

# composition des services web

TAOURIRIT Salah Eddine

23 novembre 2014

version 0.2

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>Les services web : Vue d'ensemble</b>	<b>6</b>
2.1	Notions de base et technologies associées . . . . .	7
2.1.1	Définition et caractéristiques . . . . .	7
2.1.2	L'évolution des styles des services web . . . . .	8
2.1.3	L'architecture de référence et technologies associées . . . . .	8
2.1.3.1	Communication : SOAP . . . . .	10
2.1.3.2	Description : WSDL . . . . .	10
2.1.3.3	Découverte : UDDI . . . . .	12
2.2	Description des services web . . . . .	13
2.2.1	Description syntaxique de services . . . . .	13
2.2.2	Ajout de la sémantique . . . . .	15
2.2.2.1	Définition des services Web sémantiques . . . . .	15
2.2.2.2	WSDL-S . . . . .	16
2.2.2.3	SAWSDL . . . . .	17
2.2.2.4	OWL-S . . . . .	18
2.2.2.5	WSMO . . . . .	19
2.3	Découverte des services web . . . . .	19
2.4	Conclusion . . . . .	19
<b>3</b>	<b>La Composition des services web</b>	<b>20</b>
3.1	Définition et types de composition . . . . .	21
3.1.1	Définitions . . . . .	21
3.1.2	Types de composition . . . . .	22

3.1.2.1	Composition manuel/automatique . . . . .	22
3.1.2.2	Composition statique/dynamique . . . . .	22
3.1.3	Procédés de coordination . . . . .	23
3.1.3.1	Orchestration . . . . .	23
3.1.3.2	Chorégraphie . . . . .	24
3.2	Langages de composition des services web . . . . .	25
3.2.1	BPEL . . . . .	25
3.2.2	WS-CDL . . . . .	27
3.2.3	OWL-S . . . . .	28
3.3	Composition dynamique des services web . . . . .	29
3.3.1	Techniques de planification . . . . .	29
3.4	Conclusion . . . . .	29

# Acronyms

BPEL Business Process Execution Language. 25

BPEL4WS Business Process Execution Language for Web Services. 25

DAML the DARPA Agent Markup Language. 16, 18

DARBA Defence Advanced Research Projects Agency. 18

HTTP Hypertext Transfer Protocol. 7

SAWSDL Semantic Annotations for WSDL and XML Schema. 17

SOAP Simple Object Access Protocol. 9

UDDI Universal Description Discovery and Integration. 9, 12

W3C The World Wide Web Consortium. 7, 10, 13

WS-BPEL Web Service Business Process Execution Language. 25

WS-CDL Web Services Choreography Description Language. 27

WSOI Web Service Choreography Interface. 27

WSDL Web Services Description Language. 9, 10, 13, 27

WSFL Web Services Flow Language. 25

XLANG XML Business Process Language. 25

XML extensible markup language. 25

# Table des figures

2.1	La structure d'un message SOAP . . . . .	11
2.2	Web evolution to Semantic Web services [1]. . . . .	16
2.3	Les éléments d'une ontologie OWL-S . . . . .	18
3.1	Orchestration vs Chorégraphie selon Peltz [2]. . . . .	23
3.2	Principe de l'orchestration des services Web. . . . .	24
3.3	Principe de la chorégraphie des services Web. . . . .	24
3.4	Une vue interne d'un service de voyage BPEL [3]. . . . .	25
3.5	Le procédé d'orchestration de service de voyage BPEL [3]. . . . .	26
3.6	Une vue interne d'un service de voyage composite OWLS. . . . .	28
3.7	Document OWLS de processus composite du service de voyage BPEL. . . . .	29

# Liste des tableaux

# Chapitre 1

## Introduction

# Chapitre 2

## Les services web : Vue d'ensemble

Ce chapitre établit une étude du fondement théorique de notre travail à savoir les concepts de base du paradigme service Web. Nous commençons d'abord par présenter un tour d'horizon définissant l'architecture de référence de ce paradigme ainsi que quelques définitions proposées dans la littérature. Ensuite nous nous intéressons à montrer les limitations de l'approche syntaxique de la description des services web et l'apport de l'enrichissement sémantique de cette dernière aux processus de la découverte et la composition des services Web.



## 2.1 Notions de base et technologies associées

Les services Web constituent une approche pour mettre en œuvre le paradigme de service, et peut être vue comme une instance de l'architecture orienté service.

Dans cette section va parler aussi d'un socle technologique très sollicité, On va aussi Détailler l'architecture de base d'un service web, ensuite nous introduisons l'architecture étendus.

### 2.1.1 Définition et caractéristiques

Les services Web sont la technologie la plus connue et la plus populaire dans le monde industriel et académique pour la mise en place d'architectures à services.

Les Web services ont été proposés initialement par IBM [4] et Microsoft, puis en standardisés par le W3C<sup>1</sup> et définis [5] par :

*“Un service web est un système conçu pour permettre d'interopérabilité des applications à travers un réseau. Il est caractérisé par un format de description interprétable/compréhensible automatiquement par la machine, D'autres systèmes peuvent interagir avec le Service Web selon la manière prescrite dans sa description et en utilisant des messages SOAP, généralement transmis via le protocole HTTP et sérialisés en XML et en d'autres standards du Web ”.*

Cette définition surligne les caractéristiques clés de services Web [6] :

- **Basés sur des protocoles Internet** : L'utilisation de HTTP pour le transport des informations permet de traverser les contrôles d'accès dans un environnement hétérogène.
- **Interopérables** : Le standard SOAP [7] définit comme étant un protocole destiné à l'échange de messages structurés véhiculé généralement sur HTTP et sérialisé en XML, permettant le support pour l'interopérabilité.
- **Basés sur XML** : Le méta-langage de balisage XML *eXtensible Markup Language* est un standard Web ouvert par W3C [8] offre un cadre standard pour la définition de documents Interprétable par des machines.

---

1. <http://www.w3.org/>

M. P. Papazoglou [9] apporte une autre définition de services web :

*“Les services Web sont des éléments auto-descriptifs et indépendants des plateformes permettent la composition faible coût d’applications distribuées. Les services Web effectuent des fonctions allant de simples requêtes des processus métiers complexes. Les services Web permettent aux organisations d’exposer leurs programmes résultats sur Internet (ou sur un intranet) en utilisant des langages (basés sur XML) et des protocoles standardisés et de les mettre en œuvre via une interface auto-descriptive basée sur des formats standardisés et ouverts”*

Curbera et al. [10] de ça part proposent la définition suivante :

*“Un service Web est une application réseau capable d’interagir par le moyen des standards et des protocoles via des interfaces bien spécifiés, dans lequel est décrits utilisant un langage de description fonctionnel standardisé”.*

### 2.1.2 L’évolution des styles des services web

“the next section provides a short history of web services, with emphasis on the kinds of software challenges that web services are meant to address.”

### 2.1.3 L’architecture de référence et technologies associées

[11] [12] [4] [5] Cette architecture a été proposée afin de promouvoir l’interopérabilité et l’extensibilité des services Web Dans l’ensemble, une architecture complète de services Web est constitué d’un fournisseur de service<sup>2</sup>, un annuaire de services<sup>3</sup>, et un client<sup>4</sup> de service. La figure x montre comment ces trois rôles interagissent.

- **Le fournisseur** A service provider provides the interface for the Web service and the implementation of the application. The service provider is also responsible for creating the definition of the service and publishing that

---

2. Providers

3. Service Registry

4. Service Requester

definition to meet the Universal Description, Discovery, and Integration (UDDI) specification.

- **L'annuaire** A service registry is a way in which Web services are formally published. The service registry is based on the UDDI specification and reflects information about services provided by the service provider. The service registry provides a service requester with a Web Services Description Language (WSDL) service description and a Uniform Resource Locator (URL) that points to the service itself.

- **Le client** A service requester is the consumer of a Web service and uses the service registry to gain information about, and access to, a Web service.

For an application to take advantage of Web Services, three behaviors must take place : publication of service descriptions, lookup or finding of service descriptions, and binding or invoking of services based on the service description. These behaviors can occur singly or iteratively. In detail, these operations are :

- **Publish** To be accessible, a service description needs to be published so that the service requestor can find it. Where it is published can vary depending upon the requirements of the application (see “Service Publication” for more details).

- **Find** In the find operation, the service requestor retrieves a service description directly or queries the service registry for the type of service required (see “Service Discovery” for more details). The find operation can be involved in two different lifecycle phases for the service requestor : at design time to retrieve the service’s interface description for program development, and at runtime to retrieve the service’s binding and location description for invocation.

- **Bind** Eventually, a service needs to be invoked. In the bind operation the service requestor invokes or initiates an interaction with the service at runtime using the binding details in the service description to locate, contact and invoke the service.

Les services Web sont construits autour de standards qui sont SOAP, WSDL et UDDI assurant respectivement leur communication, leur description et leur découverte.

### 2.1.3.1 Communication : SOAP

Développé par IBM<sup>5</sup> et Microsoft<sup>6</sup> [7], L'approche SOAP est une recommandation W3C qui le définit comme étant un protocole destiné à l'échange de messages structurés, permettant d'invoquer des applications sur des réseaux distribués [13].

Ce protocole SOAP est basé sur XML pour mettre en place un mécanisme valable d'échange des données indépendant du modèle de programmation de l'application et du système d'exploitation.

Un message SOAP est un document XML constitué d'une enveloppe SOAP obligatoire, d'un en-tête SOAP facultatif et d'un corps SOAP obligatoire :

- **Enveloppe** : L'élément racine du message SOAP , définissant le contexte du message, son destinataire et son contenu, il englobe l'en-tête et le corps.
- **En-tête <Header>** : Un mécanisme générique permettant d'ajouter des fonctions à un message SOAP d'une manière modulaire sans accord préalable entre les parties en communication. Des exemples d'extension qui peuvent être implémentées comme des en-têtes sont des authentifications, des transactions, des paiements
- **Corps <Body>** : Contient les informations obligatoires destinées à l'ultimate destinataire du message, il sert comme un container pour les informations mandataires à l'intention du récepteur du message. SOAP définit un élément pour le corps, qui est l'élément <Fault> (Erreur) utilisé pour rapporter les erreurs.

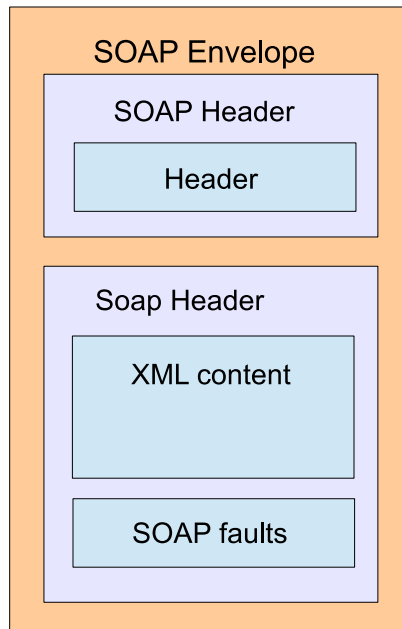
### 2.1.3.2 Description : WSDL

Le langage de description des services Web WSDL [14] est une recommandation du W3C, maintenant dans sa deuxième version. WSDL est basé sur XML pour décrire les fonctions opérationnelles de services Web. La description des WSDL sont composées d'une interface et des implémentations. L'interface est une définition abstraite et réutilisable service qui peut être référencée par plusieurs implémentations.

---

5. <http://www.ibm.com>

6. <http://www.microsoft.com>



(a) Les éléments d'un message SOAP

```

<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Body>
<symbol xmlns="http://stock.samples">XXX</symbol>
</soapenv:Body>
</soapenv:Envelope>

```

(b) Exemple de message SOAP

FIGURE 2.1 – La structure d'un message SOAP

Le WSDL sert à décrire :

- le protocole de communication (SOAP RPC ou SOAP orienté message)
- le format de messages requis pour communiquer avec ce service
- les méthodes que le client peut invoquer
- la localisation du service.

#### 2.1.3.3 Découverte : UDDI

UDDI [15] est une standardisation pour la publication et la découverte des services Web initialement conçue et spécifiée par le Consortium de standards OASIS<sup>7</sup>, et il est le résultat d'un accord d'un ensemble d'industriels Ariba<sup>8</sup>, IBM, Microsoft, etc en vue de devenir le registre standard de la technologie des services Web.

UDDI complète les technologies basiques de services Web en permettant de créer un **annuaire** permettant de localiser sur le réseau les services web recherchés, les services référencés dans UDDI sont accessibles par l'intermédiaire du protocole de communication SOAP, et la publication des informations concernant les fournisseurs et les services doit être spécifiée en XML afin que la recherche et l'utilisation soient faites de manière **dynamique** et **automatique**.

Un UDDI peut appartenir à un domaine public comme internet ou tout autre réseau accessible à un nombre non limité d'utilisateurs, comme il peut appartenir à un domaine restreint comme l'intranet d'une entreprise ou d'un groupe d'entreprise.

Les données stockés dans l'UDDI sont structurées (en XML) et organisées en trois parties connues :

**Pages blanches** : fournissent des descriptions générales sur les fournisseurs de services à savoir le nom de l'entreprise qui fournit le service, son identificateur commercial, ses adresses, etc.

**Pages jaunes** : comportent des descriptions détaillées sur les fournisseurs de services catalogués dans les pages blanches d'une de façon taxonomique (selon secteurs d'activités par exemple).

---

7. <https://www.oasis-open.org>

8. <http://www.ariba.com/>

**Pages vertes** : fournissent des informations techniques sur les services Web catalogués. Ces informations incluent la description du service, les adresses URL, du processus de son utilisation et des protocoles utilisés pour son invocation.

## 2.2 Description des services web

Une description du service Web est un document par lequel le fournisseur de services communique au client les spécifications pour invoquer le service Web. Dans cette section nous présentons les modèles de description des services web. Nous détaillons dans la première sous-section le modèle de description syntaxique WSDL [14] développé et standardisé par le W3C qui est devenu un élément essentiel dans des technologies services web. Ensuite en mettant l'accent sur les limitations majeurs de cette approche dans un environnement hétérogène qui nécessite un certain degré de dynamisme et d'automatisation. Finalement, Nous présentons les divers approches sémantiques visant à préciser la description d'un service en insistant sur les approches d'annotation sémantique et sur les ontologies de services.

### 2.2.1 Description syntaxique de services

Le langage de description de services Web WSDL [14] fournit un modèle ainsi qu'un langage basé sur XML de description de services Web. Un fichier WSDL comprend une description des fonctionnalités d'un service, mais il ne se préoccupe pas de l'implantation de celles-ci. Il contient aussi des informations concernant la localisation du service, ainsi que les données et les protocoles à utiliser pour l'invoquer. En pratique, le document WSDL<sup>9</sup> est un document XML qui se divise en deux parties [16] :

- La définition **abstraite** de l'interface du service avec les opérations supportées par le service Web, ainsi que leurs paramètres et les types des données.
- La définition **concrète** de l'accès au service avec la localisation, par une

---

9. <http://www.w3.org/TR/wsdl20/>

adresse réseau du fournisseur de service<sup>10</sup>, et les protocoles spécifiques d'accès.

Un document WSDL constitué de quatre éléments principaux [14] : `<Types>`, `<Interface>`, `<Binding>`, `<Service>`.

[17] [16] [18]

- `<Types>` : L'élément `Types` sert à un conteneur définissant les données figurant dans les messages échangés par le service. WSDL supporte des types élémentaires prédéfinis (tels que les entiers, les chaînes de caractères et les dates). Si les données échangées possèdent une structure particulière, il est possible de les décrire à travers un schéma XML [19].

- `<Interface>` : Les interfaces WDSL offrent une manière abstraite de décrire la fonctionnalité du service, Contrairement à la représentation concrète offerte par les éléments de `<Bindings>` et de `<Services>` qui sera décrit plus tard. Une interface WSDL est constitué d'un ensemble d'opérations, chacun d'entre eux décrivant d'une simple interaction entre le service et le client. Une opération décrit un séquence des messages d'entrées/sorties ou un modèle d'échange de message<sup>11</sup> suivie lorsque l'opération est invoqué. Pour chaque message contenu dans le motif<sup>12</sup>, un type de message est spécifié à l'aide des types qui ont été définis précédemment dans le document. WSDL contient huit modèles de messages prédéfinis, mais on peut facilement définir de nouveaux.

- `<Binding>` : [16] [17] L'élément `Binding` reprend les opérations de l'élément `<Interface>` et leurs associe un protocole de transfert et des spécifications des formats de données de message. La définition des protocoles de communication utilisés pour l'invocation du service Web permet d'établir le lien, d'une part, entre le document et les messages SOAP et d'autre part, entre les messages SOAP et les opérations invoquées.

- `<Service>` : Cet élément définit la localisation du service Web décrit. Pour chaque interface décrite, un élément service lui est associé. Le sous-élément `<endpoint>` définit un port d'accès en référénçant l'élément

---

10. Service Endpoint

11. message exchange pattern

12. pattern



<binding> associé et en déclarant l'URL localisant le service (avec l'attribut <address>).

### 2.2.2 Ajout de la sémantique

Malgré les améliorations apportées au standard WSDL dans son deuxième version [14], la description du service reste uniquement au niveau fonctionnel, c'est-à-dire qu'elle contient la manière dont on peut utiliser le service et non ce que fait le service, le standard WSDL est limité à l'énumération des opérations et à la description des types des paramètres d'entrée et de sortie associés, elle ne caractérise pas la sémantique de la fonctionnalité accomplie par le service. Par conséquent, la description WSDL reste insuffisante lors du processus de sélection. Pour pallier cette Difficulté, plusieurs approches proposent de rajouter une couche au dessus sémantique de WSDL complétant la description syntaxique par des précisions sémantiques.

Dans un premier temps, on va essayer de clarifier la notion d'un services Web sémantique, puis étudie les langages émergents qui permettent de décrire ce type de services Web.

#### 2.2.2.1 Définition des services Web sémantiques

L'objectif premier du Web sémantique est de définir et lier les ressources du Web afin de simplifier leur utilisation, leur découverte, leur intégration et leur réutilisation dans le plus grand nombre d'applications [20]. Le Web sémantique doit fournir l'accès à ces ressources par l'intermédiaire de descriptions sémantiques exploitables et compréhensibles par des machines. En effet, Les technologies du Web sémantique complètent le Web actuel avec des outils sémantiques. Il ne s'agit donc pas de créer un nouveau Web ou un Web séparé de l'existant : ce Web de données repose entièrement sur les technologies et concepts qui ont fait le succès du Web tel que nous le connaissons aujourd'hui [21].

La réalisation du Web sémantique trouve ces racines dans le développement des langages de balisage inspiré par des travaux issue de la communié AI [22], tels que OIL [23], DAML+OIL [24] et DAML+OTN [25] (ces deux derniers

langages sont parties de la famille DAML).

Ces langages ont une sémantique bien définies et permettent le balisage et la manipulation des taxonomie complexe et Des relations logiques entre les entités sur le Web. [26]

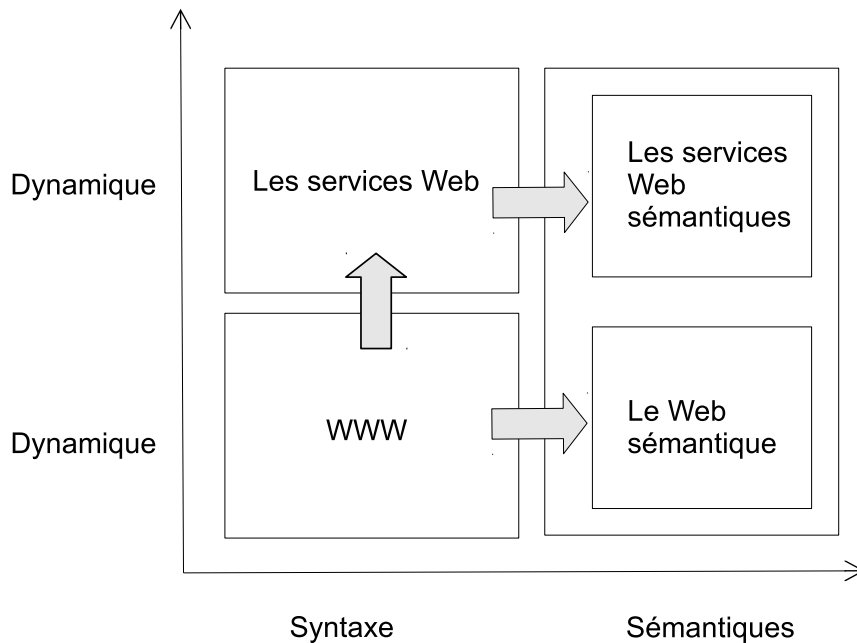


FIGURE 2.2 – Web evolution to Semantic Web services [1].

Cette description repose sur des ontologies. Selon Gruber [27], une ontologie est une spécification explicite d'une conceptualisation. Une conceptualisation est un modèle abstrait qui représente la manière dont les personnes conçoivent les choses réelles dans le monde et une spécification explicite signifie que les concepts et les relations d'un modèle abstrait reçoivent des noms et des définitions explicites. Le Web sémantique est devenu un domaine à part entière, preuve en est la création en 2001 du groupe de travail sur ce sujet par le W3C.

#### 2.2.2.2 WSDL-S

WSDL-S [28] est le résultat d'un travail collaboratif entre IBM, laboratoire LSDSI et l'université de Geogia<sup>13</sup>. La spécification a devenue une recommandation

13. <http://www.uga.edu/>

W3C depuis 2005. Son objectif principal est de fournir un processus d'annotation sémantique compatible avec les technologies existantes. Pratiquement, Le méta-modèle WSDL-S repose sur les capacités du modèle WSDL en rajoutant trois éléments majeurs `<category>`, `<effect>` et deux attributs `modelReference` et `schemaMapping`. Les éléments introduits permettent de rajouter des informations qui n'étaient pas prises en compte dans WSDL comme *les préconditions* et *les effets* d'une opération. Tandis que les attributs permettent de référencer des concepts dans des ontologies de référence, ces préconditions et effets ensemble avec les annotations sémantiques des éléments `<inputs>` et `<outputs>` permet de l'automatisation du processus de découverte de services.

- L'élément `<category>`
- `<precondition>`
- `<effect>`
- L'attribut `modelReference`
- `schemaMapping`

#### 2.2.2.3 SAWSDL

La spécification SAWSDL [29] est la suite de WSDL-S et il partage les mêmes principes de ce dernier. issue d'initiative du groupe de travail d'annotations sémantiques pour WSDL<sup>14</sup> et soumise au W3C en 2007, SAWSDL définit un mécanisme d'annoter sémantiquement les interfaces et les opérations WSDL, ainsi que les types XML SCHEMA en les reliant à des concepts dans une ontologie. Cette annotation repose sur la définition d'attributs étendant le standard de description. Les annotations sémantiques référencent des ontologies pré-existantes. Le mécanisme d'annotation de SAWSDL est indépendant de tout langage de représentation [18] d'ontologies.

SAWSDL propose deux sortes d'annotations sémantiques : une pour identifier le concept sémantique (représentée par l'attribut `modelReference`) et une autre pour faire le lien entre le concept et le document WSDL (représentée par les attributs `liftingSchemaMapping` et `loweringSchemaMapping`).

---

14. Semantic Annotations for WSDL and XML Schema

#### 2.2.2.4 OWL-S

OWL-S [30] désigné par DAML-S dans les versions antérieures [31], est un langage issue des travaux de la DARBA<sup>15</sup> et son programme DAML<sup>16</sup> en collaboration avec des chercheurs de plusieurs universités et organisations (l'Université de Toronto, Yale, Nokia, etc.). Il a été intégré au consortium W3C en 2004, au sein du groupe d'intérêt sur les services Web sémantiques, lors de la recommandation du langage OWL [24] [32]. Ankolekar *et al.* [31] présentent une ontologie pour les services web dans le but d'automatiser la *découverte*, *l'invocation*, la *composition* et la *surveillance* de l'exécution des services [33], les auteurs reprennent la notion de classes d'OWL et proposent l'ontologie OWL-S.

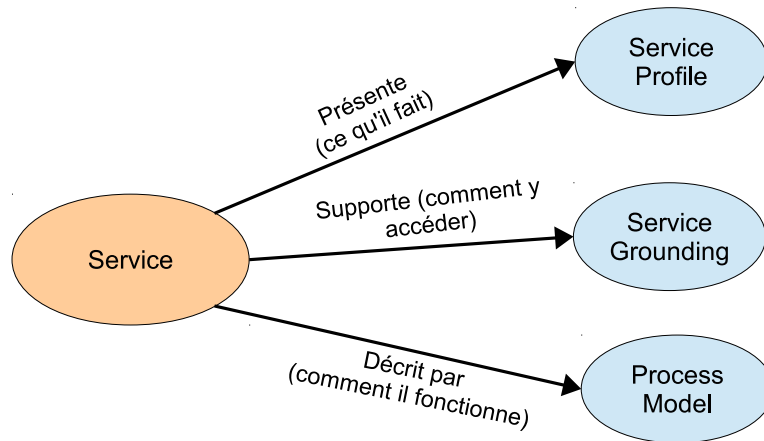


FIGURE 2.3 – Les éléments d'une ontologie OWL-S

L'objectif principal de ces recherches est d'établir une plateforme dans laquelle les descriptions des services Web sont partagés en utilisant une ontologie standard, constituée d'un ensemble de classes de base et des propriétés pour résoudre les ambiguïtés et de rendre la description d'un service compréhensible par une machine.

la figure 2.3 décrit la structure tripartite d'une ontologie OWL-S. Elle est composée de trois sous-ontologie : un *service profile*, d'un *service grounding* et d'un *process model*.

---

15. <http://www.darpa.mil/>

16. <http://www.daml.org/services/>

#### **2.2.2.5 WSMO**

[17]

### **2.3 Découverte des services web**

WS discovery is related to getting appropriate service for a request. It is one of the critical steps in the process of developing applications based on SOA. It can be done using syntactic matching or semantic matching[34].

### **2.4 Conclusion**

## Chapitre 3

# La Composition des services web

## 3.1 Définition et types de composition

Cette section a pour but d'exposer, d'une part, quelques définitions et objectifs de la composition des services Web proposées par la communauté, et d'autre part, les différents types et mécanismes de composition selon différents points de vue rencontrés dans la littérature.

### 3.1.1 Définitions

Martin *et al.* [30] définissent la composition comme étant *“le processus de sélection, de combinaison et d'exécution de services en vue d'accomplir un objectif donné”*.

Selon S. Dustdar et W. Schreiner [35] : *“ L'infrastructure de base des services Web suffit pour la mise en œuvre d'interactions simples entre un client et un service Web. Si la mise en œuvre d'une application métier implique l'invocation d'autres services web, il est nécessaire donc de combiner les fonctionnalités de plusieurs services web. Dans ce cas, nous parlons d'une composition de services Web ”*.

En d'autre terme, La composition de services Web désigne une opération qui consiste à construire de nouvelles applications ou services appelés **services composites** ou agrégats par l'assemblage ou l'agrégation de services existants nommés **services atomiques** ou élémentaires.

Il existent différentes techniques de composition de services web développées par la littérature. ces techniques sont également classés en fonction de différents critères.

Selon les travaux, les définitions des types de composition diffèrent d'une communauté de l'autre.

Barros *et al.* [36] classent la composition des services Web en trois catégories : orchestration, chorégraphie et comportementale.

[2] de ça part distingue deux ...

### **3.1.2 Types de composition**

Selon [37].

#### **3.1.2.1 Composition manuel/automatique**

#### **3.1.2.2 Composition statique/dynamique**

3.2...



### 3.1.3 Procédés de coordination

Selon Peltz [2], Nous distinguons deux catégories de procédés de coordination utilisés pour décrire la composition de services dans un flot de processus métier. des procédés implémentant l'*orchestration* de services et des procédés implémentant la *chorégraphie* des services.

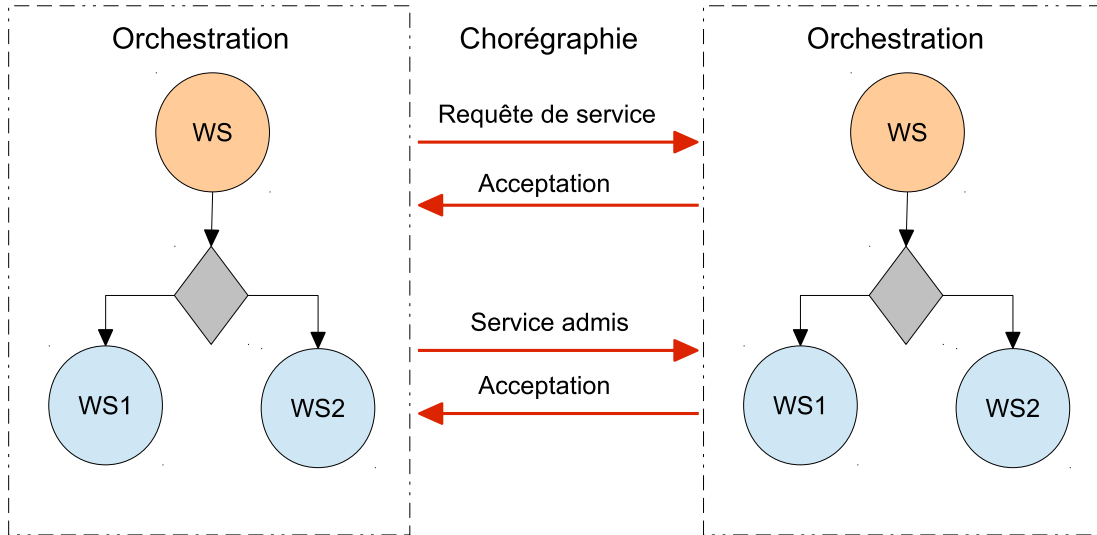


FIGURE 3.1 – Orchestration vs Chorégraphie selon Peltz [2].

Un **procédé** est représenté par un graphe orienté d'activités ou un flot de contrôle qui donne l'ordre d'exécution des activités et la logique de coordination des services. Chaque activité représente une fonctionnalité réalisée concrètement par un service [38]. La figure 3.1 illustre ces deux approches en conjonction.

#### 3.1.3.1 Orchestration

Barros *et al.* [36] définissent l'orchestration comme un ensemble de processus exécutés dans un ordre prédéfini afin de répondre à un but [18]. Ce type de composition se base sur un procédé métier exécutable permettant de décrire d'enchaînement et les interactions des différents services basiques collaborant dans une composition.

L'orchestration offre **une vision centralisée** de contrôle, le procédé est toujours contrôlé par l'un des partenaires métiers. Ce dernier joue le rôle d'un chef d'orchestre qui se charge d'appeler les services de la composition suivant l'ordre d'exécution déjà défini par le processus métier.

Le principe de l'orchestration est illustré par La figure 3.2.

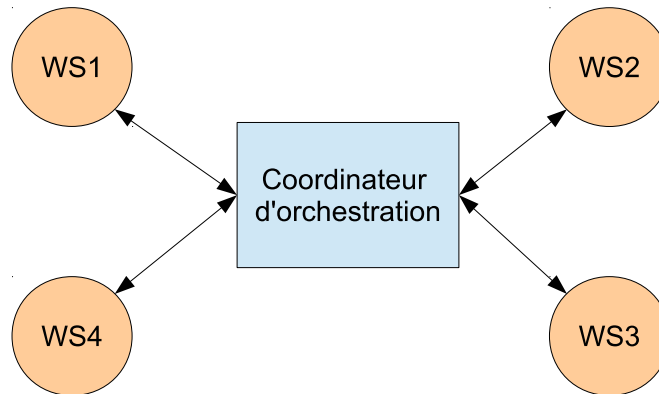


FIGURE 3.2 – Principe de l'orchestration des services Web.

### 3.1.3.2 Chorégraphie

D'après Barros *et al.* [36], la chorégraphie permet de décrire la composition comme un moyen d'atteindre un but commun en utilisant un ensemble de services Web. La collaboration entre chaque service Web de la collection (faisant partie de la composition) est décrite par des flots de contrôle [18].

La chorégraphie exprime une vue d'ensemble des services interagissant dans le cadre d'une composition de services. Selon Peltz [2], la chorégraphie illustre les différents échanges de messages entre les participants. la chorégraphie offre **une vision décentralisée** et globale.

Le principe de la chorégraphie est illustré par la figure 3.3.

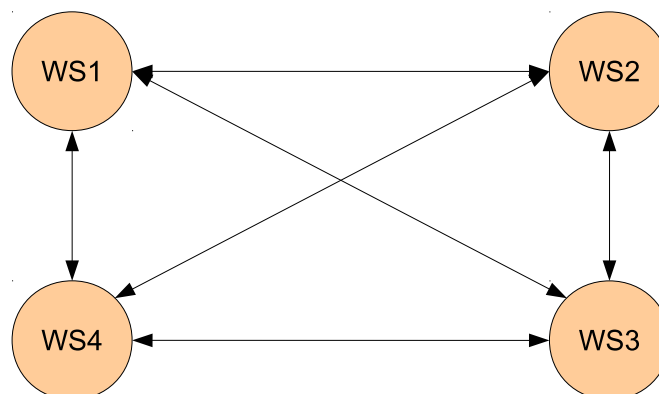


FIGURE 3.3 – Principe de la chorégraphie des services Web.

## 3.2 Langages de composition des services web

Afin de supporter la composition de services, plusieurs langages de composition de services ont été proposés comme ... Dans cette section on va détailler trois langages ... [18]

### 3.2.1 BPEL

BPEL est une spécification du consortium OASIS<sup>1</sup> issue de la fusion des spécifications XLANG Microsoft<sup>2</sup> et WSFL d'IBM<sup>3</sup>, il hérite les caractéristiques d'un langage structuré en blocs de XLANG, ainsi que les caractéristiques d'un graphe direct de WSFL [39].

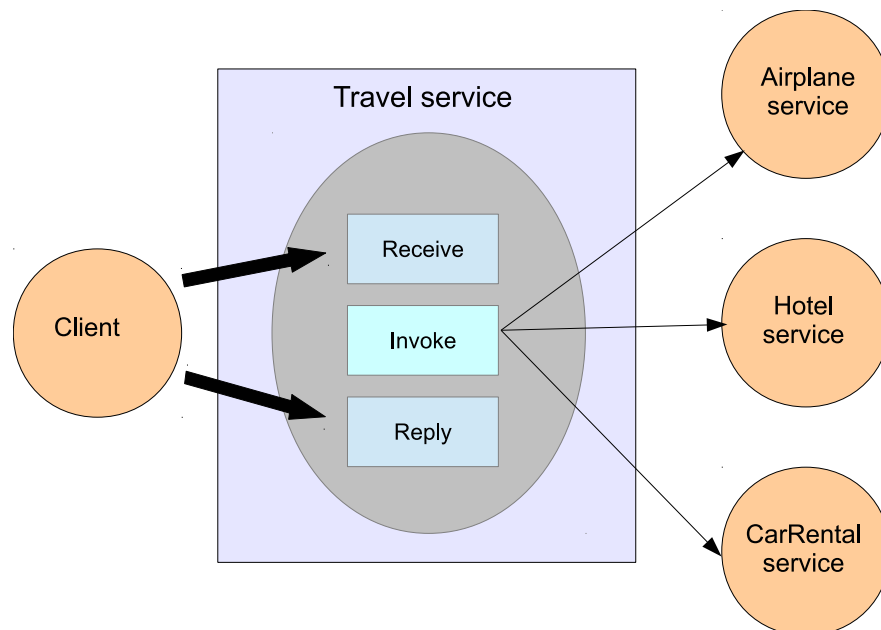


FIGURE 3.4 – Une vue interne d'un service de voyage BPEL [3].

BPEL (*appelé aussi BPEL4WS ou WS-BPEL*) est le langage d'**orchestration** le plus utilisé dans l'industrie permettant la coordination des interactions entre l'instance du service composite et ses partenaires sous forme d'un schéma XML (*le script d'orchestration*), il définit le processus, l'enchaînement et l'ordonnement des actions qui seront exécutées par le moteur d'orchestration, agissant

---

1. <https://www.oasis-open.org>

2. <http://www.microsoft.com>

3. <http://www.ibm.com>

comme une machine virtuelle capable d'exécuter **le procédé métier** intéreptable de **coordination** [38].

BPEL repose sur un modèle constitué d'activités de coordination qui peuvent être de deux types, les activités de base ou élémentaires comme l'invocation (invoke) d'un service, l'attente d'une réponse et la génération d'une réponse (reply), et les activités composites permettant du contrôle du flot de données comme les séquences (sequence), les exécutions en parallèle (flow) et les branchements (switch, if). La figure 3.5 représente une opération de type "séquence" illustrée dans le schémas 3.4.

```
<sequence name="TravelSequence">
  <receive partnerLink="client"
    portType="tns:travelPT"
    operation="trip"
    variable="request"
    createInstance="yes"/>
  <invoke name="invokeAirplane"
    partnerLink="airplane"
    portType="sairplane:Airplane"
    operation="bookAirplane"
    inputVariable="request"
    outputVariable="airplaneReturn">
  </invoke>
  <invoke name="invokeHotel"
    partnerLink="hotel"
    portType="shot:Hotel"
    operation="bookHotel"
    inputVariable="request"
    outputVariable="hotelReturn">
  </invoke>
  <invoke name="invokeCar"
    partnerLink="car"
    portType="scar:Car"
    operation="rentcar"
    inputVariable="request"
    outputVariable="totalReturn">
  </invoke>
  <reply partnerLink="client"
    portType="tns:travelPT"
    operation="trip"
    variable="carReturn"/>
</sequence>
```

FIGURE 3.5 – Le procédé d'orchestration de service de voyage BPEL [3].

### 3.2.2 WS-CDL

WS-CDL est un langage de composition de services de type **chorégraphie** qui permet de décrire une vision **globale** des collaborations entre les services Web [16], à l'instar des standards de services Web, WS-CDL est basé sur XML, il complète la description WSDL des services Web afin de décrire les interactions entre les participants (les services Web) de la composition.

WS-CDL reprend et développe la spécification WSCI décrivant les séquences ordonnées de messages impliquant plusieurs entités (services Web) engagés dans une composition visant à accomplir un objectif commun.

WS-CDL consiste à définir un fichier XML décrivant une chorégraphie, il permet de [16] :

- désigner les variables et les types de données échangées.
- décrire les activités impliquées.
- décrire les structures illustrant les interactions entre les activités.

### 3.2.3 OWL-S

[33]

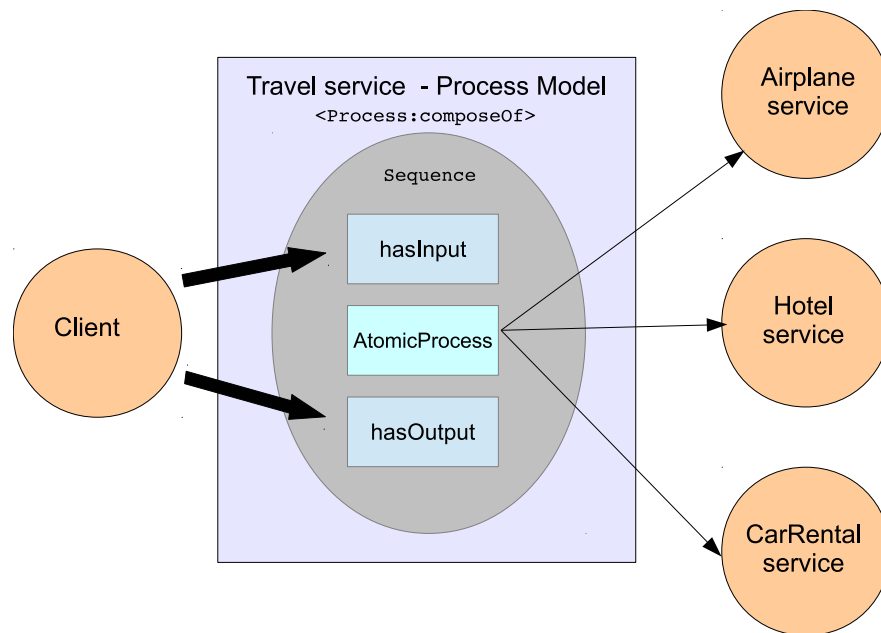


FIGURE 3.6 – Une vue interne d'un service de voyage composite OWLS.

```

<process:ProcessModel rdf:ID="TravelProcessModel">
  <service:describes
    rdf:resource="#TravelService"/>
  <process:hasProcess
    rdf:resource="#TravelProcess"/>
</process:ProcessModel>
<process:CompositeProcess rdf:ID="TravelProcess">
  <process:hasInput rdf:resource="#dt_arrival"/>
  <process:hasInput rdf:resource="#dt_departure"/>
  <process:hasInput
    rdf:resource="#destination_city"/>
  <process:hasOutput rdf:resource="#total"/>
  <process:composedOf>
    <process:Sequence>
      <process:components
        rdf:parseType="Collection">
        <process:AtomicProcess
          rdf:about="Airplane.owl#AirplaneProcess"/>
        <process:AtomicProcess
          rdf:about="Hotel.owl#HotelProcess"/>
        <process:AtomicProcess
          rdf:about="CarRental.owl#CarRentalProcess"/>
      </process:components>
    </process:Sequence>
  </process:composedOf>
</process:CompositeProcess>

```

FIGURE 3.7 – Document OWLS de processus composite du service de voyage BPEL.

## 3.3 Composition dynamique des services web

### 3.3.1 Techniques de planification

[40]

Introduire la composition dynamique basé sur le modèle graphe qui se sera détaillé dans le prochain chapitre.

## 3.4 Conclusion

# Bibliographie

- [1] Dieter FENSEL et Christoph BUSSLER : Semantic web enabled web services. *Advances in Artificial Intelligence*, page 316, 2002.
- [2] Chris PELTZ : Web services orchestration and choreography. *Computer*, 36(10):46–52, 2003.
- [3] Daniela Barreiro CLARO, Patrick ALBERS et Jin-Kao HAO : Web services composition. *In Semantic Web Services, Processes and Applications*, pages 195–225. Springer, 2006.
- [4] Heather KREGER *et al.* : Web services conceptual architecture (wsca 1.0). *IBM Software Group*, 5:6–7, 2001.
- [5] W3C Working GROUP : Web services architecture. <http://www.w3.org/TR/ws-arch/>, 2004.
- [6] Paul FREMANTLE, Sanjiva WEERAWARANA et Rania KHALAF : Enterprise services. *Communications of the ACM*, 45(10):77–82, 2002.
- [7] Don BOX, David EHNEBUSKE, Gopal KAKIVAYA, Andrew LAYMAN, Noah MENDELSON, Henrik Frystyk NIELSEN, Satish THATTE et Dave WINER : Simple object access protocol (soap) 1.1, 2000.
- [8] Tim BRAY, Jean PAOLI, C Michael SPERBERG-McQUEEN, Eve MALER et François YERGEAU : Extensible markup language (xml). *World Wide Web Consortium Recommendation REC-xml-19980210*. <http://www.w3.org/TR/1998/REC-xml-19980210>, 1998.
- [9] Mike P PAPAZOGLOU : Service-oriented computing : Concepts, characteristics and directions. *In Web Information Systems Engineering, 2003. WISE 2003. Proceedings of the Fourth International Conference on*, pages 3–12. IEEE, 2003.



- [10] Francisco CURBERA, William NAGY et Sanjiva WEERAWARANA : Web services : Why and how. *In OOPSLA 2001 Workshop on Object-Oriented Web Services*, 2001.
- [11] Francisco CURBERA, Matthew DUFTLER, Rania KHALAF, William NAGY, Nirmal MUKHI, Sanjiva WEERAWARANA *et al.* : Unraveling the web services web. *IEEE Internet computing*, 6(2):86–93, 2002.
- [12] Karl GOTTSCHALK, Stephen GRAHAM, Heather KREGER et James SNELL : Introduction to web services architecture. *IBM Systems journal*, 41(2):170–177, 2002.
- [13] Nilo MITRA, Yves LAFON *et al.* : Soap version 1.2 part 0 : Primer. *W3C recommendation*, 24:12, 2003.
- [14] Roberto CHINNICI, Jean-Jacques MOREAU, Arthur RYMAN et Sanjiva WEERAWARANA : Web services description language (wsdl) version 2.0 part 1 : Core language. *W3C recommendation*, 26:19, 2007.
- [15] Luc CLEMENT, Andrew HATELY, Claus von RIEGEN, Tony ROGERS *et al.* : Uddi version 3.0. 2, uddi spec technical committee draft. *OASIS UDDI Spec TC*, 2004.
- [16] ABI LAHOUD ELIE : *Composition dynamique de services : application à la conception et au développement de systèmes d'information dans un environnement distribué*. Thèse de doctorat, Université de Bourgogne, 2010.
- [17] George BARYANNIS et Dimitris PLEXOUSAKIS : Automated web service composition : State of the art and research challenges. *ICS-FORTH, Tech. Rep*, 409, october 2010.
- [18] Céline LOPEZ-VELASCO : *Sélection et composition de services Web pour la génération d'applications adaptées au contexte d'utilisation*. Thèse de doctorat, Université Joseph-Fourier-Grenoble I, 2008.
- [19] XML Schema PART : 2 : Datatypes. *W3C Recommendation*, 2, 2001.
- [20] Tim BERNERS-LEE, James HENDLER, Ora LASSILA *et al.* : The semantic web. *Scientific american*, 284(5):28–37, 2001.
- [21] Alexandre BERTAILS, Ivan HERMAN et Sandro HAWKE : Le web sémantique. *Réalités industrielles*, (4):84–89, 2010.

- [22] Sheila A McILRAITH, Tran Cao SON et Honglei ZENG : Semantic web services. *IEEE intelligent systems*, 16(2):46–53, 2001.
- [23] Dieter FENSEL, Frank VAN HARMELEN, Ian HORROCKS, Deborah L MCGUINNESS et Peter F PATEL-SCHNEIDER : Oil : An ontology infrastructure for the semantic web. *Intelligent Systems, IEEE*, 16(2):38–45, 2001.
- [24] Ian HORROCKS *et al.* : Daml+oil : a description logic for the semantic web. *IEEE Data Eng. Bull.*, 25(1):4–9, 2002.
- [25] Deborah L MCGUINNESS, Richard FIKES, Lynn Andrea STEIN et James A HENDLER : Daml-ont : An ontology language for the semantic web. *In Spinning the Semantic Web*, pages 65–93, 2003.
- [26] Dieter FENSEL, Jim HENDLER, Henry LIEBERMAN et Wolfgang WAHLSTER : Creating of semantic web. *Im Internet verfügbar unter <http://citeseer.nj.nec.com/481673.html>*, 2000.
- [27] Thomas R GRUBER : A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2):199–220, 1993.
- [28] Rama AKKIRAJU, Joel FARRELL, John A MILLER, Meenakshi NAGARAJAN, Amit SHETH et Kunal VERMA : Web service semantics-wsdl-s. 2005.
- [29] Jacek KOPECKY, Tomas VITVAR, Carine BOURNEZ et Joel FARRELL : Sawsdl : Semantic annotations for wsdl and xml schema. *Internet Computing, IEEE*, 11(6):60–67, 2007.
- [30] David MARTIN, Mark BURSTEIN, Jerry HOBBS, Ora LASSILA, Drew McDERMOTT, Sheila McILRAITH, Srinu NARAYANAN, Massimo PAOLUCCI, Bijan PARSIA, Terry PAYNE *et al.* : Owl-s : Semantic markup for web services. *W3C member submission*, 22:2007–04, 2004.
- [31] Anupriya ANKOLEKAR, Mark BURSTEIN, Jerry R HOBBS, Ora LASSILA, David MARTIN, Drew McDERMOTT, Sheila A McILRAITH, Srinu NARAYANAN, Massimo PAOLUCCI, Terry PAYNE *et al.* : Daml-s : Web service description for the semantic web. *In The Semantic Web ISWC 2002*, pages 348–363. Springer, 2002.
- [32] Deborah L MCGUINNESS, Frank VAN HARMELEN *et al.* : Owl web ontology language overview. *W3C recommendation*, 10(2004-03):10, 2004.

- [33] Sheila A McILRAITH et David L MARTIN : Bringing semantics to web services. *Intelligent Systems, IEEE*, 18(1):90–93, 2003.
- [34] Abrehet Mohammed OMER : *A framework for Automatic Web Service Composition based on service dependency analysis*. Thèse de doctorat, Technical University of Dresden, 2011.
- [35] Schahram DUSTDAR et Wolfgang SCHREINER : A survey on web services composition. *International journal of web and grid services*, 1(1):1–30, 2005.
- [36] Alistair BARROS, Marlon DUMAS et Phillipa OAKS : Standards for web service choreography and orchestration : Status and perspectives. *In Business process management workshops*, pages 61–74. Springer, 2006.
- [37] Matthias FLUEGGE, Ivo JG SANTOS, Neil Paiva TIZZO et Edmundo RM MADEIRA : Challenges and techniques on the road to dynamically compose web services. *In Proceedings of the 6th international conference on Web engineering*, pages 40–47. ACM, 2006.
- [38] Stéphanie CHOLLET : *Orchestration de services hétérogènes et sécurisés*. Thèse de doctorat, Université Joseph-Fourier-Grenoble I, 2009.
- [39] Maha DRISS : *Approche multi-perspective centrée exigences de composition de services Web*. Thèse de doctorat, Université Rennes 1, 2011.
- [40] Peter BARTALOS : Effective automatic dynamic semantic web service composition. *Inf. Sci. and Technol. Bulletin ACM Slovakia*, 3(1):12–14, 2011.