

LE TRIPTYQUE SOAP/WSDL/UDDI

Eric van der Vlist (vdv@dyomedea.com)

Le triptyque SOAP/WSDL/UDDI

Web Services Convention

Juin 2004

CORBA OU DCOM SUR LE WEB

Alors que les Services Web REST ne sont qu'une évolution du Web tel que nous le connaissons pour le rendre plus facilement intelligible par des machines, les Services Web SOAP/WSDL/UDDI (SWU) sont en quelque sorte un détournement du Web pour implémenter des principes architecturaux pré-existants.

Au début (98) : XML-RPC

L' ancêtre des Services Web SWU est XML-RPC. Créé en 98, XML-RPC est un vocabulaire XML décrivant des appels et des réponses de procédures distantes (RPC = Remote Procedure Call).

Le principe des RPC est bien plus ancien que cela et l'on attribue généralement à Sun la première formalisation d'un mécanisme de RPC dans les années 90.

La notion de RPC est antérieure au règne de la programmation orientée objet et est, comme son nom l'indique, purement procédurale.

XML-RPC n'est qu'une adaptation très simple de ce principe à XML et HTTP, XML et HTTP permettant d'assurer une portabilité parfaite entre plateformes logicielles et matérielles.

SOAP introduit les objets (SOAP 1.0, novembre 99)

Sous l'influence de Microsoft, XML-RPC est remanié pour être adapté au monde de la programmation orientée objets et le résultat prend le nom de SOAP.

HTTP et RPC deviennent optionnels (SOAP 1.1, mai 2000)

Les développements autour de SOAP se poursuivent et SOAP se prépare à rentrer sous l' ombrelle du W3C.

En mai 2000, une nouvelle version (SOAP 1.1) est publiée comme note W3C et introduit deux niveaux d'ouverture par rapport à SOAP 1.0 :

- SOAP 1.1 devient indépendant du protocole sur lequel il s' appuie et ses liens avec HTTP sont relégués au niveau d'un « binding ».
- SOAP 1.1 devient un mécanisme de transfert de messages dont les appels de type RPC ne sont plus qu'une des applications possibles.

SOAP accepte les attachements (SOAP with attachements, décembre 2000)

Sous la pression de ebXML qui en fait un point de blocage pour l'utilisation éventuelle de SOAP dans son architecture technique, HP et Microsoft publient une note W3C définissant l'utilisation de SOAP avec des messages attachés.

PAS AUSSI FACILE QU'IL N'Y PARAÎT

SOAP ne suffit pas à refaire Corba ou DCOM

Certes, SOAP permet l'invocation à distance de méthodes, mais entre SOAP et les technologies d'objets répartis telles que Corba (OMG) ou DCOM (Microsoft), il y a encore beaucoup de chemin à parcourir.

Pour ces systèmes, l'invocation de services à distance n'est que la partie émergée de l'iceberg et elle ne se réalise dans de bonnes conditions (au niveau conception, administration, sécurité, ...) que parce qu'elle se fait dans un cadre bien défini.

Tout le reste est à (ré-)inventer

Toutes ces fonctionnalités qui ne sont pas couvertes par SOAP sont donc à réinventer si l'on souhaite utiliser SOAP comme on utilise Corba ou DCOM ce qui est le pari fait par les promoteurs des Services Web « SWU ».

La liste des spécifications publiées ou en cours de rédaction est déjà longue, d'autant qu'il n'y a pas unanimité parmi les acteurs sur les solutions à adopter ni même sur l'organisme de standardisation à utiliser pour mener à bien ces travaux.

Consensus SOAP + WSDL + UDDI?

Parmi toutes ces spécifications, il semble se dégager un certain consensus de la part des fournisseurs (Microsoft, IBM, SUN, ...) autour des trois spécifications SOAP, WSDL et UDDI.

SOAP

Exemple de requête SOAP/HTTP:

```
POST /StockQuote HTTP/1.1
Host: www.stockquoteserver.com
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: "Some-URI"

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <m:GetLastTradePrice xmlns:m="Some-URI">
      <m:tickerSymbol>DIS</m:tickerSymbol>
    </m:GetLastTradePrice>
  </soapenv:Body>
</soapenv:Envelope>
```

Exemple de réponse SOAP/HTTP

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <m:GetLastTradePriceResponse xmlns:m="Some-URI">
      <m:price>34.5</m:price>
    </m:GetLastTradePriceResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Description de services web : WSDL

Web Services Description Language : description formelle et à deux niveaux de l'interface des Services Web

Comme son nom l'indique, WSDL permet de décrire des services web. Plus précisément, WSDL décrit de manière formelle les interfaces des services web et ce à deux niveaux : un niveau abstrait indépendant du protocole auquel est associé un protocole par « binding ».

Types -> messages -> portTypes -> bindings -> services (WSDL 1.1)

En WSDL 1.1, Une description WSDL définit d'abord des types, utilisés pour former des messages, associés dans des « ports », reliés à un protocole par des « bindings » formant des Services Web.

WSDL 2.0 sera différente

Le W3C travaille actuellement sur la version 2.0 de WSDL qui reprendra et complétera ces principes mais changera certains noms (le W3C ne semblant pas avoir été sensible à la poésie de l'image maritime du « port » préfère notamment parler de « endPoint »).

Au delà de ces modifications mineures, WSDL 2.0 apportera d'autres évolutions (notamment le support d'autres langages de schéma) et il faudra prévoir une phase migration.

Types (exemple)

```
<types>
  <schema targetNamespace="http://example.com/stockquote.xsd"
    xmlns="http://www.w3.org/2000/10/XMLSchema">
    <element name="TradePriceRequest">
      <complexType>
        <all>
          <element name="tickerSymbol" type="string"/>
        </all>
      </complexType>
    </element>
    <element name="TradePrice">
...
      </element>
    </schema>
  </types>
```

Messages (exemple)

```
<message name="GetLastTradePriceInput">
  <part name="body" element="xsd1:TradePriceRequest"/>
</message>

<message name="GetLastTradePriceOutput">
  <part name="body" element="xsd1:TradePrice"/>
</message>
```

PortType (exemple)

```
<portType name="StockQuotePortType">
  <operation name="GetLastTradePrice">
    <input message="tns:GetLastTradePriceInput"/>
    <output message="tns:GetLastTradePriceOutput"/>
  </operation>
</portType>
```

Binding (exemple)

```
<binding name="StockQuoteSoapBinding"
type="tns:StockQuotePortType">
  <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="GetLastTradePrice">
    <soap:operation
soapAction="http://example.com/GetLastTradePrice"/>
    <input>
```

```
        <soap:body use="literal"/>
    </input>
    <output>
        <soap:body use="literal"/>
    </output>
</operation>
</binding>
```

Service (exemple)

```
<service name="StockQuoteService">
  <documentation>My first service</documentation>
  <port name="StockQuotePort" binding="tns:StockQuoteBinding">
    <soap:address location="http://example.com/stockquote"/>
  </port>
</service>
```

Annuaire de services Web : UDDI

Universal Description, Discovery and Integration : l'annuaire des Services Web...

UDDI (Universal Description, Discovery and Integration) est en quelque sorte un LDAP en XML spécifique aux services Web.

Un modèle de données, une interface et un service « universel »

UDDI est à la fois un modèle de données permettant de décrire un service web, la définition d'une interface permettant de manipuler ce modèle de données et la mise à disposition de la communauté de UDDI sous forme d'un service universel et gratuit.

Le modèle de données UDDI

Plus figé que LDAP

Alors que le modèle de données LDAP est extrêmement générique et que chaque système d'information peut définir son schéma LDAP, le modèle de données UDDI est un modèle plus figé et la spécification UDDI définit explicitement quelles en sont les entités.

Si l'on pourrait représenter et stocker un annuaire UDDI dans un annuaire LDAP après avoir pris le soin de définir un schéma UDDI pour LDAP, la réciproque ne serait donc pas possible dans le cas général.

Le modèle de données UDDI est défini sous forme de schéma W3C XML Schéma.

BusinessEntity : pages blanches

Les « businessEntities » sont en quelque sorte les pages blanches d'un annuaire UDDI et elles décrivent les organisations ayant publié des services dans le répertoire.

On y trouvera notamment le nom de l'organisation, ses adresses (physiques et Web), des éléments de classification, une liste de contacts, ...

Chaque businessEntity est identifiée par une « businessKey ».

ServiceEntity : pages jaunes

Les « serviceEntities » sont en quelque sorte les pages jaunes d'un annuaire UDDI et décrivent de manière non technique les services proposés par les différentes organisations.

On y trouvera essentiellement le nom et la description textuelle des services ainsi qu'une référence à l'organisation proposant le service et un ou plusieurs « bindingTemplates ».

BindingTemplate : coordonnées des services

Les « bindingTemplates » donnent les coordonnées des services. Cherchant à être très générique en ce domaine, UDDI permet de décrire des Services Web HTTP, mais également des services invoqués par d'autres moyens (SMTP, FTP, fax, téléphone, ...).

Ils contiennent notamment une description, la définition du « point d'accès » (suivant les cas, une URL, un numéro de téléphone, ...) et les éventuels « tModels » associés.

tModel : description technique des services

Les « tModels » sont les descriptions techniques des services. UDDI n' impose aucun format pour ces descriptions qui peuvent être publiées sous n'importe quelle forme et notamment sous forme de documents textuels (XHTML par exemple).

C'est à ce niveau que WSDL intervient comme le vocabulaire de choix pour publier des descriptions techniques de services.

L'interface UDDI

L'interface UDDI est définie sous forme de documents UDDI et implémentée sous forme de Service Web SOAP. Elle est composée des modules suivants :

Interrogation (« inquiry »)

Cette interface permet de rechercher des informations dans un répertoire UDDI et de lire les différents enregistrements enregistrés suivant le modèle de données UDDI.

Publication

Cette interface permet de publier des informations dans un répertoire UDDI conformément à son modèle de données.

Sécurité

Cette interface est utilisée pour obtenir et révoquer les jetons d'authentification nécessaires pour accéder aux enregistrements protégés dans un annuaire UDDI.

Contrôle d'accès et propriété (« custody and ownership transfer »)

Cette interface permet de transférer la propriété d'informations (qui est à l'origine attribué à l'utilisateur ayant publié ces informations) et de gérer les droits d'accès associés.

Abonnement (« Subscription »)

Cette interface permet à un client de s'abonner à un ensemble d'informations et d'être avertis lors des modifications de ces informations.

Tous les répertoires UDDI doivent gérer un avertissement par polling (le client interroge le serveur pour savoir si des modifications ont eu lieu sur les données auxquelles il est abonné).

Une fonctionnalité optionnelle est également prévue permettant au client de communiquer au serveur la définition d'un Service Web sur lequel il souhaite être prévenu en cas de modification.

Réplication interne (noeuds d'un même annuaire)

A côté des interfaces utilisateurs que nous venons de voir, UDDI définit également l'interface permettant de synchroniser les noeuds d'un même annuaire UDDI.

Réplication externe (interrogation, publication, abonnement)

La réplication externe par duplication d'informations entre différents annuaires UDDI n'a pas donné lieu à la définition d'une interface spécifique mais se fait en utilisant les interfaces interrogation (pour la lecture dans un annuaire), publication (pour la publication dans un autre annuaire) et éventuellement abonnement (pour pouvoir propager les modifications ultérieures).

Le service UDDI (UBR)

UBR (UDDI Business Registry)

Les promoteurs de UDDI proposent également un répertoire UDDI disponible sous forme

d'un service gratuit.

Répliqué (IBM, Microsoft, SAP, NTT)

Ce répertoire est hébergé sur des noeuds administrés par IBM, Microsoft, SAP et NTT et ces noeuds sont synchronisés au moyen du mécanisme de réplication interne.

Annuaire publics, annuaire privés

A côté de cet annuaire public, chacun peut déployer son propre annuaire UDDI et y répliquer tout ou partie de l'annuaire UBR.

AUTRES SPÉCIFICATIONS

http://en.wikipedia.org/wiki/List_of_Web_service_specifications

La meilleure liste de spécifications associées aux Services Web que j'ai trouvée est publiée par Wikipedia. Elle liste les principales spécifications en les classant par catégories :

Directory access

Service Description

News feeds

Messaging and Function Calls

Basic Profile Specifications

Business Process Specifications

Security Specifications

Reliable Messaging Specification

Reliable Messaging Specification

Basic XML Specifications

Other

