composition des services web

TAOURIRIT Salah Eddine

7 décembre 2014

version 0.2.3

Table des matières

Acronyms

BPEL Business Process Execution Language. 28, 37

BPEL4WS Business Process Execution Language for Web Services. 28

CCS Calculus of Communicating Systems. 37

CSP C. 37

DAML the DARPA Agent Markup Language. 15, 30

DARBA Defence Advanced Research Projects Agency. 30

HTTP Hypertext Transfer Protocol. 5

MDO Model driven engineering. 36

QoS Quality of service. 23

SAWSDL Semantic Annotations for WSDL and XML Schema. 17

SOAP Simple Object Access Protocol. 8

UDDI Universal Description Discovery and Integration. 7, 8, 12

W3C The World Wide Web Consortium. 5, 11, 13

WS-BPEL Web Service Business Process Execution Language. 28

WS-CDL Web Services Choreography Description Language. 29

WSCI Web Service Choreography Interface. 29

WSDL Web Services Description Language. 7, 8, 11, 13, 29

WSFL Web Services Flow Language. 28

WSMF The Web Service Modeling Framework. 29, 37

WSMO Web service modeling ontology. 37

XLANG XML Business Process Language. 28, 37

XML Extensible Markup Language. 28

Table des figures

Liste des tableaux

Chapitre 1

Introduction

Première partie

Étude bibliographique

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 quelque définitions proposées dans la littérature. Ensuite nous nous intéressons à montrer les limitation 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 [5] et Microsoft, puis en standardisés par le W3C [1] et définis [6] 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 [7] :

- 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 [8] 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 [9] offre un cadre standard pour la définition de documents Interprétable par des machines.

1. `http://www.w3.org/`

M. P. Papazoglou [10] apporte une autre définition de services web :
"Les services Web sont des éléments auto-descriptifs et indépendants des plate-formes 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. [11] 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 décris utilisant un langage de description fonctionnel standardisé".

2.1.2 L'évolution des styles des services web

2.1.3 L'architecture de référence et technlogies associées

[12] [13] [5] [6] 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 servicee [2], un annuaire de services [3], et un client [4] de service. La figure x montre comment ces trois rôles interagissent.

- Le fournisseur Un prestataire de services fournit l'interface pour le service Web et l'implémentation de l'application. Le fournisseur de service est également responsable de la création de la définition du service et de publier cette définition pour répondre à la spécification UDDI
- L'annuaire Un registre de service est une façon dont les services Web sont officiellement publiés. Le registre de service est basée sur la spécification UDDI et reflète des informations sur les services fournis par le fournisseur de services. Le registre de service fournit un demandeur de services avec une description de service WSDL et un URL qui pointe vers le service lui-même.
- Le client Un client de services est le consommateur d'un service Web, il utilise le registre de service pour obtenir des informations et pour pouvoir accèder à un service Web.

Pour qu'une application profite des services Web, trois comportements doivent avoir lieu : la publication des descriptions du service, correspondances des descriptions du service et la liaison ou l'invocation des services basés sur la description du service. Ces comportements peuvent se produire d'une manière singulière ou demanière itérative. En détail, ces opérations sont les suivantes :

- Publish Pour être accessible, une description de service doit être publiée afin que le demandeur de service puisse la trouver. l'emplacement où elle est publiée peut varier selon les exigences de l'application.

2. Providers
3. Service Registry
4. Service Requester

- Découvrir Dans l'opération de recherche, le demandeur de service récupère une description du service directement ou interroge le registre de service pour le type de service requis. Cette opération peut être impliquée dans deux différentes phases du cycle de vie pour le demandeur de service : au moment du design pour récupérer la description d'interface de service pour le développement du programme et à l'exécution pour récupérer la liaison du service et l'emplacement de la description pour l'invocation.

- lier (bind) Finalement, un service doit être invoqué. Dans l'opération de liaison, le client de service invoque ou initie une interaction avec le service à l'exécution en utilisant les détails de liaison dans la description du service pour localiser, contacter et appeler le 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 [5] et Microsoft [6] [8], 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 [14].

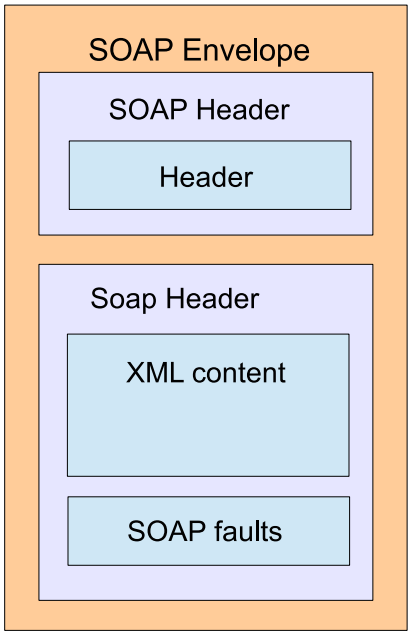


FIGURE 2.1 – Les éléments d'un message SOAP

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

5. `http://www.ibm.com`
6. `http://www.microsoft.com`

peuvent être implémentées comme des en-têtes sont des authentifications, des transactions, des paiements

- Corps `<Body>` : Contient les informations obligatoires destinées à l'ultime 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 [15] 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 desWSDL sont composées d'un interface et des implémentations. L'interface est une définition abstraite et réutilisable service qui peut être référencée par plusieurs implémentations.

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 [16] est une standardisation pour la publication et la découverte des services Web initialement conçue et spécifiée par le Consortium de standards OASIS [7], et il est le résultat d'un accord d'un ensemble d'industriels Ariba [8], 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 le 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).

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.

7. `https://www.oasis-open.org`
8. `http://www.ariba.com/`

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 [15] 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 dynamisité 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 [15] 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 [9] est un document XML qui se divise en deux parties [17] :

- 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 adresse réseau du fournisseur de service [10], et les protocoles spécifiques d'accès.

Un document WSDL constitué de quatre éléments principaux [15] : `<Types>`, `<Interface>`, `<Binding>`, `<Service>`.

9. `http://www.w3.org/TR/wsdl20/`
10. Service Endpoint

[18] [17] [19]

- <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 [20].

- <Interface> : Les interfacesWDSL 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 [11] suivie lorsque l'opération est invoqué. Pour chaque message contenu dans le motif [12], 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> : [17] [18] 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 référençant l'élément `<binding>` associé et en déclarant l'URL localisant le service (avec l'attribut `<address>`).

11. message exchange pattern
12. pattern

2.2.2 Ajout de la sémantique

Malgré les améliorations apportées au standard WSDL dans son deuxième version [15], 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 émergeants 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 [21]. 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 [22].

La réalisation du Web sémantique trouve ces racines dans le développement des langages de balisage inspiré par des travaux issue de la communié AI [23], tels que OIL [24], DAML+OIL [25] et DAML+OTN [26] (ces deux derniers langages sont parties de la famille DAML).

Ces langaes ont une sémantique bien définies et permettent le balisage et la manipulation des taxonomique complexe et Des relations logiques entre les entités

sur le Web. [27]

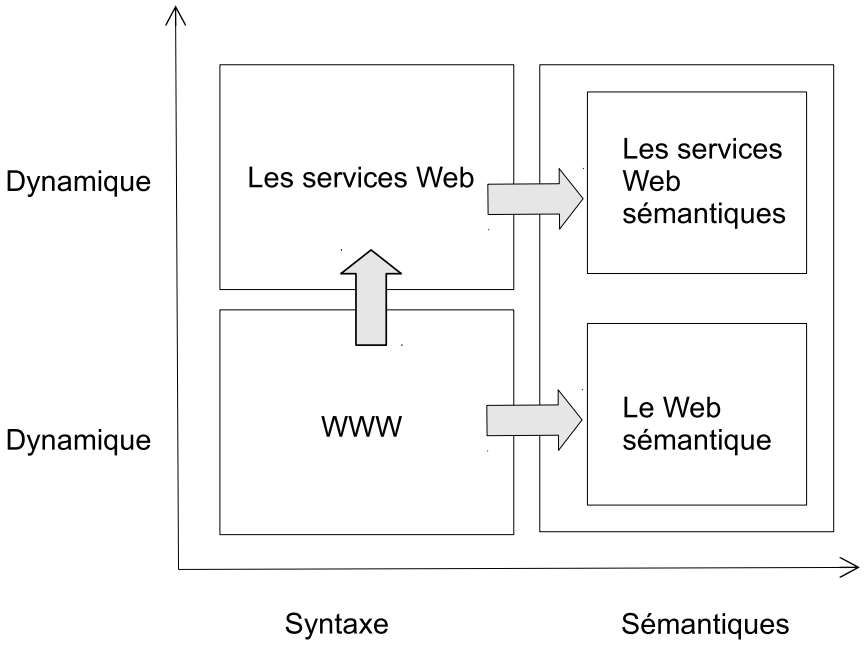


FIGURE 2.2 – Web evolution to Semantic Web services [1].

Cette description repose sur des ontologies. Selon Gruber [28], 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 [29] est le résultat d'un travail collaboratif entre IBM, laboratoire LSDSI et l'iniversité de Geogia [13]. La spécification a devenue une recommandation 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 capabilités du modèle WSDL en rajoutant trois éléments majeurs `<category>`, `<effect>` et deux attributs `modelReference`

13. `http://www.uga.edu/`

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 découvert de services.

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

2.2.2.3 SAWSDL

La spécification SAWSDL [30] 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 [14] 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 [19] 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`).

2.2.2.4 OWL-S

2.2.2.5 WSMO

[18]

14. Semantic Annotations for WSDL and XML Schema

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[31].

2.4 Conclusion

Chapitre 3

La Composition des services web

3.1 Définition et stratégies 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.* [32] 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 [33] : *" 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.

Medjahed [34]de ça part a défini un service Web composite commme un *"conglomérat de sous-traitance services Web (services appelés participants) travaillant en tandem pour offrir un service à valeur ajoutée."*

La composition de services Web vise essentiellement quatre objectifs [35] :

1. Créer de nouvelles fonctionnalités en combinant des services déjà existants.
2. Résoudre des problèmes complexes auxquels aucune solution n'a été trouvée.
3. collaborer plusieurs entreprises ensemble.
4. Optimiser et améliorer une fonctionnalité existante.

3.1.2 Cycle de vie d'une composition

Comme l'illustre la figure, le cycle de vie de la composition de services Web comprend quatre phases [2] : la phase de *définition*, La phase de *sélection*, la phase de *déploiement* et la phase d'*exécution* :.

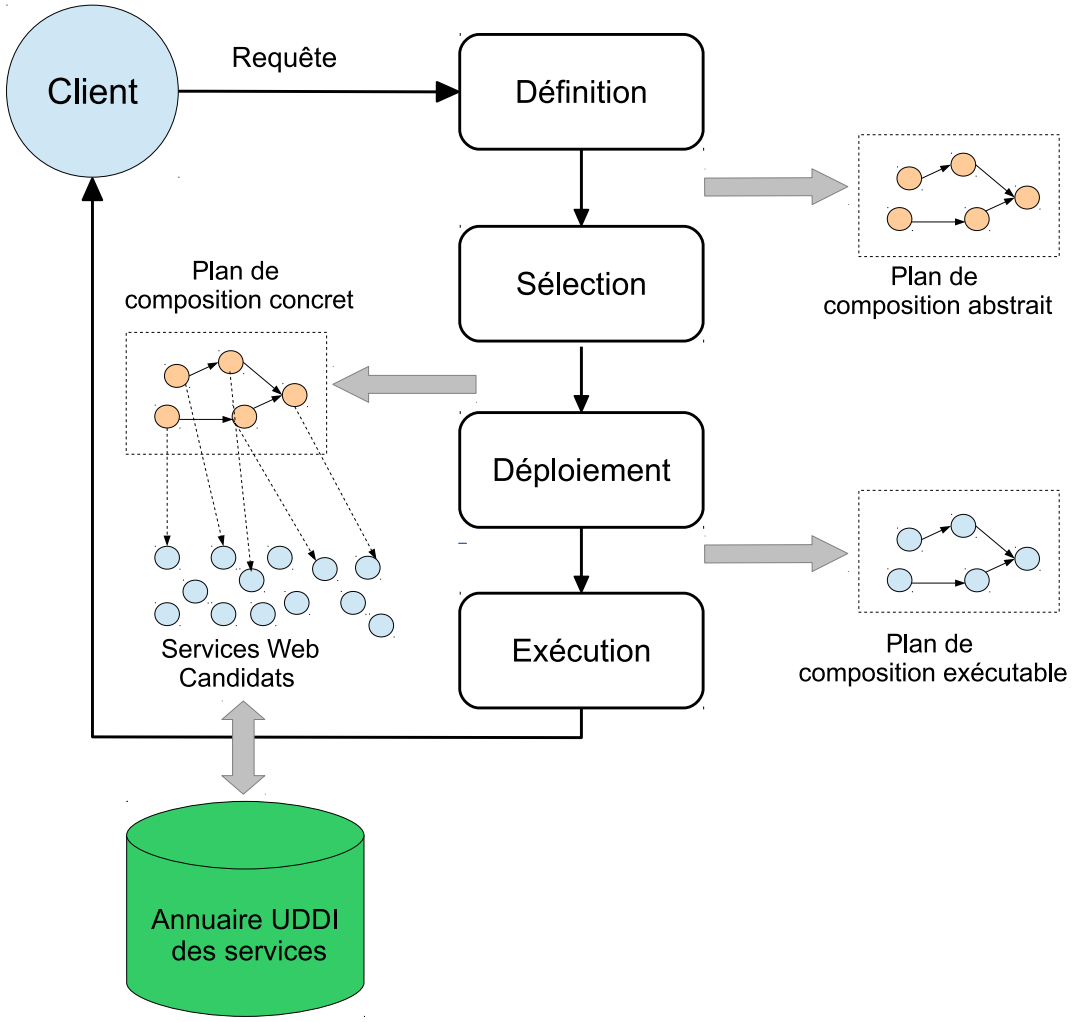


FIGURE 3.1 – Cycle de vie d'une composition des services [2].

- La phase de définition Pendant cette phase, le client de service spécifie les exigence de composition des services en termes des besoins et des préférences pour le service composite. l'exigence est ensuite *décomposé*, soit semi-automatique ou automatique, dans un modèle de **processus abstrait**(ce est à dire, le service composite abstraite), qui spécifie un

ensemble d'activités, le contrôle et le flux de données entre eux, la qualité de service QoS et la gestion des exceptions.

- La phase de séléction. Dans cette phase, pour chaque activité dans le service composite, les services Web appropriés qui répondent aux exigences de l'activité sont situés en cherchant sur le registre de service, sur la base des informations contenues dans les documents de description de service publiées. Il est probable que plus d'un service de candidat de répondre aux exigences. Par conséquent, le meilleur le service identifié doit être sélectionné. Après tous les services Web requis sont identifiés et liés aux activités correspondantes, le service composite construit est produite.

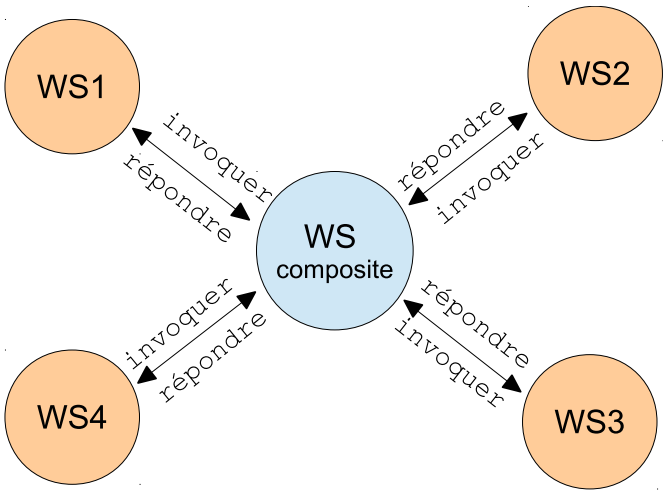
- La phase de déploiement. Dans cette phase, le service composite construit est déployé pour permettre son instanciation et l'invocation par les utilisateurs finaux. Le résultat de cette phase est le service composite exécutable.

- La phase d'exécution. Dans cette phase, l'instance de service composite est créé et exécuté par le moteur d'exécution, qui est aussi responsable de l'invocation des composants de service atomiques. Pendant l'exécution de l'instance de service composite, les tâches de surveillance, y compris le suivi d'exécution, mesure de la performance et la gestion des exceptions, doivent être effectuées.

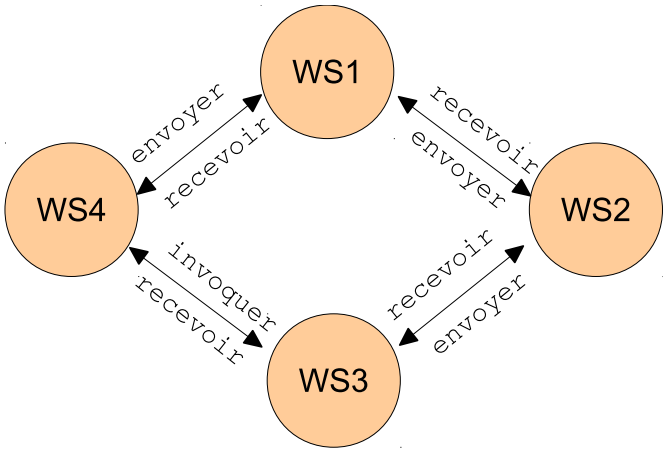
3.1.3 Procédés de coordination

Nous distinguons deux méthodes utilisés pour décrire la composition de services dans un flot de processus métier : l'*orchestration* de services et la *chorégraphie* des services. Ces deux procédés de coordination décrivent deux aspects de création des processus métiers à partir des services Web composites [36]. **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 [37].

La figure 3.2 illustre ces deux approches en conjonction.



(a) l'orchestration des services.



(b) la chorégraphie des services.

FIGURE 3.2 – Orchestration vs Chorégraphie.

3.1.3.1 Orchestration

Selon Sonia *et al.* [38] : *"L'orchestration des services Web permet de définir l'arranegement et l'enchaînement de ces services selon un canevas bien défini. Elle décrit la manière par laquelle les services peuvent interagir ensemble tout en incluant l'ordre d'exécution des différentes interactions"*.

Barros *et al.* [39] définissent l'orchestration comme un ensemble de processus exécutés dans un ordre prédéfini afin de répondre à un but [19]. 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 tou-

jours 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 3.2a.

3.1.3.2 Chorégraphie

Selon Sonia *et al.* [38] : *" La chorégraphie permet de tracer la séquence de messages échangés dans un contexte de composition de services Web. Elle est typiquement liée à la description de conversations existantes entre les services tout en impliquant plusieurs parties, incluant les clients, les fournisseurs et les partenaires".*

D'après Barros *et al.* [39], 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 [19].

La chorégraphie offre **une vision décentralisée** et **globale** du système et exprime une vue d'ensemble des services interagissant dans le cadre d'une composition de services. Selon Peltz [36], la chorégraphie illustre les différants échanges de messages entre les participants. Le principe de la chorégraphie est illustré par la figure 3.2b.

3.1.4 Stratégies de composition

Un modèle de composition de service peut être relativement complexe. Il requiert la description et l'organisation de l'interaction entre les services et nécessite la gestion de plusieurs aspects comme les échanges de données entre les services, les pannes ou erreurs éventuelles, le contexte d'interaction, le degré d'automatisation des tâches, etc.

Il existent une variété de spécifications, de langages et d'approches formelles développées par la littérature concernant la composition. Ces techniques sont également classés en fonction de différents dimensions, et selon les travaux effectués dans le champ des services web, les définitions des types de composition diffèrent d'une communauté de l'autre.

Barros *et al.* [39] classent la composition des services Web en trois catégories : la composition comportementale, l'orchestration et la chorégraphie, à l'instar de Barros *et al.*, Peltz [36] considère que les deux dernières *(orchestration, chorégraphie)*.

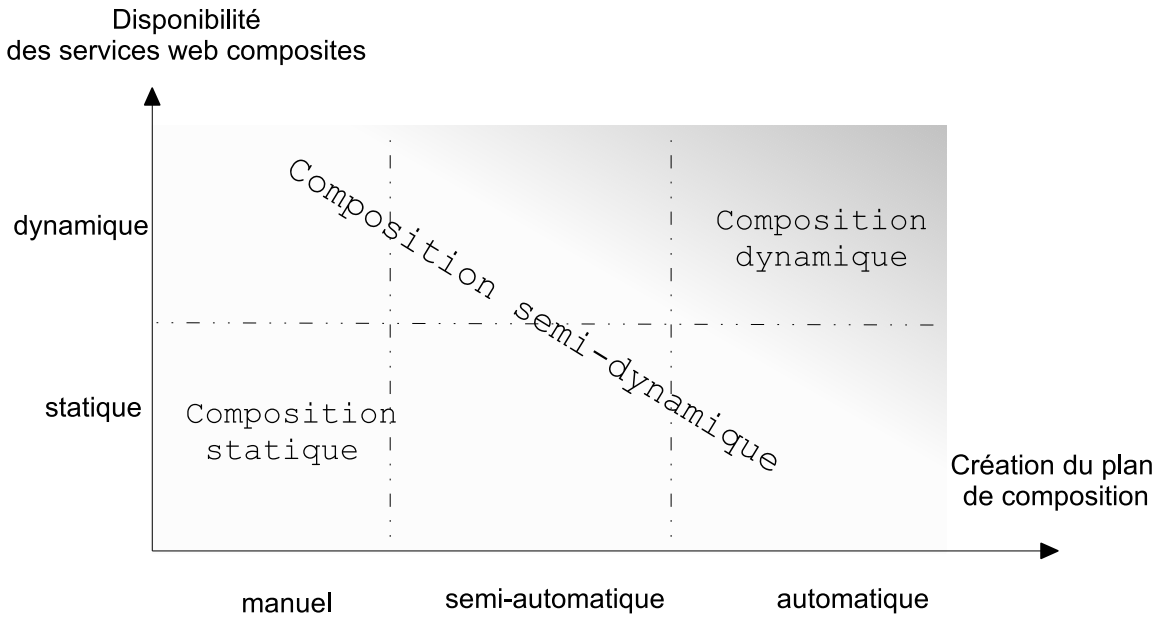


FIGURE 3.3 – Classification des stratégies de composition [3].

D'une autre façon, Fluegge *et al.*[3] dans une analyse de l'état de l'art considèrent l'orchestration et la chorégraphie comme des modèles d'exécution appliqués dans le contexte d'une composition. Il distingue trois stratégies de composition selon la disponibilité des services Web composites lors de composition et de le dégré d'automatisation : composition **statique**, **semi-dynamique** et **dynamique** (voir la figure 3.3).

3.1.4.1 Composition statique/dynamique

Selon la disponibilité des services composites, La composition des services Web peut être soit une composition statique soit une composition dynamique [35] :

- Composition statique :est appelé aussi composition *off-line*, précompil ou encore proactive. c'est une composition qui utilise des services basiques qui sont au préalablement définis d'une façon figée et qui ne peuvent pas changer en fonction du contexte du client. Ce type de composition

engendre des applications peu flexibles, parfois inappropriées avec les exigences des clients.

- Composition dynamique : appelée aussi composition *on-line*, post-compilée ou encore réactive. Elle se réfère à la sélection des services basiques à la volée. Autrement dit, la sélection des services basiques ne peut pas être définie à l'avance mais elle sera faite au moment de l'exécution en fonction des contraintes imposées par le client. Ceci permet d' élaborer différents scénario de composition qui offrent les mêmes fonctionnalités et qui tiennent compte de la dynamique de la situation du client.

3.1.4.2 Composition manuel/automatique

Classification basée sur le degré d'automatisation.

- Composition manuel : Suppose que l'utilisateur gère la composition avec sa main, via un éditeur de texte et sans l'aide d'outils dédiés.

- Composition semi-automatique : C'est un pas en avant en comparaison avec la composition manuelle, dans le sens qu'elle fait des suggestions sémantiques pour aider à la sélection des services Web dans le processus de composition.

- Composition automatique : La composition automatique (ou encore dynamique selon [3]) permet un développement plus rapide des applications à base de services. Elle consiste à préciser la requête d'un utilisateur sous forme d'objectifs à satisfaire. Un moteur de composition *"intelligent"* choisit la comabinaison de services répondant à l'objectif décrit. Il génère la composition de service adéquate de manière transparente à l'utilisateur. Ce principe a interpellé plusieurs communautés de recherche travaillant dans le domaine de l'Intelligence Artificielle. [17]

3.2 Langages pour la composition

Afin de supporter la composition de services Web, plusieurs langages de composition de services ont été proposés pour décrire et mettre en oeuvre une com-

position. Dans cette section on va faire un tour d'horizon de quelques standards et langages principaux rencontrés dans la littérature.

3.2.1 BPEL

BPEL est une spécification du consortium OASIS [1]issue de la fusion des spécifications XLANG Microsoft [2]et WSFL d'IBM [3], il hérite les caractéristiques d'un langage structuré en blocs de XLANG, ainsi que les caractéristiques d'un graphe direct de WSFL [35].

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'ordonnancement des actions qui seront exécutées par le moteur d'orchestration, agissant comme une machine virtuelle capable d'exécuter **le procédé métier** intéreptable de **coordination** [37].

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`).

3.2.2 WS-CDL

WS-CDL [4] [40] est un langage de composition de services de type **chorégraphie** qui permet de décrire une vision **globale** des collaborations entre les services Web [17], 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 points d'interactions entre les services Web engagés dans une composition. Contrairement à la spécification BPEL 3.2.1, Les interactions des services sont décrites d'une

1. `https://www.oasis-open.org`
2. `http://www.microsoft.com`
3. `http://www.ibm.com`
4. `http://www.w3.org/TR/ws-cdl-10/`

manière *peer-to-peer*, Il n'y a pas de notion de coordination ou d'un service Web principal d'orchestration.

WS-CDL reprend et développe la spécification WSCI [5] [41] décrivant les séquences ordonnées de messages impliquant plusieurs entités (services Web) engagés dans une composition visant à accomplir un objectif commun, il permet de décrire les règles selon lesquelles une collaboration doit avoir lieu par le biais d'un fichier XML décrivant une chorégraphie.

3.2.3 WSMF

WSMF [42]est un initiative européen pour fournir une plate-forme riche de modélisation décrivant plusieurs aspects de services web. Son objectif principale est de permettre le commerce éléctronique (*E-commerce*) par l'application de web sémantique aux services Web.

Le standard WSMF est centré autour de deux principes complémentaire [18] :

1. Découplage fort de différents aspetcs des applications de commerce électronique.
2. Des mécanismes de médiation permettant un dialogue automatique entre les différents composants.

WSFM comprend quatre éléments principaux [18] :

1. Des ontologies qui fournissent la terminologie utilisée par les autres éléments.
2. Un répertoire d'objectifs qui définit les problèmes qui doivent être résolus par les Web services.
3. Des descriptions des Web services qui définissent les différents aspects liés aux Web services
4. Des médiateurs qui sont en charge des problèmes d'interopérabilité.

Il y a deux principaux projets en WSMF : SWWS et WSMO, le standard SWWS fournit un cadre de description, la découverte et la médiation pour les services Web, tandis que l'ontologie de service WSMO inclut des définitions pour les objectifs, les médiateurs et les services Web.

5. `http://www.w3.org/TR/wsci/`

3.2.4 OWL-S

OWL-S [32] désigné par DAML-S [43] dans les versions antérieures, est un langage issue des travaux de DARBA [6] et son programme DAML [7] en collaboration avec des chercheurs de plusieurs universités et organisations. 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 [25, 44]. Ankolekar *et al.* [43] 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 [45], les auteurs reprennent la notion de classes d'OWL et proposent l'ontologie OWL-S.

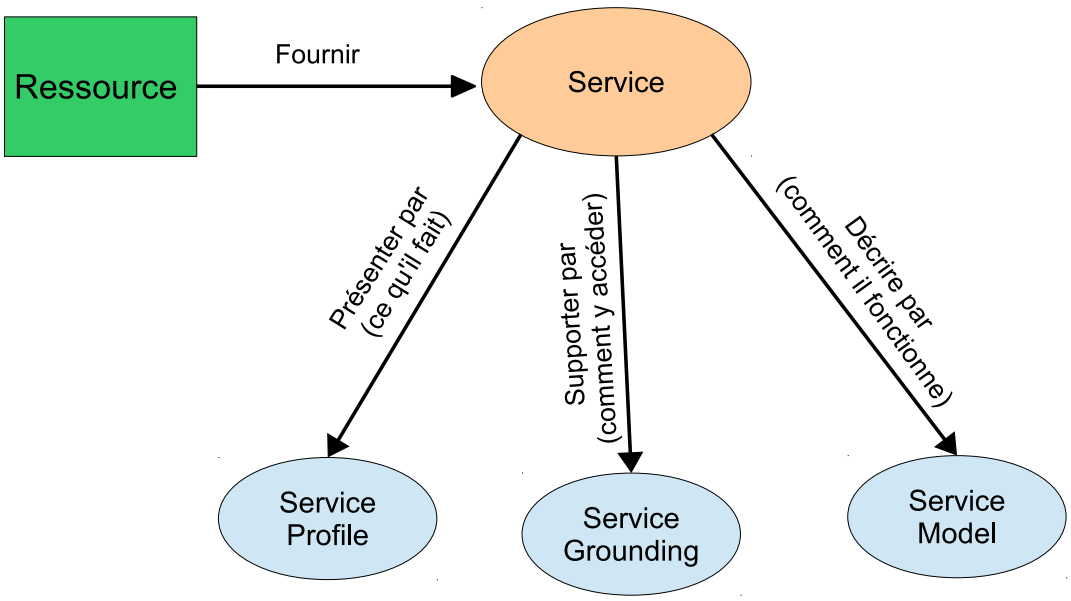


FIGURE 3.4 – Les éléments d'une ontologie OWL-S [4]

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 ensembles 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 3.4 décrit la structure tripartie d'une ontologie OWL-S. Elle est

6. `http://www.darpa.mil/`
7. `http://www.daml.org/services/`

composé de trois sous-ontologie : un *service profile*, d'un *service model* et d'un *service grounding*.

- ServiceProfile. il offre une description informelle des fonctionnalités rendues par le service (`serviceName`, `textDescription`) et des informations concernant son fournisseur (`contactInformation`). D'une aure côté, il spécifie des fonctionnalités offertes par le service (comportement fonctionnel) en terme de transformation d'information dénotée par les *Entrés/Sorties* (I/O) (`hasInput`, `hasOutput`) et de changement d'état après l'exécution du service dénoté par les *pré-conditions/effets* (P/E) (`hasPrecondition`, `hasResult`).

  Du point de vue de découverte et de la composition, Le `ServiceProfile` est la partie la plus importante de la définition du service OWL-S.

- ServiceModel. il décrit le fonctionnement du service du en indiquant comment les résultats sont produits étape par étape précisant la façon un client peut interagir avec le service afin d'atteindre sa fonctionnalité. Ceci est fait en exprimant la transformation de données avec *Entrés/Sorties* (I/O) etla transformation de l'Etat avec *pré-conditions/effets* (P/E).

  le `ServiceProfile` est généralement considéré comme un sous-ensemble du `ServiceModel`, contenant uniquement l'information nécessaire pour annoncer le service Web pour une découverte ultérieure.

- ServiceGrounding. Permet de spécifier les détails d'accès au service en précisant le protocole, le format des messages, la sérialisation et l'adressage. Il représente une correspondance *(mapping)* entre la définition abstraite d'un processus OWL-S décrivant le service et la définition WSDL concrète des éléments nécessaires pour interagir avec le service.

  Le rôle de mise en correspondance est principalement à de combler l'écart entre la description sémantique des services Web (détallée dans le deux premières sous-ontologies) et les modèles de description de service existants qui sont principalement syntaxique (WSDL).

Le `ServiceProfile` fournissent éventuellement d'autres informations supplémentaires sur le service comme la qualité qu'il assure en terme du temps de

réponse et du coût, une classification possible d'un service (`serviceCategory`), et un paramètre générique `serviceParameter`.

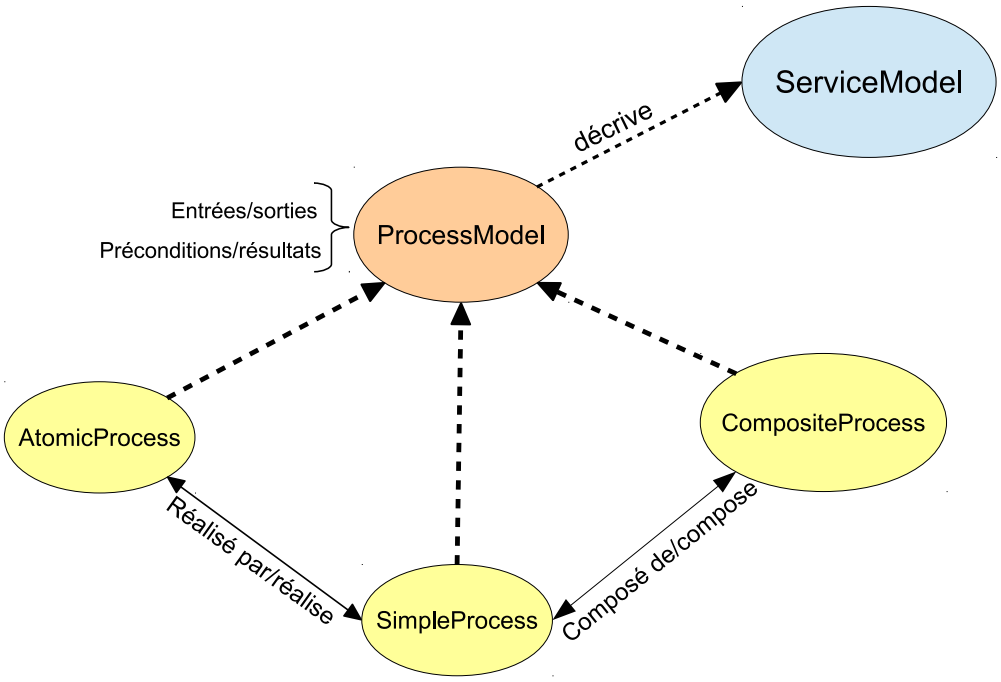


FIGURE 3.5 – Les éléments d'une ontologie ProcessModel

OWL-S définit une sous-classe de `ServiceModel` appelée `ProcessModel`. elle est utilisé pour décrire le service comme un processus. Le modèle de processus identifie trois types de processus illustrés dans la figure 3.5 :

- Le processus atomique : (`AtomicProcess`) directement invoqué par l'intermédiaire d'un `ServiceGrounding`
- Le processus composé : (`CompositeProcess`) décomposable en d'autres processus plus simples en utilisant les commandes de contrôle (par exemple : *if/else, sequences* etc.
- Le processus simple : (`SimpleProcess`) non-invoquable, il fournit simplement une vue d'un processus atomique ou une représentation simplifié d'un processus composé.

Dans [32], les auteurs discute les avantages de la description de service plus riche soutenue par OWL-S. il décrit comment OWL-S est utilisé dans le contexte d'autres standards, telles que WSDL, UDDI et BPEL.

3.2.5 Comparaison

Une composition des services Web nécessite la satisfaction de plusieurs exigences techniques [2, 46] , le tableau 3.1 montre une comparaison entre les langages et standards étudiées dans cette section selon les critères suivants :

Langages	BPLE	WS-CDL	WSFM	OWL-S
Composabilité	+	+	−	+
Representation du rôle	+	+	−	−
Support des structures complexes	+	+	−	+
Compensabilité	+	+	−	−
Support du sémantique	−	−	+	+
Support industriel	+	−	−	+

(+) support, (−) pas du support.

TABLE 3.1 – Comparaison des standards et langages de composition

- La composabilité indique la capacité d'assembler des services participants dans un processus de composition et de modéliser les interactions entre eux.
- La representation du rôle indique La représentation de rôle indique la capacité de refléter la le comportement que le participant doit présenter afin d'interagir dans le processus de composition.
- Le support des structures complexes la capacité de modéliser les structures complexes qui reflètent les règles des actions réalisées dans le processus de composition logique d'exécution et de commande.
- Compensabilité est la capacité de gérer les exceptions de processus lors de l'exécution du processus de composition
- le support du sémantique est la capacité de représenter la sémantique de services participants pour faciliter la découverte des services et la composition dynamique.
- le support industriel indiqué par la qualité des outils et le support industriel de la technologie.

D'après le tableau 3.1 Nous pouvons constater que tous les langages sauf WSMF supportent la modélisation des structures complexes. seulement BPEL et WS-CDL ont le support complet de la définition des rôles et la compensation. On peut aussi remarquer que seules le OWL-S et WSMF permettent de la composition sémantique des services Web.

3.3 Les approches de composition dynamiques des services web

La littérature quit traite la problématique de composition des services comporte une multitude d'approches visant à décrire l'interaction entre les services afin construire de nouveaux services composites répondant à un objectif donné. Selon l'approche proposée, les interprétations différent de ce que devraient être traitées dans une approche de composition, Ils diffèrent également sur le degré d'automatisation impliqué dans le processus allant des approches semi-automatisées à entièrement automatisés.

Dans cette section, nous allons décrire et classifier les approches principales de composition proposées par différents auteurs issues de plusieurs communautés de recherches. La classification présentée par la suite est basée essentiellement sur l'état de l'art fait par Baryannis *et al.*[18].Nous distinguons quatres catégories de composition dynamique des services Web :

- La composition basée sur les workflows : Ces approches éxploitent les résultats des recherches dans le domaine des *workflows*.
- La composition dirigée par les modèles : Ces approches utilisent les languages de modélisation (les réseux de Petri, UML) pour décrire la composition des services.
- La composition algébrique et mathématique des services web : Ces approches utilisent des méthodes mathématiques (logique mathématiques, algèbre).
- La composition basée sur les techniques de planification : Ces approches traitent le problème de composition comme un problème de planification.

3.3.1 La composition basée sur les workflows

Appuyant principalement du fait qu'un service composite est conceptuellement similaire à un *workflow*, il est possible d'exploiter les connaissances accumulées dans la communauté de *workflow* afin de faciliter la composition de services Web. Un *workflow* est un flux d'informations au sein d'une organisation, tel que la transmission de documents entre les personnes. Il modélise une séquence d'opérations, réalisées par différentes entités au sein de l'organisation. Pratiquement, un *workflow* est considéré comme la modélisation d'un ensemble de tâches, accomplies par différents acteurs impliqués dans la réalisation d'un processus métier. La modélisation de la composition de services sous forme d'un processus métier est de plus en plus populaire.

Les approches de composition des services Web basées sur les techniques de *workflow* étaient l'une des prmières solutions proposées pour la composition automatiques des services Web. Initialement, la plupart des travaux ont porté sur la composition statique et manuelle. Des travaux plus récents, cependant, a tenté de réaliser la composition dynamiques des services Web. En raison de la popularité de BPEL dans le milieu industriel, la plupart des approches dans cette catégorie emploient BPEL d'une manière ou d'une autre.

Majithia *et al.* [47] présentent une plateforme de construction automatique des schémas de composition.

Paws [48] est une plateforme de composition des services Web qui se concentre sur l'adaptabilité et la flexibilité d'une composition modélisé sous forme un processus métier.

fujii et *et al.* [49] [50] proposent une architecture.

En générale, nous pouvsons conclure que même si les apprcohes de composition à base de *workflow* ont évolué afin de supporter la composition automatique, les résultats sont limitées à des schémas de composition simples tels que l'éxécution séquentielle et parallèles. Cette lacune a été comblée en combinant méthodes à base de *workflow* avec des techniques de planification issues du domaine d'intélégence artificielle. Ces approches serons éxaminer dans la quatrième catégorie 3.3.4.

3.3.2 La composition dirigée par les modèles

L'ingénierie dirigée par les modèles MDO est une méthodologie de développement logiciel où l'accent est mis sur la création de modèles abstraits, indépendants de la plateforme et de la technologie. Un paradigme de modélisation doit fournir des modèles logiques du point de vue de l'utilisateur tout en restant suffisamment précis pour servir de base à l'implémentation [51]. Les approches dans cette catégorie utilisent des méthodes déjà explorées et des modèles bien établis pour représenter un système de composition afin de pâlir la complexité croissante de cette dernière utilisant une description plus abstraite au dessus de la description traditionnelle de services dans WSDL, OWL-S ou similaire.

UML [8] [52] est un langage de modélisation maintenu par l'OMG [9] qui s'est standardisé de par sa position dominante dans l'industrie du logiciel. UML est un langage polyvalent permettant de modéliser un système selon différents points de vue, Le point de vue statique ou structurel du système et le point de vue dynamique représente le comportement dynamique d'un système en montrant les interactions entre les objets ou les changements d'états au sein d'un objet.

Skogan *et al.* [53] proposent une méthode qui utilise les diagrammes d'activités UML pour modéliser la composition des services. Ces diagrammes là sont utilisés pour générer un processus exécutable d'orchestration BPEL utilisant des transformations XSLT, ce tarvail se limite à la description syntaxique de service Web utilisant les documents WSDL comme des entrées de la transformation UML. Dans [54] les auteurs essayent de soulèver cette limitation en considérant également l'enrichissement sémantique des services avec des ontologies OWLS et des annotations WSMO.

Les réseaux de Petri [55] constituent une approche bien établie pour la modélisation de processus. Un réseau de Petri est un graphe dirigé, connecté et biparti qui représente les transitions entre plusieurs états du système, ainsi que les ressources disponibles [51].

Hamadi et Benatallah dans [56] proposent une méthode de modélisation basée sur les réseaux de Petri pour modéliser le flux de contrôle et la sémantique des

8. `http://www.omg.org/spec/UML/`
9. `http://www.omg.org`

compositions de services en assignant une transition à chaque appel de service et une place à chaque état entre les appels, chaque service composite est modélisé par un réseau de Petri contenant un état d'entrée et de sortir permettant de modéliser respectivement la réception et l'émission d'informations par le service composite. Dans [57] les auteurs présentent une traduction complète de BPEL en réseaux de Petri permettant la vérification automatique des processus BPEL.

3.3.3 La composition algébrique et mathématique

Cette catégorie d'approches inclues toutes les approches ont la caractéristique commune à être fondée sur des bases mathématiques tels que le logique temporelle et linéaire, calcule formel, calcule algébrique des processus et autres méthodes mathématiques. De nombreux langages, présentés comme des algèbres de processus, ont été développés pour une compréhension formelle et la spécification d'applications à services SOA. les langages d'orchestration tels que XLANG et BPEL ont fortement inspirés de la métaphore de communication inspirée du π-calcule basé sur l'échange de messages dans un contexte distribué.

Les algèbres de processus sont des formalismes de description formelle pour la spécification de systèmes logiciels, en particulier pour les systèmes concurrentiels [51]. Elles fournissent des outils pour la description de haut niveau des interactions et des synchronisations entre les processus. Plusieurs algèbres de processus ont été décrites. Parmi les plus anciennes on peut citer CSP qui a été présenté par Hoare [58], CCS qui a été proposé par Milner [59, 60] et le π-calcule [61].

Milanovic *et al.* [62] montre que L'utilisation des langages algébriques de processus et langages formels, comme le CCS et le π-calculusa été préconisée pour la composition de services Web, car les spécifications de processus comprises dans les standards des services Web comme WSMO ou WSMF sont formellement fondées sur le algèbre des processus.

La composition algébrique permet modéliser les services comme des processus mobiles pour assurer une vérification de certaines propriétés tel que l'éxactitude, la sécurité, la vivacité *(liveness)*, et la gestion des ressources. La théorie des processus mobiles est basée sur le π-calcule, dans laquelle l'entité de base est un processus qui peut être un processus vide, un choix (branchement) entre plusieurs

opérations d'*Entrés/Sorties* et leurs continuations, une composition parallèle, une définition récursive, ou une invocation récursive [63].

Dans [64], la sémantique du langage de chorégraphie WS-CDL est décrite à l'aide de CSP pour permettre la vérification automatiques de propriétés. Un travail similaire est réalisé dans [65] où des règles sont présentées pour traduire chaque construction syntaxique de WS-CDL en CSP.

Rao *et al.* [66] proposent l'utilisation de la logique linéaire pour la composition automatique des services Web. La logique linéaire est une extension de la logique classique de premier ordre pour modéliser la notion d'évolution d'états. les auteurs proposent une méthode de traduction automatique de descriptions OWL-S à des axiomes de la logique linéaire. Ensuite, ils utilisent la démonstration automatique des théorèmes pour produire des schémas abstraits de composition, qui sont ensuite transformés à des modèles de *workflows* BPEL en utilisant un langage d'algèbre des processus inspiré par le π-calcule.

3.3.4 La composition basée sur les techniques de planification

Selon [18], [4], [67], [68], [69]

3.4 Conclusion

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

Chapitre 4

Les approches de composition dynamique des services Web sémantiques baseés sur le modèle graphe

4.1 Préliminaires

4.1.1 Définitions et terminologies de base

Le matching, l'alignement, le mapping...ect. Sont tous des termes utilisées pour référencer le concept de recherche de similarité. Voici quelques définitions de ces termes utilisés.

Definition 1 (Matching)**.** *c'est le processus de découverte des liaisons et des correspondances entre les entités de différentes représentations à travers un algorithme de matching.*

Dans le contexte d'une architecture à services, le *Matchmaking* (ou le *Matching*) est définie comme un processus qui nécessite un annuaire des services de prendre une requête en entrée, et de revenir tous les services qui peuvent satisfaire les exigences spécifiées dans la requête d'entrée [70]. Formellement, Le processus de *Matchmacking* peut se spécifié comme de suit :

Soit Φ est l'ensemble de toutes les services référencés dans un annuaire des services donné. Pour une requête $\mathtt{R}$.

Definition 2 (Alignement)**.** *on parle souvent de l'alignement des ontologies. C'est un ensemble de correspondances entre deux ou plusieurs ontologies. L'alignement est la sortie d'un processus de matching*

Definition 3 (Mapping)**.** *chercher des correspondances pour établir des transformations entre deux objets de même nature mais pas de même forme. Par conséquent, le mapping utilise les résultats du matching pour effectuer les transformations des objets.*

Comme on le voit, tous ces termes prennent bien en compte la notion de recherche de correspondance entre des concepts, dans certains contextes ils sont utilisés indifféremment. Pour les services web, on parlera du matchnig ou matchmaking pour exprimer la mise en correspondance entre deux entités, et du mapping pour exprimer la transformation ou conversion des types de données.

Nous distinguons deux techniques de matching : matching syntaxique et matching sémantique . Ils diffèrent dans la façon dont sont calculés les éléments correspondants et sur le type de relation de similitude $\mathtt{M}$ utilisée :

4.1.1.1 Matching syntactique

Utilisé pour l'extraction automatique de motifs textuels écrits en langage naturel en utilisant la méthode d'étiquetage, différentes techniques sont utilisées :

- Techniques basées string (à mots) : s méthodes prennent l'avantage de la structure des mots qui est considérée comme étant une séquence de lettres. Elles devraient, par exemple, reconnaitre une similarité entre Book et textBook. Certaines fonctions de calcul de distance convertissent une paire de chaine de caractères en un nombre réel. Si le nombre est petit (par rapport à un seuil) cela indique une grande similarité entre ces deux chaines de caractères (ex : distance de Hamming).

- Techniques basées langage : Ces méthodes considère les noms ou bien un terme comme une séquence de mots dans un langage naturel (peer-review, reviewed-paper). Ces méthodes utilisent les techniques de traitement du langage naturel pour reconnaitre les similarités entre les mots d'un terme par exemple : les techniques de tokenization (segmentation), lemmatisation...etc. Techniques utilisant des ressources linguistiques : comme les thesaurus sont utilisées pour faire correspondre des mots en se basant sur les relations linguistiques entre eux (synonymes, hyponyme). Par exemple dans le cas du matching entre les services web on fait correspondre les noms d'opérations et les noms des paramètres qui sont considérés comme des mots de langage naturel.

4.1.1.2 Matching sémanique

Consiste à chercher des correspondances sémantiques à partir des concepts, et non pas des étiquettes. Il ne suffit pas d'examiner la syntaxe des mots, mais leurs significations ou leurs interprétations dans le domaine d'application. Les ontologies de domaine et les techniques basées modèle sont utilisées dans ce type de matching.

- Ontologie de domaine spécifique : spécifique peut être utilisée comme source externe. Elles se focalisent sur un domaine particulier et utilisent des termes qui sont propres à ce domaine et qui ne sont pas liés aux concepts

similaires dans d'autres domaines. Par exemple dans le domaine de transport, on trouve : bus, avion, taxi...etc. On peut trouver des termes qui sont syntaxiquement identiques mais possédant des interprétations différentes dans des domaines différents.

- Techniques basées modèle : Appelées aussi les méthodes de déduction, les algorithmes manipulent les entrées basées sur leurs interprétations sémantiques. Si deux entités sont similaires alors ils partagent les mêmes interprétations. Parmi ces méthodes on trouve les techniques de satisfiabilité propositionnelle [Giu, 2003A] Dans le cas du matching des services web, ces derniers sont décrits à l'aide de supports

Dans le cas du matching des services web, ces derniers sont décrits à l'aide de supports de description sémantique (OWL-S par exemple). Ces descriptions contiennent des informations clés pour leur mise en correspondance (les paramètres fonctionnels et non fonctionnels).

4.1.2 Mésures de similarité

objectif des mesures de similarité sémantique est d'évaluer la proximité sémantique entre les concepts (auxquels les termes des requêtes et documents sont rattachés). Et comme le matching des services web revient à trouver les similarités entre les concepts décrivant les propriétés des services web, nous avons jugé utile d'explorer les différentes méthodes de recherche de similarité utilisées dans le domaine de la recherche d'information, en particulier.

Definition 4 (Similarité entre concepts)**.** *Une similarité $\sigma : C \times C \rightarrow [0,1]$ est une fonction de couples de concepts vers un nombre compris entre 0 et 1 exprimant le degré de similarité entre deux concepts, tel que :*

- $x \in C$, $\sigma(x,x) = 1$
- $x, y \in C$, $\sigma(x,y) = \sigma(y,x)$

Les ontologies définissent les propriétés des concepts et les relations entre concepts. Une de ces relations est utilisée dans notre application est IS–A (est-un). Cette relation est utilisée pour extraire des taxonomies à partir des ontologies

Definition 5 (Similarité entre concepts)**.** *Soit C l'ensemble de concepts définis dans une ontologie. On dit que T est une taxonomie et on écrit T(C, ≤) tel que c ≤ c' signifie que c is–a c' (le concept c est subsumé par le concept).*

De nombreuses approches ont été proposées pour évaluer la similarité sémantique entre deux concepts. Ces approches se divisent en trois catégories [Sli, 2006] : les approches basées sur les arcs, les approches basées sur le contenu informationnel et les approches hybrides.

4.2 Matching des services Web sémantiques

Le processus de matchmaking repose sur la recherche de similarités entre les paramètres descriptives de ces derniers. Il existe deux catégories de paramètres de description des services web : les paramètres fonctionnels et les paramètres non-fonctionnels que nous décrivons tout de suite.

4.2.1 Les paramètres fonctionnels

La sémantique fonctionnelle décrit les fonctionnalités qu'offre le service en terme de transformation d'information dénotée par les Entrés/Sorties (I/O) et de changement d'état après l'exécution du service dénoté par les pré-conditions/effets (ou post-conditions) (P/E). Ils sont appelés ainsi parce qu'ils sont nécessaires pour le fonctionnement du service web. Ces paramètres sont utiles dans la recherche des similarités. En effet, grâce au formalisme de description des services web (OWL-S), on trouve une description des données requises pour l'exécution du service (Inputs) et une description des résultats obtenus après exécution (Outputs). D'un autre coté, un service web peut altérer l'état du monde après son exécution. L'état du monde requis pour que le service s'exécute est la pré-condition et le nouvel état généré après son exécution est l'effet du service sur le monde. Par exemple le service de logging à un site web possède comme information d'entrées le username et le password et l'information de sortie est un message de confirmation. Après l'exécution, l'état du monde change de not_logged_in à logged_in. Prenons un autre exemple, un transfert d'argent du compte A vers le compte B ne peut se faire que si le compte A est solvable (les pré-conditions). Les effets décrivent les sorties

(Output) et l'impact d'une exécution d'un service web. Tant dis que le terme post-condition focalise souvent sur la description des conditions concernant les valeurs retournées des opérations du service web. Dans notre exemple ci-dessus, un impact de l'exécution serait le transfert réel de l'argent du compte de carte de crédit pour le compte de destination alors que la post-condition décrit qu'aucun argent ne s'est perdu lors du transfert [Car, 2007]. L'opération de matching consiste alors à relier les paramètres des différents services (faire correspondre) entre eux et cela suivant des règles de correspondance.

4.2.2 Les paramètres non-fonctionnels

Les paramètres non fonctionnels sont des paramètres qui ne sont pas reliés directement aux fonctionnalités d'un service web, mais mesure la qualité avec laquelle le service délivre ses fonctionnalités. Parmi ces paramètres on trouve :

- Le coût : C'est une propriété non-fonctionnelle qui est pertinente pour le client qui veut utiliser le service.
- La sécurité : La sécurité est une propriété non-fonctionnelle qui est pertinente à la plupart des services web. Elle applique des aspects tels que la communication et le cryptage des données.
- La qualité : La propriété qualité des services est un ensemble de propriétés différentes qui affecte tous les aspects spécifiques du service, son utilisation et sa production tel que le temps de réponse d'un appel à une opération, la capacité du service...etc

Le matching des paramètres non-fonctionnels des services web est tout aussi important dans la phase de sélections des services candidats. Les propriétés non-fonctionnelles sont typiquement représentés comme des politiques de services web exprimées par le langage WS- Policy [Car, 2007].

Généralement ces approches de matching des politiques peuvent être utilisées pour l'optimisation et l'évaluation des plans de composition. Les services web correspondent à des entités dynamiques. Ils peuvent devenir disponibles à tout moment de l'exécution du système et découverts dynamiquement par leurs clients. Les fournisseurs et les consommateurs de services doivent donc disposer

d'un moyen commun et fiable pour effectuer la publication et la recherche de services. On entend par découverte dynamique la possibilité de localiser automatiquement un ensemble de services web, sélectionner un service spécifique qui répond à des besoins particuliers de l'utilisateur. Plusieurs initiatives ont traité la problématique de la découverte des services web dont les deux principales modèles suivants :

4.2.3 Le modèle syntaxique de matching des services Web

4.2.4 Le modèle sémantique de matching des services Web

4.2.5 Graphe de dépendance

[31]

4.3 Travaux connexes

4.3.1 Génération Online du graphe de dépendance

4.3.2 Génération Offline du graphe de dépendance

4.3.3 Autres travaux basés sur le graphe matching

4.4 Vers les bases de données graphe

4.5 Conclusion

Deuxième partie

L'approche proposée

Chapitre 5

Une méthode de composition dynamique des services Web sémantiques utilisant neo4j

5.1 Reformulation mathématique du problème

5.1.1 Définitions de base

5.1.2 Hypothèses du travail

5.1.3 Objectifs

5.2 Example d'illustration

5.3 Annotation sémantiques des services web

5.4 Architecture du système

5.5 Conclusion

Chapitre 6

L'Implémentation d'un prototype

6.1 Example d'illustration

6.2 Outil d'annotaion sémantiques des services

6.3 Publication des services web atomiques

6.4 Découvert d'un service web composites

6.5 Conclusion

Bibliographie

[1] Dieter Fensel et Christoph Bussler : Semantic web enabled web services. *Advances in Artificial Intelligence*, page 316, 2002.

[2] Quan Z Sheng, Xiaoqiang Qiao, Athanasios V Vasilakos, Claudia Szabo, Scott Bourne et Xiaofei Xu : Web services composition : A decade's overview. *Information Sciences*, 280:218–238, 2014.

[3] 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.

[4] Peter Bartalos : Effective automatic dynamic semantic web service composition. *Inf. Sci. and Technol. Bulletin ACM Slovakia*, 3(1), 2011.

[5] Heather Kreger *et al.* : Web services conceptual architecture (wsca 1.0). *IBM Software Group*, 5:6–7, 2001.

[6] W3C Working Group : Web services architecture. `http://www.w3.org/TR/ws-arch/`, 2004.

[7] Paul Fremantle, Sanjiva Weerawarana et Rania Khalaf : Enterprise services. *Communications of the ACM*, 45(10):77–82, 2002.

[8] Don Box, David Ehnebuske, Gopal Kakivaya, Andrew Layman, Noah Mendelsohn, Henrik Frystyk Nielsen, Satish Thatte et Dave Winer : Simple object access protocol (soap) 1.1, 2000.

[9] 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.

[10] 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.

[11] Francisco Curbera, William Nagy et Sanjiva Weerawarana : Web services : Why and how. *In OOPSLA 2001 Workshop on Object-Oriented Web Services*, 2001.

[12] 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.

[13] Karl Gottschalk, Stephen Graham, Heather Kreger et James Snell : Introduction to web services architecture. *IBM Systems journal*, 41(2):170–177, 2002.

[14] Nilo Mitra, Yves Lafon *et al.* : Soap version 1.2 part 0 : Primer. *W3C recommendation*, 24:12, 2003.

[15] 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.

[16] 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.

[17] 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.

[18] George Baryannis et Dimitris Plexousakis : Automated web service composition : State of the art and research challenges. *ICS-FORTH, Tech. Rep*, 409, october 2010.

[19] 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.

[20] XML Schema Part : 2 : Datatypes. *W3C Recommendation*, 2, 2001.

[21] Tim Berners-Lee, James Hendler, Ora Lassila *et al.* : The semantic web. *Scientific american*, 284(5):28–37, 2001.

[22] Alexandre Bertails, Ivan Herman et Sandro Hawke : Le web sémantique. *Réalités industrielles*, (4):84–89, 2010.

[23] Sheila A McIlraith, Tran Cao Son et Honglei Zeng : Semantic web services. *IEEE intelligent systems*, 16(2):46–53, 2001.

[24] 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.

[25] Ian Horrocks *et al.* : Daml+oil : a description logic for the semantic web. *IEEE Data Eng. Bull.*, 25(1):4–9, 2002.

[26] 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.

[27] 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.

[28] Thomas R Gruber : A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2):199–220, 1993.

[29] Rama Akkiraju, Joel Farrell, John A Miller, Meenakshi Nagarajan, Amit Sheth et Kunal Verma : Web service semantics-wsdl-s. 2005.

[30] 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.

[31] Abrehet Mohammed Omer : *A framework for Automatic Web Service Composition based on service dependency analysis*. Thèse de doctorat, Technical University of Dresden, 2011.

[32] David Martin, Mark Burstein, Jerry Hobbs, Ora Lassila, Drew McDermott, Sheila McIlraith, Srini Narayanan, Massimo Paolucci, Bijan Parsia, Terry Payne *et al.* : Owl-s : Semantic markup for web services. *W3C member submission*, 22:2007–04, 2004.

[33] Schahram Dustdar et Wolfgang Schreiner : A survey on web services composition. *International journal of web and grid services*, 1(1):1–30, 2005.

[34] Brahim Medjahed : *Semantic Web Enabled Composition of Web Services*. Thèse de doctorat, Virginia Polytechnic Institute and State University, 2004.

[35] Maha Driss : *Approche multi-perspective centrée exigences de composition de services Web*. Thèse de doctorat, Université Rennes 1, 2011.

[36] Chris Peltz : Web services orchestration and choreography. *Computer*, 36(10):46–52, 2003.

[37] Stéphanie Chollet : *Orchestration de services hétérogènes et sécurisés*. Thèse de doctorat, Université Joseph-Fourier-Grenoble I, 2009.

[38] Sonia Jamal *et al.* : *Environnement de procédé extensible pour l'orchestration-Application aux services web*. Thèse de doctorat, Université Joseph-Fourier-Grenoble I, 2005.

[39] 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.

[40] Nickolas Kavantzas, David Burdett, Gregory Ritzinger, Tony Fletcher, Yves Lafon et Charlton Barreto : Web services choreography description language version 1.0. *W3C candidate recommendation*, 9, 2005.

[41] Assaf Arkin, Sid Askary, Scott Fordin, Wolfgang Jekeli, Kohsuke Kawaguchi, David Orchard, Stefano Pogliani, Karsten Riemer, Susan Struble, Pal Takacsi-Nagy *et al.* : Web service choreography interface (wsci) 1.0. *Standards proposal by BEA Systems, Intalio, SAP, and Sun Microsystems*, 2002.

[42] Dieter Fensel et Christoph Bussler : The web service modeling framework wsmf. *Electronic Commerce Research and Applications*, 1(2):113–137, 2002.

[43] Anupriya Ankolekar, Mark Burstein, Jerry R Hobbs, Ora Lassila, David Martin, Drew McDermott, Sheila A McIlraith, Srini 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.

[44] Deborah L McGuinness, Frank Van Harmelen *et al.* : Owl web ontology language overview. *W3C recommendation*, 10(2004-03):10, 2004.

[45] Sheila A McIlraith et David L Martin : Bringing semantics to web services. *Intelligent Systems, IEEE*, 18(1):90–93, 2003.

[46] Antonio Bucchiarone et Stefania Gnesi : A survey on services composition languages and models. *In International Workshop on Web Services–Modeling and Testing (WS-MaTe 2006)*, page 51, 2006.

[47] Shalil Majithia, David W Walker et WA Gray : A framework for automated service composition in service-oriented architectures. *In The Semantic Web : Research and Applications*, pages 269–283. Springer, 2004.

[48] Danilo Ardagna, Marco Comuzzi, Enrico Mussi, Barbara Pernici et Pierluigi Plebani : Paws : A framework for executing adaptive web-service processes. *IEEE software*, 24(6):39–46, 2007.

[49] Keita Fujii et Tatsuya Suda : Semantics-based dynamic web service composition. *International Journal of Cooperative Information Systems*, 15(03):293–324, 2006.

[50] Keita Fujii et Tatsuya Suda : Semantics-based context-aware dynamic service composition. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 4(2):12, 2009.

[51] C Dumez : *Approche dirigée par les modéles pour la spécification, la vérification formelle et la mise en oeuvre de services Web composés, Université de Technologie de Belfort-Montbéliard*. Thèse de doctorat, thése de doctorat, 2010.

[52] James Rumbaugh, Ivar Jacobson et Grady Booch : *Unified Modeling Language Reference Manual, The*. Pearson Higher Education, 2004.

[53] David Skogan, Roy Grønmo et Ida Solheim : Web service composition in uml. *In Enterprise Distributed Object Computing Conference, 2004. EDOC 2004. Proceedings. Eighth IEEE International*, pages 47–57. IEEE, 2004.

[54] Roy Grønmo et Michael C Jaeger : Model-driven semantic web service composition. *In Software Engineering Conference, 2005. APSEC'05. 12th Asia-Pacific*, pages 8–pp. IEEE, 2005.

[55] Carl Adam Petri : Kommunikation mit automaten. 1962.

[56] Rachid Hamadi et Boualem Benatallah : A petri net-based model for web service composition. *In Proceedings of the 14th Australasian database conference-Volume 17*, pages 191–200. Australian Computer Society, Inc., 2003.

[57] Chun Ouyang, Eric Verbeek, Wil MP Van Der Aalst, Stephan Breutel, Marlon Dumas et Arthur HM Ter Hofstede : Formal semantics and analysis of control flow in ws-bpel. *Science of Computer Programming*, 67(2):162–198, 2007.

[58] Charles Antony Richard Hoare : Communicating sequential processes. *Communications of the ACM*, 21(8):666–677, 1978.

[59] Robin Milner : *A Finite Delay Operator in Synchronous CCS.* University of Edinburgh Department of Computer Science, 1982.

[60] Robin Milner : *Communication and concurrency*. Prentice-Hall, Inc., 1989.

[61] Robin Milner, Joachim Parrow et David Walker : A calculus of mobile processes, ii. *Information and computation*, 100(1):41–77, 1992.

[62] Nikola Milanovic et Miroslaw Malek : Current solutions for web service composition. *IEEE Internet Computing*, 8(6):51–59, 2004.

[63] CHOUIREF Zahira : *Un système de programmation fonctionnelle pour la composition de services Web.* Thèse de doctorat, Université M'hamed BOUGARA de BOUMERDES, 2008.

[64] Mariya Koshkina et Franck van Breugel : Modelling and verifying web service orchestration by means of the concurrency workbench. *ACM SIGSOFT Software Engineering Notes*, 29(5):1–10, 2004.

[65] Jing Li, Jifeng He, Huibiao Zhu et Geguang Pu : Modeling and verifying web services choreography using process algebra. *In Software Engineering Workshop, 2007. SEW 2007. 31st IEEE*, pages 256–268. IEEE, 2007.

[66] Jinghai Rao, Peep Kungas et Mihhail Matskin : Logic-based web services composition : from service description to process model. *In Web Services, 2004. Proceedings. IEEE International Conference on*, pages 446–453. IEEE, 2004.

[67] KS May Chan, Judith Bishop et Luciano Baresi : Survey and comparison of planning techniques for web services composition. *University of Pretoria Pretoria*, 2007.

[68] Joachim Peer : Web service composition as ai planning-a survey. *University of St. Gallen*, 2005.

[69] Pablo Rodriguez-Mier, Manuel Mucientes et Manuel Lama : Automatic web service composition with a heuristic-based search algorithm. *In Web Services (ICWS), 2011 IEEE International Conference on*, pages 81–88. IEEE, 2011.

[70] Lei Li et Ian Horrocks : A software framework for matchmaking based on semantic web technology. *International Journal of Electronic Commerce*, 8(4), 2004.