# Description De Services : Langage WSDL

## I. INTRODUCTION :

### 1) Définition:

Le protocole SOAP met à la disposition des services web un moyen standard de structuration et d'échange de messages XML. Il ne fournit aucune indication sur la structure que le message doit respecter vis à vis du service web sollicité. C'est toujours dans le but de rendre les services web faiblement couplés et autonomes, que la spécification WSDL a vu le jour. Contrairement aux architectures monolithiques où la description des composants ainsi que les moyens de les invoquer dépendent fortement de l'infrastructure utilisée, la spécification WSDL offre une grammaire qui décrit l'interface des composants services web de telle façon qu'ils se suffisent à eux-mêmes.

WSDL est un langage de description des capacités de services web basé sur XML. Un document WSDL décrit essentiellement le nom de la méthode utilisé, son nombre de paramètres, et leur type, ce qu'un service web offre, où il réside et comment on peut l'invoquer (1].

### 2) Concepts du langage WSDL (*Web Service Description Language*) [1][2] :

Les premiers travaux du W3C concernant un langage de description universelle de services web ont vu le jour en 2001 lors de l'émergence de cette technologie. Il a fallu trouver un langage de description utilisable et compréhensible par le plus grand nombre afin que les services web conçus soient interopérables. Le W3C a rapidement proposé le langage WSDL.

La description d'un service doit inclure la définition des composants nécessaire au protocole de communication (SOAP pour les services web) et à l'interaction avec un client ou un autre service web. Les problématiques de réutilisation et d'interaction ont guidé le W3C afin de définir les catégories d'information à prendre en compte dans la description d'un service web. Les différents éléments décrits dans WSDL sont les suivants :

- Les **opérations** proposées par le service web.
- Les **données** et **messages** échangés lors de l'appel d'une opération.
- Le **protocole de communication.**
- Les **ports d'accès** au service.

Dans WSDL, il existe une séparation entre deux niveaux indépendants, respectivement nommés abstrait et concret. Le niveau abstrait regroupe les informations pouvant être réutilisées (non spécifique à un service), tandis que le niveau concret est constitué de la description des protocoles d'accès au service web (information particulière à un service). Le niveau abstrait est utilisé principalement lors du processus de sélection, tandis que le niveau concret est seulement utilisé lors de l'invocation des méthodes du service web.

#### 1. Niveau abstrait:

Il décrit les informations propres aux méthodes proposées par le service, ainsi que les informations traitant des messages et des données échangés lors de l'invocation du service. Si deux services proposent les mêmes méthodes, le niveau abstrait de description WSDL peut être réutilisé. Ce niveau est composé des informations suivantes :

a) **Les types de données (*types*):** Le document WSDL permet de décrire les types de données échangées. WSDL supporte les types élémentaires (tels que les entiers, les chaînes de caractères) et les types complexes. Si les données échangées possèdent une structure

particulière (*i.e.* un type complexe), il est possible de les décrire par l'intermédiaire d'un schéma XML.

b) **Les messages (*message*):** Un message correspond aux données qui sont véhiculées selon les méthodes invoquées. Chaque opération du service possède deux définitions de message : la première correspond à la requête et la seconde correspond à la réponse. La description d'un message contient le nom de l'élément en paramètre – d'entrée ou de sortie selon le type du message – et son type.

c) **Les opérations (*portType*)**: Une opération représente une unité de travail, c'est-à-dire une méthode proposée par le service web décrit. Chaque opération est identifiée par son nom.

### 2. Niveau concret:

Il décrit la manière dont le client accède à un service en particulier, et, est de ce fait, non réutilisable (propre à un service unique). Les informations décrites dans le niveau concret sont les suivantes :

a) **Le protocole de communication (*binding*)**: La description des protocoles de communication permet de définir le protocole à utiliser pour l'appel des méthodes du service. Si nécessaire, le document WSDL peut contenir autant de descriptions de protocole que d'opérations, étant donné que le protocole de communication peut être différent pour chaque opération du service décrit.

b) **Les ports d'accès au service (*service*)**: Dans un document WSDL, l'accès au service est défini par une collection de ports d'accès. Chaque port représente la localisation du service (i.e. son URL). Un même service peut être accessible sur plusieurs ports différents.

Les informations contenues dans WSDL constituent la description du profil fonctionnel du service. Le client peut invoquer le service par le biais de sa description abstraite (méthodes disponibles, paramètres d'entrée et sortie) et concrète (description des protocoles de communication et des points d'accès du service).

## II. Structure d'un document WSDL:

La présentation de la structure d'un document WSDL est illustrée par des exemples de la description d'un service web nommé ***GlobalWeather*** proposé par le fournisseur de service web ***WebserviceX.net***. Selon la ville et le pays, ce service web renvoie un ensemble d'informations concernant la météorologie (vitesse du vent, visibilité, condition météorologique, température, etc.) [4].

### 1. Document WSDL 1.1 [3][4]:

Un document WSDL est un document XML composé d'un élément racine (***definitions***) et de cinq sous éléments obligatoires (***types, message, portType, binding, service***). La figure 1.1 illustre la structure générale d'un document WSDL 1.1 et établit le lien entre les éléments XML et les catégories d'information définies précédemment.

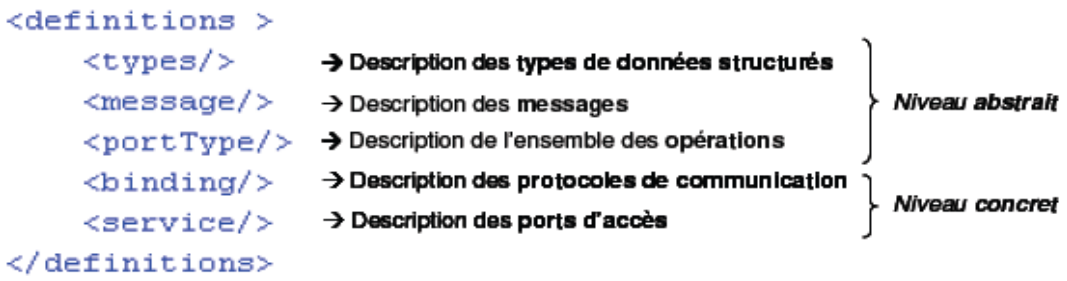


**Figure 1.1: Structure générale d'un document WSDL 1.1**

La suite de l'étude de WSDL 1.1 est composée de la définition de chaque élément principal composant un document WSDL (les éléments *definitions*, *types*, *message*, *portType*, *binding* et *service*), et de leurs illustrations par des exemples reposant sur la description WSDL 1.1 du service web *GlobalWeather*.

## 1. L'élément « *definitions* »:

L'élément *definitions* est la racine du document WSDL. Les espaces de noms permettent de connaître la version de SOAP (ligne 3), les définitions de schémas XML (ligne 4), et d'autres espaces de noms à utiliser lors de l'appel du service web décrit (lignes 2 et 5 à 8) dans la figure 1.2. Le fournisseur du service web décrit est identifié par l'espace de noms le localisant (l'attribut *targetNamespace*, ligne 9). L'espace de noms *WSDL* (ligne 10) est, par défaut, l'espace de nom de tout élément ne possédant pas d'espace de noms dans un document WSDL.

```
1  <wsdl:definitions
2      xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
3      xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
4      xmlns:s="http://www.w3.org/2001/XMLSchema"
5      xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
6      xmlns:tns="http://www.webserviceX.net"
7      xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
8      xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
9      targetNamespace="http://www.webserviceX.NET"
10     xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
11     ...
12 </wsdl:definitions>
```

**Figure 1.2: Utilisation de l'élément *"definitions"* dans la description du service web *GlobalWeather***

## 2. L'élément « *types* » :

L'élément *types* définit les structures de données contenues dans les messages échangés lors de l'appel du service web décrit. Dans l'exemple suivant nous pouvons voir que les messages échangent quatre types structurés (*GetWeather, GetWeatherResponse*, *GetCitiesByCountry*, *GetCitiesByCountryResponse* – respectivement lignes 3, 11, 12 et 13) et un type simple nommé string de type string (ligne 14). D'après l'extrait du document WSDL, nous pouvons savoir que le type complexe *GetWeather* est composé de deux éléments (nommés respectivement *CityName* et *CountryName*, lignes 6 et 7), tous deux de type string. Ce sont les types de paramètres d'entrée d'une des méthodes du service web.

```
1  <wsdl:types>
2    <s:schema elementFormDefault="qualified" targetNamespace="http://www.webserviceX.NET">
3      <s:element name="GetWeather">
4        <s:complexType>
5          <s:sequence>
6            <s:element minOccurs="0" maxOccurs="1" name="CityName" type="s:string"/>
7            <s:element minOccurs="0" maxOccurs="1" name="CountryName" type="s:string"/>
8          </s:sequence>
9        </s:complexType>
10     </s:element>
11     <s:element name="GetWeatherResponse">...</s:element>
12     <s:element name="GetCitiesByCountry">...</s:element>
13     <s:element name="GetCitiesByCountryResponse">...</s:element>
14     <s:element name="string" nillable="true" type="s:string"/>
15   </s:schema>
16 </wsdl:types>
```

**Figure 1.3: Utilisation de l'élément *"types"* dans la description du service web *GlobalWeather***

## 3. L'élément « *message* »:

L'élément *message* introduit les types de messages supportés par le service et décrit les données transmises lors des appels des méthodes du service web. Dans la figure1.4 nous pouvons voir que dans le service web *GlobalWeather*, quatre messages sont échangés entre le client et le service lui-même (*GetWeatherSoapIn*, *GetWeatherSoapOut*, *GetCitiesByCountrySoapIn, GetCitiesByCountrySoapOut* – respectivement lignes 1, 4, 5 et 6). Le message *GetWeatherSoapIn* a pour paramètre l'élément *GetWeather* défini dans l'élément types.

```
1  <wsdl:message name="GetWeatherSoapIn">
2    <wsdl:part name="parameters" element="tns:GetWeather"/>
3  </wsdl:message>
4  <wsdl:message name="GetWeatherSoapOut">...</wsdl:message>
5  <wsdl:message name="GetCitiesByCountrySoapIn">...</wsdl:message>
6  <wsdl:message name="GetCitiesByCountrySoapOut">...</wsdl:message>
```

**Figure 1.4: Utilisation de l'élément *"message"* dans la description du service web *GlobalWeather***

## 4. L'élément « *portType* »:

Cet élément décrit l'ensemble des opérations proposées par le service web. La description WSDL d'un service web prévoit un sous-élément *operation* pour chaque opération supportée par le service. La figure 1.5 illustre la description des opérations proposées par le service web *GlobalWeather*, pour le *portType GlobalWeatherSoap*, nommées respectivement *GetWeather* (ligne 2) et *GetCitiesByCountry* (ligne 6). L'opération *GetWeather* reçoit le message *GetWeatherSoapIn* lors de son appel (ligne 3) et renvoie le message *GetWeatherSoapOut* lors de sa réponse (ligne 4).

```
1  <wsdl:portType name="GlobalWeatherSoap">
2    <wsdl:operation name="GetWeather">
3      <wsdl:input message="tns:GetWeatherSoapIn"/>
4      <wsdl:output message="tns:GetWeatherSoapOut"/>
5    </wsdl:operation>
6    <wsdl:operation name="GetCitiesByCountry">...</wsdl:operation>
```

**Figure 1.5: Utilisation des éléments *"portType"* et *"operation"* dans la description du service web *GlobalWeather***

### 5. L'élément « *binding* »:

L'élément *binding* définit les protocoles de communication et les spécifications des formats de données pour les ensembles d'opérations (ou *portType*). La figure 1.6 illustre le fait que, pour l'opération *GetWeather* du *portType GlobalWeatherSoap*, le protocole de communication à utiliser est SOAP (ligne 2).

```
1  <wsdl:binding name="GlobalWeatherSoap" type="tns:GlobalWeatherSoap">
2    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
3    <wsdl:operation name="GetWeather">
4      <soap:operation soapAction="http://www.webserviceX.neti/GetWeather" style="document"/>
5      <wsdl:input>...</wsdl:input>
6      <wsdl:output>...</wsdl:output>
7    </wsdl:operation>
8    <wsdl:operation name="GetCitiesByCountry">...</wsdl:operation>
9  </wsdl:binding>
```

**Figure 1.6: Utilisation de l'élément *"binding"* dans la description du service web *GlobalWeather***

### 6. L'élément « *service* »:

Cet élément décrit la collection des ports d'accès du service. Le but de cet élément est de pouvoir localiser le service web proprement dit, et ainsi pouvoir faire appel aux méthodes disponibles.

Dans l'exemple suivant du service web *GlobalWeather,* le service est disponible à l'URL suivant : *"http://www.webservicex.com/globalweather.asmx"* (ligne 3).

```
1  <wsdl:service name="GlobalWeather">
2    <wsdl:port name="GlobalWeatherSoap" binding="tns:GlobalWeatherSoap">
3      <soap:address location="http://www.webservicex.com/globalweather.asmx"/>
4    </wsdl:port>
5  </wsdl:service>
```

**Figure 1.6: Utilisation de l'élément *"service"* dans la description du service web *GlobalWeather***

## 2. Document WSDL 2.0 [2][4]:

En 2002, le consortium W3C inscrit une nouvelle activité de recherche dans ses préoccupations : l'activité sur les services web (*Web Services Activity*). De cette activité découlent quatre groupes de travail dont le groupe de travail sur la description de services web (*Web Services Description Working Group*). Ce groupe a été créé principalement en vue de standardiser WSDL qui n'était qu'une note en 2002, date de fondation de ce groupe de travail. A la suite de la standardisation du WSDL 2.0 en 2007, les études du groupe de travail sur la description de services web ont pris fin, l'objectif final du groupe était atteint.

À l'instar de WSDL 1.1, WSDL 2.0 repose sur le langage semi-structuré XML. Sa structure générale est composée d'un élément racine (*description*) et de quatre sous-éléments obligatoires (*types, interface, binding* et *service*). La figure suivante illustre la structure générale d'un document WSDL 2.0 et établit le lien entre les éléments XML et les catégories d'information définies dans la 1ère section.

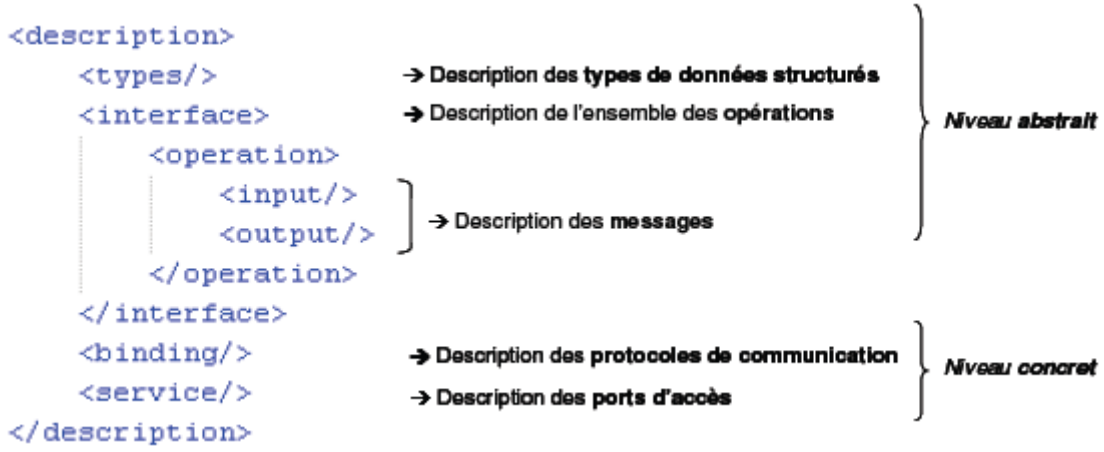


**Figure 2.1: Structure générale d'un document WSDL 2.0**

Dans la suite de cette section, nous illustrons l'utilisation de chaque élément XML composant un document WSDL 2.0 (*description, types, interface, binding* et *service*) avec l'exemple du service web *GlobalWeather.* De plus, nous établissons à la fin les liens syntaxiques entre les versions 1.1 et 2.0 de WSDL.

## 1. L'élément « *description* »:

Cet élément, élément racine d'un document WSDL 2.0, à l'instar de l'élément *definitions* de la version 1.1, est utilisé afin de déclarer les espaces de nom utilisés tout au long du document. L'espace de nom *xmlns="http://www.w3.org/ns/wsdl"* (ligne 2) est l'espace de nom utilisé par défaut.

```
1  <description
2      xmlns="http://www.w3.org/ns/wsdl"
3      xmlns:tns="http://www.webserviceX.NET"
4      targetNamespace="http://www.webserviceX.NET">
5
6  </description>
```

**Figure 2.2: Utilisation de l'élément *"description"* dans la description du service web *GlobalWeather***

## 2. L'élément « *types* »:

Cet élément décrit les types de messages que le service envoie et reçoit lors de l'appel d'une des méthodes. Cet élément est analogue à l'élément homonyme utilisé dans la version 1.1, à la différence que des espaces de noms nécessaires à la définition de la structure de données sont inclus à ce niveau du document (tel que l'espace de nom *xmlns:s="http://schemas.w3.org/2001/XMLSchema"* – ligne 4). La figure 2.3 montre que la définition de l'élément structuré *GetWeather* (lignes 11 à 18) est identique à celle utilisée dans la version 1.1.

```
1  <types>
2    <s:schema xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
3              xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
4              xmlns:s="http://www.w3.org/2001/XMLSchema"
5              xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
6              xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
7              xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
8              xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
9              elementFormDefault="qualified"
10             targetNamespace="http://www.webserviceX.NET">
11       <s:element name="GetWeather">
12          <s:complexType>
13             <s:sequence>
14                <s:element minOccurs="0" maxOccurs="1" name="CityName" type="s:string"/>
15                <s:element minOccurs="0" maxOccurs="1" name="CountryName" type="s:string"/>
16             </s:sequence>
17          </s:complexType>
18       </s:element>
19       <s:element name="GetWeatherResponse">...</s:element>
20       <s:element name="GetCitiesByCountry">...</s:element>
21       <s:element name="GetCitiesByCountryResponse">...</s:element>
22       <s:element name="string" nillable="true" type="s:string"/>
23    </s:schema>
24 </types>
```

**Figure 2.3: Utilisation de l'élément *"types"* dans la description du service web *GlobalWeather***

## 3. L'élément « *interface* »:

Cet élément décrit l'ensemble des fonctionnalités, appelées opérations, fournies par le service web. Une opération (sous-élément *operation*) représente une simple interaction entre le client et le service. Dans la version 2.0 de WSDL les messages sont définis au niveau de l'opération et sont de deux types : soit des messages que le service reçoit (sous-élément *input*), soit des messages que le service envoie au client (sous-élément *output*). De plus, l'élément *operation* possède un attribut *pattern* qui définit la séquence selon laquelle les messages sont transmis.

Dans l'exemple du service web *GlobalWeather* (figure 2.4), l'interface *GlobalWeatherSoap* contient deux opérations (*GetWeather* et *GetCitiesCountry*). L'opération *GetWeather* est composée de deux messages échangés entre le service et le client dont les paramètres sont *GetWeather* et *GetWeatherResponse* (leur structure de données est définie dans l'élément type dans la figure 2.3 lignes 11 à 18 pour la structure de données de l'opération *GetWeather* ).

```
1  <interface name="GlobalWeatherSoap">
2     <operation name="GetWeather"
3        pattern="http://www.w3.org/ns/wsdl/in-out">
4           <input element="tns:GetWeather"/>
5           <output element="tns:GetWeatherResponse"/>
6     </operation>
7     <operation name="GetCitiesByCountry"
8        pattern="http://www.w3.org/ns/wsdl/in-out">
9           <input element="tns:GetCitiesByCountry"/>
10          <output element="tns:GetCitiesByCountryResponse"/>
11    </operation>
12 </interface>
```

**Figure 2.4: Utilisation de l'élément *"interface"* dans la description du service web *GlobalWeather***

## 4. L'élément « *binding* »:

Cet élément décrit comment accéder au service. Son rôle est le même que l'élément du même nom dans la version 1.1. L'espace de nom définissant le protocole à utiliser (dans cet exemple SOAP figure 2.5, ligne 1) est défini comme attribut de cet élément.

```
1  <binding xmlns:wsoap="http://www.w3.org/ns/wsdl/soap"
2   name="GlobalWeatherSoap"
3   interface="tns:GlobalWeatherSoap">
4     <operation ref="tns:GetWeather" wsoap:soapAction="http://www.webserviceX.NET/GetWeather"/>
5     <operation ref="tns:GetCitiesByCountry"
6                 wsoap:soapAction="http://www.webserviceX.NET/GetCitiesByCountry"/>
7  </binding>
```

**Figure 2.5: Utilisation de l'élément *"binding"* dans la description du service web *GlobalWeather***

### 5. L'élément « *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)*. Ceci permet qu'une interface d'un service possède plus d'une localisation (i.e. plus d'un élément *endpoint*) pour répondre aux problèmes d'indisponibilité. L'exemple de l'élément service de la description du service web *GlobalWeather* est illustré par la figure 2.6.

```
1  <service name="GlobalWeather"
2  interface="tns:GlobalWeatherSoap">
3     <endpoint name="GlobalWeatherSoap" binding="tns:GlobalWeatherSoap"
4               address="http://www.webservicex.com/globalweather.asmx"/>
5  </service>
```

**Figure 2.6: Utilisation de l'élément *"service"* dans la description du service web *GlobalWeather***

## III. COMPARAISON ENTRE LA STRUCTURE DES DOCUMENTS WSDL 1.1 ET LA STRUCTURE DES DOCUMENTS WSDL 2.0

Les concepts inhérents à WSDL (catégories d'information à prendre en compte et niveaux d'abstraction) sont présents dans les versions 1.1 et 2.0. Trois éléments du document WSDL (*types, binding* et *service*) sont identiques en termes d'informations décrites (respectivement type de données structurées, protocole de communication et ports d'accès) [4].

| | Concepts | WSDL 1.1 | WSDL 2.0 |
|---|---|---|---|
| Niveau abstrait | Types de données | `<types/>` | `<types/>` |
| | Opérations | `<portType/>` | `<interface>`<br>`<operation/>`<br>`</interface>` |
| | Messages | `<message/>` | `<input/>`<br>`<output/>` |
| Niveau concret | Protocole de communication | `<binding/>` | `<binding/>` |
| | Ports d'accès | `<service/>` | `<service/>` |

**Figure3.1: Tableau de Comparaison des éléments XML entre WSDL1.1 et WSDL 20**

Il existe cependant deux différences notoires entre les versions 1.1 et 2.0. La première différence, qui se situe au niveau structurel du langage, concerne la description des messages échangés entre le service et le client lors de l'appel des opérations disponibles. La seconde différence concerne le niveau d'extensibilité du langage.

## 1. La description des messages:

Dans la version de WSDL 1.1, la description des messages repose sur l'élément message, alors que dans la version 2.0 elle se situe dans l'élément *operation* (lui-même sous-élément d'*interface*). La version 2.0 décrit de manière plus lisible les messages. En effet, la description des opérations et celle des messages se situent toutes deux dans l'élément *interface*, alors que dans la version 1.1 le message est décrit par un élément à part entière (l'élément *message*), et le lien entre l'opération et les messages se fait par un référencement dans l'élément *operation* (sous-élément de *portType*). Cette modification semble faciliter la lisibilité de la description du service web.

## 2. L'extensibilité du WSDL:

Dans les spécifications de WSDL 1.1, il n'existe pas de mécanisme d'extensibilité permettant d'ajouter des attributs aux éléments existants. Cette limitation a été levée dans la version 2.0. Cette possibilité d'extension est mise en pratique par la recommandation du W3C portant sur l'annotation sémantique de WSDL – SAWSDL. SAWSDL utilise l'extensibilité de WSDL 2.0 afin d'ajouter de l'information (par exemple, un paramètre de type *float* qui représente un prix est associé à une devise) aux propriétés fonctionnelles des services décrites par ce standard.

Des différences certaines existent entre les versions 1.1 et 2.0, non seulement sur le plan structurel mais aussi sur le plan de l'utilisation des services web. La recommandation WSDL 2.0 est certes plus compréhensible et offre davantage de flexibilité d'utilisation du service web. Néanmoins, aujourd'hui, les services web existants sont décrits par les spécifications WSDL 1.1 et les plates-formes d'applications web (telles que **.NET**) génèrent des descriptions WSDL 1.1. Le **W3C** et la fondation **Apache** offrent des outils permettant de convertir les descriptions de services web relevant de la version 1.1 en WSDL 2.0 et de gérer (analyser et valider) les documents WSDL 2.0 (convertisseur de version de W3C13 et le projet **Woden** d'**Apache**).

# Références :

[1] Web Service(WS) Security Tutorial with SOAP Example : «https://www.guru99.com/wsdl-web-services-description-language.html ».

[2] W3C Web Services Description Language (WSDL 2.0) « https://www.w3.org/TR/wsdl20/ »

[3] W3C Web Services Description Language (WSDL 1.1) « https://www.w3.org/TR/wsdl.html »

[4] Céline Lopez-Velasco: « Sélection et composition de services Web pour la génération d'applications adaptées au contexte d'utilisation». Thèse de doctorat soutenue en 2008 Université de Joseph Fourier.