
Veille Technologique - Discel

Auteur(s) : Matthieu Bouix - Vincent Boutour

Encadrants : Dominique Seret (Paris Descartes) & Amin Elsaleh (Kaprod)

MODULE ENSEIGNEMENT : PROJET PLURIDISCIPLINAIRES

DATE : 31 JANVIER 2010

Sommaire

1	INTRODUCTION : LE PROJET DI-SC-EL	3
1.1	Définition des concepts de base.....	3
1.2	Résumé du Projet di-sc-el.....	3
2	LES STANDARDS D'UNE ARCHITECTURE WEB-SERVICES	5
2.1	XML	5
2.2	Message - SOAP.....	5
2.3	Découverte.....	6
2.3.1	UDDI.....	6
2.3.2	ebXml	7
2.4	Description des services	7
2.4.1	WSDL.....	7
2.4.1.1	WSDL-S.....	9
2.4.2	SAWDSL.....	10
2.5	Description de processus	10
2.5.1	BPEL4WS	11
2.5.1.1	Objectifs.....	11
2.5.1.2	Le langage.....	11
2.5.2	XPDL.....	11
2.5.2.1	Présentation.....	11
2.5.2.2	Outils.....	12
2.5.2.3	Exemple	12
2.5.3	BPMN.....	12
2.6	Sécurité	13
2.6.1	WS-SECURITY	13
2.7	Schéma d'articulation des différents standards d'une architecture Web-Services.....	14
3	WEBSPHERE VS GLASSFISH : ETUDE COMPARATIVE DE DEUX SERVEURS D'APPLICATION	15
4	SYNTHESE : LES BASES DE DI-SC-EL	17
5	ANNEXES	18
5.1	Webographie.....	18
5.2	Annexe : Illustration de la configuration de WebSphere	20
5.3	Table des illustrations	25

1 INTRODUCTION : LE PROJET DI-SC-EL

Titre : Comment un document intelligent définit et gère un ou des scénarii e-Learning ? Cas du précepteur virtuel.

1.1 DÉFINITION DES CONCEPTS DE BASE

WEB SEMANTIQUE

L'objet du projet di-sc-el est de permettre aux entreprises de créer des Services WEB autour de processus d'affaire innovants qui alimenteront l'économie en réseau du 21ème siècle. Pour accomplir cet objectif, nous avons mis en place:

- ✓ Un langage de description de processus qui décrit des spécifications par des utilisateurs métiers (assurance, banque, automobile, etc.) non techniciens.
- ✓ Une infrastructure qui peut transformer ces spécifications en des processus d'affaire exécutables. La connectivité sémantique entre les processus d'affaire et un ensemble de processus légaux.

SPDF

Le projet di-sc-el offre une solution avancée pour l'utilisation de la technologie des services WEB grâce à la création d'un nouveau langage de description de processus (SPDF : Format Standard de Description de Processus). Il est envisagé grâce à SPDF qu'un concepteur puisse définir son modèle de document qui couvre la représentation syntaxique et sémantique d'un document ainsi que les standards d'échange de documents électronique.

SPDF et Ontologie du service WEB

SPDF prend en compte les ontologies du service WEB dont la définition précise est la suivante: « Les ontologies sont un vocabulaire partagé qui définit les concepts dans un domaine y compris leurs propriétés et les liens entre eux. »

Document intelligent

SPDF permet au concepteur/utilisateur métier de définir son modèle de document « intelligent » qui couvre la représentation syntaxique et sémantique ainsi que les modalités d'échange de documents électronique. Le concepteur/utilisateur commence par définir des éléments (schémas XML) à l'intérieur d'un référentiel intelligent » en utilisant un fichier de configuration.

1.2 RÉSUMÉ DU PROJET DI-SC-EL

La vision de di-sc-el est d'offrir des services WEB à valeur ajoutée en s'appuyant sur un langage de description de processus (SPDF) permettant à un utilisateur métier le pilotage de scénarios métier sur un serveur en ligne.

Le pré requis à l'utilisation de SPDF est :

1. comment structurer les données au sein d'une entreprise?
2. comment construire le document intelligent avec les liens inter données permettant de les sécuriser et assurer leur fiabilité?

Introduction : le projet DI-SC-EL

3. comment gérer un processus automatisé à partir du document intelligent? C'est la partie informatique du projet qui utilisera des ressources « Open Source ».

Notre projet se limitera dans ses champs d'application à la construction d'un environnement WEB-e-Learning qui offre la possibilité de partager les connaissances à une large audience grâce à une collaboration entre :

- le précepteur physique et,
- le précepteur virtuel (un processus automatique piloté par le document intelligent).

2 LES STANDARDS D'UNE ARCHITECTURE WEB-SERVICES

2.1 XML

Le langage X.M.L.¹ est le dénominateur commun de l'ensemble des normes et spécifications d'aujourd'hui. Dans le domaine des *Web-Services*, celui-ci contribue à disposer d'interfaces normalisées et interopérables. Ce langage offre des possibilités sans limite quant au stockage ou au transfert de données. Il structure de façon hiérarchique un ensemble d'informations au moyen de balises personnalisables par l'utilisateur. Ces balises pouvant être elles-mêmes paramétrées par l'utilisateur.

Ce langage fait l'objet d'une normalisation par le W3C et l'ISO en 2001. Il a été révisé récemment en fin d'année 2009 afin d'y inclure la notion d'espaces de nom concernant la version XML 1.0. La version la plus aboutie est la 1.1 qui corrige les limitations de la première concernant l'encodage des caractères ou la présence de types complexes tels que les énumérés. La version 1.1 demeure peu utilisée compte tenu du faible apport à l'usage courant du langage.

Afin de structurer un document XML, il existe un langage de description de la structure d'un fichier balisé appelé D.T.D.² Le fichier de description restreint le nom des balises, des paramètres ainsi que les valeurs que peuvent prendre ces composantes afin de présenter un fichier XML dans un format attendu. Ce langage fait l'objet d'une normalisation par l'ISO et a été révisé au premier trimestre 2009 pour intégrer la notion d'espaces de nom. Cette révision a donné lieu à D.S.D.L.³. DTD reste cependant le plus utilisé par l'industrie du fait de sa stabilité.

Le W3C apporte un autre langage de description qui est *XML Schema*. L'un comme l'autre fonctionnent peu ou prou sur le même principe. Il est à noter toutefois que *XML Schema* n'a pas été révisé depuis 2004, marquant une certaine stabilité, soit par absence de maintenance soit par complétude.

2.2 MESSAGE - SOAP

Le protocole S.O.A.P.⁴ est bâti sur XML et permet l'appel de procédures distantes (R.P.C.⁵). SOAP offre un échange de messages entre serveurs via un message unique qui se décompose en deux parties : un entête et un corps du message. Son fonctionnement est analogue à une lettre postée :

- le format est libre, il est possible de mettre dans le corps de message les informations que l'on souhaite sous la forme que l'on souhaite ;
- l'enveloppe est indépendante de l'infrastructure qui l'achemine.

¹ X.M.L. (Extended Markup Language)

² D.T.D. (Document Type Definition)

³ D.S.D.L. (Document Schema Definition Languages)

⁴ S.O.A.P. (Simple Object Access Protocol)

⁵ R.P.C. (Remote Procedure Call) : Protocole permettant une communication entre serveurs d'applications via des procédures publiques. Aussi appelé protocole client-serveur.

L'utilisation de SOAP via H.T.T.P.⁶ facilite la communication car ce dernier se charge de gérer l'ensemble des contraintes d'expéditions d'un message, ce que ne font pas d'anciennes technologies.

SOAP a été développé conjointement par IBM et Microsoft dans le cadre d'une architecture S.O.A.⁷. Il est désormais une recommandation du W3C. Ce protocole n'a pas fait l'objet de révisions depuis 2007 et est très largement utilisé dans l'industrie informatique et plus particulièrement dans le B2B. Il est une évolution de XML-RPC qui n'offrait pas la notion d'enveloppe de SOAP.

2.3 DECOUVERTE

Une architecture informatique doit être en mesure de décomposer une action complexe en unité de traitement triviale. Nous avons vu précédemment de quelle manière un message était émis d'un client pour parvenir jusqu'au serveur. XML et SOAP sont des standards dont **SPDF** ne pourra pas se passer car ceux-ci se retrouvent comme pré-requis d'un bon nombre des standards sur lesquels l'architecture est bâtie.

La première activité, une fois le message reçu, est d'aiguiller ce message vers la bonne unité de traitement comme le ferait un routeur avec des paquets dans un réseau informatique. Cette couche de découverte des services est l'un des éléments clés de l'architecture des *Web-Services*.

2.3.1 UDDI

L'annuaire U.D.D.I.⁸ est bâti sur XML et s'oriente résolument vers des architectures SOA. La description de sa structure XML est à ce titre faite via XML Schema. Il repose sur un ensemble de technologies que sont SOAP et WSDL (que nous aborderons plus tard dans ce document). L'annuaire est actuellement en version 3 est à ce titre un standard de OASIS depuis 2005, sans évolution connue depuis. Apache, Oracle, IBM et Microsoft proposent des serveurs UDDI.

La consultation de l'annuaire peut s'effectuer de trois manières différentes et ce, récursivement. La démarche de recherche est alors analogue à ce qui est fait de façon naturelle lors de la recherche d'un numéro de téléphone dans un annuaire téléphonique conventionnel :

- les pages blanches comportant la liste des entreprises ainsi que les informations associées ;
- les pages jaunes comportant la liste des services web de chacune des entreprises en se basant sur une description au format WSDL ;
- les pages vertes comportant la liste des informations techniques de chaque service web.

On remarque ici une dépendance technique entre UDDI et WSDL dans la mesure où la description d'un service s'effectue dans ce langage. Il faudra veiller à ce que l'un comme

⁶ H.T.T.P. (HyperText Transfer Protocol) : Protocole le plus couramment utilisé dans les infrastructures client-serveur. Le serveur HTTP utilisé dans le langage Java est Tomcat.

⁷ S.O.A. (Service Oriented Architecture) : Architecture informatique orientée vers les services utilisateurs

⁸ U.D.D.I. (Universal Description Discovery and Integration)

l'autre soit un référent solide sur le long terme. UDDI est utilisé par le gouvernement fédéral américain, ce qui peut s'avérer être un garant. **SPDF** s'appuiera sur ces deux standards.

UDDI a été conçu dans l'optique d'être implémenté dans la suite de spécifications ebXML.

2.3.2 EBXML

La suite ebXML⁹ offre une infrastructure entièrement basée sur XML destinée à faciliter les échanges électroniques B2B dans une optique de commerce électronique. Cette suite de spécifications a été éditée par OASIS et ensuite normalisée par l'ISO en 2005 dans sa version finale et actuelle. Les travaux se poursuivent toujours afin d'atteindre un niveau de maturité permettant son intégration dans les SOA.

L'intérêt de l'infrastructure est de proposer un ensemble souple et flexible répondant au plus grand nombre de besoins des entreprises tant que celles-ci n'ont pas des volumétries d'échanges importantes. Cette flexibilité facilite la découverte de partenaires par une plateforme de communication inter opérable.

L'infrastructure se compose de trois couches : information métier, processus métier, découverte de partenaires. Actuellement, cette infrastructure n'est pas très répandue dans les administrations en tant que norme et « *best practice* ». Nous n'avons pas trouvé d'informations concernant son utilisation au sein des sociétés.

2.4 DESCRIPTION DES SERVICES

Le cœur d'une application basée sur les *Web-Services* est bien évidemment le service web en lui-même. Au-delà de l'existence de ce service, il est nécessaire de bien définir en amont ses fonctions et contextes d'appels. SPDF aura besoin d'ajouter des informations sémantique à cette description afin qu'une notion d'ontologie apparaisse.

Les ontologies sont le pivot indispensable à la réalisation d'un document intelligent. Leur maîtrise ainsi que leur enrichissement devront être suffisants pour être à même de parler d'intelligence dans le cadre de SPDF. La couche de description des services contient toute l'intelligence de l'application, indépendamment du processus qui animera ces services.

2.4.1 WSDL

Le langage W.S.D.L.¹⁰ est un langage basé sur XML. Il est très largement utilisé en synergie avec SOAP. Ce standard a été rédigé par Microsoft et IBM parallèlement à la rédaction des spécifications de SOAP. Le but initial était de présenter un exemple de *Web-Services* que SOAP serait à même de gérer. Les deux standards restent donc étroitement liés.

Le format XML du langage WSDL est décrit via « XML Schema ». La description d'un service se structure en deux niveaux. Une section dite « abstraite » comportant les informations d'interface analogue à ce qui est fait dans le langage Java et une section dite « concrète » comportant les informations techniques d'implémentation.

⁹ ebXML (Electronic Business using eXtensible Markup Language)

¹⁰ W.S.D.L. (Web Services Description Language)

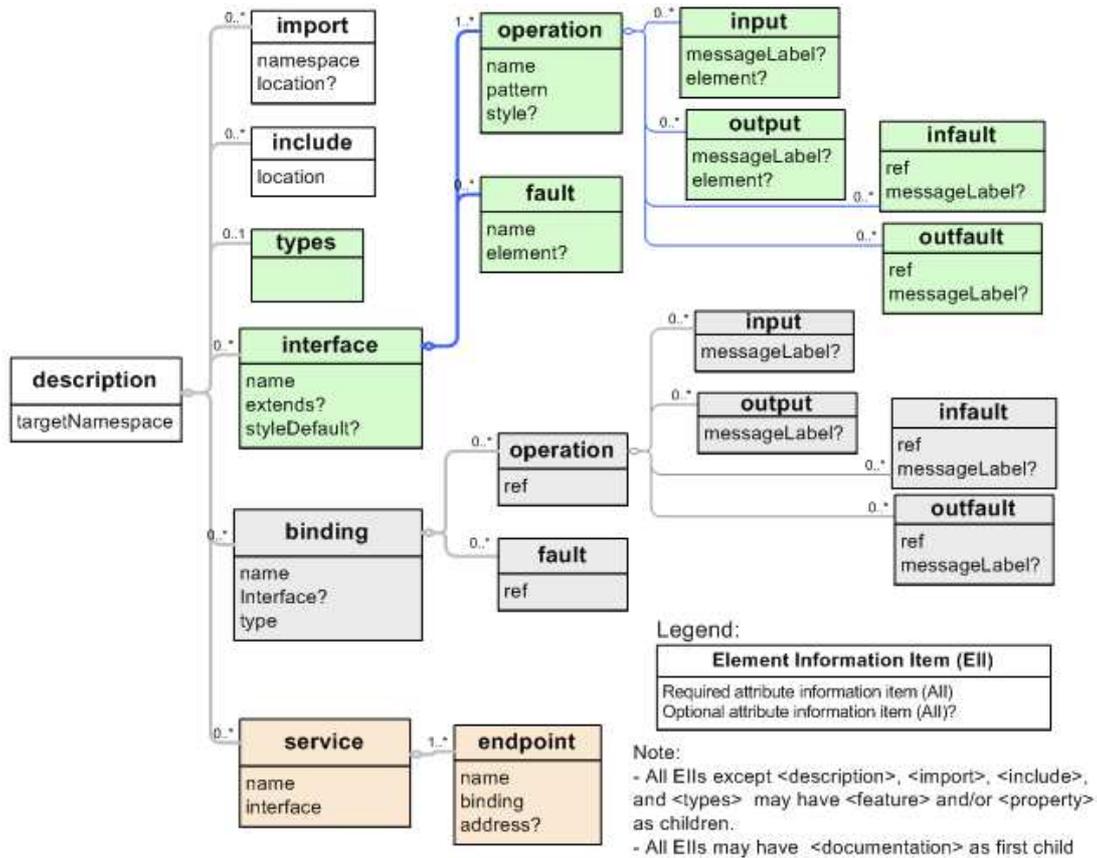


Figure 1 : Structuration WSDL 2.0 © W3C

Ces spécifications ont été acceptées par le W3C dans sa deuxième version (2.0) et ne semble pas avoir évolué depuis 2007.

Exemple d'utilisation de WSDL

```

<message name="getTermRequest">
  <part name="term" type="xs:string"/>
</message>

<message name="getTermResponse">
  <part name="value" type="xs:string"/>
</message>

<portType name="glossaryTerms">
  <operation name="getTerm">
    <input message="getTermRequest"/>
    <output message="getTermResponse"/>
  </operation>
</portType>

<binding type="glossaryTerms" name="b1">
  <soap:binding style="document"
  transport="http://schemas.xmlsoap.org/soap/http" />
  <operation>
    <soap:operation soapAction="http://example.com/getTerm"/>
    <input><soap:body use="literal"/></input>
    <output><soap:body use="literal"/></output>
  </operation>
</binding>

```

Pour les besoins de SPDF, cette première mouture de WSDL ne permet pas de répondre à la notion de sémantique requise afin d'élaborer un document intelligent. Une version intégrant cette notion a été éditée peu de temps après son adoption par le W3C.

2.4.1.1 WSDL-S

Le langage WSDL-S, S pour *Semantic*, reprend l'ensemble des composantes de WSDL en ajoutant une information sémantique dans la description même d'un service web ou plus finement, sur l'opération de ce service.

Exemple d'annotation Entrée/Sortie sur une interface d'ordre d'achat.

L'entrée est représenté par l'élément Purchase Order Request, la sortie par l'élément Purchase Order Response.

```
<interface name="Purchase Order">
  <Operation name="processPurchaseOrder" pattern=wsdl in-out
  .....
  <input message label="processPurchaseOrderRequest"
    element=tns:" processPurchaseOrderRequest"/>
  <output message label="processPurchaseOrderResponse"
    element=" processPurchaseOrderResponse"/>
  <!--Precondition and Effect are added as extensible elements on an
  operation
  .....
</Operation>
</interface>
```

Cette notion sémantique se traduit par la présence de mots-clés mais également d'une interdépendance entre différents services web sous la forme de pré et post conditions. Cette dernière modification offre une contractualisation des services web.

La notion d'ontologie est également introduite comme une part importante du document WSDL-S. Cette intégration se fait par l'utilisation de O.W.L.-S.¹¹ qui est un standard du W3C depuis la fin de l'année 2009.

Le langage WSDL-S n'a pas été standardisé par le W3C ou d'autre organismes de standardisations et ne le sera pas à cause de la sortie d'un standard analogue par le W3C. Cependant, il peut se révéler plus approprié pour un formalisme de synthèse tel que SPDF sachant que son pendant standardisé, SAWSDL, est un candidat plus stable dans le temps.

¹¹ O.W.L.-S. (Web Ontology Language for Web Services)

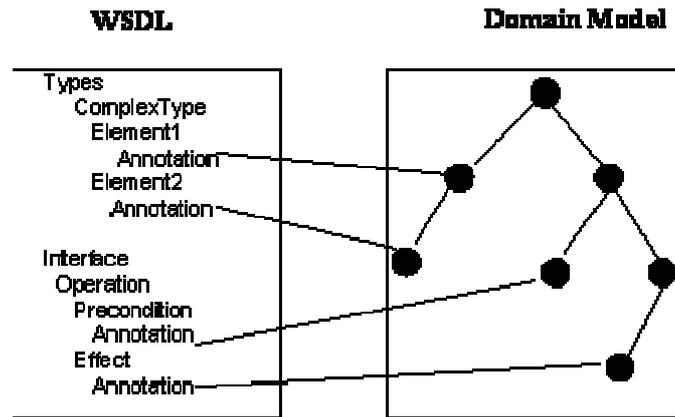


Figure 2 : Extrait du standard WSDL-S

Le schéma de la figure ci-dessus nous rapproche de la description suivante du document intelligent « *The key to semantic discovery of web services is having semantics in the description itself (i.e. formally selfdescribed and machine processable) and then using semantic matching algorithms to find the required services* ».

2.4.2 SAWDSL

Le langage S.A.W.S.D.L.¹² est donc analogue à WSDL-S à la différence que celui-ci est un standard du W3C et qu'il s'applique à la fois aux les services, étant une surcouche de WSDL, mais également à XML Schema afin de donner une valeur sémantique à la structure d'un document (chose que ne prévoyait pas WSDL-S). Ce langage a été standardisé à la mi-2007 et n'a pas évolué depuis.

L'introduction de l'information sémantique se fait en utilisant un standard W3C concernant la spécification de méta données nommé R.D.F.¹³. Ce modèle de description peut être sérialisé en utilisant XML et se veut comme le socle de base de toute représentation sémantique. Ce standard a été édité partiellement par IBM.

2.5 DESCRIPTION DE PROCESSUS

Il existe deux types de langages de description de processus : les langages graphiques et les langages d'exécution.

Les premiers ont pour but de représenter le plus clairement possible les différentes étapes de la procédure afin qu'elles soient comprises d'une part par les techniciens et d'autre part par les simples utilisateurs. Ces étapes sont représentées par différentes icônes et liaisons.

Les seconds se présentent sous la forme d'un fichier au format XML et sont destinés à être interprétés par un moteur d'exécution.

Nous étudierons ici les plus utilisés dans l'industrie, BPEL4WS, XPDL et BPMN.

¹² S.A.W.S.D.L. (Semantic Annotations for WSDL and XML Schema)

¹³ R.D.F. (Resource Description Framework)

2.5.1 BPEL4WS

Né de la fusion des langages WSFL et XLANG, développés respectivement par IBM et Microsoft, BPEL4WS est un langage d'exécution standardisé en 2003 par le consortium OASIS (qui l'a renommé « WS-BPEL » pour être en harmonie avec les autres standards touchant aux services Web). L'appellation « BPEL » est maintenant la plus utilisée.

2.5.1.1 Objectifs

BPEL a été conçu pour répondre à une dizaine d'objectifs :

- définir des processus métiers qui interagissent avec des entités externes via les opérations décrites dans le standard WSDL 1.1 ;
- utiliser XML sans définir de représentation graphique ni de méthodes de conception de processus ;
- définir un ensemble de concepts pour l'orchestration de services web destinés à être utilisés à l'extérieur comme à l'intérieur du processus. Ce type de processus définit le comportement d'une entité autonome, agissant typiquement au sein d'autres entités de même type. *« Such a business process defines the behavior of a single autonomous entity, typically operating in interaction with other similar peer entities. It is recognized that each usage pattern (i.e. abstract view and executable view) will require a few specialized extensions, but these extensions are to be kept to a minimum and tested against requirements such as import/export and conformance checking that link the two usage patterns. »*
- fournir à la fois une représentation hiérarchique et une représentation graphique des règles de contrôles (« *control regimes*») et permettre de les mixer facilement afin de réduire la fragmentation du processus ;
- fournir des fonctions simples de manipulation de données pour gérer les données du processus et les structures de contrôle (« *control flow* ») ;
- comprendre un mécanisme d'authentification pour les instances de processus au niveau des messages de l'application ;
- supporter la création et la destruction implicites d'instances et le cycle de vie basique des instances. Les opérations avancées, comme la pause et la reprise, pourront être implémentées dans des versions futures ;
- définir un modèle de transaction pour le long terme basé sur des techniques éprouvées ;
- utiliser les services Web en tant que modèle pour la décomposition et l'assemblage ;
- utiliser les standards des services Web autant que possible de manière réutilisable et modulaire.

2.5.1.2 Le langage

Une présentation détaillée du langage (syntaxe, structures de contrôle, etc.) est fournie dans le document intitulé « BPEL4WS, outils et validation » [BPEL4WS.ppt].

2.5.2 XPDL

2.5.2.1 Présentation

XPDL (XML Process Definition Language ou langage (dérivé du) XML de définition de processus), est un standard de la Workflow Management Coalition qui permet de définir un

processus métier ou processus d'affaires à l'aide du langage XML, processus métier qui sera ensuite utilisé par un moteur de workflow.

La définition d'un processus comporte les principaux éléments (balises) suivant(e)s :

- les marques de début et de fin du ou des processus ;
- les activités ;
- leurs inter-relations (les transitions) ;
- les attributs qualifiant certains comportements de l'activité ;
- les participants / rôles / groupes ;
- les interactions / relations entre les acteurs et les activités ;
- etc.

La définition ne comporte pas de façon native des attributs de positionnement (exemple : attributs (X, Y) d'une activité du diagramme représentant le processus) mais inclut la notion d'attributs étendus (la balise Extended Attribute) pour la plupart des composants. Certains éditeurs XPDL s'en servent pour mettre des attributs de positionnement.

XPDL 1.0 a été annoncé en décembre 2002.

XPDL 2.0 a été annoncé le 3 octobre 2005.

2.5.2.2 Outils

Nous avons trouvé deux éditeurs XPDL : Jawe et Jped (cf webographie pour plus d'informations).

2.5.2.3 Exemple

Extraits du standard XPDL (XML Process Definition Language)

```
<xsd :element name= " Task Receive">
  < xsd:annotation >
    <xsd:documentation>BPMN</xsd:documentation>
  </xsd :annotation>
  <xsd :complextyp>
    <xsd:sequence>
      <xsd:element name="Message type"="xpd:Message Type"min
occurs="0">
```

2.5.3 BPMN

BPMN est un langage graphique (notation par éléments graphiques et diagrammes), utilisée pour représenter un processus métier en séparant les informations métier des informations techniques. Il fournit une correspondance vers des langages d'exécution. C'est l'équivalent d'UML appliqué à la gestion des processus. Une modélisation basée sur BPMN peut ensuite être traduite en BPML ou en BPEL4WS.

Cette notation a été créée par le BPMI (Business Process Management Initiative), un consortium d'entreprises créé à l'initiative d'Intalio qui rassemble les leaders du marché de la modélisation de processus métier. Il s'est affirmé en innovateur depuis quelques années sur le

Les standards d'une architecture Web-Services

BPM. Membre de l'OASIS, OMG, W3C et WfMC, son premier objectif est d'établir une notation compréhensible par les utilisateurs, des analystes aux développeurs, en passant par les acteurs qui gèrent les processus.

BPMN définit des diagrammes de processus, appelés « flowcharts », c'est-à-dire des modèles graphiques où s'enchaînent des activités et des indicateurs.

2.6 SECURITE

2.6.1 WS-SECURITY

WS-Security est un protocole de communications qui permet d'appliquer de la sécurité aux services web. Il est né du besoin de faire remonter la gestion de la sécurité, auparavant gérée par la couche transport (HTTPS), vers les couches supérieures.

Développé à l'origine par IBM, Microsoft, VeriSign et Forum Systems, le protocole est maintenant officiellement appelé WSS et est développé via un comité dans Oasis-Open.

Le protocole contient des spécifications sur la façon dont l'intégrité et la confidentialité peuvent être appliquées aux messages de services web. Le protocole WSS inclut des détails sur l'utilisation de SAML et Kerberos, et des formats de certificat comme X.509.

WS-Security est en version 1.1 depuis le 17 février 2006. La version 1.0 avait été validée par le consortium OASIS-Open le 19 avril 2004.

WS-Security est une route à deux voies. Quand un serveur de services Web destinataire a traité les parties de l'entête de WS-Security, ce destinataire peut décrypter le message lui-même et le traiter comme il le ferait normalement. La réponse au client utilise la même sémantique WS-Security, et donc le client jouit des mêmes garanties et protections que le serveur.

2.7 SCHEMA D'ARTICULATION DES DIFFERENTS STANDARDS D'UNE ARCHITECTURE WEB-SERVICES

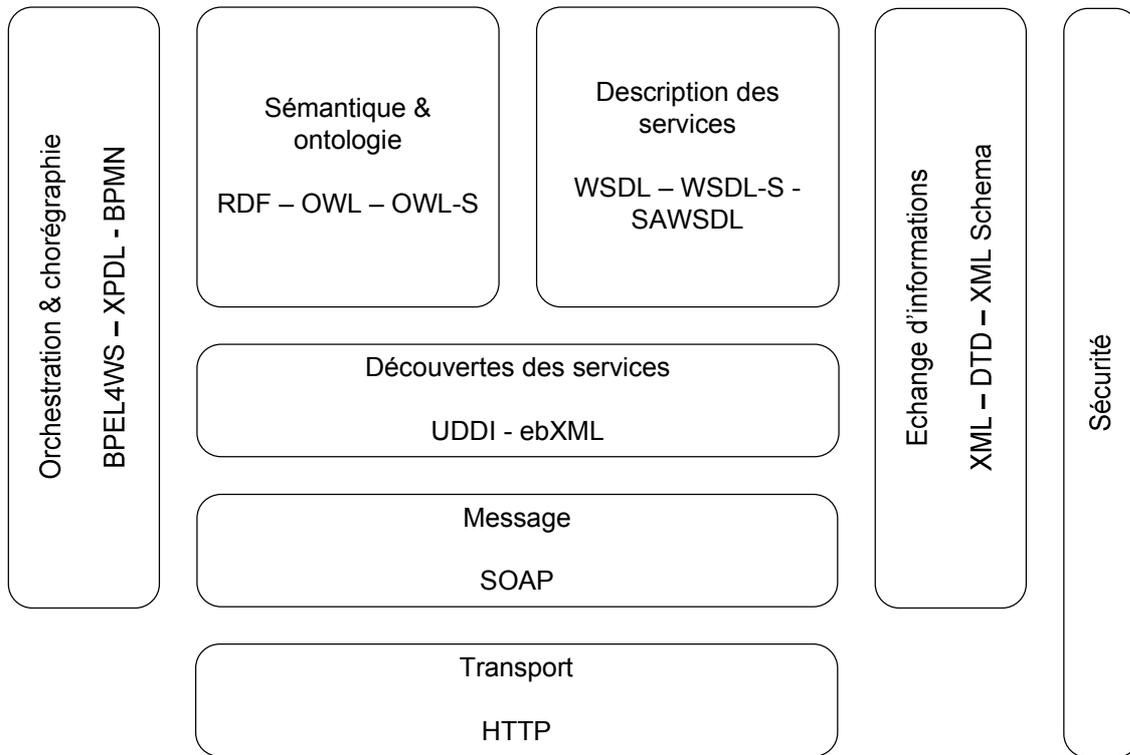


Figure 3 : Représentation graphique de la technologie des Services Webs

3 WEBSPHERE VS GLASSFISH : ETUDE COMPARATIVE DE DEUX SERVEURS D'APPLICATION

L'architecture des services web est basée, dans son principe même, sur des relations client/serveur. Si le client demeure bien un navigateur internet ou un client léger indépendant du système d'exploitation, le serveur constitue le cœur névralgique de l'application sur laquelle le projet Discel repose.

Les deux principaux serveurs d'applications disponibles répondant à des exigences de performance et de haute disponibilité sont d'une part GlassFish, un produit *open source* fourni par Sun Microsystems, et d'autre part WebSphere Application Server (WAS), un produit commercial fourni par IBM.

GlassFish est réputé pour offrir de meilleures performances. L'avantage de sa gratuité est indéniable face à des produits commerciaux très chers. Cependant, le rachat récent de Sun Microsystems par Oracle pourrait bien compromettre l'avenir de Glassfish, puisque ce dernier commercialise un produit concurrent, Weblogic.

Nous avons eu la chance d'avoir accès à un serveur IBM WebSphere durant notre veille. Nous avons pris des captures d'écrans des écrans de configuration ainsi que de la console d'administration du serveur. Ces captures se trouvent en annexes de ce document (fig 5 à 9). Le produit d'IBM est riche en fonctionnalités et est largement reconnu pour sa maturité. (voir lien en annexes).

Extraits du document « business process choreography in websphere » :

BPEL+ business process are developed with websphere studio. The user develops J2EE components with a java editor. The J2EE components can then be offered as web services and composed into business processes.

For each "invoke, receive, reply, and pick-on Message activity, the process modeler specifies which partner link is associated with the activity. The name of each partner link is later associated with a particular web service or java end point.

Business process data are represented by BPEL or BPEL+ variables, which can be defined with WSDL messages, XML schema types or java types. After the process variables have been defined, the BPEL activities can use the variables, for example as request or response messages of invoke activities.

Exemple très proche du concept « Document Intelligent » :

Le « Document intelligent » permet au concepteur/utilisateur de couvrir la représentation syntaxique et sémantique ainsi que les modalités d'échange de documents électroniques.

Figure 4 Correlation set representation—A. WSDL example and B. BPEL example

```

1 <bpws:property name="confirmationNumber" type="xsd:integer"/>
2 <bpws:propertyAlias propertyName="travel:confirmationNumber"
  messageType="travel:bookOutputMessage"
  part="confirmationNo" query=""/>
3 <bpws:propertyAlias propertyName="travel:confirmationNumber"
  messageType="travel:cancelInputMessage"
  part="confirmationNo" query=""/>
4 <process name="TravelBookingProcess"...>
  ...
  <correlationSets>
5    <correlationSet name="TravelBooking"
      properties="travel:confirmationNumber"/>
  </correlationSets>
  <flow>
  ...
6    <reply partnerLink="traveller"
      portType="travel:TravelAgentPT" operation="book"
      variable="output">
7      <correlations>
        <correlation set="TravelBooking" initiate="yes"/>
      </correlations>
    </reply>
  ...
8    <receive partnerLink="traveller"
      portType="travel:TravelAgentPT" operation="cancel"
      createInstance="no" variable="cancelInput">
9      <correlations>
        <correlation set="TravelBooking" initiate="no"/>
      </correlations>
    </receive>
  ...
  </flow>
</process>

```

Figure 4 : Exemple d'intégration de WSDL et BPEL au sein de WebSphere

4 SYNTHESE : LES BASES DE DI-SC-EL

Les standards existants à ce jour permettent de décrire des processus au sein d'une architecture orientée service (SOA). La description de processus, telle qu'elle est rendue possible au travers des outils BPEL+, se définit comme un assemblage d'éléments de taille modeste, en un plus conséquent.

Une des premières difficultés de SPDF sera de définir suffisamment de composants que l'on qualifiera de « services élémentaires ». La définition de processus devra résulter de l'assemblage de deux composantes : des « services élémentaires » et des processus utilisateur. Cette granularité de gestion autorise l'utilisateur métier à gérer la modularité de son document intelligent et offre la souplesse nécessaire à une utilisation plus large de SPDF.

L'ensemble de cette architecture et standards forme un système de Web-Services, basé sur le langage XML. Ce langage est identifié comme un pré-requis technique indispensable à tout développement de SPDF. Le second est que les spécifications devront s'appuyer sur des schémas XML. Ces deux pré-requis sont les garanties de la cohésion et de la bonne utilisation du standard.

L'architecture technique sur laquelle reposera SPDF ne semble pas avoir d'incidence sur ses spécifications, au-delà de l'utilisation de XML : les standards sont suffisamment génériques pour s'adapter à tout type d'environnement technique. Le choix du serveur d'application, en l'occurrence GlassFish ou IBM WebSphere n'est donc pas déterminant de même que le protocole de transmission des messages vers ce serveur. Au dessus de la couche transport, ce sont les standards des Web Services qui prennent le relais.

Toute l'intelligence sur laquelle repose le document d'intelligent est basée sur les annotations, la sémantique et, plus globalement, les ontologies. A ce niveau, si la description des services web est en mesure de gérer cette information sémantique qu'il s'agisse de WSDL-S ou SAWSDL, la couche de description de processus ne semble pas prendre en compte cette fonctionnalité. L'intérêt de SPDF se trouve peut-être dans ce fossé de l'architecture.

SPDF offrira une vision métier des processus d'entreprises. Le concepteur/utilisateur sera en mesure de réaliser une application, d'un point de vue technique, par un assemblage de composant métier de haut niveau. SPDF interprétera ce processus en une somme de composants métiers agissant ensembles au travers de l'intelligence et de la sémantique effectuées dans la description de ces services.

Le standard doit être en mesure de juger de façon dynamique le meilleur service web à appeler pour mener à bien le processus de bout en bout. Ce choix s'opère sans intervention de la part de l'utilisateur final, et c'est là toute la valeur ajoutée de SPDF : vulgariser la création d'applications métiers riches sans connaissances poussées en programmation.

Les machines un jour pourront résoudre tous les problèmes, mais jamais aucune d'entre elles ne pourra en poser un ! Albert Einstein

5 ANNEXES

5.1 WEBOGRAPHIE

Description	Url
Liste des standards du W3C	http://www.w3.org/TR/tr-technology-stds.html
Wikipédia - XML	http://fr.wikipedia.org/wiki/Extensible_markup_language
Wikipédia - DTD	http://fr.wikipedia.org/wiki/Document_Type_Definition
Wikipédia – XML Schema	http://fr.wikipedia.org/wiki/XML_Schema
W3C – Spécifications XML	http://www.w3.org/TR/xml/
W3C – Spécifications XML Schema	http://www.w3.org/TR/xmlschema-2/
Liste des composantes des Web Services selon IBM	http://www.ibm.com/developerworks/webservices/standards/
Liste des composantes des Web Services selon Wikipedia	http://fr.wikipedia.org/wiki/Liste_des_spécifications_des_services_web_WS-*
Wikipédia – SOAP	http://fr.wikipedia.org/wiki/SOAP
Wikipédia – HTTP	http://fr.wikipedia.org/wiki/HTTP
W3C – SOAP	http://www.w3.org/TR/soap12-part1/
OASIS – Spécifications UDDI	http://www.oasis-open.org/committees/uddi-spec/doc/spec/v3/uddi-v3.0.2-20041019.htm
Wikipédia – UDDI	http://fr.wikipedia.org/wiki/UDDI
Site communauté UDDI	http://uddi.xml.org/
Wikipédia – ebXML	http://fr.wikipedia.org/wiki/EbXML
Wikipédia – Ancêtre de ebXML	http://fr.wikipedia.org/wiki/EDIFACT
Présentation de WSDL-S	http://lstdis.cs.uga.edu/projects/meteor-s/wsdl-s/WSDL-S.pdf
W3C – OWL-S	http://www.w3.org/TR/2009/REC-owl2-rdf-based-semantics-

	20091027/
Wikipédia – RDF	http://fr.wikipedia.org/wiki/Resource_Description_Framework
W3C – RDF	http://www.w3.org/TR/rdf-syntax-grammar/
W3C – SAWSDL	http://www.w3.org/TR/sawSDL/
Thlas – Démarche analogue à SPDF	http://www.chatelp.org/thesis/jfo2007_presentation.pdf
Wikipédia – WSDL	http://fr.wikipedia.org/wiki/Web_Services_Description_Language
Spécifications BPEL4WS	http://www.ibm.com/developerworks/library/specification/ws-bpel/
BPEL4WS sur Wikipedia	http://en.wikipedia.org/wiki/Business_Process_Execution_Language
XLANG, WSFL, BPEL4WS, XPD	http://www.guideinformatique.com/fiche-orchestration_processus_xlang_wsfl_bpel4ws_xpd-380.htm
BPEL4WS, outils et validation	http://www.latece.uqam.ca/media/presentation_vf.ppt
XPDL sur Wikipedia	http://fr.wikipedia.org/wiki/XML_Process_Definition_Language
Jawe, éditeur XPDL	http://www.enhydra.org/workflow/jawe/index.html
JPED, éditeur XPDL basé sur Jawe	http://www.jpel.org/
WS-SECURITY : rôle et fonctionnement	http://systemi.itpro.fr/Dossiers-par-Theme/2007/7/16/020312872-WS-SECURITY-:-role-et-fonctionnement.htm
Article Wikipedia sur WS-Security	http://fr.wikipedia.org/wiki/WS-Security
Comparatif serveurs d'applications	http://www.journaldunet.com/developpeur/outils/actualite/avec-un-indice-de-410-websphere-se-place-en-tete-du-classement-evans-data.shtml
Fusion Oracle – Sun	http://www.dng-consulting.com/blogs/index.php/2009/04/20/oracle-rachaegrave-te-sun-queelles-consae?blog=1

5.2 ANNEXE : ILLUSTRATION DE LA CONFIGURATION DE WEBSPHERE

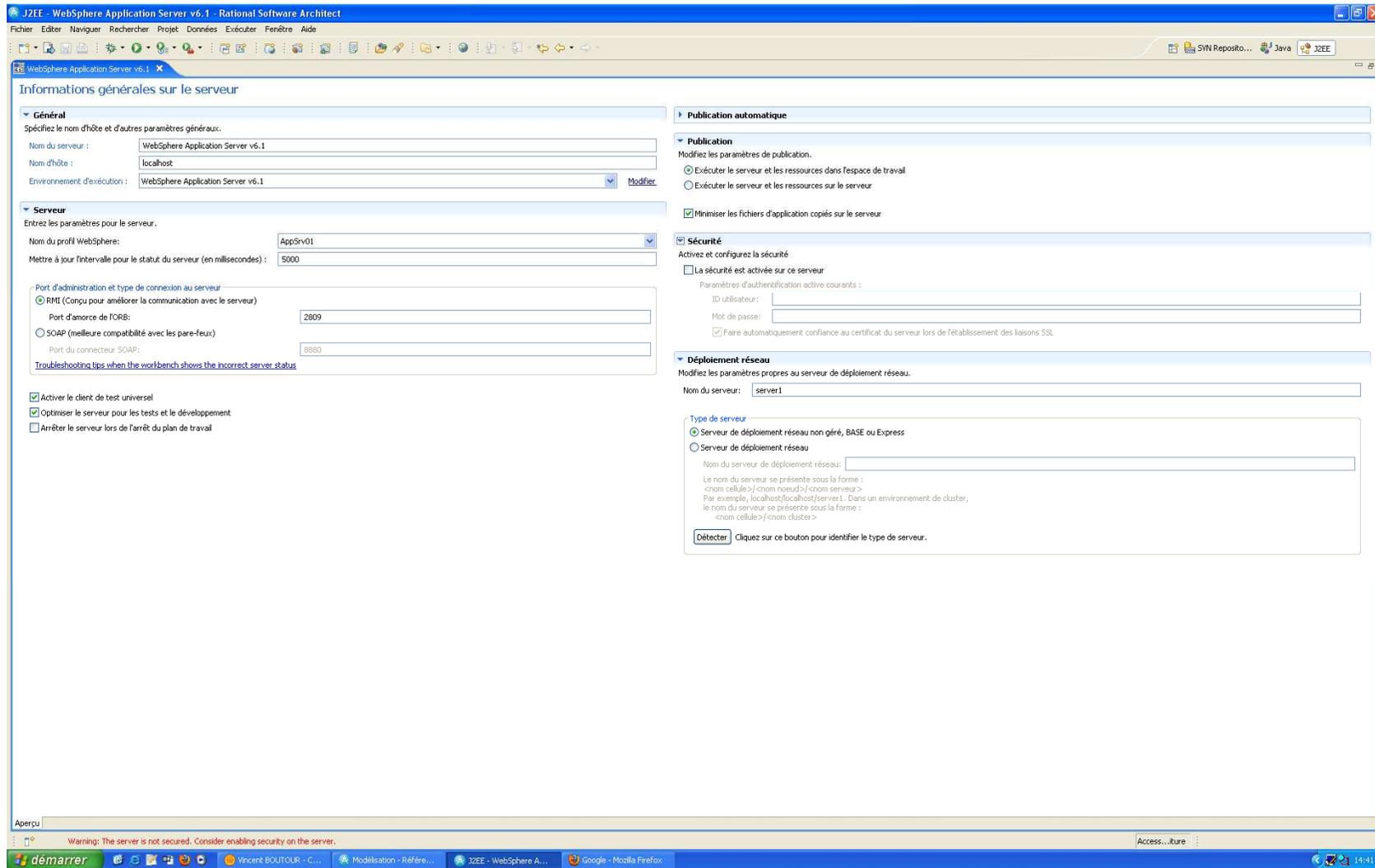


Figure 5 : Configuration du serveur WAS

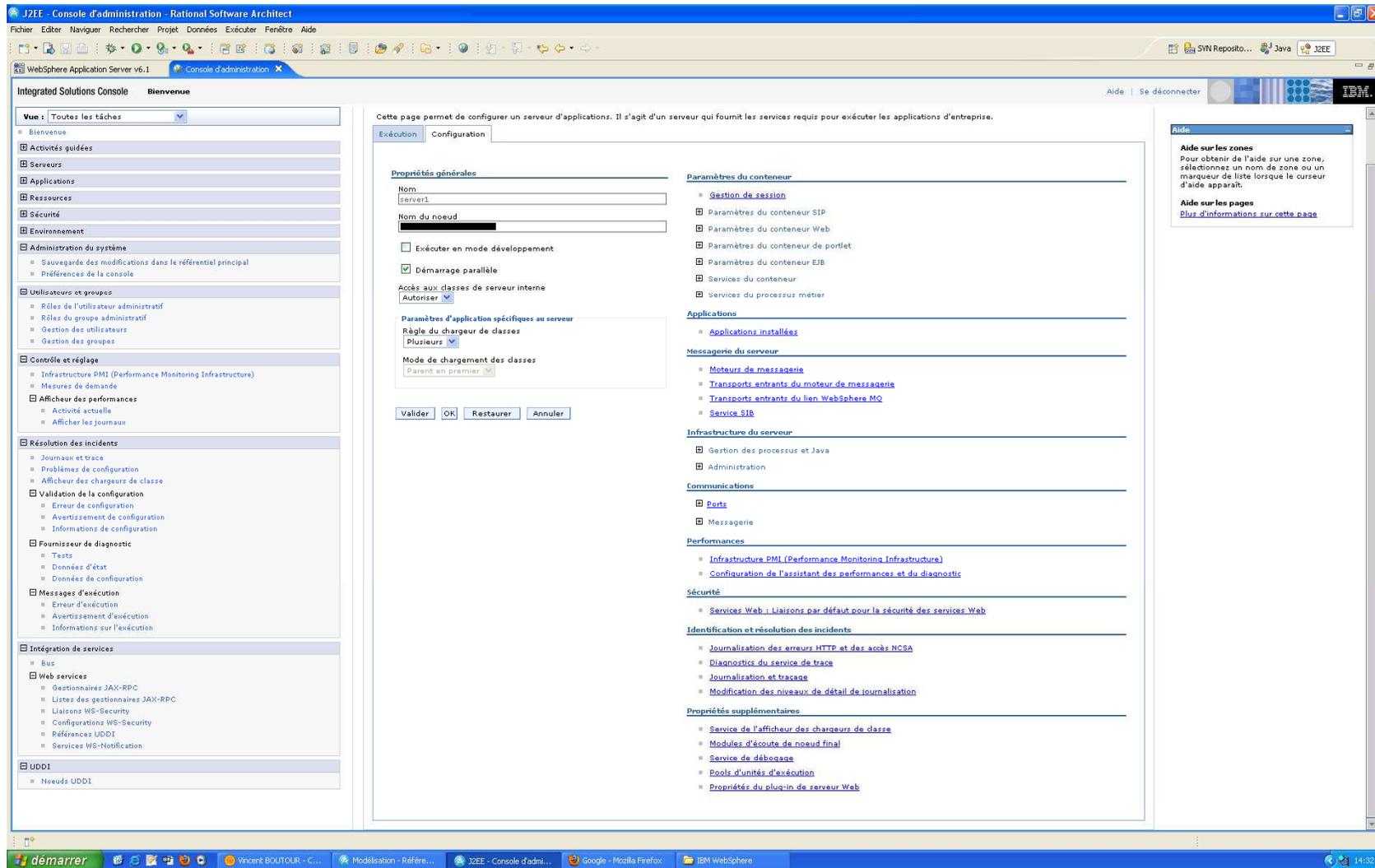


Figure 6 : Console d'administration de WebSphere

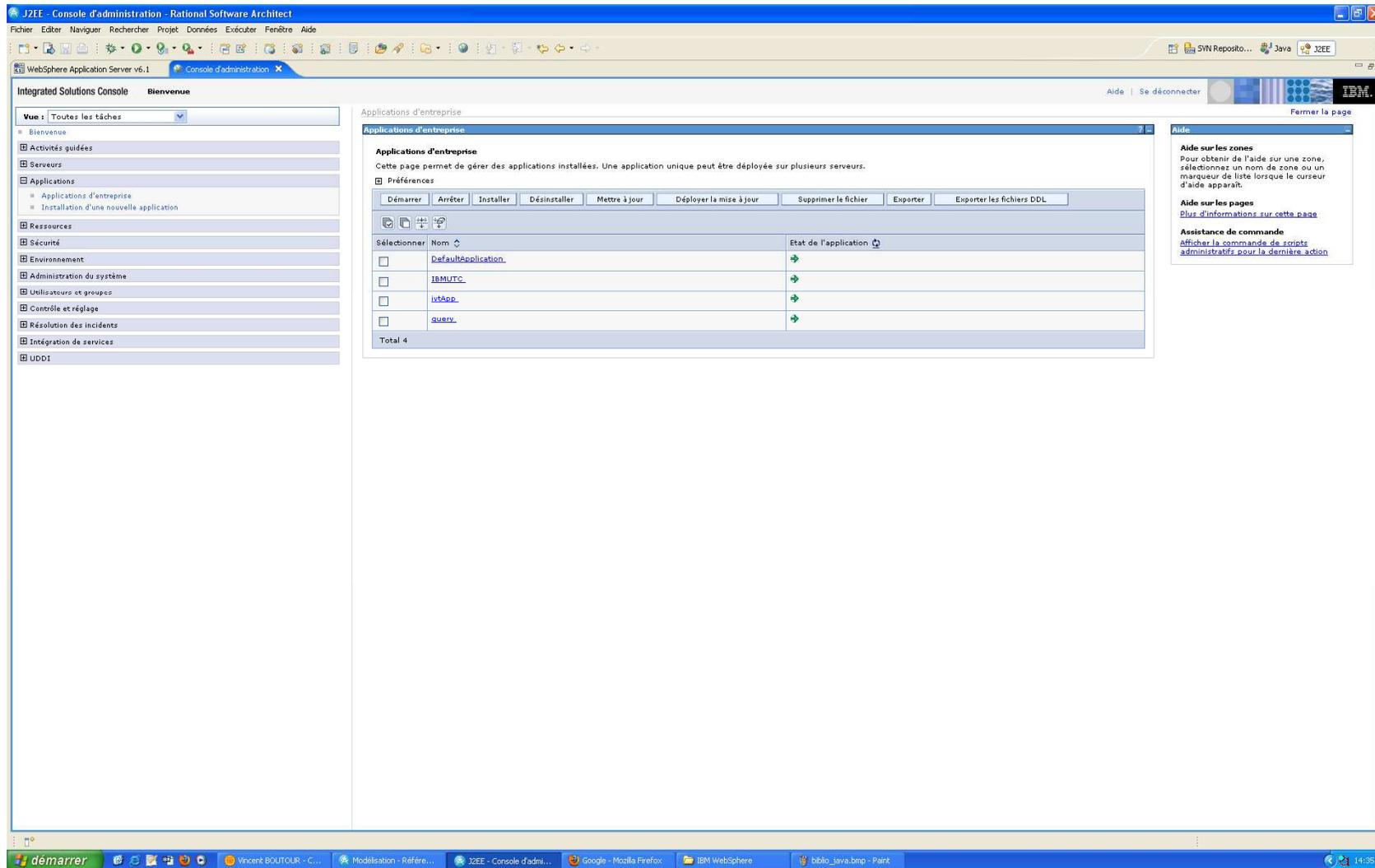


Figure 7 : Administration d'applications sous WebSphere

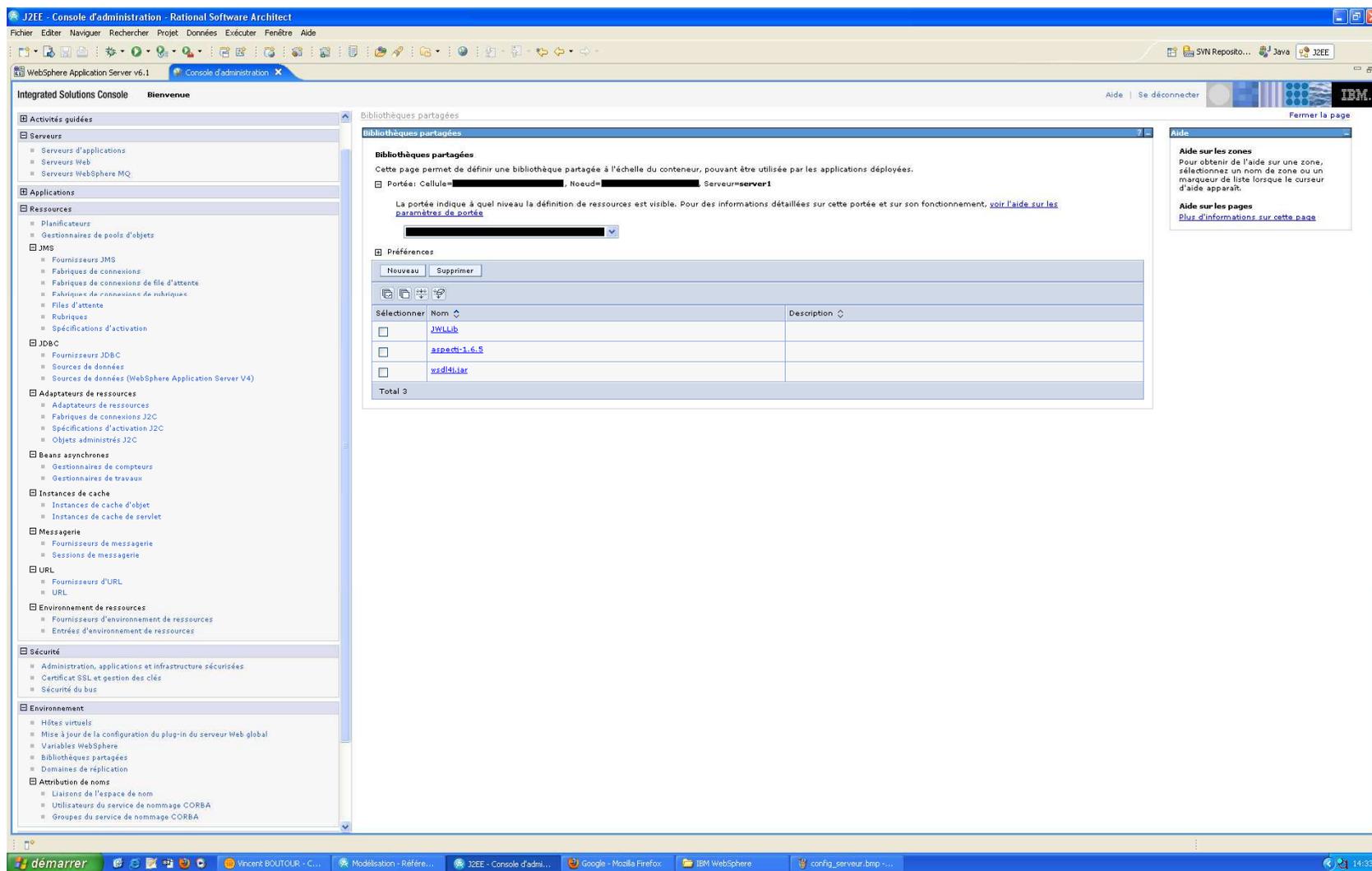


Figure 8 : Gestion des bibliothèques partagées

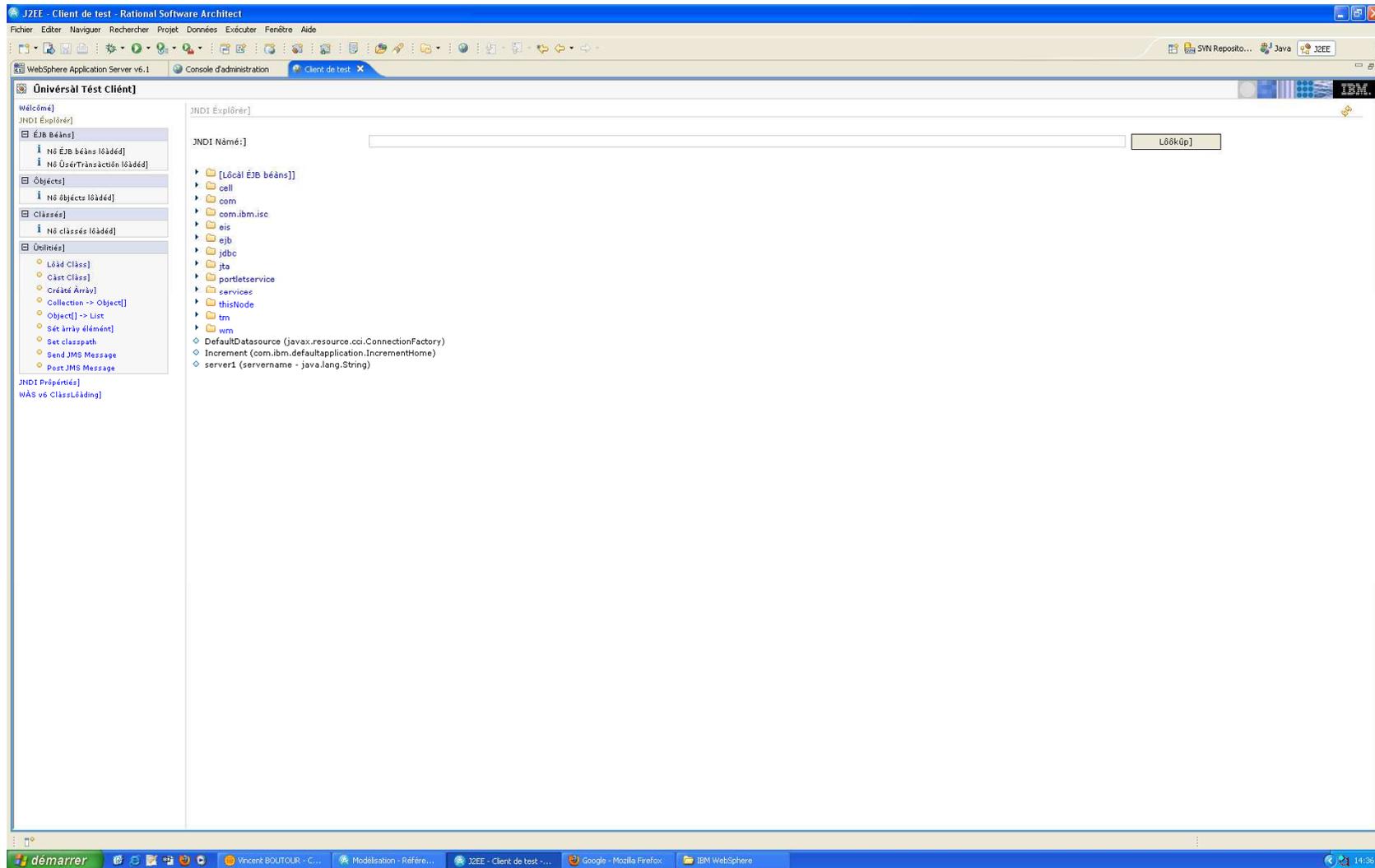


Figure 9 : Client de test universel WebSphere

5.3 TABLE DES ILLUSTRATIONS

Figure 1 : Structuration WSDL 2.0 © W3C	8
Figure 2 : Extrait du standard WSDL-S	10
Figure 3 : Représentation graphique de la technologie des Services Webs	14
Figure 4 : Exemple d'intégration de WSDL et BPEL au sein de WebSphere	16
Figure 5 : Configuration du serveur WAS	20
Figure 6 : Console d'administration de WebSphere	21
Figure 7 : Administration d'applications sous WebSphere.....	22
Figure 8 : Gestion des bibliothèques partagées	23
Figure 9 : Client de test universel WebSphere	24