies. La première est une ontologie qui définit les catégories de services géospatiaux et les relations entre eux et la deuxième catégorise les types de données d'*E/S* des services géospatiaux et les relations entre eux. Nous décrivons dans ce manuscrit la méthodologie adoptée ainsi que l'organisation des deux ontologies de services et de données.

**L'instanciation** : l'objectif est d'instancier le méta-modèle proposé dans la première étape, dans la version 2.0 du standard *WPS*. Précisément nous allons étendre le modèle de description de *process* proposé par le standard *WPS*, qui ne présente actuellement qu'une description syntaxique de la fonctionnalité et des paramètres d'*E/S* des services

*WPS*. L'instanciation dans la description *WPS* est réalisée à l'aide d'une approche par annotation.

À l'issue de cette phase, le but est de définir un nouveau modèle de description de *SWG*, appliqué à la norme *WPS*. Par conséquent, cette étape adresse le premier objectif de recherche (OR1) et est un élément clé pour l'amélioration du processus de découverte de services, elle permet alors de préparer un bon environnement pour proposer une approche de découverte sémantique afin d'adresser notre principal objectif (OR2).

### 3.2.2 Vision d'ensemble de l'approche de découverte sémantique de *SWG*

Afin d'adresser notre principal objectif d'amélioration du mécanisme de découverte des services *SWG*, nous proposons une méthode d'appariement sémantique hybride ainsi qu'un processus de découverte de *SWG* (voir figure 3.1). Concernant la méthode d'appariement sémantique hybride, le caractère d'appariement sémantique est lié à la prise en considération de la description sémantique de *SWG*, plus particulièrement de services *WPS*, et le caractère hybride est lié à la combinaison de méthodes d'appariement sémantique logique et d'une mesure de similarité non logique.

Cette méthode d'appariement sémantique hybride est exploitée dans les deux premières étapes du processus de découverte de *SWG* intégrant le calcul de scores fonctionnels. Les deux premières étapes consistent en un appariement fonctionnel, une première étape d'appariement de la fonctionnalité et une seconde étape d'appariement des *E/S*. La dernière étape consiste à l'intégration d'un score non fonctionnel de qualité de service.

Cette approche de découverte sémantique ou de matchmaking nous a permis de concevoir et réaliser un nouveau *Matchmaker* sémantique (appariereur sémantique) pour les services *WPS*. Ce *Matchmaker* sémantique est adapté aux services Web décrits en standard *WPS 2.0*.

Ainsi, cette phase s'adresse au deuxième objectif de recherche (OR2).

### 3.2.3 Vision d'ensemble de l'approche de composition automatique de services *SWG*

Le besoin de composition de services Web provient de la nécessité de satisfaire un besoin complexe d'un utilisateur recherchant une fonctionnalité pour laquelle aucun service atomique n'est disponible (voir figure 3.1). Dans ce contexte, cette phase vise à fournir une approche pour la composition automatique de services *WPS* sans états et qui ont été décrits sémantiquement selon l'approche de description proposée dans le cadre de cette thèse. L'approche consiste à associer le problème de composition de services Web au problème de planification en *IA*. Dans ce contexte, nous proposons une nouvelle solution pour la composition de services *WPS*, basée sur la planification en utilisant le langage de programmation logique par contraintes *ASP*[95]. Ceci nécessite la représentation des éléments de composition sous formes d'atomes et règles logiques pour les intégrer dans un programme logique *ASP*. Les solutions de la composition seront représentées sous forme d'ensembles d'atomes, appelés modèles stables [47]. Dans ce contexte pour la composition automatique de *SWG*, nous avons défini trois principaux éléments de la composition qui doivent être représentés en *ASP*, et une stratégie de sélection de compositions candidates :

Graphes de dépendance : une dépendance de service représente un moyen par lequel un service peut être composé avec un autre service en passant ses sorties aux entrées de l'autre service. Ce graphe est conçu en exploitant les approches de description et d'appariement sémantiques proposées des *SWG* disponibles.

Requête et contraintes utilisateur : ceci consiste en la sélection d'un sous-ensemble de services qui peuvent être utilisés en début et en fin d'une chaîne de services ainsi que des contraintes pour satisfaire des intentions particulières des utilisateurs.

Règles de composition : les règles décrivent la solution de composition dans le but de calculer les modèles stables représentant les différentes solutions de compositions candidates.

Stratégie de sélection : nous définissons une stratégie de sélection de composition de services candidates selon des critères fonctionnels et non fonctionnels.

Ainsi, la phase de composition automatique de services *WPS* répond au troisième objectif de recherche (OR3).



## Chapitre 4

# Description sémantique de *SWG*

### Sommaire

---

4.1	Terminologie, service et process . . . . .	68
4.2	Méta-modèle pour la description sémantique des services géospatiaux . . . . .	69
4.3	Ontologies de domaine : services et données géospatiales . . . . .	71
4.3.1	Ontologie de services géospatiaux . . . . .	72
4.3.2	Ontologie de données géospatiales . . . . .	73
4.4	Instanciation du méta-modèle par extension du modèle de description <i>WPS</i>	
2.0	. . . . .	74
4.4.1	Principe . . . . .	74
4.4.2	Définition des rôles . . . . .	75
4.4.3	Exemple de description . . . . .	76
4.5	Conclusion . . . . .	78

---

### Introduction

Les *SWG* sont basés sur des spécifications et des standards conçus pour être utilisés en tant que règles et lignes directrices. Le *OGC WPS*[104] est un standard qui fournit un modèle de description pour les processus de traitement des données géospatiales sur le Web (calcul d'itinéraire, profil d'élévation, etc.). Le standard *WPS* fournit une description syntaxique des propriétés fonctionnelles des services (informations de base sur la fonctionnalité, les entrées et sorties du service) assurant ainsi la prise en charge l'interopérabilité syntaxique des services. Ceci est nécessaire mais n'est pas suffisant pour une découverte plus précise de *SWG*. Deux limitations des normes actuelles peuvent être mises en évidence. La première est le manque de sémantique dans la description des propriétés fonctionnelles du service. Comme indiqué dans le chapitre de l'état de l'art, certaines propositions venant du monde de la recherche académique, ont utilisé des langages comme *OWL-S* pour décrire sémantiquement les *SWG*. Le principal inconvénient de cette démarche est qu'elle nécessite le renoncement à l'utilisation du modèle de description syntaxique proposé par les standards comme *WPS* et la redéfinition des descriptions existantes dans d'autres langages, ce qui risque d'être coûteux et difficile à mettre en oeuvre en raison des différentes correspondances de concepts à assurer. Une seconde limitation dans la description actuelle des *SWG* est la quasi-absence de description des propriétés non fonctionnelles également appelées attributs de *QoS*. Même lorsque celles-ci sont préconisées par des directives comme INSPIRE<sup>1</sup>, elles se limitent à des attributs

---

1. La directive INSPIRE vise à créer une infrastructure de données spatiales de l'Union européenne.

génériques tels que la disponibilité et la fiabilité et ne traitent pas de manière adéquate les attributs spécifiques au domaine, tel que la précision des coordonnées dans les bases de données de répertoires géographiques. En vue de nos objectifs pour l'amélioration de la découverte sémantique des *SWG* ayant besoin d'un modèle de description sémantique de *SWG*, nous considérons les limitations ci-dessus comme des obstacles. Ainsi, en premier lieu, il est important de faire évoluer le modèle de description des services *WPS* avec les technologies actuelles du Web sémantique (ontologie, annotation sémantique, etc.) offrant ainsi un support pour le codage de la sémantique des propriétés fonctionnelles dans la description du service. Et, en second lieu, il est nécessaire de proposer un modèle de description permettant la représentation des propriétés non fonctionnelles des services géospatiaux.

Dans ce chapitre nous présentons notre solution pour résoudre les problématiques liées aux descriptions syntaxiques de *SWG*. Elle consiste principalement en une approche à base d'annotations sémantiques qui permet la description sémantique des propriétés fonctionnelles de *SWG*. De plus nous proposons la prise en charge de la description des propriétés non fonctionnelles (QoS) de *SWG*. Nous estimons qu'une approche à base d'annotations est légère puisque elle permet d'ajouter de la sémantique sans passer par un nouveau langage de description, ainsi, elle incarne une solution adéquate pour réduire et maîtriser la complexité liée à la découverte de *SWG* et leurs compositions tout en respectant la compatibilité avec les standards actuels.

La première section présente des définitions autour des services *WPS*, essentiels à la compréhension des approches présentées. La seconde section présente un méta-modèle qui décrit une abstraction de notre approche d'annotation. Ce méta-modèle permet d'exprimer les éléments pertinents d'une description des *SWG*. Dans la troisième section nous présentons la méthodologie adoptée ainsi que l'organisation des deux ontologies de domaines conçues dans le cadre de l'annotation sémantique des *SWG*. Dans la quatrième section nous présentons l'instanciation du méta-modèle par extension du modèle de description *WPS* par des annotations sémantiques. La cinquième section conclut ce chapitre et introduit le prochain chapitre.

## 4.1 Terminologie, service et process

Dans le cadre de la présentation de notre solution appliquée aux services Web OGC, et plus précisément aux services *WPS*, nous allons présenter des définitions autour des services *WPS*. La norme *WPS* définit un modèle de processus (*Process*) conçu pour fournir une description syntaxiquement interopérable des fonctions de traitement géospatial. Un service fournit un ou plusieurs *Process*. Le modèle de *Process* a été conçu de sorte qu'il peut être utilisé dans des catalogues pour la découverte de services géospatiaux.

Voici quelques définitions concernant un process dans la norme *WPS 2.0* :

**Définition 7. *Process* :** *un Process est une fonction qui, pour chaque entrée, renvoie une sortie correspondante. Dans la spécification WPS, les arguments d'un Process sont appelés entrées de Process et les valeurs de résultat sont appelées sorties de Process.*

**Définition 8. *Description du process* :** *une description de Process est le modèle d'information qui spécifie l'interface d'un Process. Une description de Process est utilisée pour une description, lisible par la machine, du Process lui-même, mais fournit également quelques*

informations de base sur les entrées et les sorties du Process. Elle est utilisée pour enregistrer le service dans un catalogue et comme modèle pour la découverte des services géospatiaux.

**Définition 9. Modèle abstrait de process :** le modèle de Process abstrait spécifie les exigences génériques pour la description des Process. En d'autres termes il représente une abstraction de la description du Process.

Selon les définitions présentées ci-dessus, la description et la découverte des *SWG* sont centrées sur la description et la découverte des *Process*. Pour faciliter la lecture du document, nous allons alors, utiliser le terme *Process* géospatial et service Web géospatial indifféremment pour désigner un service Web qui fournit une opération de traitement géospatial simple ou complexe qui peut contenir plusieurs paramètres d'entrées ou de sorties.

Ainsi, l'apport à ajouter pour l'amélioration du mécanisme de découverte et de composition de services géospatiaux, se concentre sur l'amélioration du modèle abstrait de *Process* et de la description du *Process*.

## 4.2 Méta-modèle pour la description sémantique des services géospatiaux

La description sémantique des *SWG* permet d'interpréter le sens des traitements et données proposés par ces services pour mieux les exploiter. Ceci est possible grâce à la fourniture de descriptions formelles de la fonctionnalité et des *E/S* des *SWG* indiquant la signification des traitements et données proposés. De plus, la prise en compte des propriétés non fonctionnelles dans la description des *SWG* est importante pour la recommandation des *SWG* les plus appropriés.

Dans cette section nous présentons l'ensemble des notions et concepts constituant la nouvelle proposition de description des *SWG* à travers un méta-modèle (voir figure 4.1) sous forme d'un diagramme de classes UML.

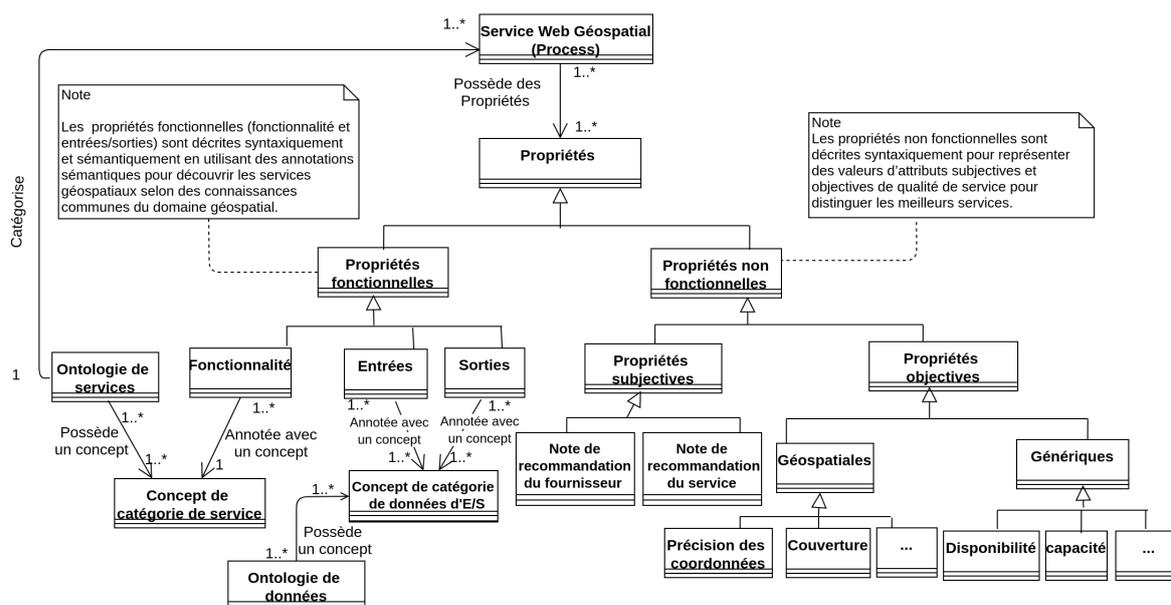


FIGURE 4.1 – Diagramme de classe UML décrivant le méta-modèle pour la description des services Web géospatiaux.

Le méta modèle proposé ( voir figure 4.1) met l'accent sur l'annotation sémantique de la description de *SWG*. Comme énoncé dans le schéma, chaque **Service Web géospatial (process)** possède des **propriétés** qui représentent la description des *SWG*. En particulier, nous avons deux types de propriétés : les propriétés fonctionnelles et les propriétés non fonctionnelles.

**Les propriétés fonctionnelles** représentent principalement les informations sur la fonctionnalité du service et les informations sur ses paramètres d'*E/S*.

- **La fonctionnalité** : la fonctionnalité du service sera annotée sémantiquement avec **un concept de catégorie de service** ou plus précisément d'un concept d'une **ontologie de services** géospatiaux (voir section 4.3.1). Cela permet d'une part de relier les différents services selon des catégories communes et d'autre part de raisonner sur ces connaissances pour améliorer le processus de découverte.
- **Les *E/S*** : les *E/S* seront annotées sémantiquement avec des **concepts de catégorie de données d'*E/S*** ou plus précisément d'un concept d'une **ontologie de données** géospatiaux que peuvent s'échanger les *SWG* (voir section 4.3.2).

**Les propriétés non fonctionnelles** sont divisées en deux catégories :

- **Propriétés Subjectives** : représentent la *QoS* en fonction des niveaux de satisfaction des utilisateurs. Dans cette perspective nous considérons deux notes de recommandation. La première est une **note de recommandation de service**, qui exprime les avis des utilisateurs sur le service. La seconde est une **note de recommandation de fournisseur** qui sera automatiquement calculée à partir des notes de tous les services fournis par ce fournisseur.
- **Propriétés Objectives** : représentent la *QoS* en fonction des propriétés **génériques** et des propriétés **géospatiales**. Les propriétés génériques représentent la *QoS* selon le respect des exigences générales en termes de disponibilité, de fiabilité, etc. Les propriétés géospatiales représentent la *QoS* selon le respect des exigences géospatiales. Elles diffèrent selon la catégorie de service considérée. Par exemple, pour les services de visualisation, la qualité du zoom est importante. Alors que pour les services de géocodage, la précision des coordonnées, la fréquence de mise à jour des répertoires géographiques utilisés, et la couverture sont des propriétés qui peuvent être prises en compte.

Dans la littérature [66, 102], nous trouvons divers attributs de *QoS*. Dans cette thèse, nous considérons 5 propriétés génériques de base présentées dans le tableau 4.1.

Concernant les propriétés objectives géospatiales, la spécification de toutes les propriétés spécifiques à chacune des catégories de services géospatiaux n'est pas traité dans le cadre de cette thèse. Cependant, nous nous intéressons à montrer l'intérêt de la prise en compte des propriétés objectives géospatiales quand celles-ci existent.

Nous pouvons citer par exemple, les propriétés spécifiques à la catégorie de géocodage. Celles-ci dépendent de la qualité des gazetiers utilisés pour le géocodage. Les travaux [8] se sont intéressés à la présentation de propriétés non fonctionnelles de quatre gazetiers et ont calculé les valeurs de ces propriétés. Le tableau 4.2 présente des exemples de propriétés non fonctionnelles. Celles-ci sont issues des travaux cités ci-dessus et peuvent être considérées comme propriétés non fonctionnelles spécifiques à la catégorie de géocodage.

TABLE 4.1 – Propriétés non fonctionnelles génériques

Propriété	Description de la propriété
<b>Disponibilité</b>	Limite inférieure du pourcentage de temps qu'un service Web est estimé être disponible sur une base annuelle.
<b>Performance</b>	Durée maximale pendant laquelle une demande au service peut être effectuée dans une situation de charge hors pointe.
<b>capacité</b>	Limite inférieure du nombre maximum de demandes simultanées qui peuvent être traitées dans les limites de la performance déclarée.
<b>Prix</b>	Gratuité ou coût d'un service Web qui peut être catégorisé en prix bas, moyen ou élevé par exemple selon la fonctionnalité offerte.
<b>Sécurité</b>	Sécurisation ou non sécurisation d'un service Web qui peut être catégorisé en sécurisation faible, moyenne ou élevé par exemple selon plusieurs critères comme l'utilisation d'une couche de chiffrement, l'authentification et le cryptage de données .

TABLE 4.2 – Propriétés non fonctionnelles spécifiques au géocodage.

Propriété	Description de la propriété
<b>Couverture</b>	Elle dépend de la base de données utilisée, si celle-ci inclut un champ communal, régional/national ou mondial.
<b>Fréquence de mise à jour</b>	Elle fait référence à la périodicité de mise à jour des sources utilisées. Par exemple, sans mise à jour, annuelle, mensuelle, hebdomadaire ou quotidienne.
<b>Précision des coordonnées</b>	Elle dépend du système de coordonnées utilisé. Selon les systèmes la précision peut être métrique, centimétrique ou décimétrique.
<b>Richesse des annotations</b>	Elle dépend de la quantité et détail des informations descriptives comme le nom et le type de l'annotation. L'annotation peut être très détaillée, moyennement détaillée, ou faiblement détaillée.
<b>Lineage</b>	Il dépend de la diversité des sources puisque les données peuvent provenir de diverses sources. Il peut y avoir une, deux ou plusieurs sources.

### 4.3 Ontologies de domaine : services et données géospatiaux

Dans le contexte de notre proposition pour la description sémantique des *SWG* et qui se base sur une approche d'annotation sémantique, les éléments de description seront annotés par des concepts d'ontologies. Les ontologies présentées dans le méta-modèle, en accord avec la typologie définie dans [49], sont considérées comme des ontologies de

domaine puisque elles décrivent un vocabulaire lié au domaine géospatial. Le rôle des ontologies de domaine est de fournir un ensemble structuré de concepts représentant un ensemble de connaissances du domaine géospatial ainsi que des relations entre ces concepts. Ces ontologies seront employées pour raisonner sur les requêtes utilisateurs et les descriptions de services géospatiaux pour améliorer les mécanismes de découverte et de composition.

Le méta-modèle sémantique présenté dans la section 4.2 expose l'annotation sémantique de la fonctionnalité d'un côté et l'annotation des *E/S* d'un autre.

Dans ce contexte, nous avons conçu deux ontologies. La première est une ontologie qui définit les catégories de services géospatiaux et les relations entre eux, et la deuxième qui catégorise les types de données d'*E/S* des services géospatiaux et les relations entre eux. Dans le cadre de cette thèse nous considérons exclusivement, les relations de sub-somption entre les concepts de chacune des ontologies. Nous décrivons dans ce qui suit la méthodologie adoptée ainsi que l'organisation des deux ontologies de services et de données.

### 4.3.1 Ontologie de services géospatiaux

Notre approche pour la conception de notre ontologie de services géospatiaux, vise à assurer un bon équilibre entre une catégorisation générique des *SWG* pour favoriser la réutilisation et une catégorisation spécifique pour répondre à différents besoins métiers incluant les besoins métiers dans le cadre du projet Choucas. La méthode suivie pour concevoir cette ontologie se résume en deux étapes principales :

1. Formaliser le vocabulaire de base (catégories génériques de services géospatiaux) proposées par la révision de 2016 de l'ISO 19119 : 2016.
2. Enrichir le vocabulaire avec de nouvelles catégories de services et les rattacher aux catégories génériques.

Plus en détails, la première étape identifie six principales catégories de services géospatiaux :

- **Services géographiques avec interaction de limite/interaction humaine** : les services géographiques portant sur la gestion des interfaces utilisateur et d'interaction avec des éléments comme les capteurs physiques.
- **Services de gestion des modèles/informations géographiques** : les services géographiques portant sur la gestion du développement, de la manipulation et du stockage de métadonnées, de schémas conceptuels et d'ensembles de données.
- **Services de gestion du workflow/des tâches géographiques** : les services géographiques portant sur la gestion des chaînes de services.
- **Services de traitement géographique** : les services géographiques de calcul et de traitement de données géospatiales.
- **Services de communication géographique** : les services géographiques portant sur le codage et le transfert de données sur les réseaux de communication.
- **Services de gestion de système géographique et de sécurité** : les services portant sur la sécurité des systèmes, des applications et des réseaux et également la gestion des comptes utilisateurs.

Dans la perspective d'étendre le vocabulaire de catégorisation des *SWG*, nous avons créé de nouvelles sous catégories identifiées à partir de différentes sources. En premier lieu, nous avons considéré des sous catégories suggérées dans la norme ISO 19119, en second lieu, nous avons créé de nouvelles sous catégories qui ont été identifiées pour les besoins du projet Choucas. En troisième lieu, nous nous sommes appuyés sur des travaux existants, et plus particulièrement ceux du projet 52 °North *WPS* [1], qui propose une liste de services de traitement géospatiaux. A partir des fonctionnalités de services proposées dans le cadre de ce projet, nous avons identifié de nouvelles sous catégories de services géospatiaux et nous les avons formalisées dans l'ontologie. Pour respecter la conformité à la norme ISO 19119, toute nouvelle catégorie de service a été associée comme sous-classe de l'une des six classes de catégories génériques ou à l'une de leurs sous-classes.

### 4.3.2 Ontologie de données géospatiales

Notre approche pour la conception de l'ontologie de catégories de données d'*E/S* de services géospatiales, vise à définir les différents concepts qui représentent la sémantique des données d'*E/S* de services géospatiaux. Contrairement à l'ontologie des services, il n'existe pas de standard de référence qui définit les principales classes génériques de types de données d'*E/S* de services géospatiales. Par conséquent, nous avons suivi le même processus que pour la spécification des catégories de *SWG*. Nous nous sommes inspirés des travaux existants dans la littérature, plus particulièrement ceux du projet 52 °North *WPS* [1], ainsi que d'autres travaux qui ont été réalisés dans le cadre du projet Choucas [42] basés sur le langage de description TEI[3], et enfin des travaux de [153] qui fournissent un formalisme ontologique du format Geography Markup Language (GML).

Dans un premier temps deux principales catégories de données d'*E/S* ont été identifiées et ensuite des sous-catégories ont été rattachées à celles-ci :

- **Géométrie** : la représentation des données d'*E/S* se fait par l'intermédiaire de la classe *Geometry*. Celle-ci contient trois autres sous-classes dépendantes du type géométrique. Par exemple, la sous-classe *GeometricPrimitive* contient différentes primitives géométriques, notamment : *Point*, *Curve*, *Solid* et *Surface*.
- **Analyse Spatiale** : elle contient plusieurs sous-classes qui représentent les différentes catégories de données d'*E/S* que les services d'analyse spatiale peuvent manipuler. Par exemple, la sous-classe *Distance* qui contient différents types de distances qui peuvent être fournis en *E/S* d'un service, notamment : *EuclideanDistance*. La sous-classe *SpatialAnnotation* qui contient différents types d'annotations spatiales qui peuvent être fournies en *E/S* d'un service, notamment : *SpatialNamedEntity* et *GeographicCoordinates*.

Bien que simple et représentant des catégories de haut niveau de données d'*E/S* géospatiales, cette ontologie nous a permis de représenter un ensemble d'éléments nécessaires à la description sémantique des données d'*E/S* de différents services géospatiaux qui seront utilisées dans le cadre de nos expérimentations. Cette ontologie est aussi ouverte à l'extension. Nous citons par exemple les travaux proposés par la norme ISO 19107 [5], qui peuvent être exploités pour étendre les sous catégories de la classe Géométrie.

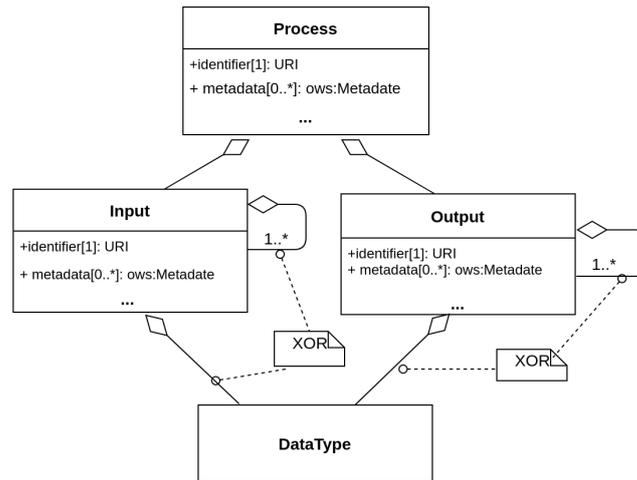


FIGURE 4.2 – Diagramme de classe UML décrivant une abstraction de la description du process WPS [104].

## 4.4 Instanciation du méta-modèle par extension du modèle de description WPS 2.0

Le modèle de description actuel des *SWG* proposé par la norme *WPS* (voir le modèle d’abstraction présenté dans la figure 4.2) est un modèle de description syntaxique et ne prend en charge que les propriétés fonctionnelles. Des efforts ont été faits dans la nouvelle version de cette norme pour proposer un nouveau concept ( *process concept*). Il s’agit d’un objet ajouté en tant que métadonnée de service pour fournir une documentation sur un groupe de services (lié à une page *HTML* ou à des formats multimédias similaires). Cependant, la proposition reste dans la notion de description syntaxique de la fonctionnalité de service, de plus elle est difficile à exploiter dans la perspective d’améliorer le processus de découverte, vis à vis de la notion de description sémantique précédemment introduite et plus particulièrement, la notion de raisonnement sur les concepts décrivant les fonctionnalités de *SWG*. Ainsi, dans cette section nous allons instancier le méta-modèle proposé (voir section 4.2) dans le modèle de description du *Process* de la norme *WPS 2.0*. Nous commençons par introduire le principe utilisé pour l’instanciation, ensuite, nous allons définir les nouveaux rôles proposés, suivi par une illustration de la nouvelle approche de description à travers un exemple.

### 4.4.1 Principe

Rappelons que selon le méta-modèle proposé, dans le cadre de description sémantique des propriétés fonctionnelles, la fonctionnalité du service sera annotée sémantiquement selon un concept d’une ontologie pour la catégorisation des services géospatiaux. Les *E/S* seront annotées sémantiquement selon les concepts de l’ontologie de données géospatiales. Aussi, dans le cadre de la prise en charge des propriétés non fonctionnelles de *SWG*, nous définissons une annotation faisant référence à la description de propriétés non fonctionnelles. Ainsi, dans le contexte de l’application de cette approche par annotation, sa mise en œuvre, est principalement basée sur l’utilisation de la balise *metadata*. Cette balise possède deux attributs : l’attribut *role* et l’attribut *href*. L’attribut *role*, indique le rôle de la métadonnée et l’attribut *href* référence la métadonnée avec le type de valeur *HTTP-URI*. Par conséquent, les annotations seront enregistrées dans la balise *metadata* sous

TABLE 4.3 – Rôles étendus pour la balise de métadonnées

Rôle	Définition	Exemple
<code>http://www.opengis.net/spec/wps/2.0/def/process/description/documentation</code>	Pour référencer toute documentation. Ce rôle existe déjà dans la norme WPS.	<code>&lt;ows:Metadata xlink:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation" xlink:href="http://erig.univ-pau.fr/PERDIDO/api.jsp"/&gt;</code>
<code>http://choucas.ign.fr/spec/wps/2.0/def/process-profile/concept_functionality</code>	Pour référencer un concept de catégorie de service (process) lié à l'un des concepts de l'ontologie de service. L'objectif est de fournir une référence de haut niveau sur un groupe générique de services (process) qui ont la même fonctionnalité.	<code>&lt;ows:Metadata xlink:role="http://choucas.ign.fr/spec/wps/2.0/def/process-profile/concept_functionality" xlink:href="http://choucas/ontologie_service/2018/6/untitled-ontology-6#Direct_geocoding_service"/&gt;</code>
<code>http://choucas.ign.fr/spec/wps/2.0/def/process-profile/concept_Input</code>	Pour référencer un concept de catégorie de données d'entrée, lié à l'un des concepts de l'ontologie de données.	<code>&lt;ows:Metadata xlink:role="http://choucas.ign.fr/spec/wps/2.0/def/process-profile/concept_Input" xlink:href="http://choucas/ontologie_data/2020/0/untitled-ontology-25#SpatialEntity"/&gt;</code>
<code>http://choucas.ign.fr/spec/wps/2.0/def/process-profile/concept_Output</code>	Pour référencer un concept de catégories de données de sortie, lié à l'un des concepts de l'ontologie de données.	<code>&lt;ows:Metadata xlink:role="http://choucas.ign.fr/spec/wps/2.0/def/process-profile/concept_Output" xlink:href="http://choucas/ontologie_data/2020/0/untitled-ontology-25#GeographicCoordinates"/&gt;</code>
<code>http://choucas.ign.fr/spec/wps/2.0/def/process-profile/Non-functional_properties</code>	Pour référencer les informations sur les propriétés non fonctionnelles liées au service.	<code>&lt;ows:Metadata xlink:role="http://choucas.ign.fr/spec/wps/2.0/def/process-profile/Non-functional_properties" xlink:href="http://choucas.ign.fr/NonFunctionalDescription/NFGetToponym.xml"/&gt;</code>

la forme de liens avec un identifiant de rôle approprié.

#### 4.4.2 Définition des rôles

Nous avons spécifié quatre nouveaux rôles (voir tableau 4.3<sup>2</sup>) et réutilisé un rôle proposé dans la norme (la première ligne du tableau).

La nouvelle proposition de description, étend et raffine la description de *Process* WPS grâce aux nouvelles annotations. Elle précise la nature des relations entre les concepts et les éléments annotés grâce à l'attribut *role*, c'est à dire la nature de l'annotation. Les quatre nouveaux rôles proposés vont annoter le document de description du *Process*. Ainsi, l'approche que nous proposons offre une expressivité de la description sémantique des propriétés fonctionnelles et une expressivité de la description des propriétés non fonc-

2. Les *HTTP-URI* présentés ne sont pas accessibles sur le web. Ils sont mentionnés à titre d'exemple à des fins d'illustration.

```

<wps:Process>
  <ows:Title>Geocoding</ows:Title>
  <ows:Abstract>This service returns the list of toponyms found in the input text with their geolocation. The input text must be
  annotated with spatial entities</ows:Abstract>
  <ows:Identifier>http://choucas.ign.fr/profileregistry/implementation/GetToponym</ows:Identifier>
  <ows:Metadata xlink:role="http://choucas.ign.fr/spec/wps/2.0/def/process-profile/concept_functionality" xlink:href="http://
  choucas/ontology_service/2018/6/untitled-ontology-6#Direct_geocoding_service" />
  <ows:Metadata xlink:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation" xlink:href="http://
  erig.univ-pau.fr/PERDIDO/api.jsp" />
  <wps:Input minOccurs="1">
    <ows:Title>Content spatial entities</ows:Title>
    <ows:Metadata xlink:role="http://choucas.ign.fr/spec/wps/2.0/def/process-profile/concept_input" xlink:href="http://choucas/
    ontologie_data/2020/0/untitled-ontology-25#SpatialEntity" />
    <ows:Abstract>Text with spatial entities</ows:Abstract>
    <ows:Identifier>content</ows:Identifier>
    <wps:ComplexData>
      <wps:Format mimeType="text/xml" encoding="UTF-8" schema="http://choucas.ign.fr/Schema-xml/Schema-NER.xsd" />
    </wps:ComplexData>
  </wps:Input>
  <wps:Output>
    <ows:Title>content coordinates</ows:Title>
    <ows:Metadata xlink:role="http://choucas.ign.fr/spec/wps/2.0/def/process-profile/concept_Output" xlink:href="http://
    choucas/ontology_data/2020/0/untitled-ontology-25#GeographicCoordinates" />
    <ows:Abstract>List of toponyms with geolocation (geographic coordinates)</ows:Abstract>
    <ows:Identifier>content</ows:Identifier>
    <wps:ComplexData>
      <wps:Format mimeType="text/xml" encoding="UTF-8" schema="http://choucas.ign.fr/Schema-xml/Schema-GetToponym.xsd"
      default="true" />
    </wps:ComplexData>
  </wps:Output>
  <ows:Metadata xlink:role="http://choucas.ign.fr/spec/wps/2.0/def/process-profile/Non-functional_properties"
  xlink:href="http://choucas.ign.fr/NonFunctionalDescription/NFGetToponym.xml" />
</wps:Process>

```

Annotation sémantique de la fonctionnalité de service

Annotation sémantique de l'entrée du service

Annotation sémantique de la sortie du service

Annotation pour référencer les propriétés non fonctionnelles

FIGURE 4.3 – Extrait d'une description de Process WPS étendue

tionnelles. L'approche est aussi ouverte à l'extension c'est à dire la possibilité d'ajout de nouveaux rôles en intégrant soit de nouveaux éléments de descriptions soit des concepts plus précis. Bien que dans l'esprit de conception des deux ontologies présentées dans la section 4.3 nous avons mis en avant le critère de généralisation pour la réutilisation de ces ontologies dans d'autres projets relatifs au domaine géospatial, l'approche proposée offre aussi une liberté sur le choix des ontologies. De même pour la description de propriétés non fonctionnelles, nous nous sommes basés sur un schéma recommandé par les directives d'INSPIRE [102] pour la description de ces propriétés afin de favoriser la réutilisation de ces descriptions, mais il est possible d'utiliser un autre schéma de description. Enfin, il existe un autre avantage, très important pour favoriser la réutilisation de cette approche, c'est le respect de la conformité à la norme *WPS*. Les extensions et les enrichissements proposés n'auront aucun impact sur la conformité à la norme ni sur le processus d'exploitation de la description syntaxique.

#### 4.4.3 Exemple de description

Dans le contexte du projet Choucas, l'une des tâches en cours est de proposer un processus de mise en correspondance automatique des traces, des textes de description de randonnées et des ressources géographiques (répertoires toponymiques et bases de données géographiques)[42]. L'accomplissement de cette tâche implique le développement de fonctionnalités comme l'identification d'entités spatiales et le géocodage qui seront exposées sous forme de services Web.

Conformément à notre proposition de description sémantique, nous allons maintenant présenter un exemple de description d'un service de géocodage illustré dans la figure 4.3. Comme mentionné auparavant, l'ensemble des annotations sont réalisées, en utilisant la balise *metadata* et ses deux attributs obligatoires *xlink:role* et *xlink:href*.

L'annotation sémantique de la fonctionnalité du service (c'est-à-dire le géocodage), a

été réalisée selon les concepts de l'ontologie de services. Comme le montre 4.3 et le détaille le tableau 4.4, l'attribut *xlink :role* indique le rôle de cette annotation en spécifiant qu'il s'agit d'un concept pour l'annotation de la fonctionnalité. L'attribut *xlink :href* référence le concept de l'ontologie de service (*Direct\_geocoding\_service*) auquel est apparié ce service.

TABLE 4.4 – Exemple d'annotation de la fonctionnalité

Nom de l'attribut	Valeur de l'attribut
<i>xlink :role</i>	<a href="http://choucas.ign.fr/spec/wps/2.0/def/process-profile/concept_functionality">http://choucas.ign.fr/spec/wps/2.0/def/process-profile/concept_functionality</a>
<i>xlink :href</i>	<a href="http://choucas/ontologie_service/2018/6/untitled-ontology-6#Direct_geocoding_service">http://choucas/ontologie_service/2018/6/untitled-ontology-6#Direct_geocoding_service</a>

L'annotation sémantique des *E/S* a été réalisée selon les concepts de l'ontologie de données. Comme le montre la figure 4.3 et le détaille les tableaux 4.5 et 4.6, l'attribut *xlink :role* indique le rôle de cette annotation en spécifiant qu'il s'agit d'un concept pour l'annotation de l'entrée ou de la sortie. L'attribut *xlink :href* référence le concept de l'ontologie de données auquel est apparié l'entrée ou la sortie du service. L'entrée du service de géocodage est appariée au concept *SpatialEntity* (un concept faisant référence aux entités spatiales) et la sortie du service de géocodage est appariée au concept *GeographicCoordinates* (un concept faisant référence aux coordonnées géographiques).

TABLE 4.5 – Exemple d'annotation de l'entrée d'un service

Nom de l'attribut	Valeur de l'attribut
<i>xlink :role</i>	<a href="http://choucas.ign.fr/spec/wps/2.0/def/process-profile/concept_Input">http://choucas.ign.fr/spec/wps/2.0/def/process-profile/concept_Input</a>
<i>xlink :href</i>	<a href="http://choucas/ontologie_data/2020/0/untitled-ontology-25#SpatialEntity">http://choucas/ontologie_data/2020/0/untitled-ontology-25#SpatialEntity</a>

TABLE 4.6 – Exemple d'annotation de la sortie d'un service

Nom de l'attribut	Valeur de l'attribut
<i>xlink :role</i>	<a href="http://choucas.ign.fr/spec/wps/2.0/def/process-profile/concept_Output">http://choucas.ign.fr/spec/wps/2.0/def/process-profile/concept_Output</a>
<i>xlink :href</i>	<a href="http://choucas/ontologie_data/2020/0/untitled-ontology-25#GeographicCoordinates">http://choucas/ontologie_data/2020/0/untitled-ontology-25#GeographicCoordinates</a>

En outre, comme le montre la figure 4.3 et le détaille le tableau 4.7, une annotation reliant le service à un fichier *XML* qui décrit les propriétés non fonctionnelles est fournie.

TABLE 4.7 – Exemple d'annotation de propriétés non fonctionnelles

Nom de l'attribut	Valeur de l'attribut
<i>xlink :role</i>	<a href="http://choucas.ign.fr/spec/wps/2.0/def/process-profile/Non-functional_properties">http://choucas.ign.fr/spec/wps/2.0/def/process-profile/Non-functional_properties</a>
<i>xlink :href</i>	<a href="http://choucas.ign.fr/spec/wps/2.0/def/process-profile/Non-functional_properties">http://choucas.ign.fr/spec/wps/2.0/def/process-profile/Non-functional_properties</a> " <i>xlink:href</i> ="http://choucas.ign.fr/NonFunctionalDescription/NFGetToponym.xml

Ce fichier décrit les propriétés non fonctionnelles selon un schéma recommandé par les directives d'INSPIRE pour la description d'attributs de *QoS* basé sur des métadonnées de

```

<gmd:report>
  <gmd:DQ_ConceptualConsistency>
    <gmd:nameOfMeasure>
      <gco:CharacterString>availability</gco:CharacterString>
    </gmd:nameOfMeasure>
    <gmd:measureIdentification>
      <gmd:MD_Identifier>
        <gmd:code>
          <gmx:Anchor
            xlink:href="http://inspire.ec.europa.eu/registry/metadata-codelist/
QualityOfServiceCriteriaCode/availability">availability</gmx:Anchor>
          </gmd:code>
        </gmd:MD_Identifier>
      </gmd:measureIdentification>
    <gmd:measureDescription>
      <gco:CharacterString>Lower limit of the percentage of time the service
is estimated to be available on a yearly basis</gco:CharacterString>
    </gmd:measureDescription>
    <gmd:result>
      <gmd:DQ_QuantitativeResult>
        <gmd:valueUnit xlink:href="urn:ogc:def:uom:OGC::percent"/>
        <gmd:value>
          <gco:Record xmlns:xs="http://www.w3.org/2001/XMLSchema"
            xsi:type="xs:double">100</gco:Record>
        </gmd:value>
      </gmd:DQ_QuantitativeResult>
    </gmd:result>
  </gmd:DQ_ConceptualConsistency>
</gmd:report>

```

FIGURE 4.4 – Extrait de la description des propriétés non fonctionnelles

service de l'ISO/TS 19139 :2007 [102]. La figure 4.4 montre un extrait de la description des propriétés non fonctionnelles. Plus précisément, elle présente la description de la disponibilité d'un service (*availability*). De la même façon nous avons décrit le reste des propriétés non fonctionnelles. La description inclut principalement, le nom de l'attribut, une description textuelle, l'unité de mesure (quand le type de résultat n'utilise pas d'unité de mesure, la valeur *unity* peut être utilisée), le type de valeur et la valeur de l'attribut.

## 4.5 Conclusion

Dans ce chapitre, nous avons présenté une approche de description sémantique des services géospatiaux appliquée à la norme *WPS*. Le principal objectif est l'intégration d'une couche sémantique dans la description de *SWG* en se basant sur une approche par annotation et l'utilisation d'ontologies de domaine.

Dans un premier temps, nous avons exposé l'ensemble des notions et concepts constituant les éléments pertinents de la nouvelle proposition de description des *SWG* à travers un méta-modèle (voir 4.1). Ce méta-modèle décrit une abstraction de notre approche de description. Le méta modèle proposé met l'accent sur l'annotation sémantique des propriétés fonctionnelles des *SWG* d'un côté et sur la description de propriétés non fonctionnelles des *SWG* d'un autre.

Dans un second temps, nous avons présenté la méthodologie adoptée pour la conception des deux ontologies de domaine conçues dans le cadre de l'annotation des propriétés fonctionnelles des *SWG*. La première ontologie dite "ontologie de service" présente des

concepts définissant les catégories de *SWG* et les relations de subsumption entre eux. La deuxième ontologie dite "ontologie de données", contient des concepts définissant la sémantique des *E/S* des services géospatiaux et les relations de subsumption entre eux.

Ensuite, nous avons présenté la méthodologie adoptée pour instancier le méta-modèle proposé dans le modèle de description de *Process* de la norme *WPS* 2.0. Dans ce contexte, nous avons commencé par expliquer le principe utilisé pour l'instanciation, suivi de la définition des nouveaux rôles proposés, et enfin une illustration de la nouvelle approche de description à travers un exemple.

La nouvelle proposition de description, étend et raffine la description des services *WPS* grâce aux annotations et précise la nature des relations entre les concepts et les éléments annotés dans une description *WPS*, c'est à dire la nature de l'annotation pour une bonne exploitation des métadonnées ajoutées. Elle offre une expressivité de la description sémantique des propriétés fonctionnelles et la prise en compte des propriétés non fonctionnelles. L'approche est aussi ouverte à l'extension c'est à dire la possibilité d'ajout de nouveaux rôles en intégrant soit de nouveaux éléments de descriptions soit des concepts plus précis. Enfin, il existe un autre avantage, très important pour favoriser la réutilisation de cette approche, c'est le respect de la conformité à la norme *WPS*. Les extensions et les enrichissements proposés n'auront aucun impact sur la conformité à la norme ni sur le processus d'exploitation de la description syntaxique (par exemple le Parsing des fichiers de description ou les outils d'invocation de services). Ceci est grâce à l'utilisation de la balise *metadata*, proposée par la norme *WPS*, pour l'ajout des annotations dans le contexte de la mise en œuvre de cette approche.

Dans le chapitre suivant, nous montrons comment la nouvelle approche de description ainsi que les ontologies conçues peuvent intervenir pour améliorer le processus de découverte des *SWG* et plus particulièrement des services *WPS*.



## Chapitre 5

# Découverte sémantique de *SWG*

### Sommaire

---

5.1	Méthode d'appariement sémantique hybride . . . . .	82
5.1.1	Méthode non-logique . . . . .	82
5.1.2	Méthode Logique . . . . .	84
5.1.3	Méthode hybride . . . . .	85
5.2	Processus de découverte en trois étapes . . . . .	86
5.2.1	Calcul de score d'appariement fonctionnel : fonctionnalité . . . . .	87
5.2.2	Calcul de score d'appariement fonctionnel : <i>E/S</i> . . . . .	88
5.2.3	Calcul de score non fonctionnel . . . . .	89
5.3	Conclusion . . . . .	90

---

### Introduction

De nombreuses études rapportent que la découverte des services géospatiaux est l'un des défis les plus immédiats et les plus contraignants [148, 27] indiquant que ce processus peut prendre jusqu'à 50% du temps de l'utilisateur [106, 147].

Le manque de sémantique dans les standards de description des *SWG* et la quasi-absence de la représentation des propriétés non fonctionnelles, également connues sous le nom d'attributs de *QoS*, contraint la découverte de *SWG*. Cette problématique a été soulevée et une proposition d'un nouveau modèle de description de *SWG* a été proposé (voir le chapitre 4). D'un autre côté, plusieurs études de la sémantique géospatiale ont mis en évidence le défi de l'amélioration du processus de découverte des *SWG* en intégrant de la sémantique. Au meilleur de nos connaissances, les approches de découverte proposées, se basent sur des technologies et langages de description sémantique génériques négligeant ainsi les spécificités de formats et standards tel que le langage *WPS* qui propose son propre modèle de description de *SWG*. De plus les techniques d'appariement (matching) sémantique utilisées dans ce contexte restent naïves et n'exploitent pas convenablement les sémantiques des *SWG*.

Les différentes limites citées nous ont orientés vers la recherche d'une solution permettant l'amélioration du processus de découverte des *SWG*.

Dans ce chapitre nous présentons notre approche pour la découverte sémantique des *SWG* [53]. L'approche exploite le modèle de description sémantique et comprends trois étapes (voir figure 5.1). Les deux premières étapes concernent le calcul de scores d'appariement sémantiques fonctionnels, et la troisième étape concerne le calcul d'un score non fonctionnel ou score de *QoS*.

L'appariement sémantique fonctionnel compare la description sémantique fonctionnelle d'un service requis (requête utilisateur) à celle d'un service offert, cela nécessite d'effectuer un appariement sémantique à chaque niveau d'éléments des descriptions. Dans la première étape d'appariement nous effectuons un appariement élémentaire entre fonctionnalité requise et offerte. Dans la deuxième étape d'appariement nous allons faire un appariement entre chaque entrée et sortie requise et offerte. L'ensemble des résultats de tous les appariements élémentaires des *E/S* peut ensuite être agrégé afin de calculer le degré global d'appariement sémantique (appariement entre l'ensemble des entrées et sorties requises et offertes). Dans ce cadre, nos travaux ont conduit à la proposition d'une approche d'appariement élémentaire et à une approche d'agrégation.

Pour la troisième étape de notre processus de découverte, nous avons proposé une fonction pour le calcul de score non fonctionnel. Cette fonction recherche parmi l'ensemble des services candidats sélectionnés dans les deux premières étapes d'appariement fonctionnel. Cette étape exploite les propriétés non fonctionnelles des services (proposées dans le méta-modèle de description de *SWG*).

Enfin, nous utilisons les scores fonctionnels et non fonctionnels pour recommander à l'utilisateur la sélection des services les plus appropriés. Ce chapitre est divisé en trois sections. Dans la première section nous présentons la méthode d'appariement sémantique hybride (logique et non logique) proposée. La deuxième section présente le processus de découverte et sélection de services divisée en trois sous sections que nous détaillerons ce processus. La troisième section conclue ce chapitre et introduit le prochain chapitre.

## 5.1 Méthode d'appariement sémantique hybride

Nous avons proposé une approche hybride dans le cadre de l'appariement élémentaire de la fonctionnalité et *E/S* de *SWG*.

La fonction proposée, appelée *SimCalc*, calcule la similarité entre deux concepts représentant les éléments de description de service et de requête en combinant une mesure de similarité non-logique et une méthode logique. Plus précisément nous allons nous baser sur l'utilisation de la mesure de similarité Wu et Palmer [149] combinée à des liens individuels logiques en prenant en compte la spécificité du type d'élément de description à appairer (fonctionnalité, entrée ou sortie).

### 5.1.1 Méthode non-logique

Wu et Palmer (*SimWP*) [149] est une mesure de similarité qui permet de calculer la similarité sémantique entre deux concepts dans une structure hiérarchique en se basant sur la profondeur de ces concepts ce qui lui donne une capacité à calculer la similarité sémantique des concepts de manière fidèle, c'est à dire en respectant la représentation hiérarchique des connaissances. Dans ce contexte, cette mesure peut être utilisée pour calculer la similarité sémantique entre deux concepts dans une ontologie en se basant sur la structure hiérarchique de l'ontologie. Elle permet d'estimer la similarité sémantique entre deux concepts par exemple  $C_1$  et  $C_2$  à partir de leurs positions. Plus précisément, en comparant la profondeur du plus petit subsumant commun  $CS$  de  $C_1$  et  $C_2$ . Elle est définie comme suit : considérons une ontologie  $\Omega$  formée par un ensemble de concepts et un concept racine  $R$ ,  $C_1$  et  $C_2$  représentent deux de ses concepts sur lesquels la similarité sera calculée. Le principe du calcul est basé sur les distances  $D_0$ ,  $D_1$  et  $D_2$ , séparant le concept racine  $R$ , le concept  $C_1$  et le concept  $C_2$  de l'ascendant commun le plus proche

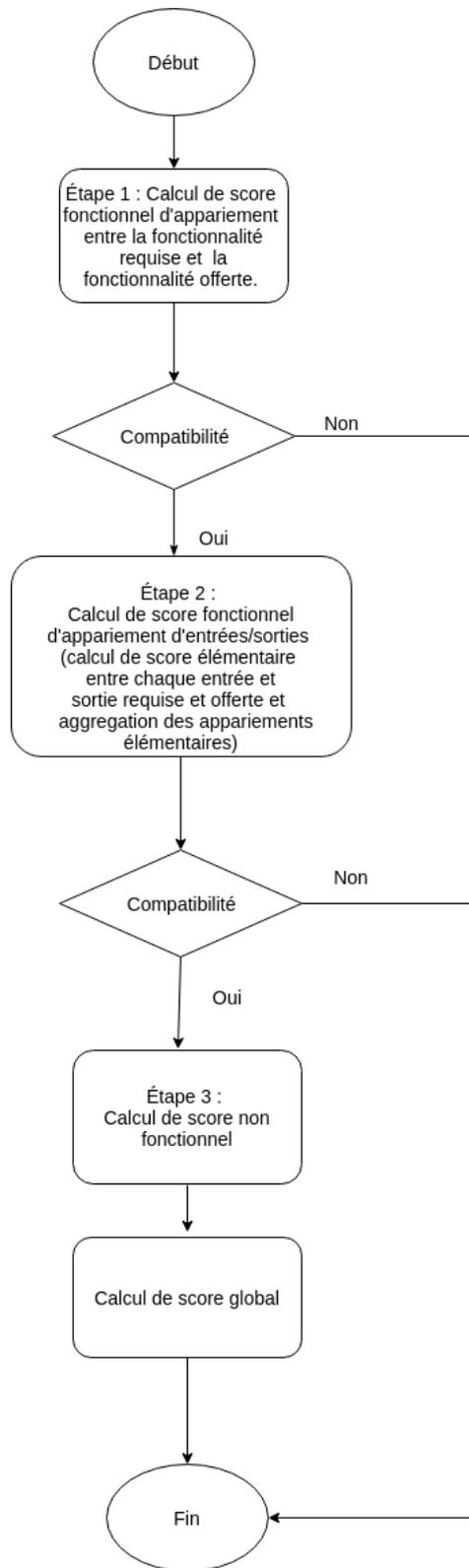


FIGURE 5.1 – Étapes de découverte

*CS* (voir figure 5.2). La mesure *SimWP* qui attribue un score  $\in [0, 1]$  est définie comme suit (équation 5.1) :

$$SimWP(C_1, C_2) = \frac{2.D_0}{D_1 + D_2 + 2.D_0} \quad (5.1)$$

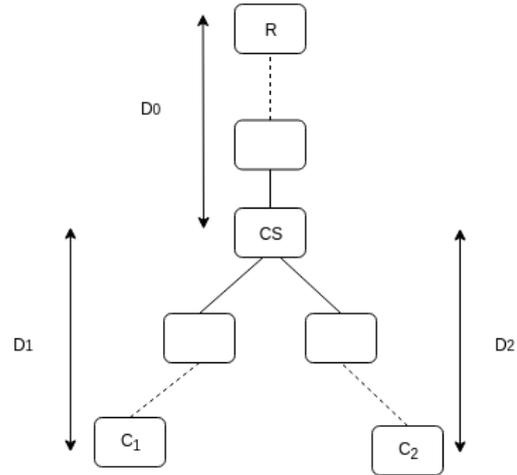


FIGURE 5.2 – Extrait d'une ontologie

Cette mesure ne peut pas être utilisée directement pour faire correspondre des services Web puisqu'elle est symétrique ( $SimWP(C_1, C_2) = SimWP(C_2, C_1)$ ). Les types de concepts à faire correspondre doivent être différenciés selon le rôle du concept dans la description du service (la fonctionnalité, l'entrée ou la sortie). Par exemple si  $C_1 = SpatialEntity$  et  $C_2 = SpatialNamedEntity$  et  $C_2$  est une sous classe de  $C_1$ . Si dans une correspondance  $C_1$  correspond au concept d'entrée du service et  $C_2$  correspond au concept d'entrée de la requête, et dans une autre correspondance  $C_1$  correspond au concept d'entrée de la requête et  $C_2$  correspond au concept d'entrée du service, on ne peut pas attribuer un score similaire à  $SimWP(C_1, C_2)$  et  $SimWP(C_2, C_1)$ . Une entrée de service doit être plus générique qu'une entrée d'une requête. De plus cette mesure peut attribuer le même score ou donner la priorité à des concepts voisins (concepts ayant le même ascendant), par rapport aux concepts appartenant à la même hiérarchie (une classe de concepts et ses sous-classes). Pour éviter cet effet, nous proposons de combiner cette mesure avec une méthode d'appariement logique.

### 5.1.2 Méthode Logique

Les méthodes logiques ou le matchmaking basé sur la logique a été utilisé dans divers travaux par plusieurs études de recherche pour vérifier si les paramètres d'*E/S* d'un service sont compatibles avec les paramètres d'*E/S* d'une requête [70]. Une approche commune de l'appariement basée sur la logique consiste à définir un ensemble de règles (filtres) qui déterminent quels types de relations logiques sont acceptables entre les paramètres d'*E/S* d'un service et les paramètres *E/S* d'une requête [73]. Ce type d'appariement tient compte de l'ensemble de la signature d'*E/S*, de sorte que le degré d'appariement entre un service et une requête ne peut être calculé. Une approche plus souple que les filtres de correspondance rigides (précédemment évoqués) est nécessaire pour pouvoir évaluer le degré de correspondance entre un service et une requête. Pour ces raisons, nous proposons une méthode d'appariement logique basée sur des liens individuels entre les paramètres du service et de la requête. Pour la création des liens proposés, nous nous sommes inspirés des travaux de [82, 20]. Les liens proposés seront utilisés à des fins d'appariement élémentaire de la fonctionnalité et des *E/S* de services géospatiaux et seront combinés à la mesure

*SimWP* en prenant en considération le type d'élément de description à appairer dans la phase d'appariement élémentaire et dans la phase d'agrégation.

Étant donné une ontologie  $\Omega$  formée par un ensemble de concepts, le lien logique  $LogLink(SP, RP)$  entre un concept de paramètre de service  $SP$  et un concept de paramètre de requête  $RP$ , peut appartenir à l'une des cinq catégories de filtres détaillées dans le tableau 5.1.

TABLE 5.1 – Catégories de filtres logiques

Nom	Définition	Rep. Formelle
<i>Exact</i>	si la RP et SP sont deux concepts équivalents	$\Omega \models RP \equiv SP$
<i>Plug – In</i>	si RP est une sous-classe de SP	$\Omega \models RP \sqsubseteq SP$
<i>Subsumes</i>	si RP est une super-classe de SP	$\Omega \models RP \sqsupseteq SP$
<i>Fail</i>	si RP et SP sont tous les deux spécifiés mais qu'aucun filtre ne peut être déterminé parmi ceux mentionnés ci-dessus	$\Omega \models SP \cap RP \sqsupseteq \perp$
<i>Unknown</i>	si RP ou SP ou les deux ne sont pas spécifiés	Aucune

Les filtres sur des liens individuels entre les différents éléments de description, permettent de calculer la similarité entre un service et une requête d'une manière plus flexible et plus précise par rapport aux méthodes logiques qui définissent des filtres sur l'ensemble de la signature du service. Cependant ces filtres peuvent tolérer un grand nombre de faux positifs au niveau du score de subsumption surtout si la chaîne de subsumption, reliant les deux concepts à comparer, est longue. L'utilisation de ces nouveaux filtres combinée à *SimCalc* permet d'attribuer un score numérique qui représente la similarité sémantique entre deux concepts en respectant la logique définie par ces filtres selon le type d'élément de description dont on va calculer la similarité.

### 5.1.3 Méthode hybride

Afin de calculer le score de correspondance entre un paramètre  $SP$  et un paramètre  $RP$ , nous présentons ci-dessous la méthode d'appariement hybride qui combine les deux méthodes précédentes (logique et non logique). Nous définissons *SimCalc* (voir l'équation 5.2) une fonction dont le résultat est un score  $\in [0, 1]$  obtenue après l'évaluation de deux autres fonctions (*LogLink* et *SimWP*) et en prenant en compte le type d'élément de description dont on va calculer le score (fonctionnalité-fonctionnalité (*func – func*), entrée-entrée (*in – in*), ou sortie-sortie (*out – out*)).

$$\text{SimCalc}(\text{LogLink}(SP, RP)) = \begin{cases} 1; & \text{si } \text{LogLink}(SP, RP) = \text{Exact} \\ \text{SimWP}(SP, RP); & \text{si } (\text{LogLink}(SP, RP) = \text{Plug} - \text{In}) \\ & \text{et } (\text{in} - \text{in}) \\ \text{SimWP}(SP, RP); & \text{si } (\text{LogLink}(SP, RP) = \text{Subsumes}) \\ & \text{et } ((\text{out} - \text{out}) \text{ ou } (\text{func} - \text{func})) \\ 0; & \text{si } (\text{Loglink}(SP, RP) = \text{Fail}) \\ & \text{ou } (\text{Loglink}(SP, RP) = \text{Unknown}), \\ & \text{ou aucune de ces conditions n'est observée.} \end{cases} \quad (5.2)$$

Le score attribué à un type (*func – func*) ou (*out – out*) dépend de la requête. Nous prenons en compte les fonctionnalités et les sorties des services qui sont identiques ou plus spécifiques que ceux mentionnés dans la requête. Par conséquent, la méthode n'accepte que les filtres *exact*, et *subsumes* pour l'appariement de ces paramètres. Inversement, puisque le score attribué au type (*in – in*) dépend du service. Nous considérons les entrées de service qui sont identiques ou plus génériques que celles spécifiées dans la requête. Par conséquent, la méthode n'accepte que les filtres *exact* et *plug – in*.

## 5.2 Processus de découverte en trois étapes

Le processus de découverte, nécessite la formalisation de la description de service et celle de la requête utilisateur.

**Définition 10.** *Un service géospatial  $S$  est représenté par un quadruplet  $(FS, ES, SS, QS)$*

- *$FS$  : représente la fonctionnalité du service et est associée à un concept (ou classe) de notre ontologie de services géospatiaux.*
- *$ES$  : représente une liste ordonnée d'entrées du service  $[ES_1, ES_2, \dots, ES_n]$  où chaque entrée est rattachée à un concept de l'ontologie de données.*
- *$SS$  : représente une liste ordonnée de sorties du service  $[SS_1, SS_2, \dots, SS_m]$  où chaque sortie est rattachée à un concept de l'ontologie de données.*
- *$QS$  : est un couple  $(ObjP, SubP)$  qui représente deux listes de valeurs de propriétés non fonctionnelles objectives et subjectives.*
  - *$ObjP$  : est une liste ordonnée de valeurs de propriétés non fonctionnelles objectives  $[ObjP_1, ObjP_2, \dots, ObjP_n]$ .*
  - *$SubP$  : est une liste ordonnée de valeurs de propriétés non fonctionnelles subjectives  $[SubP_1, SubP_2, \dots, SubP_m]$ .*

**Définition 11.** *Une requête utilisateur  $R$  est représentée par le triplet  $(FR, ER, SR)$*

- *$FR$  : représente la fonctionnalité du service recherchée et est associée à un concept (ou classe) de notre ontologie de services géospatiaux.*
- *$ER$  : représente une liste ordonnée d'entrées fournies par l'utilisateur  $[ER_1, ER_2, \dots, ER_p]$  où chaque entrée est rattachée à un concept de l'ontologie de données.*
- *$SR$  : représente une liste ordonnée de sorties recherchées  $[SR_1, SR_2, \dots, SR_q]$  où chaque sortie est rattachée à un concept de l'ontologie de données.*

Le processus de découverte et sélection que nous avons proposé passe par trois étapes de la manière suivante : les deux premières étapes consistent en un calcul de scores d'appariement fonctionnel (une étape d'appariement de la fonctionnalité et une étape d'appariement des  $E/S$ ). La dernière étape consiste à un calcul d'un score non fonctionnel de qualité de service pour enfin calculer un score global pour la sélection des services les plus appropriés. Par exemple, si la requête consiste à la recherche d'un service de géocodage avec une entité spatiale comme entrée du service et des coordonnées géographiques en sortie du service. La première étape consiste au calcul du score d'appariement fonctionnel entre le concept de la fonctionnalité *Direct Geocoding service* et le concept de la fonctionnalité de chacun des services de notre catalogue. La deuxième étape consiste au calcul de scores d'appariement fonctionnel entre le concept d'entrée *SpatialEntity*, le concept de sortie *GeographicCoordinates* et le(s) concept(s) d'entrée(s), le(s) concept(s) de sortie(s) de chacun des services sélectionnés lors de la première étape. La dernière étape consiste à un calcul de score non fonctionnel de qualité de service de chacun des services sélectionnés lors des deux étapes précédentes suivi par le calcul du score global. Deux exemples, plus complets seront présentés dans la section 7.2. Pour mieux expliquer ce processus de découverte (voir algorithme 1), les trois prochaines sous sections décrivent en détail chaque étape du processus.

---

**Algorithme 1** Calcul de score global pour la découverte d'un service Web géospatial

---

**Entrée:** R : Requête, S : Service

**Sortie:** *SelScore* : Score global pour la découverte d'un service Web géospatial

```

1: SelScore  $\leftarrow$  0;
2: FonctionnalitéScore  $\leftarrow$  SimCalc(FS, FR);
3: si (FonctionnalitéScore  $\geq$  SeuilFonctionnalité) alors
4:   FonScore  $\leftarrow$  (EMatch(ES, ER) + SMatch(SS, SR))/2;
5:   si (FonScore  $\geq$  SeuilES) alors
6:     NonFonScore  $\leftarrow$   $\alpha \cdot (SUM(ObjP)/|ObjP|) + (1 - \alpha) \cdot (SUM(SubP)/|SubP|)$ ;
7:     SelScore  $\leftarrow$   $\beta \cdot FonScore + (1 - \beta) \cdot NonFonScore$ ;
8:   fin si
9: fin si
10: retourne SelScore;

```

---

### 5.2.1 Calcul de score d'appariement fonctionnel : fonctionnalité

Dans la première étape, nous allons procéder à l'appariement élémentaire entre la fonctionnalité de service recherché  $FR$  et chacune des fonctionnalités  $FS$  des services offerts. Seuls les services dont le score est égal ou supérieur à un seuil donné *SeuilFonctionnalité*  $\in [0, 1]$  sont retenus. L'appariement est basé sur la fonction *SimCalc* proposée dans la section 5.1 avec la restriction de type d'élément à appairier (*func* – *func*). Par exemple, si le concept *Geocoding service* est demandé, toutes les instances de service annotées avec un concept qui a un score d'appariement  $\geq$  *SeuilFonctionnalité* avec le concept *Geocoding service* sont retenues à cette étape. La valeur du seuil peut être choisie parmi trois catégories (par exemple, un seuil d'acceptation strict pour *SeuilFonctionnalité* = 1 c'est à dire avec une compatibilité exacte, un seuil d'acceptation moyen pour *SeuilFonctionnalité* = 0.5 c'est à dire avec une compatibilité entre 0.5 et 1 et un seuil d'acceptation flou pour *SeuilFonctionnalité* = 0 c'est à dire large avec une compatibilité entre 0 et 1).

Cette étape vise à exclure rapidement un grand nombre de services dont la fonctionnalité offerte ne correspond pas à la fonctionnalité recherchée et ainsi minimiser les faux

positifs et augmenter la précision de recherche.

### 5.2.2 Calcul de score d'appariement fonctionnel : $E/S$

Dans la deuxième étape nous allons procéder à l'appariement des ensembles d' $E/S$   $((ES, ER), (SS, SR))$  de la requête et la liste des services offerts sélectionnés dans la première étape.

En premier nous allons procéder à un appariement élémentaire entre chaque paire d'entrées  $(ES_i, ER_j)$ , et chaque paire de sorties  $(SS_l, SR_k)$  des ensembles  $((ER, ER), (SS, SR))$ . L'appariement est basé sur la fonction *SimCalc* proposée dans la section 5.1 avec la restriction (*in – in*) ou (*out – out*) selon le type de paramètre à appairer.

L'ensemble des résultats de tous les appariements élémentaires des entrées et des sorties peut ensuite être agrégé afin de calculer le degré global d'appariement sémantique (appariement entre l'ensemble des entrées et sorties).

Nous proposons une fonction qui optimise l'agrégation en calculant le maximum sur les résultats d'appariement élémentaires et en proposons une méthode différente pour l'agrégation partielle des entrées à celle des sorties.

Le score attribué lors de l'agrégation partielle des résultats d'appariement élémentaire sur les ensembles d'entrées  $(ES, ER)$  est basé sur le nombre d'entrées du service. La priorité ici est de s'assurer que le plus grand nombre d'entrées requises pour le service doivent être satisfaites et ça reste acceptable si l'une des entrées spécifiées par la requête n'est pas utilisée. Par conséquent, nous additionnons les scores maximaux d'appariement élémentaire pour chaque concept d'entrée du service et nous divisons le résultat par le nombre de paramètres d'entrées requises par le service (voir l'équation 5.3b). Inversement, le score attribué lors de l'agrégation partielle des résultats d'appariement élémentaire sur les ensembles de sorties  $(SS, SR)$  est basé sur le nombre de sorties spécifiées dans la requête. La priorité ici est de s'assurer qu'un service offre un grand nombre de sorties requises par la requête. Dans les cas où le service génère des sorties supplémentaires, cela reste acceptable. Par conséquent, le score maximal de chaque appariement élémentaire des sorties de la requête est additionné puis divisé par le nombre de sorties requises par la requête (voir l'équation 5.3c). Enfin pour le calcul du score d'appariement fonctionnel global d' $E/S$  entre le service  $S$  et la requête  $R$  nous additionnons le score d'appariement des entrées et le score d'appariement des sorties et nous le divisons par 2 (voir l'équation 5.3a).

$$FonScore(S, R) = \frac{EMatch(ES, ER) + SMatch(SS, SR)}{2} \quad (5.3a)$$

$$EMatch(ES, ER) = \frac{\sum_{i=1}^n \max_{j=1}^m (SimCalc(LogLink(ES_i, ER_j)))}{n} \quad (5.3b)$$

$$SMatch(SS, SR) = \frac{\sum_{k=1}^p \max_{l=1}^q (SimCalc(LogLink(SS_l, SR_k)))}{p} \quad (5.3c)$$

Où :

$\{FonScore(S, R) \in \mathbb{R} \mid 0 \leq FonScore(S, R) \leq 1\}$  est le score d'appariement fonctionnel global entre  $S$  et  $R$

$\{EMatch(ES, ER) \in \mathbb{R} \mid 0 \leq EMatch(ES, ER) \leq 1\}$  est le score d'appariement fonctionnel entre  $ES$  et  $ER$  de  $S$  et  $R$

$\{SMatch(SS, SR) \in \mathbb{R} \mid 0 \leq SMatch(SS, SR) \leq 1\}$  est le score d'appariement fonctionnel entre  $SS$  et  $SR$  de  $S$  et  $R$

Comme à l'étape précédente, seuls les services dont le score d'appariement est égal ou supérieur à un seuil  $SeuilES \in [0, 1]$ , sont retenus. La valeur du seuil peut être choisie parmi les trois catégories (strict pour  $SeuilES = 1$ , moyen pour  $SeuilES = 0.5$  et flou pour  $SeuilES = 0$ ).

### 5.2.3 Calcul de score non fonctionnel

La troisième étape, étend le processus de découverte des services  $WPS$ , en introduisant un score de qualité de service en exploitant les valeurs de propriétés non fonctionnelles. Afin d'atteindre cet objectif, nous avons proposé une fonction de calcul de ce score. Cette fonction est appliquée sur l'ensemble des descriptions de services candidats sélectionnées dans les deux premières étapes et utilise les valeurs de propriétés non fonctionnelles proposées dans le méta-modèle de description de service (voir la section 4.2).

L'équation 5.4 calcule la valeur du score non fonctionnel des services candidats. Nous utilisons la pondération  $\alpha \in [0, 1]$  afin d'augmenter ou de diminuer la considération du score de propriétés objectives et subjectives.

$$NonFonScore(S) = \alpha \cdot \frac{\sum_{i=1}^n ObjP(i)}{n} + (1 - \alpha) \frac{\sum_{j=1}^m SubP(j)}{m} \quad (5.4)$$

Où :

$\{NonFonScore(S) \in \mathbb{R} \mid 0 \leq NonFonScore(S) \leq 1\}$  est le score non fonctionnel du service  $S$   
 $\{ObjP(i) \in \mathbb{R} \mid 0 \leq ObjP(i) \leq 1\}$  est la valeur de la propriété objective  $i$   
 $\{SubP(j) \in \mathbb{R} \mid 0 \leq SubP(j) \leq 1\}$  est la valeur de la propriété subjective  $j$   
 $\{\alpha \in \mathbb{R} \mid 0 \leq \alpha \leq 1\}$  est la valeur de pondération

Enfin, une fois que le score fonctionnel et le score non fonctionnel des services candidats ont été calculés, les valeurs d'évaluation finale pour la découverte des services sont déterminées. Le calcul de la valeur d'évaluation finale que nous appelons score d'aide à la sélection ( $SelScore$ ) utilise à la fois le score fonctionnel et le score non fonctionnel comme décrit dans (équation 5.5). Dans cette équation, nous utilisons la pondération  $\beta \in [0, 1]$  afin d'augmenter ou de diminuer la considération du score fonctionnel et non fonctionnel. La pondération permet de donner à l'utilisateur la possibilité de paramétrer le score final puisque l'intérêt des utilisateurs concernant particulièrement l'aspect non fonctionnel est différent, certains préfèrent le prendre en compte et d'autres non.

$$SelScore(S) = \beta.FunScore(S, R) + (1 - \beta)NonFunScore(S) \quad (5.5)$$

Où :

$\{SelScore(S) \in \mathbb{R} \mid 0 \leq SelScore(S) \leq 1\}$  est le score de sélection pour le service  $S$

$\{\beta \in \mathbb{R} \mid 0 \leq \beta \leq 1\}$  est la valeur de pondération

La prise en compte des propriétés non fonctionnelles, même de manière élémentaire, apporte une réelle valeur ajoutée dans la proposition finale pour la sélection des services découverts. L'ensemble des critères considérés permettent à l'utilisateur d'effectuer un choix lorsque plusieurs services Web sont découverts, ce qui permet de mieux satisfaire les attentes du client.

### 5.3 Conclusion

Dans ce chapitre nous avons présenté une approche pour la découverte de *SWG* basée principalement sur une méthode d'appariement sémantique hybride pour l'appariement des propriétés fonctionnelles, et étendue au calcul d'un score non fonctionnel de qualité de service géospatiaux. Nous avons proposé à la fin un score d'évaluation finale, que nous avons appelé score de sélection qui prend en compte à la fois le score fonctionnel et non fonctionnel. Cependant il existe des cas où le besoin de l'utilisateur est complexe et ne peut pas être réalisé par un service atomique d'où l'émergence de la composition des services Web. Elle se réfère au processus de création d'une chaîne de services proposant une nouvelle fonctionnalité, à partir de services atomiques existants. Notre première et seconde approches, sur la description et la découverte de *SWG* sont des éléments clés pour notre approche de composition automatique de services Web géospatiaux. Nous en discuterons dans le chapitre qui suit.

## Chapitre 6

# Composition automatique de *SWG*

### Sommaire

---

6.1	Principe global de la solution . . . . .	92
6.2	Présentation des éléments de composition . . . . .	94
6.2.1	Graphe de dépendance sémantique . . . . .	94
6.2.2	Requête et contraintes utilisateur . . . . .	97
6.2.3	Règles de composition . . . . .	99
6.2.4	Sélection des compositions de services candidates . . . . .	101
6.3	Conclusion . . . . .	103

---

### Introduction

La composition de services Web est le processus qui consiste à combiner plusieurs services pour satisfaire un besoin complexe de l'utilisateur [121]. Les approches de composition de services Web, proposées dans la littérature (voir par exemple [118, 110, 109, 84, 67, 82]), sont diverses mais nous pouvons distinguer deux grandes principales classes : les approches manuelles et les approches automatisées. Dans cette thèse nous nous intéressons à la composition automatique de services *WPS* sans états, i.e. des services synchrones qui ont une interaction simple (une requête - une réponse).

Les approches de composition de services automatisés exploitent le Web sémantique et des techniques d'intelligence artificielle (*IA*). En donnant un ensemble de services et une requête utilisateur, une spécification d'une chaîne de service peut être générée automatiquement [125].

La réalisation d'une composition automatique de *SWG* reste un défi et l'un des principaux problèmes est le manque de description sémantique des *SWG* qui empêche la compréhension complète de leurs sémantiques, ce qui affecte largement la composition automatique des *SWG* [83, 139].

La composition de services nécessite au moins trois tâches obligatoires qui doivent être prises en compte : la tâche consistant à spécifier les objectifs d'une composition ; la tâche de préciser les contraintes qui doivent être respectées ; la tâche de fournir des moyens qui aident à analyser et à comparer les solutions possibles les unes par rapport aux autres.

Dans ce contexte, nous présentons dans ce chapitre notre approche qui prend en compte ces trois tâches pour proposer une solution pour la composition de *SWG*. Nous exploitons le modèle de description sémantique proposé, la méthode d'appariement sémantique hybride et une technique de planification basée sur la programmation logique et plus particulièrement la programmation par ensembles réponses *ASP* [95]. Les techniques

de programmation logiques sont particulièrement adaptées pour construire les modèles de systèmes et formuler divers types de requêtes auxquelles le modèle peut répondre. Outre les propriétés de la programmation logique, le choix de la programmation par ensembles réponses dans ce contexte a été motivé par plusieurs points. Premièrement, le cadre déclaratif d'*ASP* pour la résolution de problèmes. Il est basé sur un langage de règles simple mais expressif qui permet de modéliser les tâches de compositions, que nous avons évoqué auparavant, sous forme de programmes logiques compacts et purement déclaratifs. De plus, des règles peuvent facilement être ajoutées ou modifiées, sans avoir à considérer l'ordre de ces règles, ni comment les exécuter, comme on aurait inévitablement à le faire lors de l'utilisation d'une approche impérative. Deuxièmement, il permet de générer la planification avec différents types de contraintes, ce qui permet d'ajouter et supprimer avec souplesse des critères de composition. Troisièmement, la recherche de solutions, appelée aussi calcul de modèles stables, est gérée par de puissants solveurs *ASP*. Dans le cadre de la composition, ceci permet de présenter un ensemble de services et des règles associées au solveur et les sorties du solveur sont collectées et présentées comme les solutions de composition.

Plus précisément, notre approche de composition automatique consiste à représenter d'un côté un graphe de dépendance de service, qui lie les services qui peuvent être composés en passant les sorties des uns aux entrées des autres, et d'un autre côté, à représenter la requête et contraintes utilisateur ainsi que des règles qui définissent les critères de composition, dans un programme logique *ASP*. La représentation du graphe de dépendance et de la requête utilisateur nécessitent des calculs de scores d'appariement sémantiques. Dans ce contexte nous exploitons la méthode d'appariement sémantique hybride proposée dans le cadre de notre contribution pour la découverte de *SWG*. Nous définissons également une stratégie de sélection de composition de services candidats en calculant des scores globaux d'appariement sémantique, et des scores globaux de QoS des services participants à chacune des compositions candidates.

Ce chapitre est divisé en trois sections. Dans la première section nous présentons le principe global de l'approche proposée pour la composition automatique de *SWG*. La deuxième section présente formellement les différents éléments de l'approche proposée. La troisième section conclue ce chapitre et introduit le prochain chapitre.

## 6.1 Principe global de la solution

La figure 6.1 montre l'aperçu de notre approche de composition automatique des services *WPS*. La solution est utilisée pour calculer les compositions possibles répondant à la requête de l'utilisateur, et à différentes contraintes spécifiées par l'utilisateur.

La solution que nous proposons consiste à mapper le problème de composition de services à un problème de planification basée sur l'utilisation du langage de programmation logique *ASP*. Les préliminaires sur *ASP* et la sémantique des modèles stables essentiels à la compréhension de l'approche proposée pour la composition automatique de *SWG* sont présentées dans l'annexe A.

Tout problème de planification, doit présenter, (1) les objets du monde qui nous intéressent, (2) les propriétés des objets qui nous intéressent, (3) un ou plusieurs état(s) initial(aux), (4) un ou plusieurs état(s) but(s), (5) et un ensemble d'actions que le planificateur peut effectuer pour atteindre le ou les état(s) but(s).

Le tableau 6.1 présente une correspondance des différents éléments de planification dans le contexte de la composition de services Web. Comme mentionné dans le tableau

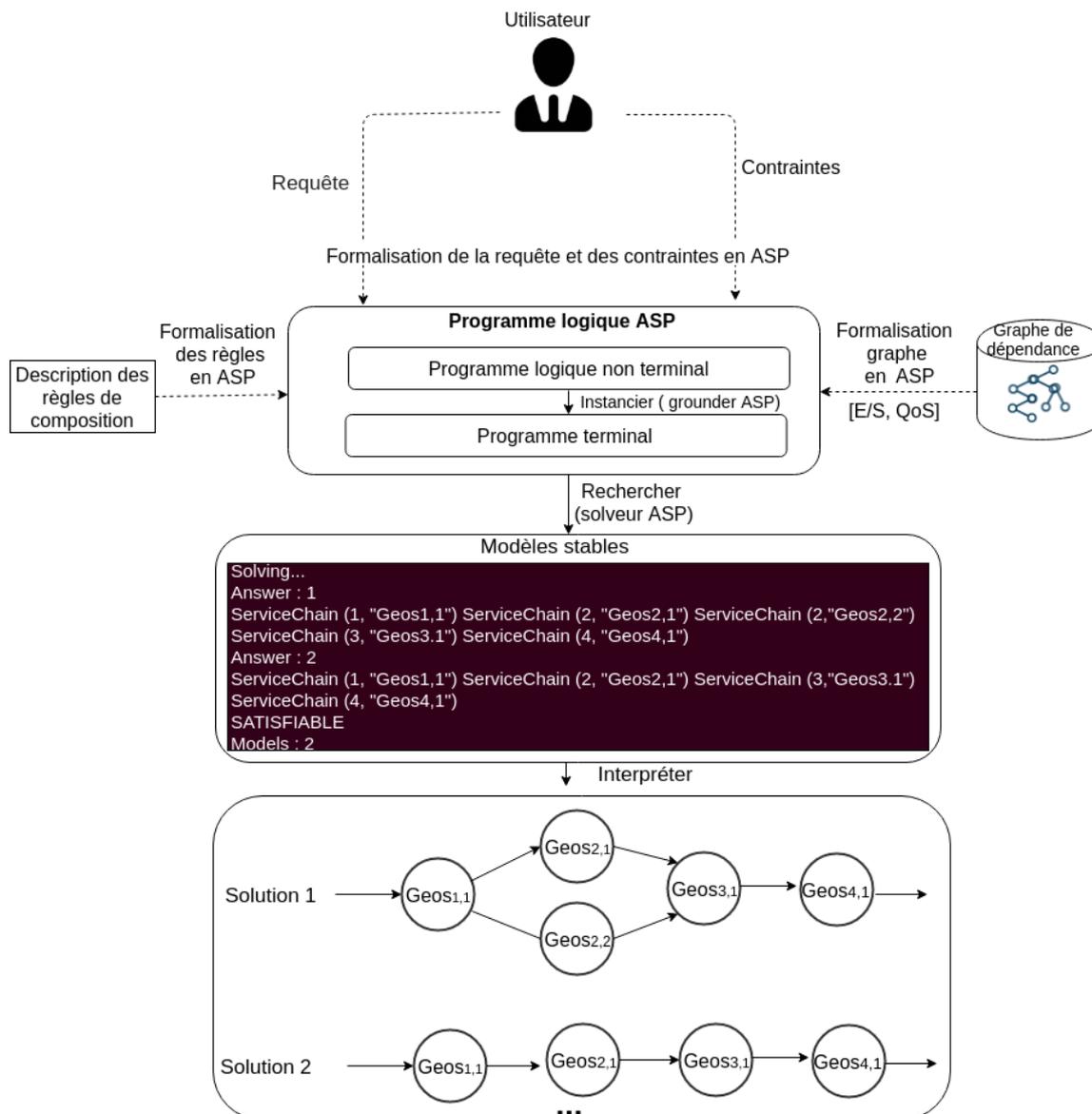
FIGURE 6.1 – Vu d'ensemble de l'approche de composition de *SWG*

TABLE 6.1 – La planification pour la composition de services Web

Planification	Composition de services Web
<b>Objets du monde</b>	Ils sont représentés par les services Web.
<b>Propriétés des objets</b>	Elles sont représentées par les propriétés fonctionnelles et non fonctionnelles des services Web et des propriétés de dépendance entre les services.
<b>États initiaux</b>	Ils sont représentés par les services qui peuvent être en début de chaîne.
<b>États buts</b>	Ils sont représentés par les services buts c'est à dire les services qui peuvent être en fin de chaîne.
<b>Ensemble d'actions</b>	Elles seront formalisés par des règles. Ces règles représentent les différents critères de composition.

(1) les objets du monde seront représentés par les services Web (voir Geos1,1, Geos1,2, etc dans la figure 6.1 qui représentent dans notre contexte des services Web géospatiaux), (2) les propriétés des objets seront représentées par les propriétés fonctionnelles et non fonctionnelles des services et aussi des propriétés entre les services. Particulièrement, nous nous intéressons à la représentation de la dépendance entre les services. Comme le montre la figure 6.1 nous avons procédé à la création d'un graphe global de dépendance. Un graphe qui représente les dépendances possibles entre tous les services *WPS* disponibles. Dans ce graphe les sommets représentent les services et les arcs représentent les dépendances entre les services, et ils sont valués. Les valeurs représentent les scores de dépendances entre les différentes *E/S* de services (voir section 6.2.1). Ce graphe de dépendance, ainsi que des propriétés de services appartenant à ce graphe seront formalisés dans langage *ASP* et seront inclus dans le programme logique *ASP* comme données essentielles à l'élaboration de la solution. Concernant (3) le(s) état(s) initiaux et (4) le(s) état(s) but(s) pour la planification dans le cadre de la composition de services, ils sont représentés par les services qui peuvent être en début et en fin de chaîne. Les services de début et les services de fin ou services buts sont calculés à partir de la requête utilisateur (c'est-à-dire les services dont les entrées sont satisfaites par les entrées fournies par la requête utilisateur et les services fournissant les sorties recherchées par la requête utilisateur). Aussi, afin de satisfaire des exigences utilisateurs supplémentaires, nous avons défini des contraintes fonctionnelles et structurelles. La requête et les contraintes seront formalisées dans langage *ASP* et seront incluses dans le programme logique *ASP* comme données essentielles à l'élaboration de la solution. (5) L'ensemble des actions à effectuer afin d'atteindre les services buts seront formalisées par des règles *ASP* que nous avons proposées. Ces règles représentent les différents critères de composition.

Une fois que les éléments cités ci-dessus seront intégrés dans un seul et même programme logique, un solveur *ASP* va calculer un ensemble de modèles stables qui seront interprétés en tant que solutions de composition possibles. Les résultats de compositions proposés contiennent des actions en séquence et/ou en parallèle.

## 6.2 Présentation des éléments de composition

Dans les sous sections qui suivent nous présentons les différents éléments impliqués dans la description du problème de composition de services *WPS*.

### 6.2.1 Graphe de dépendance sémantique

Dans le contexte de la solution de composition de *SWG*, pour créer le graphe de dépendance sémantique de services *WPS*, nous nous basons sur la description de services *WPS* définie dans le chapitre 4 et formalisée dans le chapitre 5 (voir définition 10), ainsi que sur la fonction *SimCalc* (voir équation 5.2) définie dans le chapitre 5.

Le graphe de dépendance sera représenté en *ASP* à l'aide du prédicat suivant :

- ***link\_dep(S1,S2,SS,ES,SC)*** : ce prédicat est utilisé pour décrire une règle *ASP* pour représenter les liens de dépendances entre les services. Plus précisément, ça indique qu'un service avec un identifiant *S1* qui possède une sortie annotée sémantiquement avec un concept *SS* peut être composé avec un service avec un identifiant *S2* qui possède une entrée annotée sémantiquement avec le concept *ES* qui satisfait la sortie *SS* avec un score de similarité sémantique égal à *SC*. Par exemple :  
`link_dep("NERC","TGN","SpatialNamedEntity", "PlaceName",91)`

est interprété comme le service qui a un identifiant "NERC" possède une sortie qui est annotée avec le concept "SpatialNamedEntity" et peut être composé avec le service qui a un identifiant "TGN" pour satisfaire l'entrée annotée avec le concept "PlaceName" avec une valeur de dépendance sémantique entre les deux concepts égale à 0.91.

Fondamentalement, une dépendance de service dans ce graphe représente une manière dont un service  $S_m$  peut être connecté à un autre service  $S_k$  en passant sa sortie à l'entrée du service  $S_k$ . Ainsi nous allons calculer le score de similarité sémantique entre chaque concept de sortie  $SS_j$  de  $S_m$ , et chaque concept d'entrée  $ES_i$  de  $S_k$ . Sur la base de l'observation ci-dessus, nous proposons d'adapter la fonction *SimCalc* d'appariement sémantique hybride (voir équation 5.2) présentée dans le chapitre 5 pour calculer le score de dépendance  $SC_{ij}$  sémantique entre chaque  $ES_i$  et  $SS_j$ . Si nous projetons ce contexte de dépendance dans un contexte de découverte nous pouvons imaginer que la sortie  $SS_j$  est l'une des entrées fournies par une requête utilisateur et que  $ES_i$  est l'une des entrées d'un service offert. Ainsi, pour le calcul du graphe de dépendance sémantique (voir algorithme 2) nous utilisons la fonction *SimCalc* (voir équation 5.2) avec un type de paramètre (*in – in*) où le paramètre *RP* est représenté par une sortie  $SS_j$  de  $S_m$  et le paramètre *SP* est représenté par une entrée  $ES_i$  de  $S_k$ . De plus, pour une optimisation, et dans le même principe que pour la méthode de découverte, nous maintiendrons les dépendances dont le score est le maximum entre  $ES_i$  et  $SS_j$ . La figure 6.2 présente un aperçu d'un extrait du graphe de dépendance sémantique résultant.

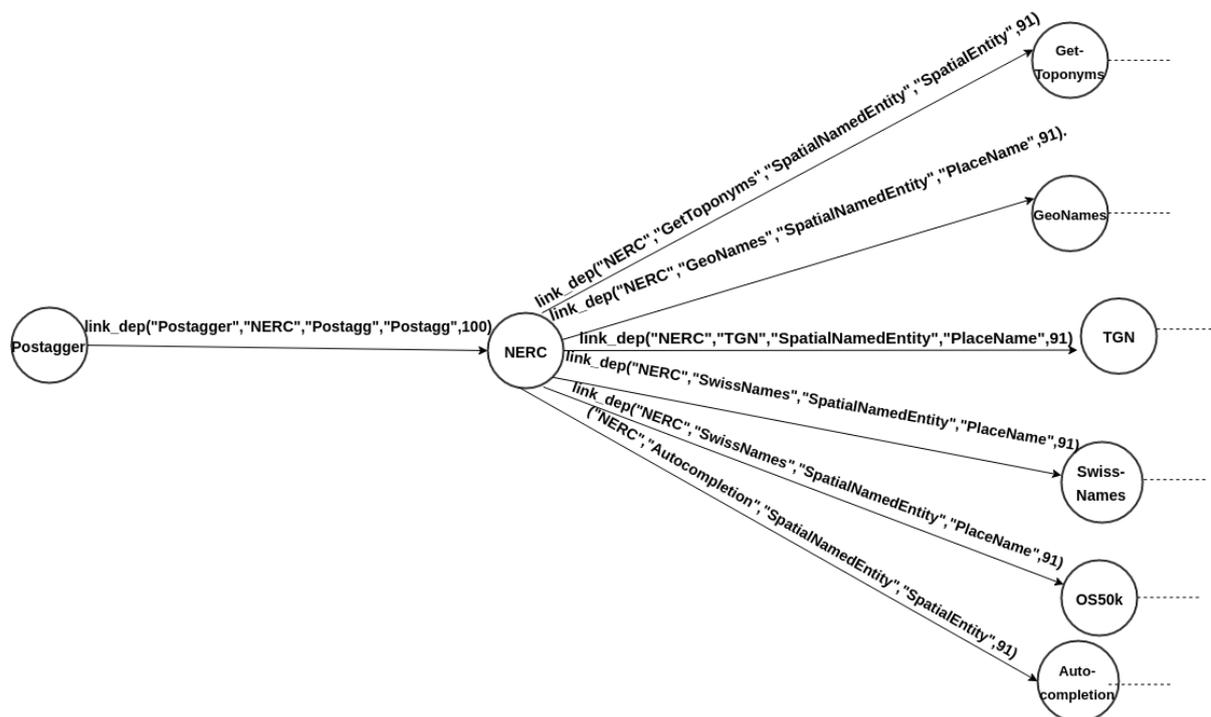


FIGURE 6.2 – Aperçu d'un extrait du graphe de dépendance sémantique

La création du graphe de dépendance est gourmande en calculs et en temps, mais il est créé en avance dans une étape de pré-traitement, plutôt qu'à la volée lors de la recherche de solution de composition dans un but de réduction du temps de recherche de solution de composition. Le graphe est mis à jour lorsqu'un nouveau service est ajouté au

---

**Algorithme 2** Calcul du graphe de dépendance sémantique
 

---

**Entrée:**  $S$  : Ensemble de services dans le registre

**Sortie:**  $G$  : Graphe de dépendances

```

1:  $G \leftarrow \emptyset$ ;
2:  $MaxSC\_Value \leftarrow 0$ ;
3:  $MaxSC\_Concept \leftarrow \emptyset$ ;
4: pour tout  $S_k, S_m \in S$  et  $k \neq m$  faire
5:   pour tout  $ES_i \in S_k$  faire
6:     pour tout  $SS_j \in S_m$  faire
7:        $SC_{ij} \leftarrow SimCalc(ES_i, SS_j)$ ;
8:       si ( $SC_{ij} > MaxSC\_Value$ ) alors
9:          $MaxSC\_Value \leftarrow SC_{ij}$ ;
10:         $MaxSC\_Concept \leftarrow SS_j$ ;
11:       fin si
12:     fin pour
13:     si ( $MaxSC\_Value \geq SeuilComposition$ ) alors
14:        $G \leftarrow G \cup Link\_dep(S_m, S_k, MaxSC\_Concept, ES_i, MaxSC\_Value)$ ;
15:     fin si
16:    $MaxSC\_Value \leftarrow 0$ ;
17:    $MaxSC\_Concept \leftarrow \emptyset$ ;
18: fin pour
19: fin pour
20: retourne  $G$ ;
    
```

---

registre ou lorsqu'un service existant est mis à jour ou supprimé du registre. Le graphe de dépendance est stocké dans le fichier du programme logique *ASP*.

Dans le cadre de notre solution et dans un but de satisfaire des critères de composition (voir section 6.2.3) nous représentons également les concepts d'entrées d'un service et la qualité d'un service formellement à l'aide des prédicats suivants :

- ***service\_in(S,ES)*** : ce prédicat est utilisé pour décrire une règle ASP pour représenter les entrées d'un service. Plus précisément, pour présenter un service avec un identifiant  $S$ , qui possède une entrée annotée sémantiquement avec un concept  $ES$ . Ainsi chaque entrée de la liste des entrées d'un service sera représentée en utilisant ce prédicat. Par exemple :

`service_in("géo_1", "PlaceName")`

représente le service qui a un identifiant "géo\_1" et qui possède une entrée annotée avec le concept "PlaceName".

- ***service\_nfs(S,Q)*** : ce prédicat est utilisé pour décrire une règle ASP pour représenter le score de qualité d'un service. Plus précisément, pour représenter un service avec un identifiant  $S$  possédant un score  $Q$  non fonctionnel (un score de *QoS*). Ainsi, pour chaque service un score non fonctionnel sera calculé en utilisant la fonction présentée dans l'équation 5.4 dans le cadre de la découverte de services 5. Les scores de *QoS* de l'ensemble des services présents dans le graphe de dépendance sémantique sont calculés dans une étape de pré-traitement avant l'arrivée et le traitement d'une requête utilisateur. Ces scores sont mis à jour lorsqu'un nouveau service est ajouté ou supprimé ou lorsque les scores sont mis à jour.

### 6.2.2 Requête et contraintes utilisateur

Une requête utilisateur a été définie dans le chapitre 5 par un triplet,  $(FR, ER, SR)$ , où  $FR$  représente la fonctionnalité recherchée et est associée à un concept de l'ontologie de service,  $ER$  une liste ordonnée d'entrées que l'utilisateur peut fournir et  $SR$  représente une liste ordonnée de sorties recherchées où chaque entrée et sortie est rattachée à un concept de l'ontologie de données. Dans le cadre de la composition des services, la définition de  $FR$  est étendue à une liste contenant une ou plusieurs fonctionnalité(s) recherchée(s). Ainsi, nous allons procéder au calcul de similarité sémantique entre les fonctionnalités recherchées et les fonctionnalités des services offerts incluses dans le graphe de dépendance. De même que dans le processus de découverte, l'appariement est basé sur l'utilisation de la fonction *SimCalc* proposée dans la section 5.1 avec la restriction de type  $(func - func)$ . Les services dont le score est égal ou supérieur à un seuil donné  $\in [0, 1]$  sont retenus. Cette étape vise à inclure exclusivement les services retenus, dans notre chaîne de composition de services. Ceci est représenté en *ASP* à l'aide des prédicats suivants :

- **satisfy\_fonc(S,F)** : ce prédicat est utilisé pour décrire une règle ASP pour représenter un service qui peut être utilisé dans la chaîne puisqu'il satisfait une fonctionnalité recherchée par l'utilisateur. Plus précisément, le service avec l'identifiant S, peut être inclus dans la chaîne de service, dans le cadre de la satisfaction de la fonctionnalité recherchée F. Par exemple  
`satisfy_fonc("TGN", "Direct_Geocoding_Service")`  
 est interprété comme le service qui a un identifiant "TGN" satisfait la fonctionnalité "Direct\_Geocoding\_Service".
- **include\_fonc(F)** : ce prédicat est utilisé pour décrire une règle ASP pour représenter un fonctionnalité recherchée par un utilisateur. Plus précisément, la fonctionnalité F est recherchée par un utilisateur et doit être satisfaite par au moins un service de la chaîne. Des règles supplémentaires vont être ajoutées au programme logique pour satisfaire cette exigence.

Concernant  $ER$  et  $SR$  la liste des entrées et sorties de la requête, elle engendre le calcul d'un ou de plusieurs services de début et d'un ou de plusieurs services buts. Les services de début représentent les services dont toutes les entrées peuvent être satisfaites avec la liste des entrées  $ER$  fournie dans la requête utilisateur et qui peuvent ainsi figurer en début de la chaîne de composition. Ainsi, nous allons procéder au calcul de similarité sémantique entre la liste des entrées de la requête  $ER$  et la liste des entrées  $ES$  de chacun des services offerts inclus dans le graphe de dépendance. L'appariement est basé sur l'utilisation de la fonction *EMatch*( $ES, ER$ ) proposée dans la section 5.1. Nous avons utilisé la fonction d'agrégation puisque l'ensemble des entrées  $ES$  requises par un service doivent être satisfaites par la liste  $ER$ .

Les services de sorties représentent les services qui peuvent fournir une ou plusieurs sorties requises (de la liste  $SR$ ) par la requête. Ainsi, nous allons procéder au calcul de similarité sémantique en utilisant la fonction *SimCalc* proposée dans la section 5.1 avec la restriction de type de paramètre  $(out - out)$ . Pour une optimisation, et dans le même principe que pour le calcul du graphe de dépendance, les sorties avec les scores d'appariement élémentaire maximum sont retenus. De plus, les services dont le score de sortie est égal ou supérieur à un seuil de composition donné  $\in [0, 1]$  sont retenus. Pour chaque chaîne de services produite comme une composition possible, nous vérifions si toutes les sorties recherchées sont bien fournies.

---

**Algorithme 3** Calcul des services de début et des services buts
 

---

**Entrée:**  $G$  : Graphe de dépendances

**Sortie:**  $SD$  : services de début,  $SB$  : services buts

```

1:  $SD \leftarrow \emptyset$ ;
2:  $SB \leftarrow \emptyset$ ;
3:  $MaxSC_B\_Value \leftarrow 0$ ;
4:  $MaxSC_B\_Concept \leftarrow \emptyset$ ;
5: pour tout  $S_i \in G$  faire
6:    $SC_D \leftarrow EMATCH(ES_i, ER)$  //  $ES_i$  : la liste des entrées du service  $S_i$ ,  $ER$  : la liste
     des entrées de la requête  $R$ .
7:   si ( $SC_D \geq SeuilComposition$ ) alors
8:      $SD \leftarrow SD \cup service\_start(S_i, SC_D)$ ;
9:   fin si
10:  pour tout  $SR_k \in SR$  faire
11:    pour tout  $SS_l \in SS_i$  faire
12:       $SC_B \leftarrow SimCalc(SS_l, SR_k)$  //  $SS_l$  : une sortie de la liste des sorties  $SS_i$  du service
         $S_i$ ,  $SR$  : la liste des sorties de la requête  $R$ ,  $SR_k$  : une sortie de la liste des sorties
        de la requête  $SR$ .
13:      si ( $SC_B > MaxSC_B\_Value$ ) alors
14:         $MaxSC_B\_Value \leftarrow SC_B$ ;
15:         $MaxSC_B\_Concept \leftarrow SS_l$ ;
16:      fin si
17:    fin pour
18:    si ( $MaxSC_B\_Value \geq SeuilComposition$ ) alors
19:       $SB \leftarrow SB \cup service\_goal(S_i, SR_k, MaxSC_B\_Concept, MaxSC_B\_Value)$ ;
20:    fin si
21:     $MaxSC_B\_Value \leftarrow 0$ ;
22:     $MaxSC_B\_Concept \leftarrow \emptyset$ ;
23:  fin pour
24: fin pour
25: retourne  $SD, SB$ ;
    
```

---

Les services de début, les services buts et les sorties requises par la requête utilisateur sont représentés en *ASP* à l'aide des prédicats suivants :

- **$service\_start(S, SC)$**  : ce prédicat est utilisé pour décrire une règle ASP pour représenter un service qui peut être utilisé en début de chaîne. Plus précisément, le service avec un identifiant  $S$  peut être utilisé en début de chaîne et ses concepts d'entrées sont satisfaits par les concepts d'entrées fournis par la requête avec un score de similarité sémantique entre les concepts d'entrées de la requête et les concepts d'entrées du service égal à  $SC$ .
- **$service\_goal(S, SR, SS, SC)$**  : ce prédicat est utilisé pour décrire une règle ASP pour représenter un service but qui peut être utilisé en fin de chaîne. Plus précisément, le service avec identifiant  $S$  peut être utilisé en fin de chaîne comme un service but puisqu'il possède un concept de sortie  $SS$  qui satisfait le concept de sortie  $SR$  recherché par la requête avec un score de similarité sémantique entre  $SS$  et  $SR$  égal à  $SC$ .
- **$request\_out(SR)$**  : ce prédicat est utilisé pour décrire une règle ASP pour représenter un concept de sortie recherché par un utilisateur. Plus précisément, le concept de sortie  $SR$  qui est recherché par l'utilisateur.

Nous décrivons ensuite des contraintes comme une partie étendue de la requête pour satisfaire les intentions particulières de l'utilisateur. Plus précisément, il s'agit de prendre en compte des exigences supplémentaires dans la conception de la solution de composition. Elles sont représentées en *ASP* à l'aide des prédicats suivants :

- **use\_service(S)** : ce prédicat est utilisé pour décrire une règle ASP pour représenter un service recherché par un utilisateur. Plus précisément, cela exprime qu'un service avec un identifiant S doit être utilisé dans la chaîne de service.
- **priority\_service(S1, S2)** : ce prédicat est utilisé pour décrire une règle ASP pour représenter une priorité entre des services recherchés par un utilisateur. Plus précisément, cela exprime qu'un service avec un identifiant S1, doit être utilisé dans la chaîne de service avant le service avec un identifiant S2.

### 6.2.3 Règles de composition

Les codages *ASP* de la description du graphe de dépendance et de la requête utilisateur en utilisant les prédicats présentés ci-dessus, sont utilisés comme des atomes (des *faits*) pour le programme de composition des services. Ces atomes seront utilisés en conjonction avec un ensemble de règles que nous définissons. Notons que ces règles utilisent également un ensemble supplémentaire de prédicats nécessaires à la construction de la solution comme suit :

- **chain(N,S)** : ce prédicat est utilisé pour représenter un service qui est inclut dans une chaîne de composition solution. Plus précisément, le service avec identifiant S est inclut dans une solution de composition et sa position dans la chaîne est N.
- **validate\_in(S,E)** : ce prédicat est utilisé pour représenter qu'une entrée I d'un service S appartenant à la chaîne est satisfaite par l'un des services le précédant dans la chaîne.
- **validate\_goal(SR)** : ce prédicat est utilisé pour représenter qu'une sortie SR recherchée par une requête utilisateur est satisfaite.
- **validate\_fonc(F)** : ce prédicat est utilisé pour représenter qu'une fonctionnalité F exigée dans la requête utilisateur est satisfaite.
- **chain\_long(N)** : ce prédicat est utilisé pour représenter la longueur N de la chaîne.

Concernant les règles de composition, en premier, nous définissons une règle pour positionner au moins un service de début avec un identifiant S en début de chaîne (à la position 1).

$$\mathbf{r1} : 1 \{ \text{chain}(1, S) : \text{service\_start}(S, \_) \} .$$

Ensuite, nous devons définir une cible de composition, cela nécessite la définition d'une règle pour la création d'un lien entre au moins un service but et un service virtuel "end-service" qui représente la cible de la chaîne de composition.

$$\mathbf{r2} : 1 \{ \text{link\_dep}(S, \text{endservice}, SR, SS, SC) : \text{service\_goal}(S, SR, SS, SC) \} .$$

Plus précisément, la règle montre qu'au moins un service but avec un identifiant S ayant une sortie SS qui satisfait la sortie SR recherchée par une requête, avec un score de similarité sémantique SC, est lié au service virtuel "endservice".

Pour calculer les différentes solutions de compositions automatiquement à partir de(s) service(s) de début jusqu'au(x) service(s) but(s), nous avons introduit les règles (r3, r4, r5, r6, r7, r8, r9, r10, r11 et r12) qui vont générer toutes les combinaisons possibles<sup>1</sup>. La règle r3 positionne les différents services dans la chaîne de composition si la longueur maximale *chain\_maxlen* de la chaîne n'est pas atteinte et si il existe un lien de dépendance entre un service S1 et un service S2, sachant que S1 appartient à la chaîne à la position N, et S2 peut succéder S1 et appartenir à la chaîne à la position N+1. Si deux services possèdent la même position dans une chaîne, ceci exprime qu'ils peuvent être exécutés en parallèle.

**r3** :  $\{chain(N + 1, S2) : chain(N, S1), link\_dep(S1, S2, \_, \_), S1! = endservice\}$   
 :-  $chain(N, \_), N < chain\_maxlen$ .

La règle r4 vérifie qu'une entrée d'un service appartenant à la chaîne de composition est bien satisfaite par l'un des services le précédant dans la chaîne. Plus précisément, le concept d'entrée E d'un service S2 appartenant à la chaîne à la position M, est validé s'il existe un lien de dépendance sémantique avec l'une des sorties d'un service S1 le précédant dans la chaîne avec une position  $N < M$ .

**r4** :  $validate\_in(S2, E) :- service\_in(S2, E), chain(N, S1), chain(M, S2),$   
 $link\_dep(S1, S2, \_, E, \_), N < M$ .

La règle r5 vérifie que toutes les entrées d'un service appartenant à la chaîne, doivent être validées, sauf si c'est un service de début ses entrées sont alors validées par les entrées fournies par la requête utilisateur.

**r5** :  $:- not validate\_in(S, E), service\_in(S, E), chain(N, S), N! = 1$ .

La règle r6 vérifie qu'une sortie recherchée par une requête utilisateur est satisfaite. Plus précisément, si un service but S fournissant une sortie SR recherchée par la requête, appartient à la chaîne, alors la sortie SR est validée.

**r6** :  $validate\_goal(SR) :- service\_goal(S, SR, \_, \_), chain(\_, S),$   
 $link\_dep(S, endservice, SR, \_, \_)$ .

La règle r7 vérifie que toutes les sorties recherchées par la requête utilisateur sont validées.

**r7** :  $:- not validate\_goal(SR), request\_out(SR)$ .

La règle r8 vérifie qu'une fonctionnalité F exigée dans la requête utilisateur est validée en vérifiant si il existe un service S de la chaîne satisfaisant cette fonctionnalité.

**r8** :  $validate\_fonc(F) :- include\_fonc(F), chain(\_, S), satisfy\_fonc(S, F)$ .

---

1. Une règle supplémentaire, pour certains cas particuliers, pourrait également être ajoutée pour que chaque modèle stable ne représente qu'une solution de composition ( :-  $chain(N1, S1), chain(N2, S2), link\_dep(S1, Z, \_, M, \_), link\_dep(S2, Z, \_, M, \_), S1 != S2, N1 = N2$ ).

La règle r9 vérifie que toutes les fonctionnalités exigées par la requête utilisateur sont validées.

**r9** :  $:- \text{not validate\_fonc}(F), \text{include\_fonc}(F).$

La règle r10 vérifie que tous les services appartenant à la chaîne doivent satisfaire au moins une fonctionnalité exigée.

**r10** :  $:- \text{chain}(\_, S), \text{not satisfy\_fonc}(S, \_), s! = \text{endservice}.$

La règle r11 vérifie que toutes les chaînes candidates aboutissent aux services buts en vérifiant qu'ils doivent avoir comme dernier service, le service virtuel cible "endservice"<sup>2</sup>.

**r11** :  $:- \text{chain}(N, S), \text{chain\_long}(N), \text{not } S = \text{endservice}.$

La règle r12 élimine les cycles.

**r12** :  $:- \text{chain}(N, S), \text{chain}(M, S), N! = M.$

Pour le calcul de la longueur d'une chaîne de composition nous avons introduit la règle r13.

**r13** :  $\text{chain\_long}(N) :- \text{chain}(N, \_), \text{not chain}(N + 1, \_).$

Pour prendre en compte les contraintes utilisateurs, nous introduisons les règles (r14, r15 et r16). La règle r14 vérifie qu'un service S exigé par un utilisateur est validé en vérifiant si il appartient à la chaîne.

**r14** :  $\text{validate\_service}(S) :- \text{use\_service}(S), \text{chain}(\_, S).$

La règle r15 vérifie que tous les services S exigés par un utilisateur sont validés.

**r15** :  $:- \text{use\_service}(S), \text{not validate\_service}(S).$

La règle r16 vérifie l'ordre de services exigé par un utilisateur.

**r16** :  $:- \text{priority\_service}(S1, S2), \text{chain}(N, S1), \text{chain}(M, S2), N > M.$

#### 6.2.4 Sélection des compositions de services candidates

Nous définissons une stratégie de sélection des compositions candidates. Une composition de services, consiste en une chaîne de services constituée d'un ensemble de services et un ensemble de dépendances de services.

Notre objectif est de trouver le meilleur schéma de composition (la meilleure chaîne) parmi les compositions candidates selon des critères d'optimisation. Ces critères concernent le degré d'appariement sémantique entre les services participants, et entre la requête et les services de début et de fin ainsi que la qualité des services (score non fonctionnel).

---

2. Une règle supplémentaire, pourrait également être ajoutée pour supprimer toutes les chaînes qui n'aboutissent pas. (  $:- \text{chain}(\_, S1), S1! = \text{endservice}, \text{not link\_dep}(S1, \_, \_, \_)$  ).

Pour chaque chaîne candidate nous calculons un score (voir équation 6.1a) en additionnant (i) un score qui représente la somme des scores non fonctionnels des services inclus dans la composition (voir équation 6.1b), (ii) et un score qui représente la somme des scores de dépendances reliant les paires de services dans la chaîne, additionné à la sommes des scores d'appariement des services de début (voir équation 6.1c). Nous notons que les scores d'appariement de services buts avec la requête sont inclus dans les scores de dépendance puisque tous les services buts sont reliés à un service virtuel *endservice* qui représente la cible de la chaîne de composition.

$$scoreCmp = \beta \cdot scoreCmp\_F + (1 - \beta) \cdot scoreCmp\_NF \quad (6.1a)$$

$$scoreCmp\_NF = \frac{\sum_{i=1}^n scoreNF_i}{n} \quad (6.1b)$$

$$scoreCmp\_F = \frac{\sum_{j=1}^m scoreDp_j + \sum_{k=1}^o scoreIn_k}{m + o} \quad (6.1c)$$

où :

- $scoreCmp$  = Score global de la composition
- $scoreCmp\_NF$  = Score non fonctionnel de la composition
- $scoreNF_i$  = Score non fonctionnel d'un service
- $scoreCmp\_F$  = Score fonctionnel de la composition
- $scoreDp_j$  = Score de dépendance sémantique entre deux services
- $scoreIn_k$  = Score d'appariement sémantique d'un service de début
- $\beta$  = Un nombre réel représentant la pondération,
- et
- $0 \leq scoreCmp, scoreCmp\_NF, scoreNF_i, scoreCmp\_F, scoreDp_j, scoreIn_k, \beta \leq 1$

Les scores fonctionnels et non fonctionnels de la composition de service ( voir équation 6.1a) peuvent être multipliés, chacun, par une valeur de pondération ce qui permet d'augmenter ou de diminuer la considération du score fonctionnel et non fonctionnel. Le poids permet de donner à l'utilisateur la possibilité de paramétrer le score final puisque l'intérêt des utilisateurs concernant l'aspect non fonctionnel est différent, certains préfèrent en tenir compte et d'autres non.

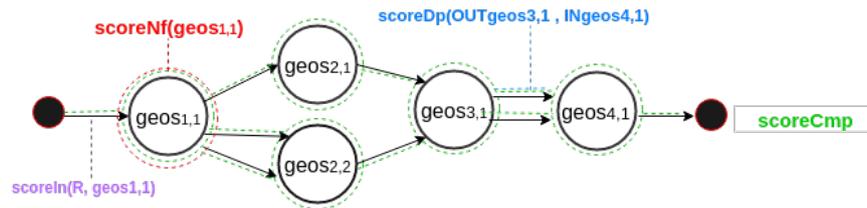


FIGURE 6.3 – Exemple d'une solution de composition montrant le score non fonctionnel, le score de dépendance, le score de service de début, et le score global de composition

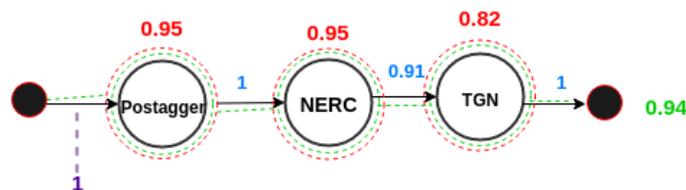


FIGURE 6.4 – Exemple d’une solution de composition avec les valeurs de scores fonctionnels, non fonctionnels et global

Un exemple d’une solution de composition théorique est donné dans la figure 6.3 et un exemple d’une solution de composition réelle est donnée dans la figure 6.4. Sur la base de la solution proposée chaque service a un score non fonctionnel  $scoreNF$  calculé en fonction de l’ensemble des valeurs de propriétés non fonctionnelles du service (attributs de  $QoS$ ). En outre, chaque lien de dépendance entre deux concepts de services est valué par un score de similarité sémantique  $ScoreDp$  (par exemple, entre le concept représentant la sortie  $OUTgeos3,1$  du service  $geos3,1$  et le concept représentant l’entrée  $INgeos4,1$  du service  $geos4,1$ ). Chaque service de début a un score d’appariement sémantique  $ScoreIn$ . Enfin un score global  $scoreCmp$  est calculé et attribué à chaque composition de services candidate (voir l’encerclement en vert dans la figure 6.3).

### 6.3 Conclusion

Dans ce chapitre nous avons présenté une approche pour la composition automatique de  $SWG$ . L’approche proposée consiste à mapper le problème de composition de services  $WPS$  à un problème de planification défini sous forme d’un programme logique  $ASP$  et résolu par le calcul de modèles stables  $ASP$ .

Dans un premier temps, nous avons décrit le principe global de la solution pour exprimer les éléments pertinents de la planification dans le cadre d’une composition de services. Ensuite nous avons proposé une sémantique basée sur la terminologie  $ASP$  pour la modélisation des différents éléments de composition en terme de graphe de dépendance et de requête et contraintes utilisateur, tout en expliquant les méthodes de calcul de similarités sémantiques utilisées pour la représentation des différents éléments. Celles-ci sont principalement basées sur les méthodes d’appariement sémantique présentées dans le chapitre 5 et adaptées au contexte de composition de services. Aussi nous avons défini les règles de composition s’appliquant au différents éléments de composition afin de générer les modèles stables représentant les compositions possibles. Les résultats de compositions proposés contiennent des actions en séquence ou en parallèle. Enfin nous avons défini une stratégie de sélection de compositions candidates basée sur le calcul de scores fonctionnels et non fonctionnels globaux. Dans le chapitre suivant, nous décrivons l’environnement implémentant notre solution pour l’amélioration du processus de découverte de  $SWG$  de la description à la composition de services, suivi par les différentes expérimentations menées, et une discussion des résultats obtenus.



## Chapitre 7

# Implémentation et expérimentations

### Sommaire

---

7.1	Mise en œuvre de l’approche de description de <i>SWG</i> . . . . .	106
7.1.1	Architecture fonctionnelle de la description de <i>SWG</i> . . . . .	106
7.1.2	Création d’un corpus de services WPS annotés . . . . .	106
7.2	Mise en oeuvre de l’approche de découverte sémantique de <i>SWG</i> . . . . .	108
7.2.1	Architecture fonctionnelle de la découverte de <i>SWG</i> . . . . .	108
7.2.2	Expérimentations sur l’approche de découverte . . . . .	108
7.3	Mise en oeuvre de l’approche de composition automatique de <i>SWG</i> . . . . .	116
7.3.1	Architecture fonctionnelle de la composition de <i>SWG</i> . . . . .	116
7.3.2	Cas d’étude et expérimentations sur l’approche de composition . . . . .	116

---

### Introduction

Dans les précédents chapitres de contributions, nous avons présenté nos approches de description, découverte et composition de *SWG*. Nous avons présenté une approche par annotation pour l’extension de la description WPS. Cette approche intègre une couche sémantique pour décrire les propriétés fonctionnelles d’un service *WPS*, de même, elle intègre une couche de description de propriétés non fonctionnelles. Nous avons par la suite présenté une approche de découverte sémantique et de sélection qui exploite la première approche pour la description sémantique des services WPS. Cette approche est basée principalement sur une méthode d’appariement sémantique hybride que nous avons proposée. Enfin, nous avons présenté une approche de composition sémantique automatique de *SWG* qui exploite les deux premières approches de description et de découverte sémantiques et une méthode de planification basée sur le langage de programmation logique *ASP*.

Dans ce dernier chapitre dédié aux contributions, nous implémentons les différentes approches proposées et nous effectuons des expérimentations des approches de découverte sémantique et de composition automatique. Dans la section 7.1, nous introduisons l’architecture fonctionnelle de description de *SWG* et le corpus de services *WPS* annotés selon l’approche proposée. Dans la section 7.2, nous présentons l’architecture fonctionnelle de découverte de *SWG* et nous présentons les expérimentations de l’approche proposée. Dans la section 7.3, nous présentons l’architecture de composition de *SWG*, puis nous présentons un cas d’étude dans le cadre du projet Choucas et des premières expérimentations de cette approche de composition. Dans la section 7.4 nous concluons enfin ce chapitre.

## 7.1 Mise en œuvre de l'approche de description de *SWG*

### 7.1.1 Architecture fonctionnelle de la description de *SWG*

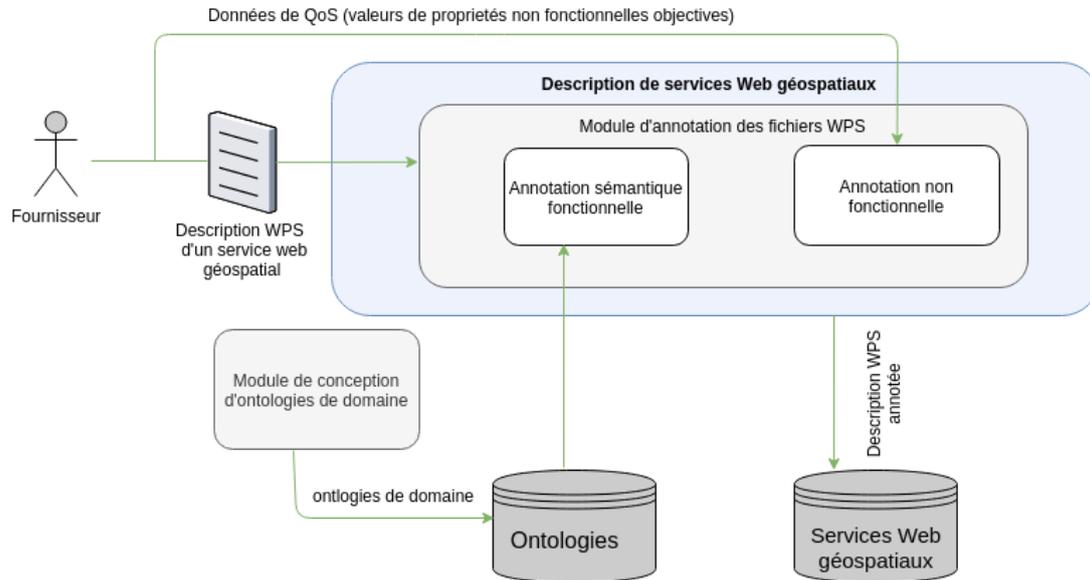


FIGURE 7.1 – Architecture fonctionnelle de la description de *SWG*

L'architecture de description de *SWG* (Figure 7.1), regroupe principalement les deux modules :

- Module de conception d'ontologies de domaine : ce module construit les deux ontologies de domaines décrites dans le chapitre 4. Il fera appel au logiciel Protégé.
- Module d'annotation des fichiers WPS : les points d'entrées de ce module sont : les ontologies conçues par le premier module ; les fichiers WPS décrivant syntaxiquement les propriétés fonctionnelles d'un service géospatial et les informations sur les propriétés non fonctionnelles objectives. Ce module applique l'approche d'annotation décrite dans le chapitre 4. Il permet par la suite de créer un répertoire de description de services WPS annotés.

Dans le cadre d'application de notre approche de description sémantique de *SWG*, afin d'effectuer des expérimentations des approches de découverte et de composition sémantiques qui exploitent le modèle de cette approche, nous avons créé un corpus de 94 *Process* WPS annotés. La section qui suit décrit la démarche de construction de ce corpus.

### 7.1.2 Création d'un corpus de services WPS annotés

Pour être en mesure d'effectuer des expérimentations sur nos contributions, nous avons créé un corpus de 94 *Process* WPS décrit syntaxiquement selon le modèle de description WPS 2.0. Nous avons, par la suite décrit sémantiquement ces *Process* suivant l'approche d'annotation proposée.

Pour obtenir un corpus diversifié, nous avons sélectionné des services avec des caractéristiques différentes : fonctionnalités offertes, types de paramètres d'*E/S*, nombres de paramètres d'*E/S* etc. Tous les services sélectionnés sont liés au domaine géospatial.

En premier, nous avons sélectionné quatre services qui ont été développés dans le cadre du projet ANR Choucas<sup>1</sup>, deux services proposés par l’IGN et quatre services qui utilisent 4 différents gazetiers cités dans l’article [8]. Nous avons analysé les fonctionnalités et les *E/S* de ces services et ensuite nous avons décrit des variantes de ces services selon le modèle de description syntaxique WPS 2.0.

Pour enrichir notre corpus, nous avons récupéré 84 descriptions de *Process* WPS publiés en open source dans le cadre du projet 52° North, ces *Process* été décrits en WPS 1.0. Nous avons repris ces descriptions pour les réécrire selon la version actuelle 2.0 de WPS. Ils proposent plusieurs traitements liés au domaine géospatial, comme le calcul de distance, l’interpolation de surface, etc. Enfin, la démarche d’annotation sémantique et d’appariement aux ontologies de domaine est réalisée suivant les 2 étapes décrites ci-après :

1. Annotation de la fonctionnalité : pour chaque *Process*, nous avons analysé le fichier de description syntaxique WPS. La fonctionnalité offerte est analysée ensuite une analyse de l’ontologie de services géospatiaux (voir chapitre 4) est effectuée pour trouver la catégorie de service correspondante. Nous ajoutons par la suite une annotation indiquant le rôle de la méta-donnée (concept de fonctionnalité de service), et un lien vers le concept sélectionné de l’ontologie.
2. Annotations des paramètres d’*E/S* : pour chaque *Process*, nous avons analysé le fichier de description syntaxique WPS. Nous avons sélectionné et analysé les paramètres d’entrées obligatoires et les paramètres de sorties avec leurs valeurs respectives et nous avons analysé l’ontologie de données géospatiales (voir chapitre 4) pour trouver des liens possibles et décider de la meilleure alternative. Nous ajoutons par la suite une annotation indiquant le rôle de la méta-donnée à ajouter (concept d’entrée ou concept de sortie), et un lien vers le concept sélectionné de l’ontologie.

De plus, pour chaque *Process*, une annotation le liant à un fichier *XML* qui représente les propriétés non fonctionnelles a été réalisée. Les valeurs correspondantes aux propriétés non fonctionnelles sont simulées car ces informations ne sont pas encore disponibles excepté pour cinq services qui utilisent des gazetiers. Pour ces services, un ensemble de valeurs réelles de propriétés objectives spécifiques au domaine géospatial ont été prises en considération.

Dans la section qui suit nous présentons les expérimentations de nos approches de découverte et composition sémantiques automatiques de *SWG*. Les descriptions de services annotés ainsi que les ontologies de domaine sont fournis en entrée des modules de découverte et de composition. L’ensemble des expérimentations présentées dans le reste de ce chapitre ont été exécutées sur une machine équipée d’un processeur Intel i7-7820 Ocatacore cadencé à 2.90GHz, d’une mémoire vive (RAM) de 16Gb et d’un disque de stockage de type SSD de 500Gb. La première version du code de l’approche de découverte ainsi que l’ensemble des données utilisées, sont disponibles publiquement sous la licence MIT à l’adresse <https://github.com/mshalilali/Semantic-GWS-discovery>.

---

1. Dans le cadre d’effectuer l’une des tâches en cours pour proposer un processus d’appariement automatique des traces, des textes de description de randonnées et des ressources géographiques (répertoires toponymiques et bases de données géographiques) [42].

## 7.2 Mise en oeuvre de l'approche de découverte sémantique de *SWG*

### 7.2.1 Architecture fonctionnelle de la découverte de *SWG*

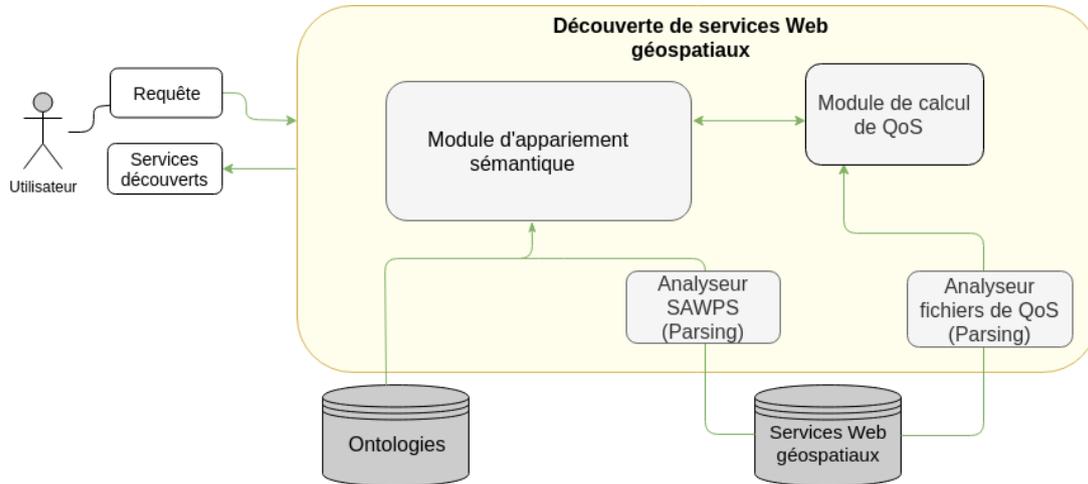


FIGURE 7.2 – Architecture fonctionnelle de la découverte de *SWG*

Pour implémenter les méthodes proposées dans le chapitre 5, nous développons un système pour la découverte de services *WPS* implémenté en Java. L'architecture fonctionnelle est illustrée à la Figure 7.2, elle regroupe principalement les quatre modules :

- Analyseur SAWPS : ce module se charge de l'extraction des concepts d'annotation de la fonctionnalité et des *E/S* associées aux descriptions *WPS* sémantiquement annotées.
- Module d'appariement sémantique : ce module effectue l'appariement sémantique (matching) entre la requête utilisateur et l'ensemble des services *WPS* offerts en comparant différents concepts selon la méthode d'appariement sémantique hybride proposée dans le chapitre 5. Il utilise les ontologies de domaine conçues. Pour inférer les relations entre les concepts il utilise le raisonneur de l'API Jena<sup>2</sup>.
- Analyseur de fichiers de *QoS* : ce module se charge de l'extraction des valeurs de propriétés non fonctionnelles (*QoS*) dans les fichiers de description de ces derniers.
- Module de calcul de *QoS* : ce module calcule un score non fonctionnel (score de *QoS*) pour chacun des services découverts par le module d'appariement sémantique.

### 7.2.2 Expérimentations sur l'approche de découverte

Pour d'effectuer les expérimentations sur notre approche de découverte, deux configurations différentes ont été testées :

1. Une configuration avec deux étapes d'appariement sémantique : appariement de la fonctionnalité et appariement des *E/S*.
2. Une configuration avec trois étapes : appariement de la fonctionnalité, appariement des *E/S* et enfin une étape non fonctionnelle pour le calcul de score de *QoS*.

Pour la première configuration nous utilisons les métriques de précision et de rappel décrites dans les équations 7.1 et 7.2

2. Jena : <https://jena.apache.org/>

$$Precision = \frac{|A \cap B|}{B} \quad (7.1)$$

$$Rappel = \frac{|A \cap B|}{A} \quad (7.2)$$

où  $A$  représente les services pertinents pour une requête et  $B$  représente l'ensemble de services retournés pour une requête. De plus nous allons calculer le temps moyen de réponse. Pour la deuxième configuration, elle nécessite le calcul de score non fonctionnel (*QoS*). Cependant, nous ne disposons pas encore des valeurs réelles de propriétés non fonctionnelles pour tout les services inclus dans le corpus test. Par conséquent, nous avons simulé les valeurs de ces propriétés et nous présentons un exemple où nous disposons de valeurs réelles de propriétés non fonctionnelles spécifiques au domaine géospatial, pour montrer l'intérêt de la prise en compte de ces propriétés.

Pour procéder au calcul des taux de rappel et précision et le temps de réponse, pour la première configuration, nous l'évaluons sur une collection de test incluant 10 requêtes pour 8 fonctionnalités de services différentes.

Nous avons également fait varier les seuils d'acceptation (voir section 5.2) :

- pour l'appariement de la fonctionnalité, un seuil strict  $\theta = 1$ , un seuil moyen  $\theta = 0.5$ , et un seuil flou  $\theta = 0$ .
- pour l'appariement des  $E/S$ , un seuil strict  $\theta = 1$ , un seuil moyen  $\theta = 0.5$ , et un seuil entre les deux  $\theta = 0.7$ .

Lors de la présentation des résultats pour l'appariement des fonctionnalités, les résultats pour le seuil 0 nous permettent de simuler la non prise en compte de cette étape avant l'étape d'appariement des  $E/S$ . En revanche, pour l'appariement des  $E/S$ , les résultats pour les seuils  $< 0.5$  ne sont pas présentés puisque en dessous de 0.5 l'étude n'est pas pertinente car il y a un risque de se retrouver avec un grand nombre de faux positifs.

Dans ce qui suit, les tableaux 7.1 et 7.2 contiennent 2 des 10 requêtes sémantiques de notre ensemble de tests.

La première requête (R1) correspond à la recherche d'un **service de calcul de distance** et la deuxième requête (R2) correspond à la recherche d'un **service de géocodage direct**. Rappelons que seuls les services de géocodages possèdent des valeurs réelles pour les propriétés non fonctionnelles objectives, c'est pourquoi la requête R2 a été choisie ici. La requête R1 représente une situation où les valeurs des propriétés non fonctionnelles ont été simulées. Dans les tableaux correspondants, les trois colonnes indiquent la fonctionnalité et les entrées/sorties du service requis. Ces tableaux sont accompagnés d'extraits de l'ontologie des services (voir les figures 7.3 et 7.5) et d'extraits de l'ontologie des données (voir les figures 7.4 et 7.6) afin de montrer pour les requêtes présentées les concepts liés aux fonctionnalités et aux entrées/sorties. Les figures montrent la hiérarchie des classes dans les ontologies décrites avec des relations **is-a** entre les classes et les sous-classes. La couleur orange montre l'équivalence de deux classes. Les couleurs violette et verte sur les flèches de relation montrent respectivement les classes enfant et parent d'une classe sélectionnée. La classe sélectionnée apparaît entourée d'une bordure bleue.

Ensuite, nous présentons les résultats associés (voir tableaux 7.3 et 7.4) avec les noms des services, leurs fonctionnalités, le score d'appariement entre la fonctionnalité requise et la fonctionnalité offerte, les  $E/S$  et les scores d'appariement entre les  $E/S$  requises et

TABLE 7.1 – Requête *R1*

Fonctionnalité	Entrées	Sorties
Distance calcula- tion service	SourceLocation, Impudence	Distance, EuclidianDis- tance

les *E/S* offertes. Enfin, nous présentons les taux de rappel moyen, de précision et le temps de réponse moyen obtenus pour l'ensemble des tests (voir les figures 7.7, 7.8 et 7.9).

 TABLE 7.2 – Requête *R2*

Fonctionnalité	Entrées	Sorties
Direct geocoding service	SpatialEntity	GeographicCoordinates

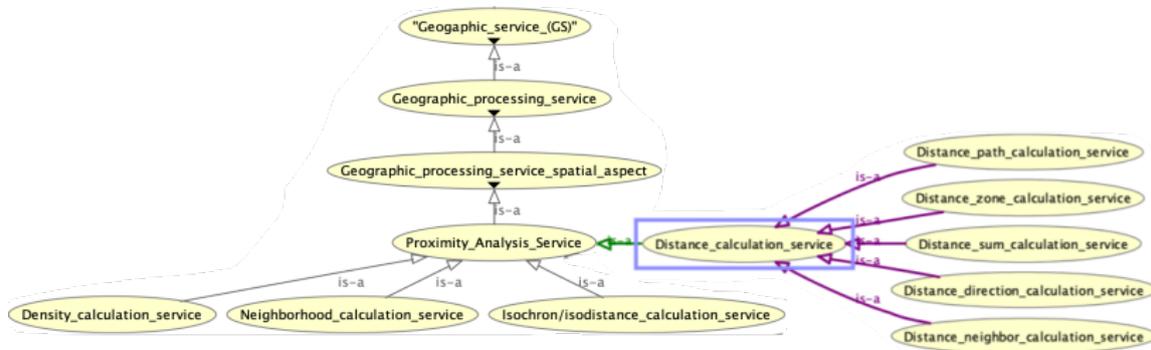

 FIGURE 7.3 – Extrait de l'ontologie de service pour l'exemple de requête *R1*

 TABLE 7.3 – Résultats associés à la requête *R1*

Service	Fonctionnalité	Score Fonctionnalité	Entrées	Sorties	Score <i>E/S</i>
Euclidean Distance	Distance calculation service	1.0	SourceLocation	EuclideanDirection, EuclideanDistance	0.97
Euclidean Direction	Source direction calculation service	0.90	SourceLocation	EuclideanDirection, EuclideanDistance	0.97
Euclidean Allocation	Distance zone calculation service	0.90	SourceLocation	EuclideanDirection, EuclideanDistance, EuclideanAllocation	0.97
Cost Distance	Distance calculation service	1.0	SourceLocation, Impedance	CostDistance, CostBackLink	0.72
Cost Back-Link	Distance neighbor calculation service	0.90	SourceLocation, Impedance	CostDistance, CostBackLink	0.72
Cost Allocation	Distance zone calculation service	0.90	SourceLocation, Impedance	CostAllocation, CostDistance, CostBackLink	0.72
Corridor	Distance sum calculation service	0.90	CostDistance, CostDistance	Corridor	0.22
Cost Path	Distance path calculation service	0.90	DestinationLocation, CostDistance, CostBackLink	CostPath	0.0



FIGURE 7.4 – Extrait de l'ontologie de données pour l'exemple de requête R1

TABLE 7.4 – Résultats associés à la requête *R2*

Service	Fonctionnalité	Score Fonctionnalité	Entrées	Sorties	Score E/S
GeoNames	Direct geocoding service	1.0	PlaceName	GeographicCoordinates	1.0
OS50k	Direct geocoding service	1.0	PlaceName	GeographicCoordinates	1.0
SwissNames	Direct geocoding service	1.0	PlaceName	GeographicCoordinates	1.0
TGN	Direct geocoding service	1.0	PlaceName	GeographicCoordinates	1.0
GetToponyms	Direct geocoding service	1.0	SpatialEntity <sup>3</sup>	GeographicCoordinates	1.0
ConvPerdido	Geographic format conversion service	0.0	GeographicCoordinates	GeographicCoordinates	0.5
Autocompletion	Autocompletion service	0.0	SpatialEntity	Adress	0.5

3. PlaceName et SpatialEntity sont des concepts equivalents (voir figure 7.6)

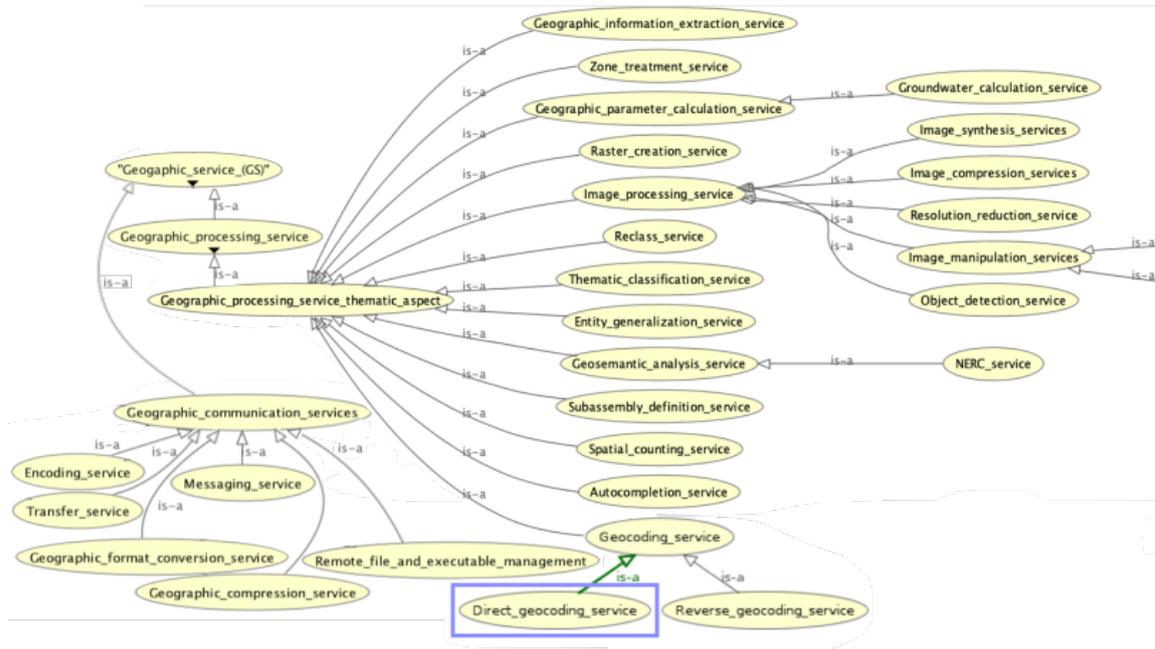


FIGURE 7.5 – Extrait de l'ontologie de service pour l'exemple de requête R2

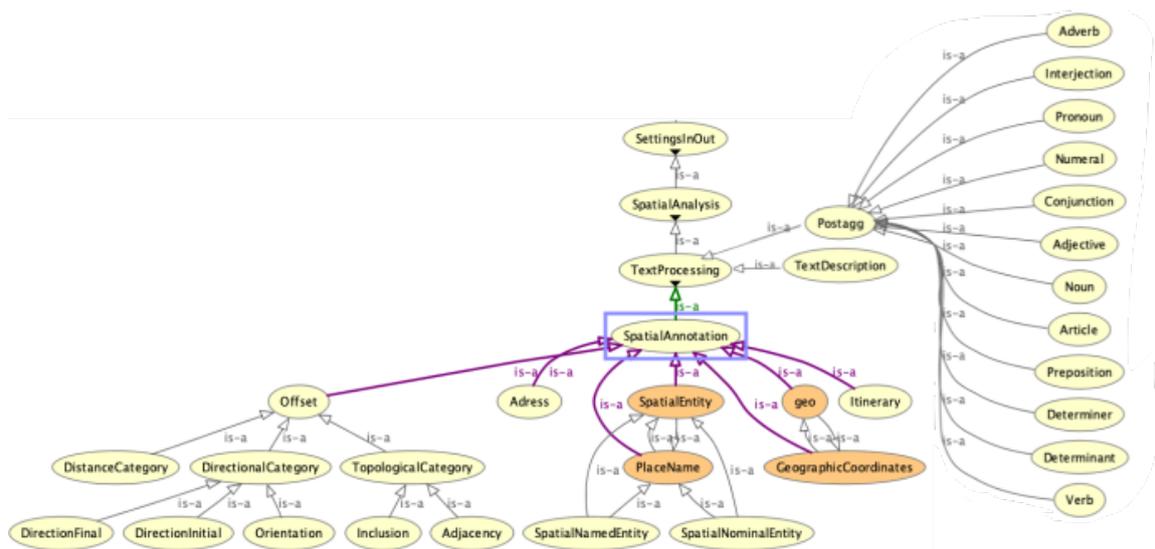


FIGURE 7.6 – Extrait de l'ontologie de données pour l'exemple de requête R2

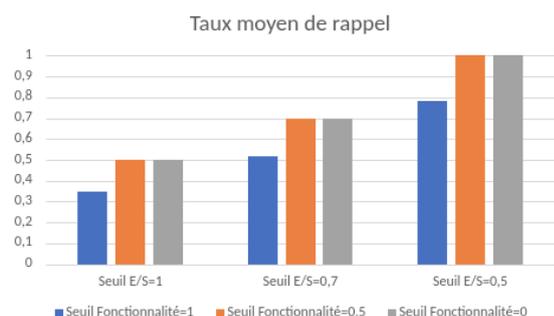


FIGURE 7.7 – Taux moyen de rappel

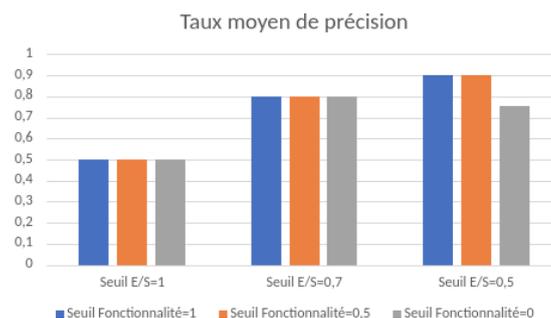


FIGURE 7.8 – Taux moyen de précision

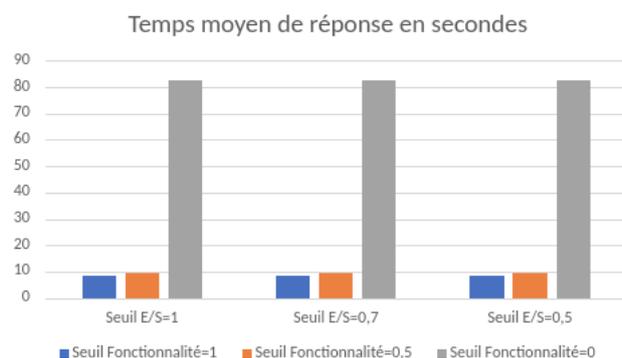


FIGURE 7.9 – Temps moyen de réponse

Dans la configuration à 2 étapes, l'évaluation basée sur l'ensemble des tests montre que l'approche de découverte offre les meilleurs taux de précision (90 %), taux de rappel (100%) et temps de réponse moyen (10s) pour un seuil de fonctionnalité et d' $E/S$  de 0.5.

Le taux de rappel le plus élevé (100%) est obtenu pour les configurations de seuil {seuil de fonctionnalité=0.5, seuil d' $E/S$  = 0.5} et {seuil de fonctionnalité=0, seuil d' $E/S$ =0.5}. Cependant, dans le cas du seuil de fonctionnalité=0 qui simule la non considération de l'étape de d'appariement de la fonctionnalité, les taux de précision ne sont pas optimaux (75%) et le temps de réponse augmente considérablement (83s). Cela montre l'importance de notre première étape d'appariement sémantique de la fonctionnalité qui exclut rapidement un grand nombre de services non pertinents comme le montrent les résultats de l'exemple de la requête R2 (voir le tableau 7.4) concernant les deux services *ConvPerdido* et *Autocompletion*. Ainsi, cette étape réduit les faux positifs de 15% et diminue le temps de réponse de 73s.

En termes de précision, l'approche atteint le taux plus élevé (90%) pour les configurations de seuil {seuil de fonctionnalité=0.5 , seuil d' $E/S$ =0.5} et {seuil de fonctionnalité=1.0 , seuil d' $E/S$ =0.5}. Cependant, dans le cas du seuil de fonctionnalité=1.0, qui simule un appariement sémantique strict et rigide de la fonctionnalité, le taux de rappel est en diminution (79%). De même, si nous appliquons une correspondance stricte des  $E/S$  (seuil  $E/S$ =1.0), cela entraîne une diminution du taux de rappel, comme le montrent les résultats de l'exemple de requête R1 (voir tableau 7.3). Ceci montre le rôle important joué par la méthode d'appariement sémantique hybride proposée qui permet de calculer des scores d'appariement numériques pour la fonctionnalité et les  $E/S$  tout en évitant les inconvénients tels que le principe de favoriser les concepts voisins, ou la non distinction le type de paramètres à apparier. Cependant, le taux de précision comprend un faible

pourcentage de faux positifs (10%). Selon notre analyse des différentes requêtes utilisées dans l'ensemble des tests, les faux positifs peuvent se produire dans le cas où toutes les entrées requises par le service sont fournies mais qu'un grand nombre de sorties requises par la requête ne sont pas fournies par le service. La raison en est le fait de donner une pondération similaire au score d'appariement sémantique des entrées et des sorties (voir équation 5.3a). A l'avenir, pour une meilleure satisfaction des besoins de l'utilisateur, il devrait être possible pour l'utilisateur de définir les valeurs de pondération afin de calculer le score global de correspondance sémantique des  $E/S$ . L'analyse combinée des taux de précision et de rappel montrent que l'approche proposée réussit à rechercher des services géospatiaux correspondants aux besoins utilisateur avec un taux de rappel maximal et un taux de précision avec un faible pourcentage de faux positifs. En termes de temps de réponse, les évaluations montrent que l'approche permet de fournir une réponse pour un temps moyen de 10 secondes, mais cette valeur varie selon le nombre d' $E/S$  de la requête et des services à appairier.

Au meilleur de nos connaissances, il n'existe pas de corpus de *SWG* décrits sémantiquement, ni de plateforme d'évaluation de *Matchmakers* pour qu'ils soient utilisés pour la comparaison de nos travaux. Pour ces raisons, nous résumons dans ce qui suit les principaux avantages de notre approche par rapport aux approches d'appariement sémantiques dans le domaine géospatial utilisant un appariement sémantique exact strict ou un appariement sémantique logique rigide sur l'ensemble des  $E/S$  de services.

1. La première étape d'appariement de la fonctionnalité permet d'exclure rapidement un grand nombre de services dont la fonctionnalité offerte ne correspond pas à la fonctionnalité recherchée et permet ainsi d'augmenter le taux de précision en diminuant les faux positifs et elle permet également de diminuer le temps de réponse.
2. L'utilisation de la méthode d'appariement sémantique hybride, et l'optimisation de la méthode d'agrégation permet d'augmenter le taux de rappel et de précision. La méthode non logique permet de calculer un score d'appariement ce qui permet de diminuer les faux positifs générés quand la chaîne de subsumption est longue. La méthode logique basée sur des liens individuels entre les paramètres du service et de la requête permet de diminuer les faux positifs qui peuvent être générés par la méthode non logique comme la non distinction du type d'élément de la description à appairier ou la favorisation des concepts voisins.
3. L'attribution d'un score d'appariement numérique permet de classer les services selon les scores d'appariement les plus élevés.

Concernant la configuration à 3 étapes, Nous pouvons rappeler en premier que nous avons mentionné l'avantage de l'approche proposée pour calculer un score de d'appariement sémantique qui permet de classer les services découverts dans un ordre de pertinence. Cependant, dans certains cas, les scores obtenus (scores fonctionnels calculés sur les paramètres d' $E/S$ ) après les deux étapes primaires sont similaires, ce qui donne des ensembles de services équivalents sans ordre. Nous pouvons l'observer sur les tableaux présentant les résultats des requêtes R1 (voir tableau 7.3) et R2 (voir tableau 7.4). En effet, en considérant un seuil=0.5 pour l'appariement de la fonctionnalité et des  $E/S$ , nous voyons deux ensembles de services équivalents pour R1 (score de 0.97 et score de 0.72) et un ensemble pour R2 (score de 1).

Dans ce contexte, la troisième étape tente d'affiner les résultats et de classer les services en calculant un score global appelé "score de sélection" en fonction des scores fonctionnels et non fonctionnels (voir l'équation 5.5). Pour calculer les scores non fonctionnels et les

scores globaux, deux pondérations  $\alpha$  et  $\beta$  sont considérées (voir équation 5.4 et équation 5.5). Dans ces expérimentations,  $\alpha$  a été fixée à 1 afin de ne considérer que les propriétés objectives et  $\beta$  à 0.5 pour une considération équitable des scores fonctionnels et non fonctionnels. Seules les propriétés objectives ont été prises en compte en raison de la faible représentation des propriétés subjectives dans le corpus de services.

Les propriétés objectives comprennent des propriétés génériques et des propriétés spécifiques aux services géospatiaux. Les valeurs des propriétés sont le plus souvent simulées de manière plus ou moins aléatoire. C'est le cas pour les services présentés en réponse à la requête R1. Mais, pour les services liés à la requête R2, nous supposons que les valeurs des propriétés objectives génériques sont similaires pour les cinq services sélectionnés et nous calculons les valeurs des propriétés objectives spécifiques à la fonctionnalité de géocodage (voir tableau 4.2). Les tableaux 7.5 et 7.6<sup>4</sup> montrent les scores de recommandation ou les scores d'aide à la sélection pour la requête R1 et la requête R2 respectivement.

TABLE 7.5 – Résultats de sélection associés à la requête R1

<b>Service</b>	<b>Score fonctionnel</b>	<b>Score non fonctionnel</b>	<b>Score global de sélection</b>
Euclidean Distance	0.97	0.95	0.96
Euclidean Allocation	0.97	0.95	0.96
Euclidean Direction	0.97	0.90	0.93
Cost Distance	0.72	0.95	0.83
Cost Allocation	0.72	0.95	0.83
Cost BackLink	0.72	0.73	0.73

TABLE 7.6 – Résultats de sélection associés à la requête R2

<b>Service</b>	<b>Score fonctionnel</b>	<b>Score non fonctionnel</b>	<b>Score global de sélection</b>
GeoNames	1.0	0.90	0.95
GetToponym	1.0	0.90	0.95
TGN	1.0	0.82	0.91
SwissNames	1.0	0.77	0.88
OS50k	1.0	0.72	0.86

Dans les deux cas, les scores de sélection ont affiné les ensembles de services équivalents. Pour les résultats de R2, chaque ensemble de services a été divisé en deux sous-ensembles. Alors que pour R1, l'ensemble initial a été divisé en quatre sous-ensembles, ce qui permet une meilleure distinction des services. On peut supposer que le résultat est meilleur pour R2 car les propriétés spécifiques sont vraiment discriminantes puisque les propriétés génériques ont les mêmes valeurs pour tous les services. De plus, les valeurs des propriétés spécifiques sont des valeurs réelles ce qui rend le résultat plus significatif. Ces exemples montrent que la prise en compte des propriétés non fonctionnelles, même de manière secondaire, apporte une réelle valeur ajoutée à l'aide à la sélection de *SWG*.

4. Nous considérons la variante du service GetToponyms qui utilise la nomenclature GeoNames.

## 7.3 Mise en oeuvre de l'approche de composition automatique de SWG

### 7.3.1 Architecture fonctionnelle de la composition de SWG

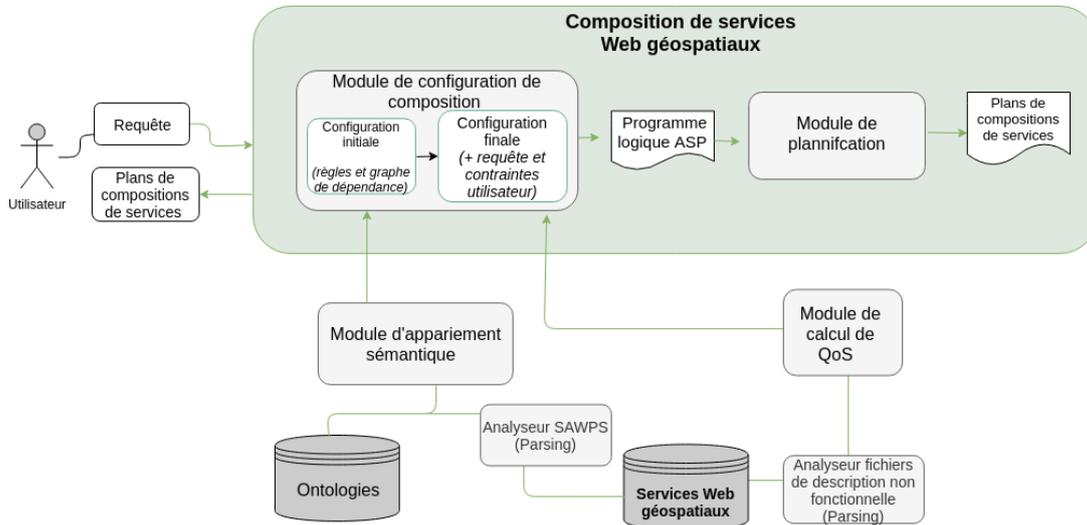


FIGURE 7.10 – Architecture fonctionnelle de la composition de SWG

Pour implémenter les méthodes proposées dans le chapitre 6, nous développons un système pour la composition de SWG implémenté en Java et en ASP. L'architecture fonctionnelle est illustrée dans la figure 7.10, elle regroupe principalement six modules. Les deux modules analyseurs ont été présentés dans la section 7.2. L'environnement de composition utilise les modules analyseurs développés pour la découverte de SWG. L'environnement de composition fait appel aussi à des méthodes réalisées par les deux modules d'appariement sémantique et de calcul de scores de *QoS* présentés dans la section 7.2. Les deux modules restants sont définis comme suit :

- Module de la configuration de composition : ce module se charge de configuration de l'environnement de composition. Dans un premier temps il définit des règles de composition écrites en ASP et il calcule un graphe de dépendance de SWG en faisant appel au module d'appariement sémantique et au module de calcul de *QoS* avec adaptation au contexte de composition. Le graphe est décrit en ASP et est conjointement inséré avec les règles de composition dans un programme logique ASP. Dans un second temps, ce module reçoit la requête et les contraintes utilisateur, calcule les services de début de composition, des services buts et décrit l'ensemble en langage ASP pour l'insérer dans le programme logique ASP.
- Module de planification et sélection ASP : ce module se charge de calculer des modèles stables ASP où chaque modèle stable représente une solution de composition possible. Ce module fait appel à un solveur ASP. Dans le cadre de cette thèse nous utilisons Clingo qui inclut les deux composants Gringo[45] et Clasp[43].

### 7.3.2 Cas d'étude et expérimentations sur l'approche de composition

Nous présentons dans ce qui suit **un cas d'étude** dans le cadre du projet Choucas. L'un des objectifs du projet Choucas est de proposer une approche pour la mise en correspon-

dance automatique des traces, des textes de description des randonnées et des ressources géographiques (répertoires géographiques et bases de données géographiques). Une des tâches nécessaires dans cette approche est l'identification des entités spatiales nommées et la correspondance des toponymes. Plus précisément, dans ce cas d'étude, nous évaluons l'attribution de la géolocalisation à des entités spatiales dans un texte. Pour l'étude de cas d'expérimentation, nous avons travaillé sur la composition d'une chaîne de services qui retourne la géolocalisation d'entités spatiales extraites d'un texte donné en entrée. Nous avons à notre disposition la description sémantique des services de notre corpus décrit dans la section 7.1 et parmi eux sept services. Un service permet d'interroger un analyseur morphosyntaxique (Postagger), un deuxième service permet l'identification des entités spatiales nommées (NERC) et les cinq services restants permettent la géolocalisation des entités spatiales. (GeoNames, OS50k, SwissNames, TGN, GetToponyms).

Nous considérons la requête utilisateur présentée dans le tableau 7.7.

TABLE 7.7 – Requête R3

Fonctionnalités	Entrées	Sorties
Geosemantic analysis service, Geographic annotation service, Direct geocoding service	TextDescription	GeographicCoordinates

Pour l'**application de notre approche de composition**, nous fixons les seuils d'appariement de la fonctionnalité et d'*E/S* à 0.5 puisque ce sont les valeurs de seuils optimales obtenues dans le cadre des expérimentations de l'approche de découverte dans la section 7.2. Également, dans ces expérimentations, pour calculer les scores non fonctionnels et les scores globaux, deux pondérations sont considérées (voir équation 5.4 et équation 6.1a).  $\alpha$  a été fixée à 1 afin de ne considérer que les propriétés objectives<sup>5</sup> et  $\beta$  à 0.5 pour une considération équitable des scores fonctionnels et non fonctionnels.

Avant la réception de la requête, **la première étape** de l'approche de composition automatique consiste à la représentation des règles de composition dans un programme logique *ASP* (voir section. 6.2.3) suivie par la création d'un graphe de dépendance entre tous les services du répertoire (voir section 6.2.1). Par exemple, une dépendance entre deux services géospatiaux est définie en *ASP* comme suit :

— `link_dep("NERC","GetToponyms","SpatialNamedEntity","SpatialEntity",91).`

Ceci se traduit par le service "NERC" peut être composé avec le service "GetToponyms" pour satisfaire l'entrée annotée sémantiquement avec le concept "SpatialEntity" avec un score d'appariement sémantique égal à 0.91.

Les scores non fonctionnels de tous les services appartenant au graphe sont également représentés en *ASP*. Par exemple, les scores des services "NERC" et "GetToponyms" sont définis comme suit :

— `service_nfs("NERC",95).`

— `service_nfs("GetToponyms",90).`

Ceci se traduit par le service "NERC" possède un score non fonctionnel égal à 0.95 et le service "GetToponyms" possède un score non fonctionnel égal à 0.90.

Après la réception de la requête utilisateur, **la seconde étape** consiste au calcul des services satisfaisant les différentes fonctionnalités, les services de début de composition et

5. Comme pour les expérimentations concernant la découverte, seules les propriétés objectives ont été prises en compte en raison de la faible représentation des propriétés subjectives dans le corpus de services.

les services buts de la composition et enfin elle représente ces informations en *ASP* (voir section 6.2.1). Par exemple, un extrait de la requête R3 est défini comme suit :

- `include_fonc("Geosemantic_analysis_service").`
- `satisfy_fonc("NERC","Geosemantic_analysis_service").`
- `request_out("GeographicCoordinates").`
- `service_start("Postagger",100).`
- `service_goal("GetToponyms","GeographicCoordinates","GeographicCoordinates",100).`

Ceci se traduit par la fonctionnalité "Geosemantic \_analysis\_service" est recherchée par la requête utilisateur, le service "NERC" satisfait cette fonctionnalité. Le concept de sortie "GeographicCoordinates" est recherché par la requête utilisateur. Le service "Postagger" peut figurer en début du schéma de composition avec un score d'appariement sémantique entre les entrées de la requête et les entrées du service "Postagger" égale à 1. Le service "GetToponyms" peut figurer en fin du schéma de composition pour satisfaire le concept de sortie "GeographicCoordinates" recherché par la requête utilisateur avec un score d'appariement sémantique égal à 1.

**La troisième étape** consiste à calculer les modèles stables *ASP* qui représentent les différentes compositions possibles. Cette étape est assurée par le solveur *ASP*. Pour la requête R3, l'ensemble d'atomes suivants, représentant les cinq modèles stables calculés est retourné :

- Composition 1 : `chain(1,"Postagger") chain(2,"NERC") chain(3,"GetToponyms")`
- Composition 2 : `chain(1,"Postagger") chain(2,"NERC") chain(3,"GeoNames")`
- Composition 4 : `chain(1,"Postagger") chain(2,"NERC") chain(3,"TGN")`
- Composition 5 : `chain(1,"Postagger") chain(2,"NERC") chain(3,"SwissNames")`
- Composition 3 : `chain(1,"Postagger") chain(2,"NERC") chain(3,"OS50k")`

Chacun des modèles stables (ensemble d'atomes retournés) représente une solution de composition possible. Chacun des atomes `chain(N,S)` représente un service et sa position dans la chaîne de composition. Dans cet exemple un schéma séquentiel est présenté mais l'approche peut proposer des schémas avec actions parallèles. Si deux services ont la même position, cela veut dire qu'ils peuvent s'exécuter en parallèle.

Concernant le calcul de scores globaux de sélection de chacune des compositions candidates qui incluent un score fonctionnel et un score non fonctionnel pour la requête R3, les scores des 5 compositions candidates sont représentés dans le tableau 7.8. Ceci a permis de présenter à l'utilisateur un classement de ces 5 compositions candidates.

TABLE 7.8 – Résultats de sélection associés à la requête R3

Composition	Score fonctionnel	Score non fonctionnel	Score global de sélection
Composition 1	0.97	0.93	0.95
Composition 2	0.97	0.93	0.95
Composition 3	0.97	0.90	0.94
Composition 4	0.97	0.89	0.93
Composition 5	0.97	0.87	0.92

De plus, des contraintes supplémentaires peuvent être appliquées pour répondre à des exigences supplémentaires de l'utilisateur comme l'utilisation d'un service spécifique

dans la chaîne de composition. Par exemple, dans ce cas d'étude, nous avons supposé que l'utilisateur souhaite utiliser le service "GetToponyms". Ceci est défini par l'atome :

— `use_service("GetToponyms")`.

Par l'ajout de cette contrainte, seule la composition 1 sera retournée.

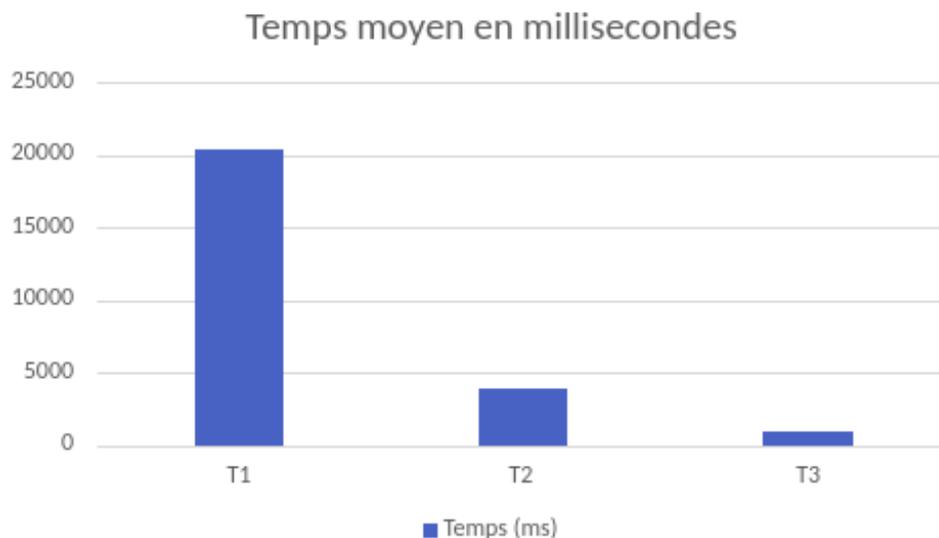


FIGURE 7.11 – Temps moyen de réponse de l'environnement de composition.

Dans l'objectif d'effectuer les premières expérimentations de notre approche de composition, nous avons mené des tests sur 10 requêtes différentes sur des compositions qui incluent des services appartenant de 2 à 5 classes de fonctionnalités. Sur l'ensemble des 10 requêtes, les taux de rappel et de précision étaient maximaux. Nous avons également calculé les temps moyens de réponse de notre environnement de composition.

Nous avons calculé trois temps T1, T2, et T3. Le temps T1 représente le temps de calcul des services satisfaisant les fonctionnalités exigées par l'utilisateur. Le temps T2 représente le temps de calcul des services de début et de fin, alors que le temps T3 représente le temps de recherche de modèles stables. La figure 7.11 montre les résultats obtenus. Nous remarquons que le temps T1 est le temps le plus élevé, il dépend fortement du nombre de fonctionnalités incluses puisque c'est le temps qui consiste à rechercher les services satisfaisant les différentes fonctionnalités et ça consiste en une activité consommatrice en temps. Le temps T2 est moins important puisqu'il consiste à la recherche des services de début et des services buts dans le sous-ensemble de services préalablement sélectionné durant le temps T1. Ainsi, il dépend du nombre de services satisfaisants les fonctionnalités exigées. Nous notons que malgré que ces temps sont importants dû au raisonnement sémantique qui est coûteux en termes de temps de traitement, nous estimons que ces temps sont acceptables. Particulièrement, pour les utilisateurs finaux qui souhaitent obtenir de nouvelles compositions de *SWG* à la demande lors de l'exécution. Nous expliquons cela par le fait que notre approche propose un niveau d'automatisation élevé par rapport aux approches qui nécessitent la conception d'un workflow abstrait ou le calcul de services abstraits avant la construction des schémas finaux de compositions. Enfin le temps T3 est le temps le plus important puisqu'il s'agit du temps de calcul de solutions de compositions possibles. L'approche calcule les solutions possibles en une moyenne de temps d'une seconde. Ces premières expérimentations, nous ont permis de montrer la faisabilité de cette approche de composition et de montrer les principaux avantages qu'elle offre. L'approche

se base sur une méthode d'appariement sémantique qui a été préalablement expérimentée dans la section 7.2 pour configurer un environnement de composition optimale. Un deuxième avantage de cette approche est son appui sur une méthode logique en *ASP* qui permet de résoudre le problème de composition après la configuration de l'environnement de composition dans un temps moyen d'une seconde, grâce aux performances du solveur *ASP*. De plus, des règles peuvent facilement être ajoutées pour générer la planification avec des contraintes, ce qui permet d'ajouter des critères de composition supplémentaires et de satisfaire au mieux les besoins utilisateurs. Enfin, comme nous l'avons cité précédemment, l'approche ne nécessite pas la conception d'un workflow abstrait ou le calcul de services abstrait, ce qui lui attribue un niveau d'automatisation plus élevé par rapport aux approches qui passent par ce type de conception.

## 7.4 Conclusion

Dans ce chapitre, nous avons proposé des expérimentations des approches de découverte, et composition de *SWG* introduites, afin de valider nos contributions. Dans un premier temps, nous avons appliqué notre approche de description sémantique sur un corpus de services *WPS*. En appliquant cette approche nous fournissons un moyen d'étendre et raffiner la description des services *WPS* grâce aux annotations et la précision de la nature de relation entre les concepts et les éléments annotés dans une description *WPS*, c'est à dire la nature de l'annotation pour une bonne exploitation des métadonnées ajoutées. Les extensions et les enrichissements proposés n'auront pas d'impact sur la conformité à la norme ni sur le processus d'exploitation de la description syntaxique. Cette approche de description sémantique, appliquée au corpus que nous avons créé, nous a permis d'exploiter les annotations sémantiques pour effectuer des expérimentations sur l'approche de découverte proposée. L'approche de découverte, permet à l'utilisateur de paramétrer la découverte selon ses exigences en introduisant un seuil minimal d'acceptance pour l'appariement sémantique de la fonctionnalité et des *E/S* ainsi que des valeurs de pondérations pour le calcul des scores non fonctionnels. Nous avons fait varier ces seuils pour effectuer les expérimentations sur notre approche. Les expérimentations ont montré que l'approche offre de meilleurs taux de précision et de rappel et également en termes de temps de réponse moyen pour un seuil de fonctionnalité et d'*E/S* égale à 0.5 avec un faible pourcentage de faux positifs. Cette approche permet ainsi de rechercher des services géospatiaux correspondants aux besoins utilisateurs et de classer ces services selon les scores d'appariement sémantique les plus élevés. Nous avons également montré que dans certains cas les scores d'appariement sémantique fonctionnels peuvent être identiques et ne permettent pas de classer les services. Nous avons procédé alors au calcul de scores de sélection avec une prise en compte de score fonctionnel et non fonctionnel. Cette dernière étape a montré l'importance de la prise en compte de propriétés non fonctionnelles pour l'aide à la sélection de *SWG*. Enfin, nous avons appliqué notre approche de composition sémantique automatique de *SWG* sur un cas d'étude dans le cadre du projet Choucas et nous avons effectué les premières expérimentations avec un calcul des temps moyens de réponse de l'environnement de composition. Les principaux avantages de cette approche est qu'elle se base sur la méthode d'appariement sémantique proposée pour configurer l'environnement de composition d'un côté, et s'appuie sur une représentation logique en *ASP* du problème de composition qui ensuite est résolu grâce aux performances du solveur *ASP*. De plus, des règles peuvent facilement être ajoutées pour générer des solutions de compositions avec des contraintes, ce qui permet d'ajouter des critères de composition

supplémentaires. Un autre avantage de cette approche est le niveau d'automatisation élevé par rapport aux approches qui nécessitent la conception d'un workflow abstrait ou le calcul de services abstraits avant la construction des schémas finaux de compositions.

Au vue des différents résultats d'expérimentations et de cas d'utilisation présentés, nous estimons que les trois approches proposées présentent un environnement cohérent pour la description, la découverte et la composition de *SWG* et spécifiquement de services *WPS*, en relevant les différents défis posés pour pallier aux problématiques identifiées dans la littérature (voir tableau 2.3).



# Conclusion générale



# Conclusion générale

## Sommaire

---

Bilan . . . . .	125
Perspectives . . . . .	127

---

## Bilan

Le principal défi de cette thèse était d'améliorer le processus de découverte de *SWG* en proposant un cadre général pour la description, la découverte et la composition de *SWG* tout en préservant la compatibilité avec les standards actuels et plus précisément le standard *WPS*. Nous avons proposé une solution pour la description de *SWG* basée sur les technologies du Web sémantique. Cette solution a été appliquée au modèle de description syntaxique *WPS* comme une proposition d'extension vers un modèle de description sémantique. Nous avons exploité cette proposition de description sémantique pour proposer une approche de découverte sémantique de *SWG* et une approche de composition automatique de *SWG*. L'approche de composition a été proposée pour pallier aux cas d'absence de services atomiques qui puissent répondre aux besoins d'un utilisateur où nous sommes amenés à composer plusieurs services afin de satisfaire une requête utilisateur.

**Approche de description sémantique de *SWG*.** Nous avons proposé une approche de description sémantique à base d'annotations sémantiques pour résoudre les problématiques liées aux descriptions syntaxiques de *SWG*. Un méta-modèle qui représente une abstraction du nouveau modèle de description a été présenté pour exprimer les éléments pertinents d'une description des *SWG*. Plus précisément, il utilise des annotations sémantiques et des ontologies de domaine pour apporter une couche sémantique pour la description de la fonctionnalité et des *E/S* des *SWG*. Également, nous avons défini une annotation pour la description de propriétés non fonctionnelles subjectives et objectives (génériques et spécifiques au domaine géospatial) pour la représentation de la qualité des *SWG*. Le méta-modèle a fait l'objet d'une proposition d'extension du modèle de description de services *WPS*. Pour effectuer des expérimentations sur nos approches pour la découverte et la composition de *SWG*, nous avons créé un corpus de 94 services *WPS* décrit syntaxiquement selon le modèle de description *WPS* 2.0. Nous avons, par la suite décrit sémantiquement ces services suivant l'approche de description proposée.

**Approche de découverte sémantique de *SWG*.** Nous avons proposé une approche pour la découverte sémantique des *SWG*. Cette approche est basée sur une méthode d'appariement sémantique hybride qui combine une mesure de similarité avec des liens logiques individuels en prenant en considération les types des éléments de description à appairer. La mesure de similarité permet de calculer un score numérique qui représente la similarité sémantique entre deux concepts dans une ontologie en se basant sur la profondeur des

concepts, ce qui lui donne une capacité à calculer la similarité sémantique des concepts de manière fidèle, c'est à dire en respectant la représentation hiérarchique des connaissances dans l'ontologie. Cependant celle-ci ne peut pas être utilisée directement pour faire correspondre des services Web puisqu'elle est symétrique, les types de concepts à appairer doivent être différenciés selon le rôle du concept dans la description du service. La méthode logique définit des filtres sur des liens individuels entre les différents éléments de description, ce qui permet de calculer la similarité entre un service et une requête d'une manière plus flexible par rapport aux méthodes logiques qui définissent des filtres sur l'ensemble de la signature du service. Cependant ces filtres peuvent tolérer des faux positifs au niveau du score de subsumption surtout si la chaîne de subsumption, reliant les deux concepts à comparer, est longue. Ainsi, la combinaison des deux méthodes leur permet de se compléter pour diminuer les faux positifs qui peuvent être générés. Le processus de découverte global implique le calcul d'un score d'appariement fonctionnel qui inclut l'appariement sémantique de la fonctionnalité et des  $E/S$  qui utilise l'approche d'appariement sémantique hybride proposée, et en plus une fonction qui optimise l'agrégation des résultats d'appariement élémentaire des  $E/S$ . Le processus de découverte implique également le calcul d'un score de qualité des  $SWG$  en utilisant les valeurs de propriétés non fonctionnelles des  $SWG$ . L'appariement de la fonctionnalité vise à exclure rapidement un grand nombre de services dont la fonctionnalité offerte ne correspond pas à la fonctionnalité recherchée, ce qui permet d'augmenter le taux de précision en diminuant les faux positifs et permet également de diminuer le temps de réponse. L'appariement des  $E/S$  permet d'affiner la découverte et ainsi augmenter davantage le taux de précision. Enfin le calcul d'un score de  $QoS$  permet de proposer à l'utilisateur les services les plus appropriés parmi ceux qui sont candidats. Cette approche a été mise en oeuvre et une série de tests sur un ensemble de services  $WPS$  a été réalisée afin d'effectuer des expérimentations sur cette approche. Les tests ont montré que la l'approche proposée, donne de bons résultats comparé aux approches de découverte sémantique de  $SWG$  qui utilisent une approche d'appariement sémantique exact strict qui engendre un faible taux de rappel et ceux qui utilisent une approche d'appariement logique rigide qui engendre un faible taux de précision à cause de la tolérance d'un nombre important de faux positifs. En outre, ces expérimentations ont mis en évidence le problème d'absence de services atomiques qui puissent répondre à une requête utilisateur et le besoin de composition automatique. Pour remédier à ce problème, nous avons proposé une approche de composition de  $SWG$ .

**Approche de composition automatique de  $SWG$ .** Nous avons exprimé une solution pour la composition automatique de  $SWG$  basée sur une technique de planification en utilisant le langage de programmation logique par contraintes  $ASP$ . Plus précisément, nous avons représenté les éléments et les critères de composition, dans un programme logique  $ASP$  et les solutions de compositions possibles sont représentées par un ensemble de modèles stables  $ASP$  calculés par un solveur  $ASP$ . La représentation de certains éléments nécessite des calculs de scores d'appariement sémantique. Dans ce contexte nous avons adapté à notre environnement de composition, la méthode d'appariement sémantique hybride proposée dans le cadre de notre contribution pour la découverte de  $SWG$ , ce qui présente le premier avantage de cette approche. Un deuxième avantage de cette approche est la flexibilité de la définition des règles et contraintes de composition grâce à l'utilisation du langage  $ASP$  pour proposer une solution de composition automatique qui génère un plan de composition qui contient des actions en séquence ou en parallèle. De plus nous avons proposé une stratégie de sélection de compositions candidates dans le but de trouver le meilleur plan de composition parmi les compositions candidates selon deux critères

d'optimisation. Le premier est le degré d'appariement sémantique entre les services participants à un plan de composition. Le deuxième est la qualité des services (score non fonctionnel) participants à un plan de composition. Un autre avantage de cette approche est le niveau d'automatisation élevé par rapport aux approches qui nécessitent la conception d'un Workflow abstrait ou le calcul de services abstraits avant la construction des plans finaux de compositions.

## Perspectives

Au terme de notre travail, nous avons identifié trois principales perspectives à court, moyen, et long termes dans le but de poursuivre les travaux scientifiques présentés dans ce manuscrit. D'un point de vue pratique, la première perspective se rapporte à l'extension de frameworks implémentant le standard WPS avec les approches proposées. D'un point de vue théorique, la deuxième perspective concerne l'enrichissement des propriétés non fonctionnelles spécifiques au domaine géospatial. Enfin, la troisième perspective concerne l'amélioration de la stratégie et des performances de sélection de compositions candidates.

**Extension de frameworks WPS existants.** Pour intégrer la solution proposée dans un framework WPS, nous devons en premier étendre les descriptions de services WPS existants selon l'approche d'annotation sémantique proposée, ensuite nous procéderons à l'intégration des composants qui implémentent nos approches de découverte sémantique et de composition automatique. Le composant de composition nécessite également la traduction du plan de sortie de composition dans un langage d'orchestration, nous citons comme exemple le langage *BPEL*<sup>6</sup>. Un moteur d'orchestration pourra ensuite exécuter les services Web appartenant à la chaîne de composition de services.

Ces méthodes supplémentaires pour les services WPS peuvent être considérées comme une extension possible de la norme WPS. Dans cette condition, les frameworks WPS seront en mesure de décrire sémantiquement les services (*Process*) WPS, de découvrir sémantiquement les services (*Process*) WPS annotés et de permettre à différents référentiels de découvrir sémantiquement ces services annotés, et enfin de composer automatiquement les services WPS annotés. Parmi les frameworks WPS les plus réputés aujourd'hui nous citons : 52° North<sup>7</sup>, PyWPS<sup>8</sup>, et ZOO-Project<sup>9</sup>. Ces Frameworks intègrent aujourd'hui les fonctionnalités de description, d'intégration, de publication et d'exécution de services WPS. L'objectif de l'extension que nous proposons est d'étendre le modèle de description syntaxique vers un modèle de description sémantique et de proposer de nouveaux composants pour la découverte sémantique et la composition automatique de services WPS.

**Enrichissement des propriétés non fonctionnelles géospatiales** Á long terme, une de nos principales pistes de recherche consiste à enrichir la description de services avec les propriétés non fonctionnelles spécifiques au domaine géospatial. Dans ce manuscrit, nous avons uniquement présenté des propriétés non fonctionnelles spécifiques à la catégorie de géocodage. Une solution serait de représenter formellement les propriétés non fonctionnelles spécifiques à chaque catégorie géospatiale dans une ontologie afin d'unifier ces connaissances dans la même philosophie que pour les propriétés fonctionnelles.

---

6. *BPEL* : <https://www.oasis-open.org/committees/download.php/23964/wsbpel-v2.0-primer.htm>

7. 52° North : <https://52north.org/>

8. PyWPS : <https://pywps.org/>

9. ZOO-Project : [ZOO-Project:http://www.zoo-project.org/](http://www.zoo-project.org/)

**Amélioration de la stratégie et des performances de sélection de compositions candidates** Nous avons proposé une stratégie de sélection de compositions de *SWG* qui inclut le calcul de score non fonctionnel dans le score final de classement de compositions candidates. Afin d'améliorer cette stratégie nous envisageons la prise en compte du profil utilisateur (tel que recommandé par [55]) pour le calcul du score non fonctionnel. Plus spécifiquement, nous envisageons la prise en considération des préférences utilisateur en s'appuyant sur l'historique (l'utilisation de caches) pour enregistrer les requêtes et les mises en correspondance déjà effectuées. Ceci peut également améliorer les performances de l'approche de composition en termes de temps de réponse et particulièrement le temps de la configuration de la composition après réception de la requête.

# Annexe A

## Préliminaires : *ASP* et modèles stables

La Programmation par ensembles réponses *ASP* [95] est une forme de programmation déclarative non monotone qui offre un bon cadre pour la formulation de problèmes en Intelligence Artificielle comme la planification [69]. Le paradigme de programmation *ASP* fournit un cadre général pour la résolution de problème d'optimisation et de décision [61]. L'approche a été rendue populaire grâce au développement de plusieurs solveurs performants dédiés à l'*ASP* comme smodels [127], DLV [86] et dernièrement Clasp [43], et notamment ceux basés sur la procédure DPLL et les solveurs SAT comme ASSAT [90].

L'idée de base de l'*ASP* est d'exprimer un problème donné sous forme d'un programme logique pour lequel on doit chercher les modèles stables qui représentent les solutions du problème originel. Pour avoir une expression concise du problème, on utilise dans un premier temps, des règles construites à base de la logique du premier ordre. En conséquence, le problème sera exprimé par un programme logique dit souvent programme non terminal contenant des prédicats  $P$ , avec des variables  $V$  et des constantes  $C$ . Des systèmes appelés en anglais "*grounders*" ont été conçus pour calculer l'ensemble des instances terminales des règles de ce programme non terminal donnant en conséquence un programme logique terminal équivalent. La figure 7.12 illustre le fonctionnement général d'un système *ASP* [69].

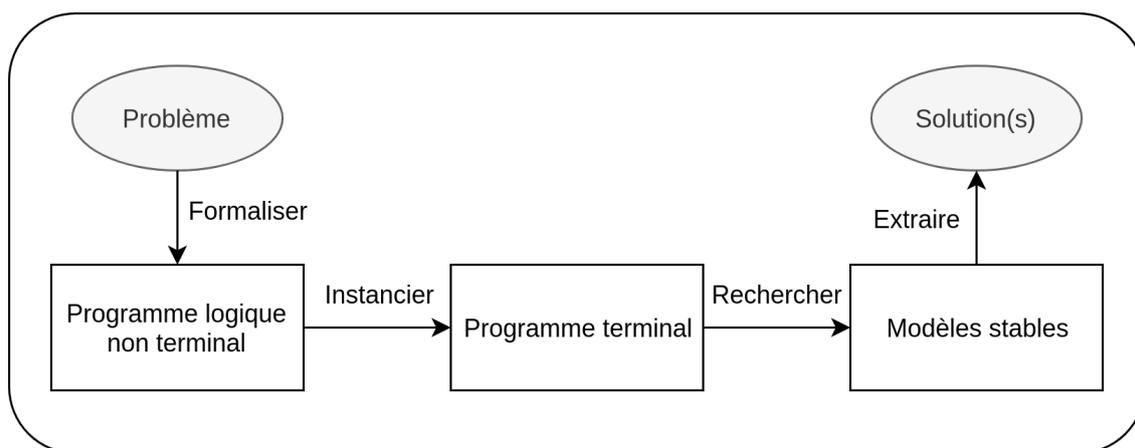


FIGURE 7.12 – Modèle conceptuel du paradigme *ASP* [69]

**Définition 12.** *Un programme logique général  $\pi$  est un ensemble de règles se terminant par un point '?. Les connecteurs ':-' et ',' apparaissent dans ces règles et peuvent être interprétés comme étant 'si' et 'et' respectivement. Précisément, le programme logique  $\pi$  est un ensemble de règles de la forme :*

$a_0 :- a_1, \dots, a_m, \text{not } a_{m+1}, \dots, \text{not } a_n$  , où  $0 \leq m \leq n$ , et chaque  $a_i$  est un atome<sup>10</sup>,  $1 \leq i \leq n$  et *not* le symbole exprimant la négation par échec.

Les côtés gauche et droit sont respectivement appelés la *tête* et le *corps* de la règle. Une règle peut avoir soit une *tête* vide, soit un *corps* vide, mais pas les deux. Les règles avec une *tête* vide sont appelées des *contraintes* puisque une tête vide représente implicitement le faux ; les règles avec un *corps* vide sont appelées des *faits*.

a.

FIGURE 7.13 – Exemple d’une relation qui est considérée vraie sans condition (un fait).

a :- b.

FIGURE 7.14 – Exemple d’une règle avec l’atome  $a$  vrai si les atomes  $b$  et  $c$  sont vrais.

:- a.

FIGURE 7.15 – Exemple d’une contrainte qui écarte tous les modèles avec l’atome  $a$  vrai.

Le langage permet l’utilisation de variables, mais seulement indirectement, en forçant l’évaluation des variables (grounding), en utilisant un système de “*grounder*”.

Les variables sont simplement considérées comme des substituts pour tout élément de  $C$  ; ainsi, une règle avec variables (appelée règle non fondée ou *non-ground rule*) est simplement un symbole syntaxique pour l’ensemble des règles obtenu en remplaçant systématiquement chaque variable par tout élément de  $C$  (appelées instances fondées ou *ground rules*).

Un ensemble d’atomes fondés (*ground atoms*)  $X$  satisfait le corps d’une règle fondée (*ground rule*) si  $\{a_{m+1}, \dots, a_n\} \cap X = \emptyset$  et  $\{a_1, \dots, a_m\} \subseteq X$ . Une contrainte est satisfaite par  $X$  si son corps n’est pas satisfait par  $X$ .  $X$  est un modèle de règle si il satisfait le *corps* de la règle, ou  $a_0 \in X$ .  $X$  est un modèle du programme s’il satisfait toutes ses règles.

Un programme positif est un programme qui ne contient pas de négation par échec ou de négation classique. Il est connu que tout programme positif admet un seul modèle stable. C’est le plus petit ensemble d’atomes obtenu par l’application des règles du programme jusqu’à un point fixe. C’est le modèle minimal de Herbrand de l’ensemble de clauses de Horn formant le programme  $\pi$  ; il est noté  $Cn(\pi)$ .

Le réduit d’un programme  $\pi$  par rapport à un ensemble d’atomes  $X$  est le programme positif  $\pi^X$  obtenu à partir de  $ground(\pi)$  en supprimant :

10. Par défaut, tout atome est considéré faux, jusqu’à ce qu’il soit explicitement statué comme vrai. Si un atome est statué à vrai c’est à dire que c’est un fait, une relation qui est considérée vraie sans condition.

- chaque instance de règle ayant un atome *not*  $a_i$  dans son corps négatif tel que  $a_i \in X$  et
- tous les atomes *not*  $a_j$  tels que  $a_j \notin X$ , dans les corps négatifs des instances des règles restantes.

La sémantique la plus utilisée pour les programmes logiques généraux est celle des modèles stables [47]. Un ensemble d'atomes  $X$  est un modèle stable de  $\pi$  si et seulement si  $X = Cn(\pi^X)$ . En d'autres mots, l'ensemble d'atomes  $X$  est un modèle stable de programme  $\pi$  ssi  $X$  est identique au modèle minimal de Herbrand du réduit  $\pi^X$  du programme  $\pi$  par rapport à  $X$ .

Plusieurs extensions ont été introduites pour faciliter l'élaboration du programme. Par exemple, les règles de choix. Syntactiquement une règle de choix peut avoir la forme  $\{a_1, a_2, \dots, a_m\} :- a_{m+1}, \dots, a_n, \text{not } a_{n+1}, \dots, \text{not } a_o$  où  $0 \leq m \leq n \leq o$  et chaque  $a_i$  est un atome,  $1 \leq i \leq o$ . Sémantiquement si le corps est satisfait, par le modèle stable courant, alors n'importe quel sous-ensemble de  $\{a_1, a_1, \dots, a_m\}$  peut être inclus dans le modèle stable. Un autre exemple de règles de choix peut avoir la forme

$l \{a_1, a_2, \dots, a_m : a_{m+1}, \dots, a_n, \text{not } a_{n+1}, \dots, \text{not } a_o\} u$  où chaque  $a_i$  est un atome et  $0 \leq m \leq n \leq o$  et  $l$  et  $u$  sont des entiers où  $l \leq u$ . Sémantiquement si le corps est satisfait, par le modèle stable courant, alors je peux choisir un ou plusieurs éléments, qui peuvent être inclus dans le modèle stable, parmi le sous-ensemble de  $\{a_1, a_1, \dots, a_m\}$ . Les valeurs de  $l$  et  $u$  à gauche et à droite de la règle, représentant respectivement la borne minimale, par défaut égale à zéro et la borne maximale, par défaut l'infini.



# Bibliographie

- [1] The 52° north wps. <https://52north.org/software/software-projects/wps/>. En ligne ; consulté le 7 novembre 2020. 36, 73
- [2] Ibm knowledge center. [https://www.ibm.com/support/knowledgecenter/fr/SSRTLW\\_9.6.1/org.eclipse.jst.ws.doc.user/concepts/cws.html](https://www.ibm.com/support/knowledgecenter/fr/SSRTLW_9.6.1/org.eclipse.jst.ws.doc.user/concepts/cws.html). En ligne ; consulté le 18 juillet 2020. 26
- [3] Tei : Text encoding initiative. <http://www.tei-c.org/>. En ligne ; consulté le 20 juin 2020. 73
- [4] Uddi version 3.0.2. technical report. <http://www.uddi.org/pubs/uddi-v3.0.2-20041019.htm>, 2004. En ligne ; consulté le 20 juillet 2020. 34
- [5] I. 19107 :2019(fr). Iso/tc 211. (2019). iso 19107 :2019(fr), information géographique - géomatique. <https://www.iso.org/fr/standard/66175.html>. En ligne ; consulté le 10 octobre 2021. 73
- [6] I. 19119 :2016(fr). Iso/tc 211. (2016). iso 19119 :2016(fr), information géographique — services. <https://www.iso.org/obp/ui/#iso:std:iso:19119:ed-2:v1:fr>. En ligne ; consulté le 20 juin 2020. 52
- [7] I. 211. Iso/tc 211. information géographique/géomatique. <https://www.iso.org/fr/committee/54904.html>. En ligne ; consulté le 24 août 2020. 35
- [8] E. Acheson, S. De Sabbata, and R. S. Purves. A quantitative analysis of global gazetteers : Patterns of coverage for common feature types. *Computers, Environment and Urban Systems*, 64 :309–320, 2017. 70, 107
- [9] R. Akkiraju, J. Farrell, J. A. Miller, M. Nagarajan, A. P. Sheth, and K. Verma. Web service semantics-wsdl-s. *W3C*, 7, 2005. 40
- [10] S. Al-Areqi, A.-L. Lamprecht, and T. Margaria. Constraints-driven automatic geospatial service composition : Workflows for the analysis of sea-level rise impacts. In *International Conference on Computational Science and Its Applications*, pages 134–150. Springer, 2016. 10, 52, 53, 57
- [11] A. AlSedrani and A. Tourir. Web service composition processes : A comparative study. *International Journal on Web Service Computing (IJWSC)*, 7(1) :1–21, 2016. 50
- [12] A. A. Alwasouf and D. Kumar. Research challenges of web service composition. In *Software Engineering*, pages 681–689. Springer, 2019. 48

- [13] M. Andrei, A. Berre, L. Costa, P. Duchesne, D. Fitzner, M. Grcar, J. Hoffmann, E. Klien, J. Langlois, A. Limyr, et al. Swing : An integrated environment for geospatial semantic web services. In *European Semantic Web Conference*, pages 767–771. Springer, 2008. 52
- [14] Y. Bai, L. Di, and Y. Wei. A taxonomy of geospatial services for global service discovery and interoperability. *Computers & Geosciences*, 35(4) :783–790, 2009. 52
- [15] P. Bartalos and M. Bieliková. Automatic dynamic web service composition : A survey and problem formalization. *Computing and Informatics*, 30(4) :793–827, 2012. 45, 48
- [16] A. Bekkouche, S. M. Benslimane, M. Huchard, C. Tibermacine, F. Hadjila, and M. Merzoug. Qos-aware optimal and automated semantic web service composition with user’s constraints. *Service Oriented Computing and Applications*, 11(2) :183–201, 2017. 49, 51
- [17] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific american*, 284(5) :34–43, 2001. 38, 48, 51
- [18] D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. F. Nielsen, S. Thatte, and D. Winer. Simple object access protocol (soap) 1.1. <https://www.w3.org/TR/2000/NOTE-SOAP-20000508/>, 2000. En ligne ; consulté le 20 juillet 2020. 32
- [19] R. Brackin and P. Gonçalves. Ogc web services context document (ows context). <https://www.ogc.org/standards/owc>. En ligne ; consulté le 10 octobre 2021. 35
- [20] G. Cassar, P. Barnaghi, W. Wang, and K. Moessner. A hybrid semantic matchmaker for iot services. In *2012 IEEE International Conference on Green Computing and Communications*, pages 210–216. IEEE, 2012. 43, 84
- [21] J.-C. Chambelland and G. Gesquière. Complex virtual urban environment modeling from citygml data and ogc web services : application to the simfor project. In *Three-Dimensional Image Processing (3DIP) and Applications II*, volume 8290, page 82901A. International Society for Optics and Photonics, 2012. 17
- [22] Y. Cheng, W. Ge, and L. Xu. Quality of geographical information services evaluation based on order-relation. In *International Conference of Pioneering Computer Scientists, Engineers and Educators*, pages 679–688. Springer, 2018. 18
- [23] R. Chinnici, J.-J. Moreau, A. Ryman, and S. Weerawarana. Web services description language (wsdl) version 2.0 part 1 : Core language. *W3C recommendation*, 26(1) :19, 2007. 32
- [24] E. Christensen, F. Curbera, G. Meredith, S. Weerawarana, et al. Web services description language (wsdl) 1.1. <https://www.w3.org/TR/wsdl.html>, 2001. En ligne ; consulté le 18 juillet 2020. 17, 26, 32
- [25] O. Corcho. Ontology based document annotation : trends and open research problems. *International Journal of Metadata, Semantics and Ontologies*, 1(1) :47–57, 2006. 40

- [26] S. A. Cruz, A. M. Monteiro, and R. Santos. Automated geospatial web services composition based on geodata quality requirements. *Computers & Geosciences*, 47 :60–74, 2012. 52, 56, 57
- [27] S. Dareshiri, M. Farnaghi, and M. Sahelgozin. A recommender geoportal for geospatial resource discovery and recommendation. *Journal of Spatial Science*, 64(1) :49–71, 2019. 18, 81
- [28] C. Davis Jr and L. Alves. *Geospatial Web Services*, pages 1270–1273. 01 2007. 30
- [29] L. M. De Sousa, J. M. de Jesus, J. Čepický, A. T. Kralidis, D. Huard, C. Ehbrecht, S. Barreto, and J. Eberle. Pywps : overview, new features in version 4 and existing implementations. *Open Geospatial Data, Software and Standards*, 4(1) :1–10, 2019. 36
- [30] K. Derbal, G. Bordogna, G. Pasi, and Z. Alimazighi. Spatial querying supported by domain and user ontologies : An approach for web gis applications. In *Flexible Query Answering Systems 2015*, pages 353–365. Springer, 2016. 17
- [31] L. Di, P. Yue, W. Yang, G. Yu, P. Zhao, and Y. Wei. Ontology-supported automatic service chaining for geospatial knowledge discovery. *Proceedings of American Society of Photogrammetry and Remote Sensing*, 2007. 48
- [32] V. Dialani, S. Miles, L. Moreau, D. De Roure, and M. Luck. Transparent fault tolerance for web services based architectures. In *European Conference on Parallel Processing*, pages 889–898. Springer, 2002. 27
- [33] Y. Ding, D. Fensel, M. Klein, and B. Omelayenko. The semantic web : yet another hip? *Data & Knowledge Engineering*, 41(2-3) :205–227, 2002. 30
- [34] S. Dustdar and M. P. Papazoglou. Services and service composition—an introduction (services und service komposition—eine einföhrung). *IT-Information Technology*, 50(2) :86–92, 2008. 26
- [35] K. Evangelidis, K. Ntouros, K. Papatheodorou, A. Konstantinidisa, and K. Perakis. geoclouds for environmental management. *Journal of Environmental Protection and Ecology*, 15(2) :623–629, 2014. 36
- [36] M. Farnaghi and A. Mansourian. Automatic composition of wsmo based geospatial semantic web services using artificial intelligence planning. *Journal of Spatial Science*, 58(2) :235–250, 2013. 48, 49, 52, 57
- [37] M. Farnaghi and A. Mansourian. Multi-agent planning for automatic geospatial web service composition in geoportals. *ISPRS International Journal of Geo-Information*, 7(10) :404, 2018. 52, 56, 57
- [38] C. Feier, A. Polleres, R. Dumitru, J. Domingue, M. Stollberg, and D. Fensel. Towards intelligent web services : The web service modeling ontology (wsmo). In *The International Conference on Intelligent Computing, ICIC, Hefei, China*. Springer Science & Business Media, 2005. 39
- [39] G. Fenoy, N. Bozon, and V. Raghavan. Zoo-project : the open wps platform. *Applied Geomatics*, 5(1) :19–24, 2013. 36

- [40] R. T. Fielding and R. N. Taylor. *Architectural styles and the design of network-based software architectures*, volume 7. University of California, Irvine Irvine, 2000. 34
- [41] J. Footen and J. Faust. *The service-oriented media enterprise : SOA, BPM, and web services in professional media systems*. CRC Press, 2012. 26
- [42] M. Gaio and L. Moncla. Extended Named Entity Recognition Using Finite-State Transducers : An Application To Place Names. In *The Ninth International Conference on Advanced Geographic Information Systems, Applications, and Services (GEOProcessing 2017)*, Nice, France, Mar. 2017. 17, 73, 76, 107
- [43] M. Gebser, B. Kaufmann, A. Neumann, and T. Schaub. clasp : A conflict-driven answer set solver. In *International Conference on Logic Programming and Nonmonotonic Reasoning*, pages 260–265. Springer, 2007. 116, 129
- [44] M. Gebser, B. Kaufmann, A. Neumann, and T. Schaub. Conflict-driven answer set solving. In *IJCAI*, volume 7, pages 386–392, 2007. 50
- [45] M. Gebser, T. Schaub, and S. Thiele. Gringo : A new grounder for answer set programming. In *International Conference on Logic Programming and Nonmonotonic Reasoning*, pages 266–271. Springer, 2007. 116
- [46] D. Geer. Taking steps to secure web services. *Computer*, 36(10) :14–16, 2003. 30
- [47] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *ICLP/SLP*, volume 88, pages 1070–1080, 1988. 64, 131
- [48] A. Gozhyj, I. Kalinina, V. Gozhyj, and V. Vysotska. Web service interaction modeling with colored petri nets. In *2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems : Technology and Applications (IDAACS)*, volume 1, pages 319–323. IEEE, 2019. 51
- [49] N. Guarino. *Formal ontology in information systems : Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy*, volume 46. IOS press, 1998. 71
- [50] Z. Gui, J. Cao, X. Liu, X. Cheng, and H. Wu. Global-scale resource survey and performance monitoring of public ogc web map services. *ISPRS International Journal of Geo-Information*, 5(6) :88, 2016. 18
- [51] M. J. Hadley. Web application description language (wadl). <https://dl.acm.org/doi/pdf/10.5555/1698142>, 2006. En ligne ; consulté le 20 juillet 2020. 35
- [52] M. Halilali, E. Gouardères, F. Devin, and M. Gaio. Plateforme logicielle pour l'intégration et la composition de services géospatiaux. In *Sageo 2018*, 2018. 19
- [53] M. Halilali, E. Gouardères, M. Gaio, and F. Devin. Geospatial web services discovery through semantic annotation of wps. *Soumis à une revue scientifique*, 2021. 81
- [54] S. A. Hamid, R. A. Abdalrahman, I. A. Lafta, and I. Al Barazanchi. Web services architecture model to support distributed systems. *Journal of Southwest Jiaotong University*, 54(6), 2019. 47

- [55] K. Hu, Z. Gui, X. Cheng, H. Wu, and S. C. McClure. The concept and technologies of quality of geographic information service : improving user experience of giservices in a distributed computing environment. *ISPRS International Journal of Geo-Information*, 8(3) :118, 2019. 18, 128
- [56] A. Huang. Similarity measures for text document clustering. In *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand*, volume 4, pages 9–56, 2008. 43
- [57] H. Huang, L. He, X. Chen, M. Yu, and Z. Wang. Automatic composition of heterogeneous models based on semantic web services. *International Journal of Parallel Programming*, 43(3) :339–358, 2015. 48
- [58] J. K. Itani, E. Gouarderes, and P. Aniorte. A framework for managing services in a virtual community context. In *2015 IEEE World Congress on Services*, pages 237–244. IEEE, 2015. 51
- [59] D. A. Jackson, K. M. Somers, and H. H. Harvey. Similarity coefficients : measures of co-occurrence and association or simply measures of occurrence? *The American Naturalist*, 133(3) :436–453, 1989. 43
- [60] M. C. Jaeger, G. Rojec-Goldmann, C. Liebetruh, G. Mühl, and K. Geihs. Ranked matching for service descriptions using owl-s. In *Kommunikation in Verteilten Systemen (KiVS)*, pages 91–102. Springer, 2005. 43
- [61] T. Janhunen and I. Niemelä. The answer set programming paradigm. *AI Magazine*, 37(3) :13–24, 2016. 129
- [62] K. Janowicz, S. Schade, A. Bröring, C. Keßler, P. Maué, and C. Stasch. Semantic enablement for spatial data infrastructures. *Transactions in GIS*, 14(2) :111–129, 2010. 52
- [63] K. Janowicz, S. Scheider, T. Pehle, and G. Hart. Geospatial semantics and linked spatiotemporal data—past, present, and future. *Semantic Web*, 3(4) :321–332, 2012. 18
- [64] S. Jayakumar, J. Singh, and K. S. Joseph. Suitable qos parameters survey for standard web services & web applications to understand their cloud deployability. *International Journal of Computational Intelligence Systems*, 4 :1–18, 2014. 27
- [65] K. S. Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 1972. 43
- [66] K. Kaewbanjong and S. Intakosum. Qos attributes of web services : A systematic review and classification. *Journal of Advanced Management Science*, 3(3), 2015. 51, 70
- [67] R. Khalaf, N. Mukhi, and S. Weerawarana. Service-oriented composition in bpel4ws. In *WWW (Alternate Paper Tracks)*, pages 27–28, 2003. 91
- [68] R. Khaled, M. T. Laskri, and S. Servigne. Geospatial web services semantic discovery approach using quality. *J. Convergence Inf. Technol.*, 5(2) :28–35, 2010. 52

- 
- [69] T. Khaled, B. Benhamou, and P. Siegel. Vers une nouvelle méthode de calcul de modèles stables et extensions en programmation logique, 2017. 11, 129
- [70] M. Klusch. Semantic web service coordination. In *CASCOM : Intelligent Service Coordination in the Semantic Web*, pages 59–104. Springer, 2008. 84
- [71] M. Klusch and P. Kapahnke. Owls-mx3 : an adaptive hybrid semantic service matchmaker for owl-s. In *Third International Workshop on Service Matchmaking and Resource Retrieval in the Semantic Web (SMR2)*, 2009. 43
- [72] M. Klusch and P. Kapahnke. isem : Approximated reasoning for adaptive hybrid selection of semantic services. In *Extended Semantic Web Conference*, pages 30–44. Springer, 2010. 44
- [73] M. Klusch and P. Kapahnke. isem : Approximated reasoning for adaptive hybrid selection of semantic services. In *Extended Semantic Web Conference*, pages 30–44. Springer, 2010. 84
- [74] M. Klusch, P. Kapahnke, S. Schulte, F. Lecue, and A. Bernstein. Semantic web service search : a brief survey. *KI-Künstliche Intelligenz*, 30(2) :139–147, 2016. 42
- [75] M. Klusch, P. Kapahnke, and I. Zinnikus. Hybrid adaptive web service selection with sawsdl-mx and wsdl-analyzer. In *European Semantic Web Conference*, pages 550–564. Springer, 2009. 42, 44
- [76] J. Kopecký, K. Gomadam, and T. Vitvar. hrests : An html microformat for describing restful web services. In *2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, volume 1, pages 619–625. IEEE, 2008. 35, 40, 41
- [77] J. Kopecký, T. Vitvar, C. Bournez, and J. Farrell. Sawsdl : Semantic annotations for wsdl and xml schema. *IEEE Internet Computing*, 11(6) :60–67, 2007. 40
- [78] K. Kurniawan, F. J. Ekaputra, and P. R. Aryan. Semantic service description and compositions : A systematic literature review. In *2018 2nd International Conference on Informatics and Computational Sciences (ICICoS)*, pages 1–6. IEEE, 2018. 41, 45, 48
- [79] R. B. Lamine, R. B. Jemaa, and I. A. B. Amor. Graph planning based composition for adaptable semantic web services. *Procedia computer science*, 112 :358–368, 2017. 50
- [80] U. Lampe and S. Schulte. Self-adaptive semantic matchmaking using cov4sws.kom and log4sws.kom. In *Semantic web services*, pages 141–157. Springer, 2012. 43, 44
- [81] U. Lampe, S. Schulte, M. Siebenhaar, D. Schuller, and R. Steinmetz. Adaptive matchmaking for restful services based on hrests and microwsmo. In *Proceedings of the 5th International Workshop on Enhanced Web Service Technologies*, pages 10–17, 2010. 44
- [82] F. Lécué, E. Silva, and L. F. Pires. A framework for dynamic web services composition. In *Emerging Web Services Technology, Volume II*, pages 59–75. Springer, 2008. 84, 91

- [83] R. Lemmens, A. Wytzisk, R. By, C. Granell, M. Gould, and P. Van Oosterom. Integrating semantic and syntactic descriptions to chain geographic services. *IEEE Internet Computing*, 10(5) :42–52, 2006. 91
- [84] A. L. Lemos, F. Daniel, and B. Benatallah. Web service composition : a survey of techniques and tools. *ACM Computing Surveys (CSUR)*, 48(3) :1–41, 2015. 48, 91
- [85] A. L. Lemos, F. Daniel, and B. Benatallah. Web service composition : a survey of techniques and tools. *ACM Computing Surveys (CSUR)*, 48(3) :33, 2016. 45, 48
- [86] N. Leone, G. Pfeifer, W. Faber, T. Eiter, G. Gottlob, S. Perri, and F. Scarcello. The dlvs system for knowledge representation and reasoning. *ACM Transactions on Computational Logic (TOCL)*, 7(3) :499–562, 2006. 50, 129
- [87] H. Li, Y. Wang, and P. Cheng. Semantic description for the taxonomy of the geospatial services. *Boletim de Ciências Geodésicas*, 21(3) :515–531, 2015. 52, 57
- [88] Z. Li, Z. Gui, B. Hofer, Y. Li, S. Scheider, and S. Shekhar. Geospatial information processing technologies. In *Manual of Digital Earth*, pages 191–227. Springer, 2020. 18, 52
- [89] D. Lin et al. An information-theoretic definition of similarity. In *Icml*, volume 98, pages 296–304, 1998. 43
- [90] F. Lin and Y. Zhao. Assat : Computing answer sets of a logic program by sat solvers. *Artificial Intelligence*, 157(1-2) :115–137, 2004. 129
- [91] N. Lin, U. Kuter, and E. Sirin. Web service composition with user preferences. In *European Semantic Web Conference*, pages 629–643. Springer, 2008. 49
- [92] Liping Di, Peisheng Zhao, Wenli Yang, Genong Yu, and Peng Yue. Intelligent geospatial web services. In *Proceedings. 2005 IEEE International Geoscience and Remote Sensing Symposium, 2005. IGARSS '05.*, volume 2, pages 1229–1232, 2005. 30
- [93] M. Lutz and E. Klien. Ontology-based retrieval of geographic information. *International Journal of Geographical Information Science*, 20(3) :233–260, 2006. 51
- [94] A. Malki, S. M. Benslimane, et al. Building semantic mashup. In *ICWIT*, pages 40–49, 2012. 41
- [95] V. W. Marek and M. Truszczyński. Stable models and an alternative logic programming paradigm. In *The Logic Programming Paradigm*, pages 375–398. Springer, 1999. 64, 91, 129
- [96] D. Martin, M. Paolucci, S. McIlraith, M. Burstein, D. McDermott, D. McGuinness, B. Parsia, T. Payne, M. Sabou, M. Solanki, et al. Bringing semantics to web services : The owl-s approach. In *International Workshop on Semantic Web Services and Web Process Composition*, pages 26–42. Springer, 2004. 38
- [97] N. Masuch, B. Hirsch, M. Burkhardt, A. Heßler, and S. Albayrak. Sema 2 : A hybrid semantic service matching approach. In *Semantic web services*, pages 35–47. Springer, 2012. 44

- 
- [98] C. Mayer, B. Stollberg, and A. Zipf. Providing near real-time traffic information within spatial data infrastructures. In *2009 International Conference on Advanced Geographic Information Systems & Web Services*, pages 104–111. IEEE, 2009. 17
- [99] D. McDermott, M. Ghallab, A. Howe, C. Knoblock, A. Ram, M. Veloso, D. Weld, and D. Wilkins. Pddl-the planning domain definition language. <https://www.csee.umbc.edu/courses/671/fall112/hw/hw6/pddl1.2.pdf>, 1998. En ligne ; consulté le 18 juillet 2020. 49, 52
- [100] S. McIlraith and T. C. Son. Adapting golog for composition of semantic web services. *Kr*, 2(200) :2, 2002. 49
- [101] B. Medjahed, A. Bouguettaya, and A. K. Elmagarmid. Composing web services on the semantic web. *The VLDB journal*, 12(4) :333–351, 2003. 49
- [102] O. Michael, A. Christian, A. Lars, B. Vincent, C. Pierluigi, C. Radoslav, V. Ine, F. Daniele, G. Christine, S. Alejandro, et al. Technical guidelines for implementing dataset and service metadata based on iso/ts 19139 : 2007. *Technical Guidelines for implementing dataset and service metadata based on ISO/TS 19139 : 2007*, 2017. 51, 70, 76, 78
- [103] J. Mtsweni. Exploiting uml and acceleo for developing semantic web services. In *2012 International Conference for Internet Technology and Secured Transactions*, pages 753–758. IEEE, 2012. 40
- [104] M. Müller and B. Pross. Ogc wps 2.0 interface standard. version 2.0. standard, Open Geospatial Consortium, 2015. 10, 11, 17, 36, 67, 74
- [105] S. D. Nagowah, H. B. Sta, and B. Gobin-Rahimbux. An overview of semantic interoperability ontologies and frameworks for iot. In *2018 Sixth International Conference on Enterprise Systems (ES)*, pages 82–89. IEEE, 2018. 40
- [106] A. Nasa. computational modeling algorithms and cyberinfrastructure. Technical report, Technical report, National Aeronautics and Space Administration (NASA), 2012. 19, 81
- [107] D. S. Nau, T.-C. Au, O. Ilghami, U. Kuter, J. W. Murdock, D. Wu, and F. Yaman. Shop2 : An htn planning system. *Journal of artificial intelligence research*, 20 :379–404, 2003. 49
- [108] S. Naujokat, A.-L. Lamprecht, and B. Steffen. Loose programming with prophets. In *International Conference on Fundamental Approaches to Software Engineering*, pages 94–98. Springer, 2012. 52
- [109] T. H. Nguyen, E. Pontelli, and T. C. Son. On repairing web services workflows. In *International Symposium on Practical Aspects of Declarative Languages*, pages 37–53. Springer, 2020. 91
- [110] T. H. Nguyen, T. C. Son, and E. Pontelli. Automatic web services composition for phylotastic. In *International Symposium on Practical Aspects of Declarative Languages*, pages 186–202. Springer, 2018. 50, 91

- [111] A.-M. Olteanu-Raimond, P.-A. Davoine, M. Gaio, E. Gouarderes, M.-D. Van Damme, M. Villanova-Oliver, M. Brasebin, C. Domingues, C. Duchêne, O. Favre, et al. Projet choucas : Intégration de données hétérogènes et raisonnement spatial pour l'aide à la localisation des victimes en montagne. In *Spatial Analysis and GEomatics*, 2017. 17
- [112] M. Paolucci, T. Kawamura, T. R. Payne, and K. Sycara. Semantic matching of web services capabilities. In *International semantic web conference*, pages 333–347. Springer, 2002. 42
- [113] M. P. Papazoglou, P. Traverso, S. Dustdar, and F. Leymann. Service-oriented computing : a research roadmap. *International Journal of Cooperative Information Systems*, 17(02) :223–255, 2008. 25
- [114] M. P. Papazoglou and W.-J. Van Den Heuvel. Service oriented architectures : approaches, technologies and research issues. *The VLDB journal*, 16(3) :389–415, 2007. 26
- [115] C. Peltz. Web services orchestration and choreography. *Computer*, (10) :46–52, 2003. 45
- [116] M. Phan and F. Hattori. Automatic web service composition using congollog. In *26th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'06)*, pages 17–17. IEEE, 2006. 50
- [117] P. Plebani and B. Pernici. Urbe : Web service retrieval based on similarity evaluation. *IEEE Transactions on Knowledge and data engineering*, 21(11) :1629–1642, 2009. 43
- [118] V. Portchelvi, V. P. Venkatesan, and G. Shanmugasundaram. Achieving web services composition—a survey. *Softw Eng*, 2(5) :195–202, 2012. 45, 48, 91
- [119] J. Puttonen, A. Lobov, M. A. C. Soto, and J. L. M. Lastra. Planning-based semantic web service composition in factory automation. *Advanced Engineering Informatics*, 29(4) :1041–1054, 2015. 48
- [120] A. Rafiee and S. Emadi. An integrated method for semantic web service composition using planning based on qualitative parameters. In *2016 Second International Conference on Web Research (ICWR)*, pages 84–89. IEEE, 2016. 48
- [121] J. Rao and X. Su. A survey of automated web service composition methods. In *International Workshop on Semantic Web Services and Web Process Composition*, pages 43–54. Springer, 2004. 91
- [122] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007*, 1995. 43
- [123] P. Rodriguez-Mier, C. Pedrinaci, M. Lama, and M. Mucientes. An integrated semantic web service discovery and composition framework. *IEEE transactions on services computing*, 9(4) :537–550, 2015. 50
- [124] S. Schade, C. Granell, G. Vancauwenberghe, C. Keßler, D. Vandenbroucke, I. Masser, and M. Gould. Geospatial information infrastructures. In *Manual of Digital Earth*, pages 161–190. Springer, 2020. 18

- [125] Q. Z. Sheng, X. Qiao, A. V. Vasilakos, C. Szabo, S. Bourne, and X. Xu. Web services composition : A decade’s overview. *Information Sciences*, 280 :218–238, 2014. 45, 46, 48, 91
- [126] A. P. Sheth, K. Gomadam, and J. Lathem. Sa-rest : Semantically interoperable and easier-to-use services and mashups. *IEEE Internet Computing*, 11(6) :91–94, 2007. 40
- [127] P. Simons, I. Niemelä, and T. Soinen. Extending and implementing the stable model semantics. *Artificial Intelligence*, 138(1-2) :181–234, 2002. 129
- [128] E. Sirin, J. Hendler, and B. Parsia. Semi-automatic composition of web services using semantic descriptions. In *1st Workshop on Web Services : Modeling, Architecture and Infrastructure*, pages 17–24, 2003. 29
- [129] E. Sirin, B. Parsia, D. Wu, J. Hendler, and D. Nau. Htn planning for web service composition using shop2. *Journal of Web Semantics*, 1(4) :377–396, 2004. 49
- [130] S. Sohrabi, N. Prokoshyna, and S. A. McIlraith. Web service composition via the customization of golog programs with user preferences. In *Conceptual Modeling : Foundations and Applications*, pages 319–334. Springer, 2009. 50
- [131] Y. Song, Q. Sun, A. Zhou, S. Wang, and J. Li. Qos-aware service composition using htn planner. In *2018 IEEE 8th International Symposium on Cloud and Service Computing (SC2)*, pages 107–110. IEEE, 2018. 49, 51
- [132] N. Srinivasan, M. Paolucci, and K. Sycara. Semantic web service discovery in the owl-s ide. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS’06)*, volume 6, pages 109b–109b. IEEE, 2006. 43
- [133] C. Szyperski, D. Gruntz, and S. Murer. *Component software : beyond object-oriented programming*. Pearson Education, 2002. 25
- [134] D. A. Taylor. *Object technology : a manager’s guide*. Addison-Wesley Longman Publishing Co., Inc., 1997. 25
- [135] I. Taylor, M. Shields, and I. Wang. Distributed p2p computing within triana : A galaxy visualization test case. In *Proceedings International Parallel and Distributed Processing Symposium*, pages 8–pp. IEEE, 2003. 47
- [136] P. Thi Quynh and H. Quyet Thang. Dynamic coupling metrics for service-oriented software. *Tp chí Khoa hc và Công ngh các trng đi hc k thut*, (73) :50, 2009. 26
- [137] E. Thomas. Soa principles of service design. *Boston : Prentice Hall*, 37 :71–75, 2007. 26
- [138] D. Tosi and S. Morasca. Supporting the semi-automatic semantic annotation of web services : A systematic literature review. *Information and Software Technology*, 61 :16–32, 2015. 39
- [139] L. Van den Brink, P. Barnaghi, J. Tandy, G. Atemezing, R. Atkinson, B. Cochrane, Y. Fathy, R. García Castro, A. Haller, A. Harth, et al. Best practices for publishing, retrieving, and using spatial data on the web. *Semantic Web*, pages 1–20, 2019. 91

- [140] S. Vinoski. Web services interaction models. current practice. *IEEE Internet Computing*, 6(3) :89–91, 2002. 30
- [141] S. Vinoski. An overview of middleware. In *International Conference on Reliable Software Technologies*, pages 35–51. Springer, 2004. 25
- [142] T. Vitvar, J. Kopecký, J. Viskova, and D. Fensel. Wsmo-lite annotations for web services. In S. Bechhofer, M. Hauswirth, J. Hoffmann, and M. Koubarakis, editors, *The Semantic Web : Research and Applications*, pages 674–689, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. 41
- [143] P. Vretanos. Opengis web feature service 2.0 interface standard, open geospatial consortium inc. Technical report, Tech. Rep. OGC 09-025r1 and ISO/DIS 19142, Version 2.0. 0, 2010. 17
- [144] W3C. Web services architecture. <https://www.w3.org/TR/ws-arch/>. En ligne ; consulté le 15 juillet 2020. 27
- [145] H. Wang, Q. Wu, X. Chen, Q. Yu, Z. Zheng, and A. Bouguettaya. Adaptive and dynamic service composition via multi-agent reinforcement learning. In *2014 IEEE international conference on web services*, pages 447–454. IEEE, 2014. 51
- [146] Y. Wang and J. S. Huang. Method of semantic web service discovery and automatic composition. *Journal of Emerging Technologies in Web Intelligence*, 6(3) :298–304, 2014. 50
- [147] Y. Wei, S.-K. Santhana-Vannan, and R. B. Cook. Discover, visualize, and deliver geospatial data through ogc standards-based webgis system. In *2009 17th International Conference on Geoinformatics*, pages 1–6. IEEE, 2009. 19, 81
- [148] N. Wiegand and C. García. A task-based ontology approach to automate geospatial data retrieval. *Transactions in GIS*, 11(3) :355–376, 2007. 18, 81
- [149] Z. Wu and M. Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics, 1994. 43, 82
- [150] B. Yang and Z. Qin. Composing semantic web services with pddl. *Information Technology Journal*, 9(1) :48–54, 2010. 49
- [151] Q. Yu, X. Liu, A. Bouguettaya, and B. Medjahed. Deploying and managing web services : issues, solutions, and directions. *The VLDB Journal*, 17(3) :537–572, 2008. 27, 29, 30
- [152] P. Yue, P. Baumann, K. Bugbee, and L. Jiang. Towards intelligent giservices. *Earth Science Informatics*, 8(3) :463–481, 2015. 51
- [153] M. P. Zafer Defne, Akm Saiful Islam. Ogc : Ontology for geography markup language (gml3.0) of open gis consortium (ogc). [http://protegewiki.stanford.edu/wiki/Protege\\_Ontology\\_Library](http://protegewiki.stanford.edu/wiki/Protege_Ontology_Library). En ligne ; consulté le 20 juin 2020. 73