

TF 8.0 - Webservice Mapping

Type diffusion
Révision 4

Auteur *Grégory PRINCE*
Date de création *21/09/2012*



L'information contenue dans ce document est fournie sans garantie d'aucune sorte, explicite ou implicite. Elle est fournie à titre éducatif et devra être relue et validée en environnement de test avant utilisation en production. L'utilisateur doit s'assurer du caractère approprié du contenu de ce document à son usage particulier, et assume le risque de son utilisation.

Société	:		Titre	:	<i>Guide de mise en oeuvre</i>
Dernière modification	:	<i>Grégory PRINCE</i>	Sujet	:	<i>TF 8.0 - Webservice Mapping</i>
Temps de modification	:	<i>59</i>	Résumé	:	
Nom du fichier		<i>GMO Webservice Mapping.docx</i>			

Liste de diffusion

Siège Social et Agence Nord
Immeuble le Verdi
6 rue du Moulin de Lezennes
BP 10215
59654 VILLENEUVE D'ASCQ Cedex
Tél. : +33 (0)3 20 41 48 00
Fax. : +33 (0)3 20 41 48 09
Service Technique :
Tél. : +33 (0)3 20 41 48 07
Fax. : +33 (0)3 20 41 48 08

Agence Région Parisienne

69/71 rue Beaubourg
75003 PARIS
Tél. : +33 (0)1 47 49 36 66
Fax. : +33 (0)1 47 51 22 17

Table des matières

1	Objectif	3
2	Principe de fonctionnement	4
3	Mise en place	5
3.1	Partie applicative GCE	5
3.1.1	configuration.xml	5
3.1.2	index.html	6
3.2	Partie applicative spécifique	7
3.2.1	classe Java d'implémentation	7
3.2.2	wSDL	8
3.3	Partie technique	9
3.3.1	sun-jaxws.xml	9
3.3.2	web.xml	9
3.4	Intégration	9
4	Annexes	10
4.1	Partie applicative GCE	10
4.1.1	configuration.xml	10
4.2	Partie application spécifique	10
4.2.1	classe Java d'implémentation	10
4.2.2	wSDL	11
4.3	Partie technique	12
4.3.1	sun-jaxws.xml	12
4.3.2	web.xml	12
4.4	Intégration	13

1 Objectif

Pour intégrer GCE dans un système d'information, plusieurs connecteurs sont possibles, en particulier celui des WebServices. A ce titre, GCE expose des webservices « standards ». Cela veut dire que les flux d'entrée/sortie sont généralistes, voir abstraits (cf CrudManager).

Nos clients ont des besoins spécifiques pour les appels à ces WS : valeurs par défaut à passer systématiquement, nom des champs, nom de service et noms d'opération exposés.

1er exemple : un client voudrait pouvoir, pour créer une facture, contacter un WS « gestionFacture » et une opération « creerFacture » ; dont l'implémentation GCE serait finalement d'invoquer le WS GCE gceService, opération assimilate.

Il veut spécifier SON flux d'entrée (les seules informations pertinentes pour lui avec ces propres noms) ; idem pour le flux de sortie attendu. Masquant ainsi la généralité des flux d'entrée/sortie des WS standard.

2ème exemple : une opération spécifique « mettreAJourContrat » pourrait techniquement revenir à invoquer une BusinessView. Virtuellement, ça revient à exposer une BusinessView sous forme de WS (utilisation du WS GCE BusinessViewService, opération execute)

Nous appellerons ce concept : mapping de webservice.

L'objectif de ce GMO est d'aider à la mise en œuvre de « webservice mapping ».

2 Principe de fonctionnement

La spécialisation de WS revient à offrir une façade personnalisée à un WS (valeurs par défaut positionnées de façon masquée, nom d'opération personnalisé, ...) ; chacune des opérations spécifiques pouvant pointer vers des WS différents.

La notion ainsi enrichie va s'appuyer sur les ClientWS ; aussi bien pour les WS GCE, que les WS externes à GCE.

Remarque : Un point d'entrée dans le module services de GCE pourra ainsi déléguer son traitement vers un WS externe à GCE pour une opération et/ou invoquer une opération standard d'un WS GCE.

Dans ce 2ème cas, la JVM « consommatrice » du WS spécifique (ou WSMapping) ne sera pas forcément la JVM « consommatrice » du WS GCE (ou le sera selon l'url spécifié) ; ceci apporte de la souplesse par rapport à la répartition de charge.

3 Mise en place

Cette nouveauté est disponible à partir du socle 8.0 ; soit en GCE161 ainsi que sur les futures versions.

3.1 Partie applicative GCE

3.1.1 configuration.xml

Spécialisation d'un WS GCE dans la configuration :

nœud `config/business/webservice/webservice_mapping_def`

Il convient de définir :

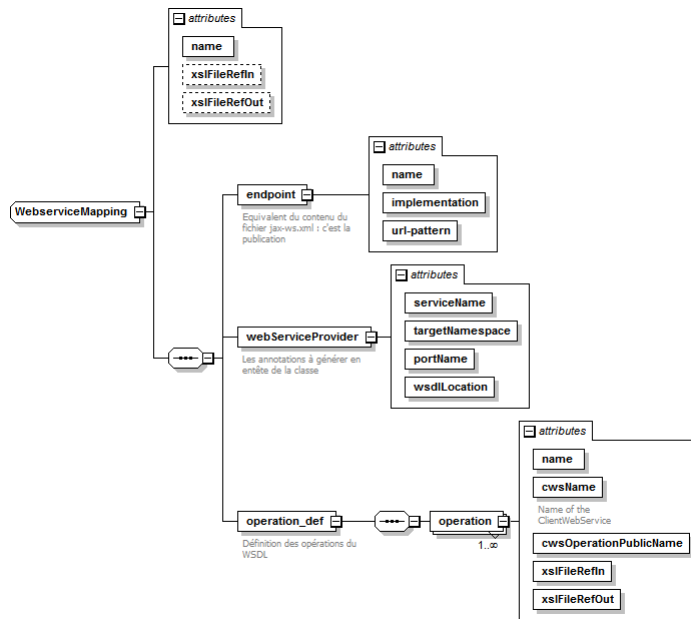
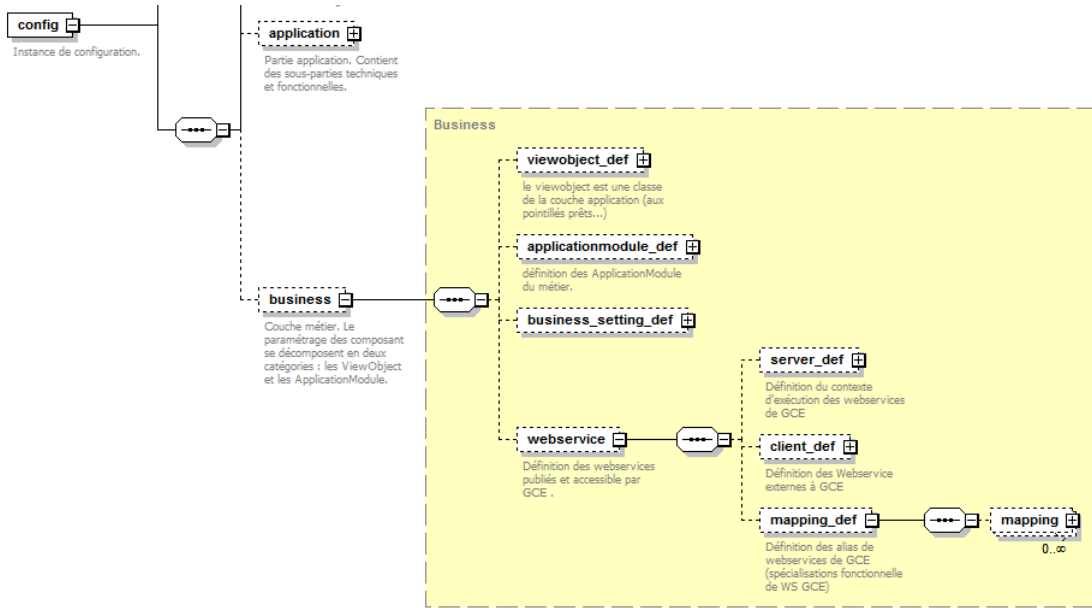
- le nom GCE du mapping
- l'url du fichier XSL permettant une transformation du flux d'entrée de WS « mapping » (permet par exemple de positionner un 1^{er} niveau de valeur par défaut, effectuer des calculs globaux, ...)
- l'url du fichier XSL permettant une transformation du flux de sortie de WS « mapping » ; c'est le « pendant » de la transformation global pour l'entrée

- Les attributs nécessaires pour une éventuelle génération, dans le futur, des descripteurs de déploiement « sun-jaxws.xml » et « web.xml » : nœud « endpoint »
- Les attributs nécessaires une éventuelle génération, dans le futur, de la classe JAVA d'implémentation du WSMapping : nœud « webServiceProvider »

- Pour chaque opération spécifique, lien avec le WS GCE :
 - o Nom de l'opération spécifique
 - o Nom du ClientWS GCE et l'opération à déclencher sur le WS GCE
 - o L'url du fichier XSL permettant la transformation du flux d'entrée du WS « mapping » en flux d'entrée du WS GCE
 - o L'url du fichier XSL permettant la transformation du flux de sortie du WS GCE en flux de sortie du WS « mapping »

Remarque : les fichiers XSL de transformation prennent en charge toute l'enveloppe SOAP (y compris SOAPFault). Ils sont intégrés dans le jar « spécifique ».

Schémas :



3.1.2 index.html

Accessoirement, mettre à jour ce fichier référençant tous les WS

3.2 Partie applicative spécifique

Ci-dessous les éléments générés (ou à la main) correspondant au WSDL spécifique

3.2.1 classe Java d'implémentation

Il s'agit ici d'un exemple d'implémentation. Rien n'empêche d'y mettre le code de son choix.

Publié sur l'URL spécifiée dans le WSDL, le WS spécifique implémente la méthode abstraite `public SOAPMessage invoke(SOAPMessage soapMessage)` de la classe abstraite `fr.generix.technicalframework.business.webservice.mapping.AbstractGenericGXWebservice` en appelant la méthode :

```
SOAPMessage
fr.generix.technicalframework.business.webservice.mapping.AbstractGenericGXWebservice.executeWS(String wsMappingName, SOAPMessage request)
```

Celle-ci réalise :

- Récupération du contenu du message au format DOM depuis `soapMessage`
- Si paramétrée, application de la feuille XSL global de transformation du WS spécifique : `<conf:business.webservice.mapping_def.mapping@xslFileRefIn>`
- Application de la feuille XSL de transformation pour l'opération du WS spécifique : `<conf.mapping.operation_def.operation@xslFileRefIn>`
- Récupération dans la configuration du ClientWebService à invoquer ainsi que l'opération : `<conf:business.webservice.clien_def.clien>`
- Invocation du WS Client sur base du flux SOAP transformé
- Application de la feuille XSL de transformation pour l'opération du WS spécifique sur la sortie : `<conf.mapping.operation_def.operation@xslFileRefOut>`
- Si paramétrée, application de la feuille XSL de transformation du WS spécifique sur la sortie : `<conf:mapping@xslFileRefOut>`
- Création d'un objet SOAPMessage de retour à partir de ce contenu transformé

Ci-dessous le « template » d'implémentation :

```
/**
 * oOo GENERATED BY GENERIX. oOo
 * Ce fichier est généré par XDME
 * Ne pas modifier manuellement.
 * Ce fichier ne doit pas être poussé dans SVN. Utiliser svn:ignore pour l'ignorer.
 */
package <Package (conf:mapping.endpoint@implementation)>;

import javax.xml.soap.SOAPMessage;
import javax.xml.ws.Service.Mode;
import javax.xml.ws.ServiceMode;
import javax.xml.ws.WebServiceProvider;

import fr.generix.technicalframework.business.webservice.mapping.AbstractGenericGXWebservice;

@WebServiceProvider(serviceName = "<conf:mapping.webServiceProvider@serviceName>", targetNamespace =
"<conf:mapping.webServiceProvider@targetNamespace>", portName = "<conf:mapping.webServiceProvider@portName>", wsdlLocation =
"<conf:mapping.webServiceProvider@wsdlLocation>")
@ServiceMode (Mode.MESSAGE)
public class <ClassName (conf:mapping.endpoint.implementation)> extends AbstractGenericGXWebservice
{
    @Override
    public SOAPMessage invoke(SOAPMessage soapMessage)
```

```

{
    return executeWS("<conf:mapping@name>", soapMessage);
}

```

3.2.2 wsdl

Il s'agit ici du contrat de service du WS : ces opérations, et pour chacune d'elles, définition des paramètres d'entrée et de sortie.

Ci-dessous le « template » :

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- oOo GENERATED BY GENERIX. oOo -->
<definitions name="<conf:mapping.webServiceProvider@serviceName>"
targetNamespace="<conf:mapping.webServiceProvider@targetNamespace>"
xmlns:tns="<conf:mapping.webServiceProvider@targetNamespace>" xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
  <types>
    <xsd:schema targetNamespace="<conf:mapping.webServiceProvider@targetNamespace>"
      <xsd:include schemaLocation="xsd/<conf:mapping.webServiceProvider@serviceName>.xsd"/>
    « Pour chaque operation de <conf:mapping.operation_def.operation >
      <xsd:element name="<operation@name>" type="tns:<JavaType (operation@name)>"/>
      <xsd:element name="<operation@name>Response" type="tns:<JavaType (operation@name)>Response"/>
    « Fin chaque operation »
  </xsd:schema>
</types>
« Pour chaque operation de <conf:mapping.operation_def.operation >
  <message name="<operation@name>">
    <part name="parameters" element="tns:<operation@name>"/>
  </message>
  <message name="<operation@name>Response">
    <part name="parameters" element="tns:<operation@name>Response"/>
  </message>
« Fin chaque operation »
  <portType name="<ClassName (conf:mapping.endpoint@implementation)>">
    « Pour chaque operation de <conf:mapping.operation_def.operation >
      <operation name="<operation@name>">
        <input message="tns:<operation@name>"/>
        <output message="tns:<operation@name>Response"/>
      </operation>
    « Fin chaque operation »
  </portType>
  <binding name="<conf:mapping.webServiceProvider@portName>Binding"
type="tns:<ClassName (conf:mapping.endpoint@implementation)>"
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  « Pour chaque operation de <conf:mapping.operation_def.operation >
    <operation name="<operation@name>">
      <soap:operation soapAction=""/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
  « Fin chaque operation »
</binding>
  <service name="<conf:mapping.webServiceProvider@serviceName>"
  <port name="<conf:mapping.webServiceProvider@portName>"
binding="tns:<conf:mapping.webServiceProvider@portName>Binding">
    <soap:address location="REPLACE_WITH_ACTUAL_URL"/>
  </port>
</service>
</definitions>

```

Les type « In » et « Out » sont externalisés dans un schéma à part et au final, seul ce schéma provient du client/projet.

Ci-dessous ces caractéristiques :

- Le nom de ce schéma est « WEB-INF/wsdl/xsd/<conf:mapping.webServiceProvider@serviceName>.xsd », xmlns et targetNamespace valent « <conf:mapping.webServiceProvider@targetNamespace> »
- Il contient 2 types « complex » par opération :
 - o « <JavaType(operation@name)> » : Notation Java (1^{ère} lettre en majuscule) de l'opération
 - o « <JavaType(operation@name)>Response » schéma

3.3 Partie technique

3.3.1 sun-jaxws.xml

C'est le fichier de paramétrage de JAXWS ; localisé dans <webmodule>/WEB-INF.
A chaque service (WS) est défini un « endpoint » ; à savoir : un nom, une classe d'implémentation et une url d'invocation.

Il convient d'ajouter aux services standard GCE les spécifiques (WSMapping).

Ci-dessous le « template » :

```
<!-- Les spécifiques -->
« Pour chaque mapping de <conf.mapping_def.mapping >
  <endpoint name="<mapping.endpoint@name>" implementation="<mapping.endpoint@implementation>" url-
  pattern="<mapping.endpoint@url-pattern>"/>
« Fin chaque mapping »
```

3.3.2 web.xml

C'est le descripteur de déploiement d'une application web ; localisé dans <webmodule>/WEB-INF. Il contient les caractéristiques et paramètres de l'application. Cela inclut la description des servlets utilisées, ou les différents paramètres d'initialisation.

Il convient d'ajouter aux mapping standard de la servlet « jaxws-servlet » les mapping d'url pour les services spécifiques (les mêmes que dans sun-jaxws.xml).

Ci-dessous le « template » :

```
<!-- Les spécifiques -->
« Pour chaque mapping de <conf.mapping_def.mapping >
  <servlet-mapping>
    <servlet-name>jaxws-servlet</servlet-name>
    <url-pattern><mapping.endpoint@url-pattern></url-pattern>
  </servlet-mapping>
« Fin chaque mapping »
```

3.4 Intégration

L'intégration passe par le Webmodule service.

Il s'agit donc de modifier le war :

- Modification des fichiers de déploiements :
 - WEB-INF/web.xml,
 - WEB-INF/sun-jaxws.xml
- Ajout des wsdl et les éventuels schémas
 - WEB-INF/wsdl
- Ajout d'un jar spécifique comprenant les feuilles de style de transformation et les classes des ws
 - WEB-INF/lib

4 Annexes

Ici les exemples d'utilisation du chapitre 3

4.1 Partie applicative GCE

4.1.1 configuration.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<config xmlns="http://www.generix.fr/technicalframework/configuration" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.generix.fr/technicalframework/configuration ../configuration.xsd">
  <version/>
  <business>
    <webservice>
      <client_def>
        <client name="gCEServiceLiteral" serviceName="GCEServiceLiteral" targetNamespace="urn:GCE"
servicePortName="WSGCEServiceLiteralPort">
          <operation_def>
            <operation name="assimilate" publicName="Assimilate"></operation>
          </operation_def>
        </client>
        <client name="crudManagerLiteral" serviceName="CRUDManagerLiteral" targetNamespace="urn:GCE"
servicePortName="WSCRUDManagerLiteralPort">
          <operation_def>
            <operation name="delete" publicName="Delete"></operation>
          </operation_def>
        </client>
        <client name="businessViewServiceLiteral" serviceName="BusinessViewServiceLiteral" targetNamespace="urn:GCE"
servicePortName="WSBusinessViewServiceLiteralPort">
          <operation_def>
            <operation name="execute" publicName="Execute"></operation>
          </operation_def>
        </client>
      </client_def>
      <mapping_def>
        <mapping name="WSFacture" xsiFileRefIn="xsl/WSFacture/gestionFactureIn.xsl"
xslFileRefOut="xsl/WSFacture/gestionFactureOut.xsl">
          <endpoint name="gestionFacture" implementation="com.surcouf.Facture" url-pattern="/facturation"/>
          <webServiceProvider serviceName="Facturation" targetNamespace="www.surcouf.com" portName="FacturationPort"
wsdlLocation="WEB-INF/wsdl/Facture.wsdl"/>
          <operation_def>
            <operation name="creerFacture" cwsName="gCEServiceLiteral" cwsOperationPublicName="Assimilate"
xslFileRefIn="xsl/WSFacture/gestionFacturecreerFactureIn_gCEServiceassimilateIn.xsl"
xslFileRefOut="xsl/WSFacture/gCEServiceassimilateOut_gestionFacturecreerFactureOut.xsl"/>
            <operation name="supprimerFacture" cwsName="crudManagerLiteral" cwsOperationPublicName="Delete"
xslFileRefIn="xsl/WSFacture/gestionFacturesupprimerFactureIn_crudManagerLiteraldeleteIn.xsl"
xslFileRefOut="xsl/WSFacture/crudManagerLiteraldeleteOut_gestionFacturesupprimerFactureOut.xsl"/>
          </operation_def>
        </mapping>
        <mapping name="WSAchat" xsiFileRefIn="xsl/WSAchat/gestionAchatIn.xsl"
xslFileRefOut="xsl/WSAchat/gestionAchatOut.xsl">
          <endpoint name="gestionAchat" implementation="org.tic.tac.AchatImpl" url-pattern="/achat"/>
          <webServiceProvider serviceName="Achat" targetNamespace="www.tac.tic.org" portName="AchatPort"
wsdlLocation="WEB-INF/wsdl/Achat.wsdl"/>
          <operation_def>
            <operation name="creer" cwsName="businessViewServiceLiteral" cwsOperationPublicName="Execute"
xslFileRefIn="xsl/WSAchat/gestionAchatcreerIn_businessViewServiceLiteralexecuteIn.xsl"
xslFileRefOut="xsl/WSAchat/businessViewServiceLiteralexecuteOut_gestionAchatcreerOut.xsl"/>
            <operation name="supprimer" cwsName="crudManagerLiteral" cwsOperationPublicName="Delete"
xslFileRefIn="xsl/WSAchat/gestionAchatsupprimerIn_crudManagerLiteraldeleteIn.xsl"
xslFileRefOut="xsl/WSAchat/crudManagerdeleteOut_gestionAchatsupprimerOut.xsl"/>
          </operation_def>
        </mapping>
      </mapping_def>
    </webservice>
  </business>
</config>
```

4.2 Partie application spécifique

4.2.1 classe Java d'implémentation

com.surcouf.Facture.java :

```
/**
 * oOo GENERATED BY GENERIX. oOo
 * Ce fichier est généré par XDME
 * Ne pas modifier manuellement.
 * Ce fichier ne doit pas être poussé dans SVN. Utiliser svn:ignore pour l'ignorer.
 */
package com.surcouf;

import javax.xml.soap.SOAPMessage;
```

```

import javax.xml.ws.Service.Mode;
import javax.xml.ws.ServiceMode;
import javax.xml.ws.WebServiceProvider;

import fr.generix.technicalframework.business.webservice.mapping.AbstractGenericGXWebservice;

@WebServiceProvider(serviceName = "Facturation", targetNamespace = "http://www.surcouf.com", portName = "FacturationPort",
wsdlLocation = "WEB-INF/wsdl/Facture.wsdl")
@ServiceMode (Mode.MESSAGE)
public class Facture extends AbstractGenericGXWebservice
{
    @Override
    public SOAPMessage invoke(SOAPMessage soapMessage)
    {
        return executeWS("WSFacture", soapMessage);
    }
}

```

4.2.2 wsdl

- Facture.wsdl :

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- oOo GENERATED BY GENERIX. oOo -->
<definitions xmlns:tns="http://www.surcouf.com" xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" name="Facturation"
targetNamespace="http://www.surcouf.com">
    <types>
        <xsd:schema targetNamespace="http://www.surcouf.com">
            <xsd:include schemaLocation="xsd/Facturation.xsd"/>
            <xsd:element name="creerFacture" type="tns:CreerFacture"/>
            <xsd:element name="creerFactureResponse" type="tns:CreerFactureResponse"/>
            <xsd:element name="supprimerFacture" type="tns:SupprimerFacture"/>
            <xsd:element name="supprimerFactureResponse" type="tns:SupprimerFactureResponse"/>
        </xsd:schema>
    </types>
    <message name="creerFacture">
        <part name="parameters" element="tns:creerFacture"/>
    </message>
    <message name="creerFactureResponse">
        <part name="parameters" element="tns:creerFactureResponse"/>
    </message>
    <message name="supprimerFacture">
        <part name="parameters" element="tns:supprimerFacture"/>
    </message>
    <message name="supprimerFactureResponse">
        <part name="parameters" element="tns:supprimerFacture"/>
    </message>
    <portType name="Facture">
        <operation name="creerFacture">
            <input message="tns:creerFacture"/>
            <output message="tns:creerFactureResponse"/>
        </operation>
        <operation name="supprimerFacture">
            <input message="tns:supprimerFacture"/>
            <output message="tns:supprimerFactureResponse"/>
        </operation>
    </portType>
    <binding name="FacturationPortBinding" type="tns:Facture">
        <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
        <operation name="creerFacture">
            <soap:operation soapAction=""/>
            <input>
                <soap:body use="literal"/>
            </input>
            <output>
                <soap:body use="literal"/>
            </output>
        </operation>
        <operation name="supprimerFacture">
            <soap:operation soapAction=""/>
            <input>
                <soap:body use="literal"/>
            </input>
            <output>
                <soap:body use="literal"/>
            </output>
        </operation>
    </binding>
    <service name="Facturation">
        <port name="FacturationPort" binding="tns:FacturationPortBinding">
            <soap:address location="REPLACE_WITH_ACTUAL_URL"/>
        </port>
    </service>
</definitions>

```

- xsd/Facture.xsd :

```

<?xml version="1.0" encoding="UTF-8"?>
<!--
    Schéma fournit par le client
-->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://www.surcouf.com"
targetNamespace="http://www.surcouf.com" elementFormDefault="qualified">
    <xsd:complexType name="CreerFacture">

```

```

        <xsd:sequence>
            <xsd:element name="tiers" type="Tiers"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CreerFactureResponse">
        <xsd:attribute name="numfacture" type="xsd:integer" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="SupprimerFacture">
        <xsd:attribute name="numfacture" type="xsd:integer" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="SupprimerFactureResponse">
        <xsd:attribute name="status" type="xsd:boolean" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="Tiers">
        <xsd:attribute name="typtie" type="xsd:string" use="required"/>
        <xsd:attribute name="sigtie" type="xsd:string" use="required"/>
    </xsd:complexType>
</xsd:schema>

```

4.3 Partie technique

4.3.1 sun-jaxws.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- oOo GENERATED BY GENERIX. oOo -->
<endpoints version="2.0" xmlns="http://java.sun.com/xml/ns/jax-ws/ri/runtime">
...
    <endpoint implementation="fr.generix.metier.bc4j.integration.WSGCEServiceImpl" name="gCEService" url-
pattern="/gCEService"/>
    <endpoint implementation="fr.generix.metier.bc4j.integration.WSGCEServiceImplLiteral" name="gCEServiceLiteral" url-
pattern="/literal/gCEService"/>
...
    <endpoint
implementation="fr.generix.technicalframework.businesscomponent.applicationmodule.WSMonitoringManagerImplLiteral"
name="monitoringManagerLiteral" url-pattern="/literal/monitoringManager"/>
...
    <endpoint
implementation="fr.generix.technicalframework.businesscomponent.applicationmodule.WSTransactionManagerImplLiteral"
name="transactionManagerLiteral" url-pattern="/literal/transactionManager"/>
    <!-- Les spécifiques -->
    <endpoint name="gestionFacture" implementation="com.surcouf.Facture" url-pattern="/facturation"/>
    <endpoint name="gestionAchat" implementation="org.tic.tac.AchatImpl" url-pattern="/achat"/>
</endpoints>

```

4.3.2 web.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- J2EE1.4 Descriptor -->
<!-- oOo GENERATED BY GENERIX. oOo -->
<web-app version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
...
    <!-- Publication Webservices -->
    <servlet>
        <display-name>JAX-WS servlet</display-name>
        <servlet-name>jaxws-servlet</servlet-name>
        <servlet-class>com.sun.xml.ws.transport.http.servlet.WSServlet</servlet-class>
        <load-on-startup>1</load-on-startup>
        <security-role-ref>
            <description>role d'accès gce global</description>
            <role-name>sr_gce_overall</role-name>
            <role-link>sr_gce_overall</role-link>
        </security-role-ref>
    </servlet>
...
    <servlet-mapping>
        <servlet-name>jaxws-servlet</servlet-name>
        <url-pattern>/gCEService</url-pattern>
    </servlet-mapping>
...
    <servlet-mapping>
        <servlet-name>jaxws-servlet</servlet-name>
        <url-pattern>/literal/gCEService</url-pattern>
    </servlet-mapping>
...
    <servlet-mapping>
        <servlet-name>jaxws-servlet</servlet-name>
        <url-pattern>/monitoringManager</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>jaxws-servlet</servlet-name>
        <url-pattern>/transactionManager</url-pattern>
    </servlet-mapping>
...
    <!-- Les spécifiques -->
    <servlet-mapping>
        <servlet-name>jaxws-servlet</servlet-name>
        <url-pattern>/facturation</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>jaxws-servlet</servlet-name>
        <url-pattern>/achat</url-pattern>
    </servlet-mapping>

```

```

</servlet-mapping>
<session-config>
...
</web-app>

```

4.4 Intégration

surcouf.jar :

com\surcouf\Facture.class
META-INF\MANIFEST.MF
org\tic\tac\AchatImpl.class
xsl\WSAchat\businessViewServiceLiteralexecuteOut_gestionAchatcreerOut.xsl
xsl\WSAchat\crudManagerdeleteOut_gestionAchatsupprimerOut.xsl
xsl\WSAchat\gestionAchatcreerIn_businessViewServiceLiteralexecuteIn.xsl
xsl\WSAchat\gestionAchatIn.xsl
xsl\WSAchat\gestionAchatOut.xsl
xsl\WSAchat\gestionAchatsupprimerIn_crudManagerdeleteIn.xsl
xsl\WSFacture\crudManagerLiteraldeleteOut_gestionFacturesupprimerFactureOut.xsl
xsl\WSFacture\gCEServiceassimilateOut_gestionFacturecreerFactureOut.xsl
xsl\WSFacture\gestionFacturecreerFactureIn_gCEServiceassimilateIn.xsl
xsl\WSFacture\gestionFactureIn.xsl
xsl\WSFacture\gestionFactureOut.xsl
xsl\WSFacture\gestionFacturesupprimerFactureIn_crudManagerLiteraldeleteIn.xsl

- xsl/WSFacture/gestionFactureIn.xsl :

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="xml" encoding="UTF-8" indent="yes" omit-xml-declaration="no"/>
  <!--
    Feuille de style permettant de modifier le flux d'entrée du WS gestionFactureIn
  -->
  <!--
    On recopie le gros de la troupe
  -->
  <xsl:template match="fournisseur">
    <tiers typtie="FOU" sigtie="AGFOU(@id)"/>
  </xsl:template>
  <xsl:template match="/">
    <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:www="www.surcouf.com">
      <soapenv:Header/>
      <soapenv:Body>
        <www:creerFacture>
          <xsl:apply-templates/>
        </www:creerFacture>
      </soapenv:Body>
    </soapenv:Envelope>
  </xsl:template>
</xsl:stylesheet>

```

- xsl/WSFacture/gestionFacturecreerFactureIn_gCEServiceassimilateIn.xsl :

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:www="www.surcouf.com" exclude-result-prefixes="www">
  <xsl:output method="xml" encoding="UTF-8" indent="yes" omit-xml-declaration="no"/>
  <!--
    Feuille de style permettant de générer un flux d'entrée gCEServiceassimilateIn à partir d'un flux
    gestionFacturecreerFactureIn
  -->
  <!--
    Template principal
  -->
  <xsl:template match="/">
    <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:GCE"
    xmlns="http://www.generix.fr/metier/bc4j/integration/common"
    xmlns:ser="http://www.generix.fr/technicalframework/business/webservice/server">
      <soapenv:Body>
        <urn:assimilate>
          <ctx entity="1312" user="DEMO" password="DEMO" endOfRequest="destroy">

```

```

        <ser:log>
          <ser:logger name="DATA" active="true"/>
          <ser:logger name="webservice.server.Timer"
active="true"/>
        </ser:log>
      </ctx>
      <GCEServiceassimilateIn>
        <transaction>
          <context>
            <entity>1312</entity>
            <identification>
              <code>DEMO</code>
            </identification>
            <target>ACDA</target>
          </context>
          <businessEvent>
            <header action="creation">
              <info>
                <thirdparty
                  <key>
                    <type><xsl:value-of select="/soapenv:Envelope/soapenv:Body/www:creerFacture/tiers/@typtie"/></type>
                    <identifier><xsl:value-of select="/soapenv:Envelope/soapenv:Body/www:creerFacture/tiers/@sigtie"/></identifier>
                    </key>
                    </thirdparty>
                    <date>20120404</date>
                    <expense>
                      <info>
                        <expense>PORT</expense>
                        <rate>10</rate>
                      </info>
                    </expense>
                  </info>
                </header>
              </businessEvent>
            </transaction>
          </GCEServiceassimilateIn>
        </urn:assimilate>
      </soapenv:Body>
    </soapenv:Envelope>
  </xsl:template>
</xsl:stylesheet>

```

- xsl/WSFacture/ gCEServiceassimilateOut_gestionFacturecreerFactureOut.xsl :

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:ns1="urn:GCE" exclude-result-
prefixes="ns1">
  <xsl:output method="xml" encoding="UTF-8" indent="yes" omit-xml-declaration="no"/>
  <!--
    Feuille de style permettant de générer un flux d'entrée gCEServiceassimilateIn à partir d'un flux
gestionFacturecreerFactureIn
  -->
  <!--
    Template principal
  -->
  <xsl:template match="/">
    <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:www="www.surcouf.com">
      <soapenv:Header/>
      <soapenv:Body>
        <www:creerFactureResponse
numfacture="/soapenv:Envelope/soapenv:Body/ns1:assimilateResponse/*[name()='GCEServiceassimilateOut']/*[name()='result']/*[n
ame()='eventHeaderKey']/*[name()='eventNumber']"/>
        </soapenv:Body>
      </soapenv:Envelope>
    </xsl:template>
  </xsl:stylesheet>

```

- xsl/WSFacture/ gCEServiceassimilateOut_gestionFacturecreerFactureOut.xsl :

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="xml" encoding="UTF-8" indent="yes" omit-xml-declaration="no"/>
  <!--
    Feuille de style permettant de modifier le flux de sortie du WS gestionFactureOut
  -->
  <!--
    On recopie le gros de la troupe
  -->
  <xsl:template match="@*|*|comment()">
    <xsl:copy>
      <xsl:apply-templates select="@*|*|text()|comment()|processing-instruction()"/>
    </xsl:copy>
  </xsl:template>
</xsl:stylesheet>

```