

AIGUEPERSE Frédéric  
BRUN Pierre  
LEFEVRE Vincent  
LIMAIEM Heythem

Promotion 2006 - Option Informatique  
Intelligence Artificielle

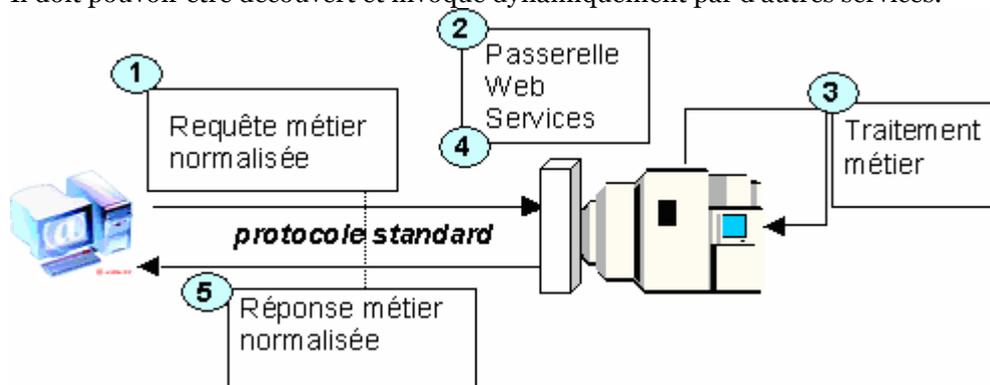


## - Web Services -

Enseignant : Alain MILLE

## 1. Qu'est-ce qu'un Web Service ?

Un Web Service est un composant implémenté dans n'importe quel langage, déployé sur n'importe quelle plate-forme et enveloppé dans une couche de standards dérivés du XML. Il doit pouvoir être découvert et invoqué dynamiquement par d'autres services.



Le concept des Web Service s'articule actuellement autour des trois acronymes suivants :

### 1.1. Le protocole d'accès : SOAP (Simple Object Access Protocol)

L'utilisateur du Service Web invoque le service par échange de messages XML : un message contient la requête, un autre le résultat de l'invocation du service (ou les éventuelles erreurs). Ces messages sont transportés par un protocole standard (HTTP, SMTP) ou tout autre mécanisme de transport (JMS, etc.). SOAP est la spécification qui formalise le format des messages XML qui sont échangés pour faire de l'invocation de services à distance : comment représenter une requête, une réponse, une erreur, etc. C'est un premier pas vers l'interopérabilité. L'invocation d'un Service Web peut donc se faire en tout langage de programmation permettant de manipuler du XML.

### 1.2. L'interface WSDL (Web Services Description Language)

L'interface d'un Service Web est décrite en XML. Elle décrit une liste d'opérations que l'utilisateur peut invoquer, et donne les détails techniques de cette invocation (protocoles, URL d'accès, ...).

### 1.3. Le référencement UDDI (Universal Description, Discovery and Integration)

SOAP et WSDL fournissent un moyen d'appeler les Services Web et de les modéliser. UDDI permet de les référencer : c'est l'annuaire des Services Web. Il s'agit d'un annuaire distribué, contenant des informations de trois types : une description des entreprises proposant des services, une description générale des services offerts, et des détails sur l'invocation de ces services (URL d'accès, protocoles utilisés, interfaces implémentées, etc.).

### 1.4. Objectif

L'objectif des Web Services est de :

- faciliter l'accès aux applications entre entreprises et ainsi simplifier les échanges de données.
- permettre aux applications d'inter-opérer à travers un réseau, indépendamment de leur plate-forme et de leur langage d'implémentation.

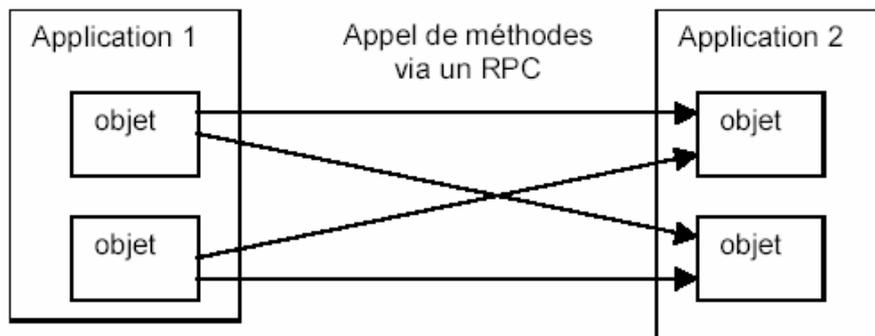
## 1.5. Principaux avantages

**Universalité :** Que les applications tournent sous Windows ou Unix, qu'elles soient codées en Java, C++, VB ou Fortran, qu'elles soient déployées sur un serveur ou non, qu'elles soient derrière un firewall ou pas, elles sont capables de communiquer à partir du moment où elles comprennent le XML.

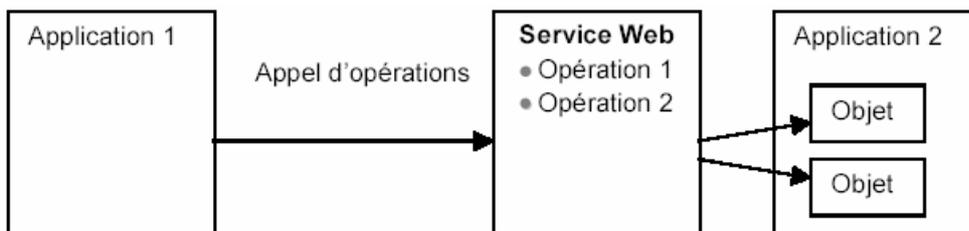
**Simplicité d'utilisation :** Utilisation des standards acceptés par tous : HTTP et XML.

**Support des pare-feux :** Un avantage significatif des Web services, relativement aux autres solutions d'architecture distribuée, est son support des pare-feux (firewalls) : l'utilisation du protocole HTTP sur le port 80, généralement ouvert, leur permet de passer sans encombre ces barrières de l'entreprise.

**Couplage souple des applications :** Le troisième atout majeur des Services Web est la possibilité de coupler les applications de manière beaucoup plus souple qu'avec Corba, RMI ou DCOM. En effet, ces RPC permettent de mettre en place des collaborations entre objets distants. Résultat, pour mettre en place une collaboration entre applications, il est nécessaire de faire collaborer les objets qui les composent :



Or ce mécanisme entraîne un couplage fort des applications qui collaborent. Il est souvent nécessaire que la structure des données soit la même des deux côtés de la connexion. Les Services Web quand à eux introduisent la notion de « service » : une application offre des fonctionnalités à d'autres applications, et on n'a plus à se soucier de la structure interne de l'application et des objets qui la composent. Les appels de méthodes au niveau objet sont la responsabilité du service, sorte de « proxy » entre l'application cliente, et les objets de l'application distante.

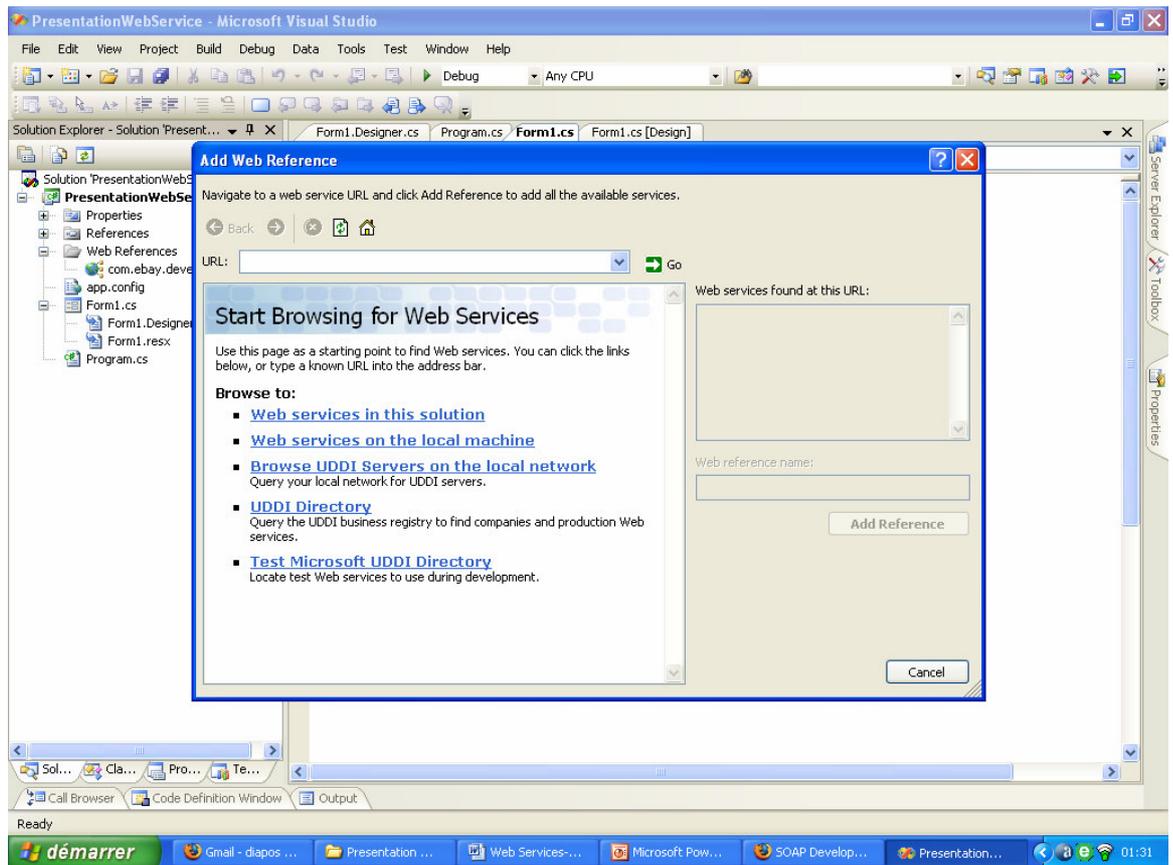


## 1.6 Un exemple d'application

Pour mieux visualiser ce que c'est un Webservice et comment les utiliser, nous allons étudier un cas précis. Nous allons réaliser une petite application qui exploite les WeServices proposés par eBay.

Pour cela, il faut d'abord obtenir un identifiant de développeur, et un jeton d'authentification auprès d'eBay, ce qui se fait sur cette page : <http://developer.ebay.com/>. Ensuite, on utilise le langage de programmation que l'on veut, et on charge le wsdl de ces webServices, disponible ici : <http://developer.ebay.com/webservices/latest/eBaySvc.wsdl>.

Dans la pratique, nous avons utilisé l'environnement .Net et programmé en C#. Pour exploiter les webServices d'eBay à partir du wsdl, il suffit d'ajouter une référence Web :



On peut soit indiquer directement l'url du wsdl, ou naviguer dans l'annuaire UDDI. Une fois ajoutée la référence web, tout se passe comme si on avait ajouté une librairie en local. On peut faire les appels des fonctions propres aux WebServices eBay comme si elles faisaient parties d'une classe locale (à l'exception près qu'ici, il faut s'identifier).

Un extrait du code, qui permet de faire des recherches parmi les objets en vente sur eBay :

```
// Create the request object (theRequest)
GetSearchResultsRequestType theRequest = new GetSearchResultsRequestType();

theRequest.Query = textBox1.Text;
PaginationType pagination = new PaginationType();
pagination.PageNumber = 1;
pagination.EntriesPerPage = 5;
theRequest.Pagination = pagination;
theRequest.Version = version;

// Variable qui contiendra le réponse
GetSearchResultsResponseType theResponse = null;

//On lance la requete
try
{
- Web Services -
- 4 / 11 -
```

```

        theResponse = service.GetSearchResults(theRequest);
for (int i = 0; i < theResponse.SearchResultItemArray.Length; i++)
{
    affich(theResponse.SearchResultItemArray[i].Item.Title);
}
        affich("Fin des résultats");
}

```

On remarque qu'aussi bien pour la requête que pour la réponse, jamais on utilise de fichiers XML, pourtant à la base des Services Web. On fait, les requêtes (comme les réponses) sont traduites en (respectivement depuis) SOAP. En réalité, voici à quoi ressemblent les requêtes et réponses réellement échangées (transparent pour l'utilisateur) :

Requête :

```

<?xml version="1.0" encoding="utf-8"?>
<GetSearchResultsRequest xmlns="urn:ebay:apis:eBLBaseComponents">
<RequesterCredentials>
<eBayAuthToken>ABC...123</eBayAuthToken>
</RequesterCredentials>
<Query>antique toy car</Query>
<PriceRangeFilter>
<MaxPrice currencyID="USD">10.00</MaxPrice>
</PriceRangeFilter>
</GetSearchResultsRequest>

```

Réponse :

```

<?xml version="1.0" encoding="utf-8"?>
<GetSearchResultsResponse xmlns="urn:ebay:apis:eBLBaseComponents">
<Timestamp>2005-01-27T00:58:56.300Z</Timestamp>
<Ack>Success</Ack>
<CorrelationID>00000000-00000000-00000000-00000000-00000000-00000000-
00000000</CorrelationID>
<Version>393</Version>
<Build>20050123212033</Build>
<SearchResultItemArray>
<SearchResultItem>
<Item>
<ItemID>4036164357</ItemID>
<ListingDetails>
<StartTime>2005-01-26T16:51:37.000Z</StartTime>
<EndTime>2005-02-02T16:51:37.000Z</EndTime>
<ViewItemURL>http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&category=307&
amp;item=4036164357&rd=1&ssPageName=WDVW</ViewItemURL>
</ListingDetails>
<SellingStatus>
<BidCount>0</BidCount>
<CurrentPrice currencyID="USD">8.0</CurrentPrice>
</SellingStatus>
<Site>US</Site>
<Title>Collectible Antique Toy Car</Title>

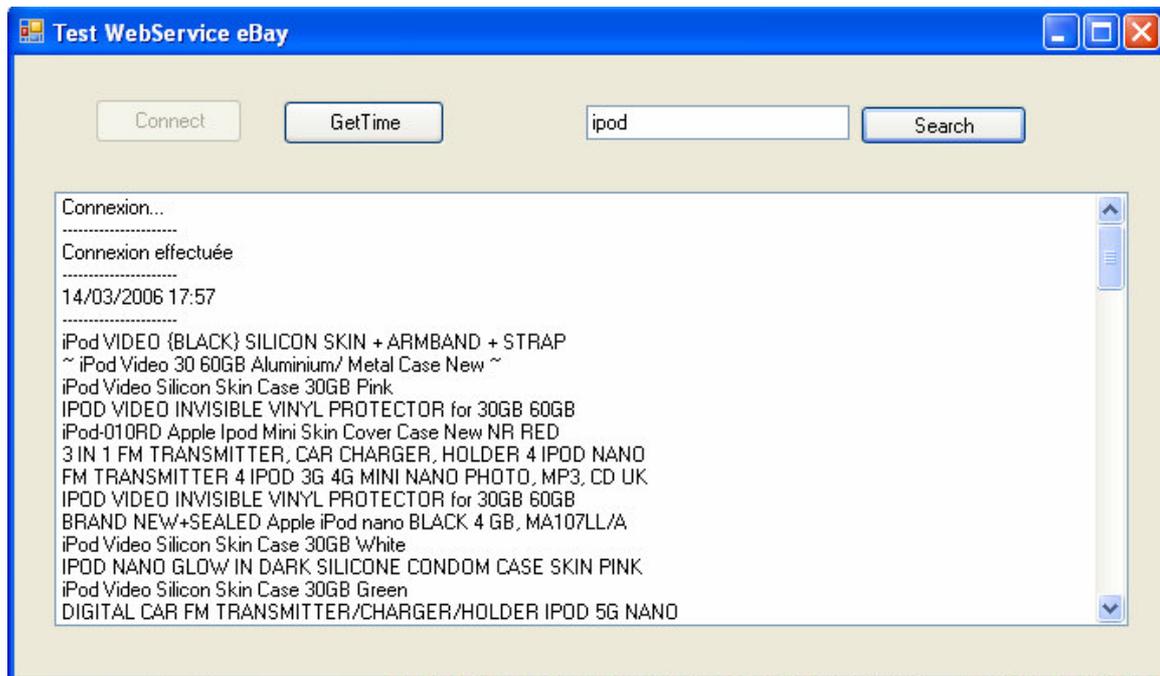
```

```

<ListingType>Chinese</ListingType>
<Currency>USD</Currency>
<SiteHostedPicture>
<GalleryType>None</GalleryType>
</SiteHostedPicture>
<SubTitle>
</SubTitle>
</Item>
</SearchResultItem>
</SearchResultItemArray>
<ItemsPerPage>100</ItemsPerPage>
<PageNumber>1</PageNumber>
<HasMoreItems>>false</HasMoreItems>
<PaginationResult>
<TotalNumberOfPages>1</TotalNumberOfPages>
<TotalNumberOfEntries>1</TotalNumberOfEntries>
</PaginationResult>
<CategoryArray />
</GetSearchResultsResponse>

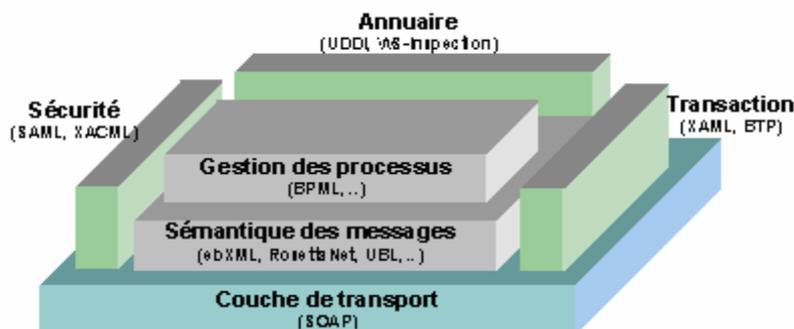
```

Voici une capture d'écran de la petite application de démo développée qui utilise le Webservice d'eBay :



## 2. Etat de l'offre actuelle

### 2.1. Etat de la normalisation



A l'heure actuelle, seule la couche de transport est réellement normalisée et ne souffre d'aucune contestation. Elle s'appuie sur le protocole SOAP pour l'échange des messages et sur le langage WSDL pour la définition du contrat de l'interface.

Les initiatives de définition de la couche de sémantiques des messages sont nombreuses et n'ont pour le moment pas conduit à une quelconque normalisation. Deux types de chantiers sont actuellement ouverts, l'un organisé selon les différents corps de métier, l'autre suivant une approche plus globale autour de consortium tel que OASIS (initiateur de ebXML) ou RosettaNet.

L'orchestration de transactions B2B complexes, fondée sur une modélisation normalisée des flux est également une initiative qui n'avance que très lentement et sur des activités non concertées. On peut citer en exemple BPML (Business Process Modeling Language de BPMI), WSFL (Web Services Flow Language, d'IBM) ou XLang (de Microsoft).

Au niveau des services, on pouvait penser que la proposition d'annuaire UDDI apporterait une solution définitive. On constate qu'il n'en est rien et que le canevas, trop global, du projet ne convient pas à une problématique d'échanges entre entreprises se connaissant. Il se voit maintenant concurrencer par WS-Inspection (proposé par IBM et Microsoft, pourtant à l'origine de UDDI). Moins ambitieux puisque consistant en une simple exposition, par agrégation, des services d'une entreprise, il est toutefois plus adapté à cette seconde problématique.

La gestion de la sécurité et des transactions est actuellement le frein le plus important à la mise en place d'architectures distribuées à base de Web Services. Plusieurs chantiers sont ouverts mais aucun n'est réellement accepté. On peut cependant penser que la norme XACML (eXtensible Access Control Markup Language) devrait supplanter SAML (Security Assertion Markup Language) au niveau sécurité et s'imposer à terme comme standard de sécurité. En ce qui concerne l'aspect transactionnel, la lutte est plus ouverte, même si BTP (Business Transaction Protocol) semble plus soutenu actuellement.

### 2.2. Exemples de web services :

- Google Web APIs : fourni la possibilité d'intégrer recherche et publicité à son application ou site en utilisant les ressources de Google. <http://www.google.com/apis/>
- Ebay : fourni la possibilité d'accéder directement à la base de données d'Ebay, on peut ainsi développer des interfaces permettant des actions telles que la mise en vente massive ou le « sniper » d'enchère qui consiste à surenchérir à la dernière seconde, <http://developer.ebay.com/common/api>
- Amazon : fourni l'accès aux bases d'Amazon pour pouvoir intégrer les articles dans des applications, [http://www.rjsoftwares.com/amazon\\_web\\_services/amazon\\_api.htm](http://www.rjsoftwares.com/amazon_web_services/amazon_api.htm)

- Flickr : offre des services de gestion de photos, <http://www.flickr.com/services/api/>
- Delicious : gestion de favoris, <http://del.icio.us/help/api/>

### 3. Futurs développements

#### 3.1. Achèvement de la normalisation

A l'heure actuelle, les services Web se basent sur l'utilisation des protocoles SOAP+WSDL+UDDI. Seuls les standards SOAP et WSDL ont bien été définis par le W3C (recommandation SOAP 1.2 en 2003 et WSDL 2.0). Pour le moment, malgré la multiplication des Web Services proposés, la norme UDDI est encore jeune. C'est pourquoi, étant donné l'intérêt suscité par ces nouvelles techniques, le W3C va s'efforcer de poser de nouveaux standards et recommandations dans les domaines suivants afin de consolider:

- Nouvelles extensions sur la norme SOAP : Web Services Internationalization
- Working Drafts sur la norme WSDL 2.0.
- Notamment grâce au nouveau groupe de travail WS2, l'alliance du Web Sémantique et des services Web va poser de nouveaux problèmes de normes et de représentation de sémantique (comment définir le sens des informations, quel sens donner, à qui est-il destiné ?).

#### 3.2. Intégration de la sémantique dans les services Web

La description sémantique est la clé pour résoudre de nombreux problèmes de haut niveau se rapportant aux services Web. Des recherches sont menées pour cibler des besoins particuliers dans le domaine, comme enregistrer des ressources ou des services, les découvrir sur le Web, composer un service en utilisant plusieurs autres services, etc. Les toutes premières solutions utilisent des agents Web, faisant souvent appel à des catalogues prédéfinis (UDDI), ou alors parcourant tout simplement le Web.

Parallèlement, l'activité Web Sémantique du W3C a développé les briques de bases des technologies du Web sémantique : la description des ressources (RDF) et le langage des ontologies Web (OWL). Le groupe d'intérêt W3C « Semantic Web Services » a montré un grand intérêt pour intégrer davantage de sémantique dans l'architecture des services Web. Le travail vise à définir une structure plus compréhensive et plus pertinente pour décrire tous les aspects des services, autorisant ainsi des outils plus puissants et une automatisation plus complète d'un large éventail d'activités de services Web, mais aussi afin d'exprimer leurs contraintes d'utilisation. Le W3C va se pencher maintenant sur la mise en place d'une infrastructure générique et extensible fondée sur les technologies du Web sémantique. Le W3C espère identifier les meilleurs cas d'utilisation qui pourront conduire à des solutions plus efficaces et plus complètes dans ces domaines notamment :

- Technologies du Web sémantique fondatrices, telles qu'RDF et OWL
- Technologies des services Web fondatrices, telles que SOAP et WSDL
- Registres, taxonomies et mécanismes de recherche, tel qu'UDDI
- Ontologies pour les services Web, tels qu'OWL-S et WSMO
- Chorégraphie de services Web, tel que WS-CDL
- Orchestration de services Web, tel que BPEL4WS

Dans le cadre de SOAP, on pourra ainsi donner du sens aux informations à différents niveaux de description, par exemple sur les arguments d'un service :

- le service prend un entier comme argument
- cet entier doit être un numéro ISBN

Mais aussi sur la réponse du service (disponibilité d'un livre selon son numéro ISBN). La sémantique peut être implicite mais aussi bien exprimée dans un langage humain ou machine. Enfin, on peut aussi imaginer comment le service fonctionne (type d'algorithme mais aussi quel type de structure de données est utilisé...).

- Web Services -

- 8 / 11 -

### 3.3. Agents Web

En créant des ontologies lisibles par les machines, et utilisées par des agents « intelligents » dans la recherche de ces services et de leur automatisation, de nombreux problèmes pourront se résoudre plus facilement. L'un des problèmes actuels est de trouver un service web en ligne. La création d'un vocabulaire contrôlé (par exemple UDDI aujourd'hui), les moteurs de recherche pourront trouver facilement les services Web. Mais ce n'est pas suffisant, et si l'on crée en fait un vocabulaire ontologique, des outils pourront à la fois utiliser une hiérarchie des informations (et les restrictions de propriétés) afin de trouver des résultats via les propriétés de classes mais aussi les liens sémantiques. Par exemple la recherche « fleur » pourra amener au résultat de « fleuriste » bien que le résultat ne soit pas explicite. Par exemple, grâce à la logique de description ou aux moyens d'inférence, un agent Web pourra trouver des catégorisations qui n'étaient pas explicites. Les agents Web seront donc un nouveau type de solution qui permettra d'automatiser des tâches mais surtout d'améliorer leur efficacité, particulièrement en combinant différents services Web mais aussi en choisissant les services Web pertinents. Dans le domaine du référencement, elles vont ainsi remplacer les méthodes de « crawling » qui récupère les informations dans les fichiers à balises (HTML et XML) à l'aide d'expressions régulières.

## 4. Conclusion

Les Web Services provoquent un intérêt évident auprès des architectes et des décideurs. La question récurrente concerne leur degré de maturité. Il est clair que sur une technologie aussi récente, le recul n'existe pas. Mais, même si l'édifice est encore fragile, il repose sur des bases solides (SOAP et WSDL) qui ont prouvé leur efficacité et leur maturité. D'ores et déjà, les Web Services ont quitté le champ des échanges inter-entreprises pour s'accaparer celui du référencement et de la mise à disposition des ressources de l'entreprise, empiétant en ce sens sur les technologies de type EAI. Cette utilisation à elle seule prouve la qualité du modèle et sa pérennité, notamment au niveau des couches les plus basses.

Par contre, la normalisation complète d'une architecture distribuée fondée sur les services Web n'est pour le moment qu'un rêve annoncé chaque année, pour la fin de l'année suivante ! Par ailleurs, ce modèle n'échappe pas à des problèmes de performance : les données sont transmises en ASCII dans une encapsulation XML elle-même intégrée dans une enveloppe SOAP puis HTTP... Le problème du choix de la bonne granularité du service, commun à toutes les architectures distribuées, se présente dans le cas des Web Services de manière plus aiguë encore.

Même s'ils n'ont pas acquis la maturité nécessaire à leur industrialisation, les services Web s'annoncent plus que jamais comme la réponse appropriée aux problématiques d'échange de données et d'intégration d'applications. Ils devront également répondre à différents défis :

- les problèmes de langages de représentation des connaissances (e.g. ontologies et descriptions " sémantiques " de ressources et de services), de leur sémantique formelle et de leur interopérabilité,
- les problèmes d'automatisation permettant un passage à l'échelle via l'exploitation de la fouille de données ("data mining"), la fouille de textes ("text mining") reposant sur des techniques de traitement automatique de la langue naturelle et de techniques statistiques, et la fouille des usages du Web (" web usage mining "),
- les problèmes liés à la manipulation de documents multimédia (comment représenter le contenu "sémantique" d'images, de sons ou de vidéos et comment annoter automatiquement des images, des sons ou des vidéos pour, entre autres, améliorer la recherche d'information ?),

- les problèmes liés à la manipulation d'objets mathématiques ou géométriques sur le Web (langages pour décrire non seulement la syntaxe, mais aussi la sémantique et la visualisation de formules mathématiques, outils interactifs et d'indexation pour manipuler et rechercher des formules mathématiques sur le Web),
- les problèmes de distribution (services distribués, architectures peer-to-peer, agents mobiles, vérification d'applications distribuées sur le Web, architectures Grid...) et les problèmes sous-jacents de sécurité,
- les problèmes liés à l'intelligence des services par la prise en compte de la "sémantique" dans la description, la découverte, la sélection et la composition de services, et par une plus grande personnalisation et adaptation à l'utilisateur de ces services,
- et les problèmes liés à la prise en compte du rôle du Web comme support à la coopération entre humains (individus, communautés ou organisations) à travers le Web.

## 5. Bibliographie

- *Foundations And Future Directions of Web Services*, Hugo Haas & Charlton Barreto  
<http://www.w3.org/2005/Talks/1114-hh-ecows/>
- *Web, sémantique et services*, Rose Dieng-Kuntz
- *Ontologies and services: Agents on the Semantic Web*, James Hendler  
<http://www.cs.umd.edu/~hendler/AgentWeb.html>
- *Site on the semantic Web*, MINDSWAP Group  
<http://www.mindswap.org>