Web Services

Olivier Coupelon

2021-2022

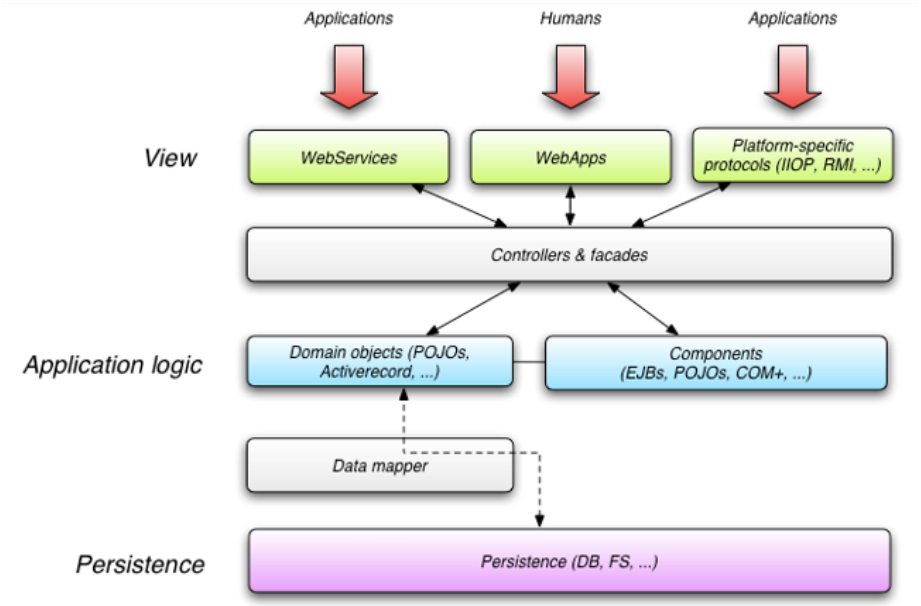

Figure 1: Application

Web Services

- Communication inter-applications :
  - Technologie non adaptée à une interrogation directe par un utilisateur.
  - Couplage faible
- Trois besoins :
  - Echange de données (SOAP)
  - Description des services d'échanges (WSDL)
  - Découverte des services (UDDI)

SOAP

- Anciennement Simple Object Access Protocol
- Enveloppe d'échange de données, comprenant :

  - Un entête (header)
  - Un corps (body)
- Enveloppe transportée sur HTTP, XMPP, ou encore SMTP

```xml
<?xml version="1.0" encoding="UTF-8"?>

<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
    <env:Header>
        <t:transaction xmlns:t="urn:mytransaction">
            <t:id>123456789</t:id>
        </t:transaction>
        <h:plop xmlns:h="urn:plop:da:plop" env:mustUnderstand="1">Coin Coin</h:plop>
    </env:Header>
    <env:Body>
        <s:sales xmlns:s="urn:sales">
            <s:product>Banana split</s:product>
            <s:quantity>5</s:quantity>
            <s:customer>Mr Bean</s:customer>
        </s:sales>
    </env:Body>
</env:Envelope>
```

Figure 2: Enveloppe SOAP

WSDL

Web Service Description Language

- Décrit les opérations disponibles d'un service web
- Contrat établi entre le client et le serveur

WSDL 1.1 vs 2.0

UDDI Universal Description Discovery and Integration * Annuaire de services : * Pages blanches : liste des enteprises, les fournisseurs de service * Pages jaunes : liste des services (WSDL) * Pages vertes : détails techniques des services, liaison avec les processus métiers associés

---

UDDI par l'exemple

1. Requête vers UDDI pour obtenir les informations techniques sur un service
   - WSDL

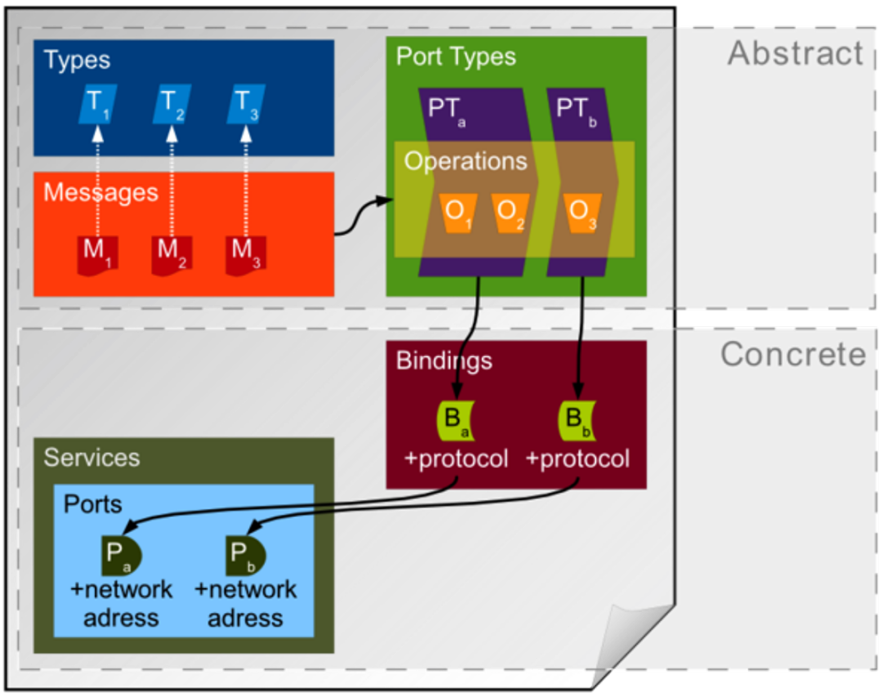

Figure 3: WSDL

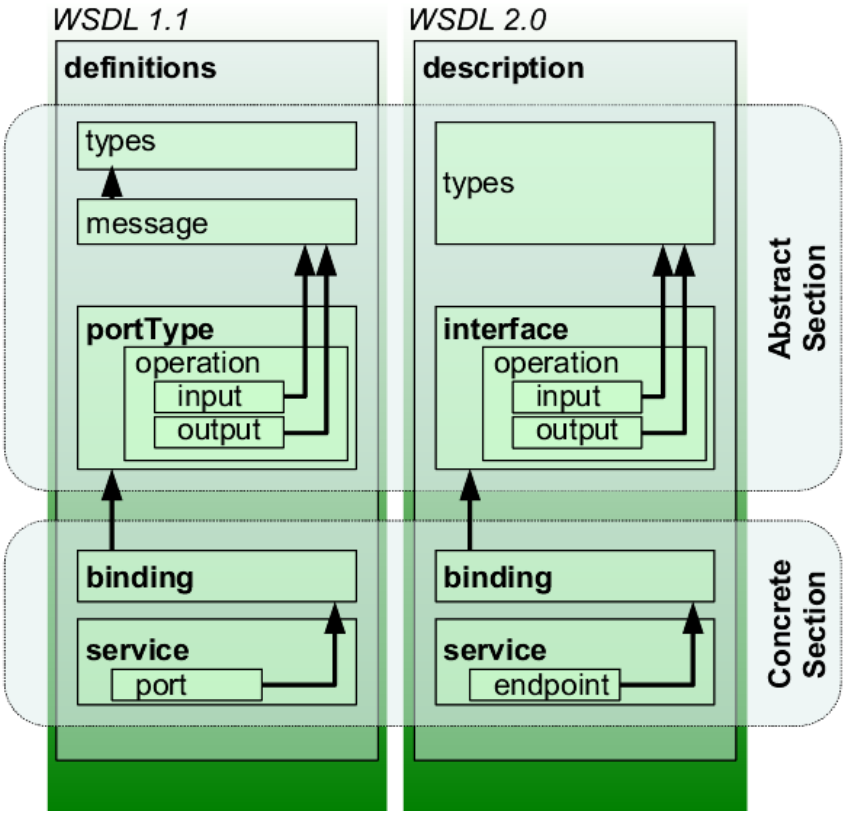

Figure 4: WSDL

- Adresse
- bindingKey : clé identifiante unique

2. Ecriture du client associé
3. En cas d'échec lors d'une utilisation, le client doit prévoir de rappeler l'annuaire avec la bindingKey afin de récupérer la nouvelle interface/adresse du service automatiquement.

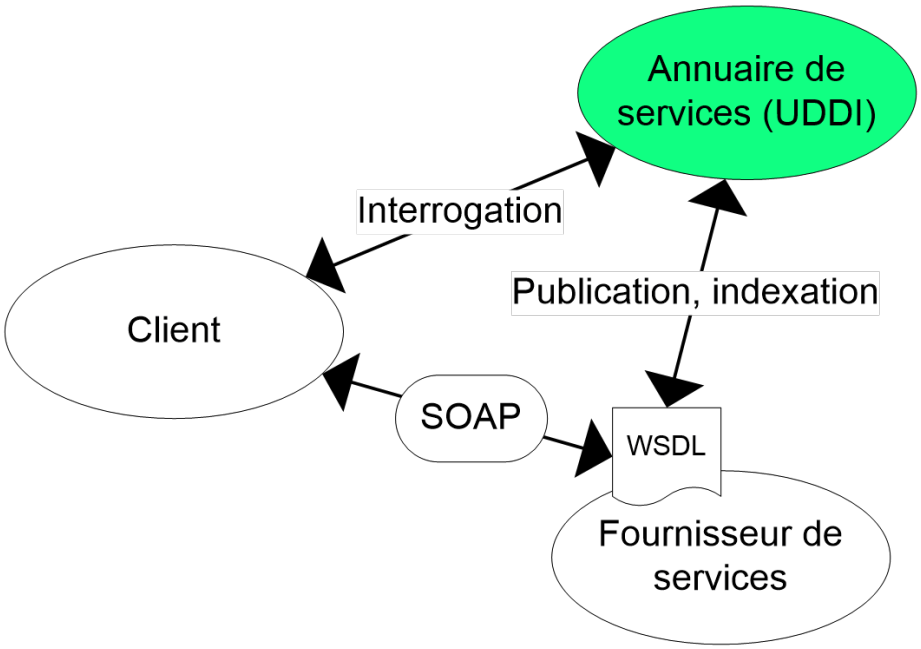

Figure 5: WSDL

Création d'un Web Service * Top-Down : 1. Ecriture du WSDL 2. Génération du code associé :

```
wsimport -d src/generated http://example.org/stock?wsdl

wsdl2java -d src/generated -server -client http://example.org/stock?wsdl
```

- Bottom-Up :

1. Ecriture du code du service
2. Génération automatique du WSDL à partir de ce code
   - Automatique avec les frameworks courants (JAX-WS)

```
wsgen     -cp . ws.Hello
```

Technologies de création d'un Web Service

- Apache Axis et Axis 2
- XFire
- JAX-WS
  - Apache CXF
  - Metro (implémentation de référence)

JAX-WS

Java API for XML Web Services

- JAX-WS 2.0 : Standard dans Java EE 5
- JAX-WS 2.2 : Java EE 6, 7 et 8

Comme JAX-RS, on utilise des annotations Java pour créer les services

---

Annotations JAX-WS

- @WebService : La classe annotée est déclarée comme étant un service
- @WebMethod : La méthode annotée doit être exposée en tant qu'opération du service
- @WebParam : Permet de spécifier les propriétés d'un paramètres
- Par défaut, les méthodes publiques de la classe sont exposées

---

Appeler un Web Service

- Doit toujours se baser sur le WSDL (Top-Down)
- Génération du code Java à partir du WSDL
- Les IDE savent générer le code Java à partir d'un WSDL automatiquement
- Exemple d'injection de web services :

```
@WebServiceRef(wsdlLocation=http://www.toto.fr/ws?wsdl) static AppService service;
```

Spécifications WS-*

http://fr.wikipedia.org/wiki/Liste_des_spécifications_des_Services_Web_WS-* https://www.innoq.com/soa/ws-standards/poster/innoQ%20WS-Standards%20Poster%202007-02.pdf

- Les spécifications sont nombreuses, parfois non maintenues, concurrentes...
- Apporte des fonctionnalités supplémentaires à la communication Web Service (fiabilité, sécurité...)

WS-Policy

- Ajoute des informations au WSDL sur les capacités du service
- Exemple : force l'authentification. Doit être intégré à la partie Bindings du WSDL

```
<wsp:Policy wsu:Id="AvecUserNameToken"
    xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy"
    xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
    <sp:SignedSupportingTokens>
        <wsp:Policy>
            <sp:UsernameToken sp:IncludeToken=
            "http://schemas.xmlsoap.org/ws/2005/07/securitypolicy/IncludeToken/AlwaysToRecipient"/>
        </wsp:Policy>
    </sp:SignedSupportingTokens>
</wsp:Policy>
```

Figure 6: WS-Policy

La sécurité dans les Web Services

1. Transport Level Security
   - Principalement HTTPS et Authentification HTTP
2. Message Level Security
   - WS-Security, qui permet de décrire :
     - XML-Encryption : crypter le contenu des messages SOAP
     - XML-Signature : authentification des interlocuteurs

WS-Security

- Permet d'ajouter une couche de sécurité aux échanges SOAP
- Utilisation de SAML, Kerberos, certificats X509...
- Exemple avec UsernameToken et Timestamp :
  - Identifiant par login/mot de passe

Ces informations sont transmises dans l'entête SOAP

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:val="http://prestation.almerys.com/schema/valerys">
   <soapenv:Header>
        <wsse:Security soapenv:mustUnderstand="true"
        xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
        xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
            <wsu:Timestamp u:Id="_0" xmlns:u="u">
                <wsu:Created>2010-01-01T00:00:00.462Z</wsu:Created>
                <wsu:Expires>2999-12-31T23:59:59.462Z</wsu:Expires>
            </wsu:Timestamp>

            <wsse:UsernameToken wsu:Id="UsernameToken-32380043">
                <wsse:Username>Kermit</wsse:Username>
                <wsse:Password
                Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">
                theFrog</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </soapenv:Header>
   <soapenv:Body>

   </soapenv:Body>
</soapenv:Envelope>
```

Figure 7: WS-Security

WS-Addressing

- Permet de contrôler de manière standard la communication SOAP.
- Ajoute des Message Information (entêtes définies par WSA)
- Indépendant de la couche de transport

Exemple WS-Addressing To : le service cible du message

```
<wsa:To> http://host/WidgetService </wsa:To>
```

From : l'emmeteur du message

```
<wsa:From>
    <wsa:Address> http://client/myClient </wsa:Address>
</wsa:From>
```

ReplyTo : le service à contacter pour répondre

```
<wsa:ReplyTo>
    <wsa:Address> http://client/myReceiver </wsa:Address>
</wsa:ReplyTo>
```

FaultTo : le service à contacter en cas d'exception

```
<wsa:FaultTo>
    <wsa:Address> http://client/FaultReceiver </wsa:Address>
</wsa:FaultTo>
```

Exemple WS-Addressing MessageID : identifiant unique du message

```
<wsa:MessageID>uuid:098765</wsa:MessageID>
```

RelatesTo : spécifie une relation avec un autre message par son MessageID

```
<wsa:RelatesTo RelationshipType="wsa:Response">
    uuid:098765
</wsa:RelatesTo>
```

Autres spécifications

- WS-Reliability : Permet de s'assurer qu'un message SOAP est bien a bien été livré
- WS-Transaction : Utilise WS-Coordination pour offrir des garanties transactionnelles aux services

---

BPMN : Business Process Modeling Notation

Utilisé pour décrire au tableau des **processus métier**

- Ensemble de tâches coordonnées dans le but de générer une plus-value
- A dérivé avec le temps en langage exécutable par une machine

---

Tâches en BPMN

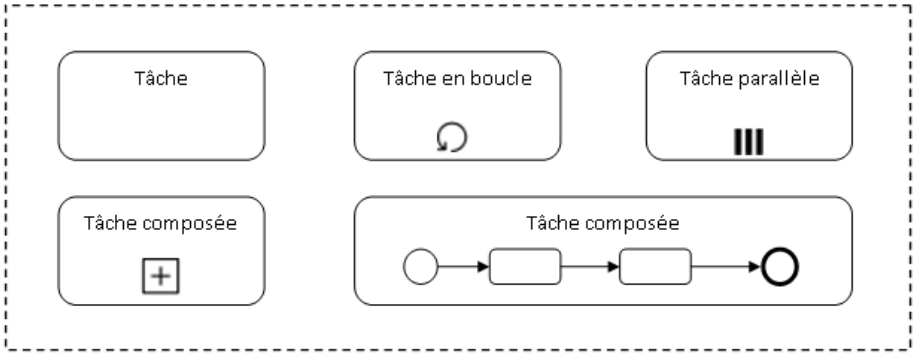

Figure 8: BPMN tasks

Conditions en BPMN

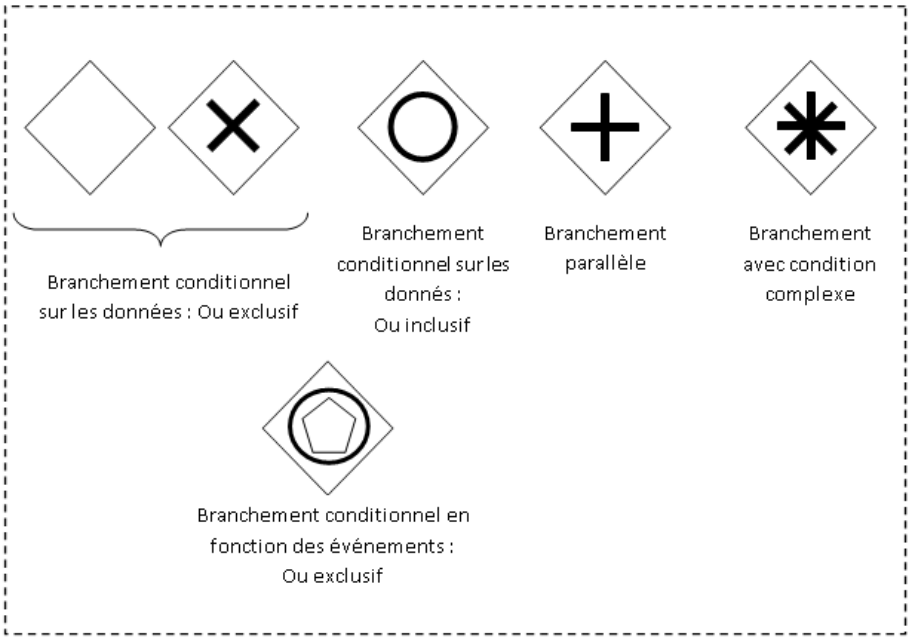

Figure 9: BPMN conditions

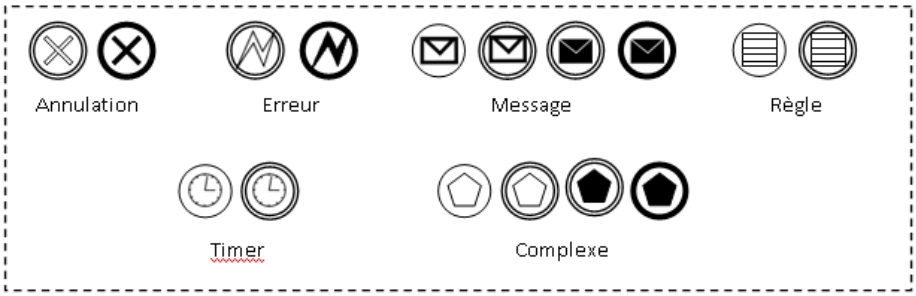

Figure 10: BPMN events

Evènements en BPMN

Connecteurs en BPMN

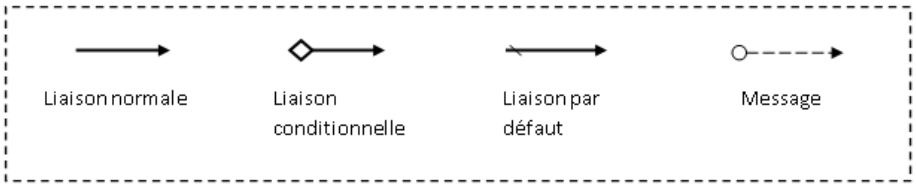

Figure 11: BPMN connectors

Couloirs et rôles en BPMN

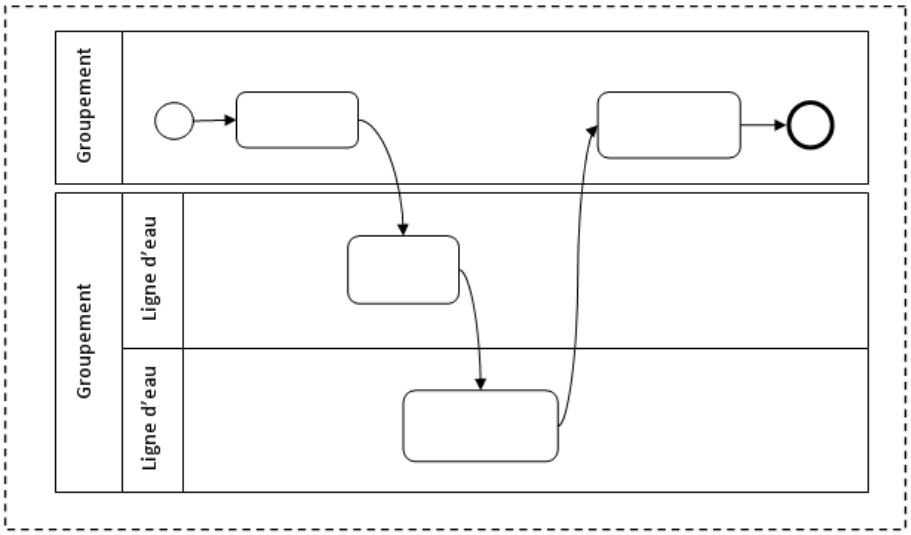

Figure 12: BPMN swimlanes

Artefacts en BPMN

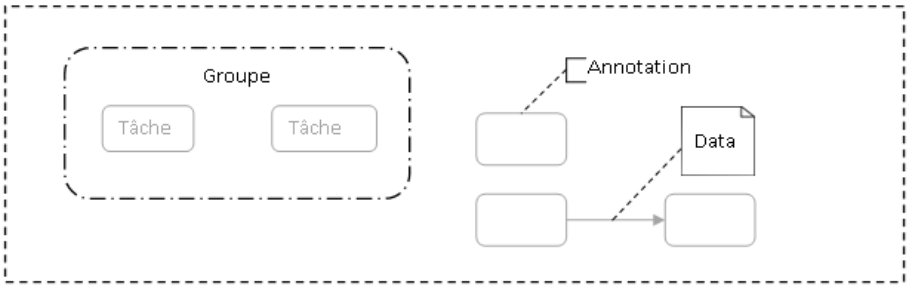

Figure 13: BPMN artefacts

Exécution BPMN

- Historiquement uniquement graphiques, donc pas exécutables
  - Théoriquement non déterministe (sémantique du OU)
- Représentés sur disque via la norme XPDL
- Avec BPMN 2.0 : exécution directe
  - jBPM
  - Oracles & co. . .