



**HAL**  
open science

## Vers la gestion de Services Web

Marie-Christine Fauvet

► **To cite this version:**

Marie-Christine Fauvet. Vers la gestion de Services Web. INFORSID 2003, 2003, Nancy, France. 2003. hal-00954132

**HAL Id: hal-00954132**

**<https://inria.hal.science/hal-00954132>**

Submitted on 21 Mar 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Vers la gestion de services Web

Marie-Christine Fauvet  
LSR-IMAG, Université Joseph Fourier

Présentation préparée avec LaTeX (paquetage prosper)

Tutoriel INFORSID, 3 juin 2003, © M.-C. Fauvet – p.1/4

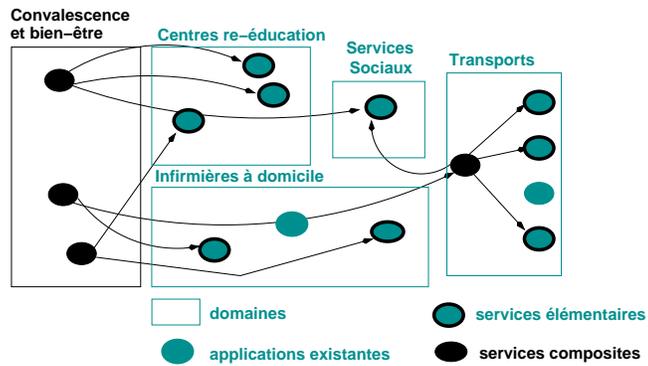
## Objectifs du tutoriel

Présenter les concepts sous-jacents aux services Web

- Etudier les fonctions d'un système de gestion de services Web
- Sans (trop) rentrer dans les détails de la technologie

Tutoriel INFORSID, 3 juin 2003, © M.-C. Fauvet – p.2/4

## Le contexte : un scénario exemple



*Des applications autonomes et hétérogènes  
encapsulées, et capables de communiquer entre elles  
Vers des services composites, à valeur ajoutée*

Tutoriel INFORSID, 3 juin 2003, © M.-C. Fauvet – p.3/4

## Plan de la présentation

- Approche
- Technologie
- Services Web :
  - Conception
  - Implantation
- Conclusion
- Références et ressources

Tutoriel INFORSID, 3 juin 2003, © M.-C. Fauvet – p.4/4

## Service Web : définition

*“Web services are a new breed of Web application. They are self-contained, self-described, modular applications that can be published, located, and invoked across the Web. Web services perform functions, which can be anything from simple requests to complicated business processes...”*

*Once a Web service is deployed, other applications (and other Web services) can discover and invoke the deployed service.”*

D'après “Cardoso et al. 2002”

Tutoriel INFORSID, 3 juin 2003, © M.-C. Fauvet – p.5/4

## 1 L'approche

Tutoriel INFORSID, 3 juin 2003, © M.-C. Fauvet – p.6/4

## Vers la gestion de services Web

Proposer un environnement pour :

- la localisation,
    - la composition,
      - l'intégration,
        - le déploiement,
          - l'exécution,
            - et le suivi
- de ressources accessibles par l'Internet.  
(contexte non contrôlé)

Tutoriel INFORSID, 3 juin 2003, © M.-C. Fauvet – p.7/4

## Problèmes clefs

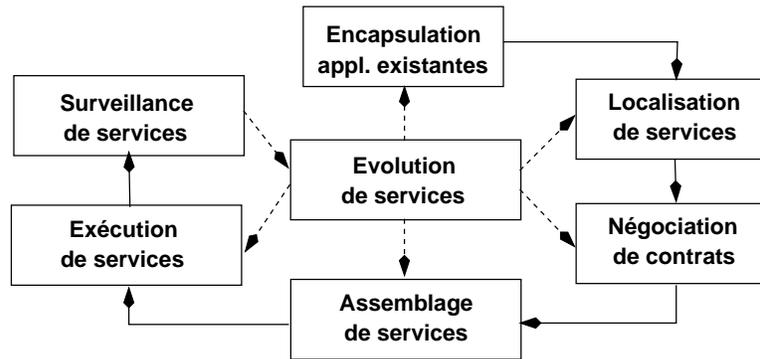
- Intégration et assemblage, rapides et à faible coût,
- Modèle d'exécution,
- Qualité,
- Sécurité
- Adaptation aux utilisateurs,
- Adaptation à l'infrastructure d'accueil.

*S'abstraire de la technologie  
Respecter le caractère  
distribué et autonome des services*

Tutoriel INFORSID, 3 juin 2003, © M.-C. Fauvet – p.8/4

# Problématique

quels outils pour la gestion de services ?



*Concevoir et gérer des services composites*

Tutoriel INFORSID, 3 juin 2003, © M.-C. Fauvet – p.9/4

## 2 La technologie

Tutoriel INFORSID, 3 juin 2003, © M.-C. Fauvet – p.10/4

## Caractéristiques des services Web

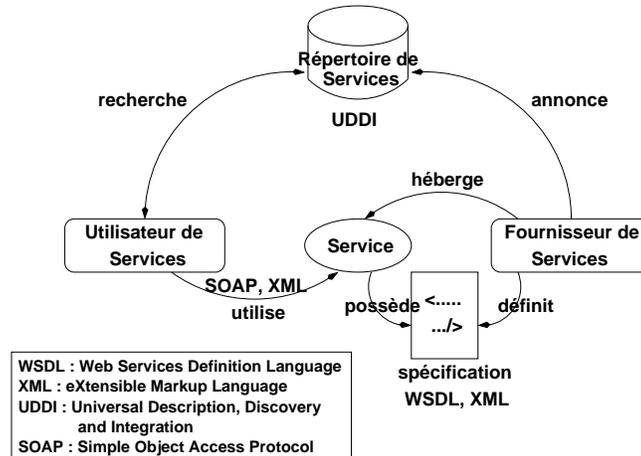
- *Basés sur des protocoles Internet* : l'utilisation de HTTP pour le transport des informations permet de traverser les contrôles d'accès, dans un environnement hétérogène.
- *Interopérables* : le standard SOAP est le support pour l'interopérabilité.
- *Basés sur XML* : le méta-langage XML (eXtensible Markup Language) offre un cadre standard pour la définition de documents interprétables par des machines.

D'après "Fremantle et al. 2002"

## Les technologies : quelques repères

- *UDDI* : (Universal Description Discovery and Integration) permet de publier et de rechercher des services Web.
- *WSDL* : (Web Services Description Language) permet de décrire des services Web dans la perspective de les retrouver et de les utiliser.
- *SOAP* : (Simple Object Access Protocol) est un standard qui définit le format de messages contenant des données encodées en XML.
- *HTTP ou TCP/IP* : sont des protocoles de transport des données.

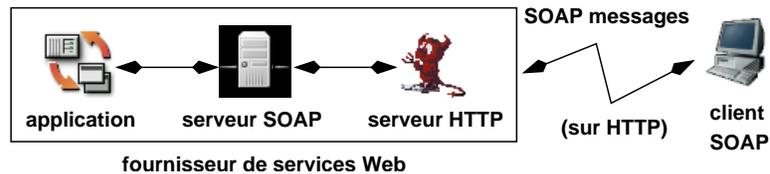
## Offre et consommation de services



D'après "A. Bouguettaya 2003"

Tutoriel INFORSID, 3 juin 2003, © M.-C. Fauvet - p.13/4

## SOAP (1)



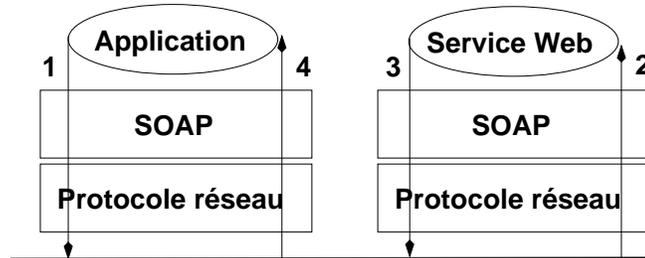
Composants nécessaires :

- Une application
- Un serveur SOAP
- Un serveur HTTP (par exemple)
- Un client SOAP

D'après "Cardoso et al. 2002"

Tutoriel INFORSID, 3 juin 2003, © M.-C. Fauvet - p.14/4

## SOAP (2)



1. Création d'un message SOAP contenant l'adresse du fournisseur de service. Le serveur SOAP transmet le message à l'infrastructure de transport.
2. Transmission des données au travers du réseau. Le message est décodé par le serveur SOAP, en termes d'objets transmis au service.
3. Le serveur Web construit une réponse et la formate en un message SOAP, qui est transmis à l'infrastructure de transport.
4. Le message est décodé par le serveur SOAP et transmis à l'application.

D'après "Benatallah et al. 2002a"

Tutoriel INFORSID, 3 juin 2003, © M.-C. Fauvet – p.15/4

## Description de services

- Pour informer les demandeurs :
  - des fonctions offertes,
  - du protocole de transport,
  - du format d'échanges des messages
  - du point d'entrée du service
- Des langages pour cela :
  - WSDL : description syntaxique
  - DAML-OIL, DAML-S, RDF : descriptions syntaxique et sémantique
    - notion d'ontologie

D'après "Cardoso et al. 2002"

Tutoriel INFORSID, 3 juin 2003, © M.-C. Fauvet – p.16/4

## WSDL

- Syntaxe XML pour la description de services.
- Définition de l'implantation du service :
  - Les opérations proposées
  - Les points d'entrée pour l'exécution des opérations
- Définition de l'interface du service :
  - Les types de données utilisés
  - Le format des entrées/sorties
- S'appuie sur SOAP

## UDDI (1)

Sorte d'annuaire pour services Web :

- Méta-données décrivant le service (catégorie, fournisseur)
- Informations pour l'accès (localisation, interface)

## **UDDI (2)**

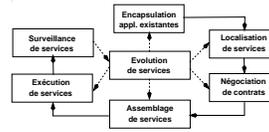
Un annuaire UDDI est constitué de :

- Pages blanches : décrivent les sociétés participantes
- Pages jaunes : classifient les participants et les services offerts
- Pages vertes : indiquent comment exécuter un service offert.

## **3 Concevoir et implanter des services**

# La méthode

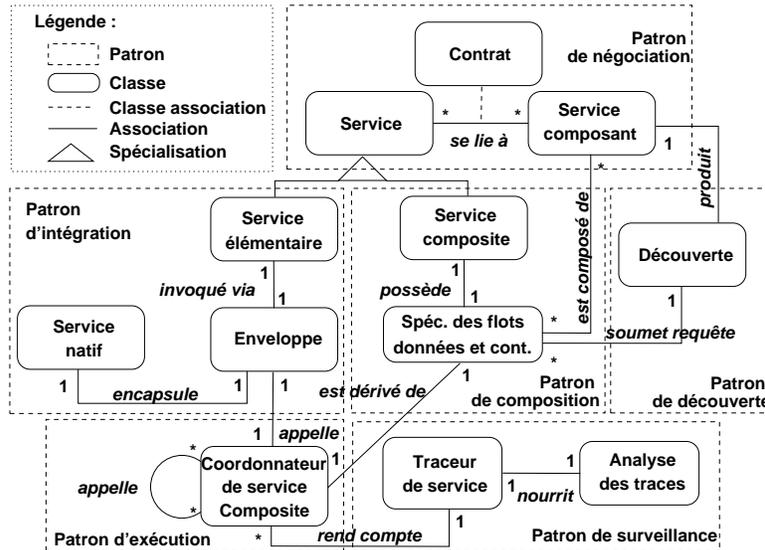
Analyser les outils en terme de patrons de conception ("design patterns").



Pour :

- S'abstraire de la technologie
- Se donner une grille d'analyse des solutions

# Conception de services Web



## Encapsulation de services (1)

*Problème posé :*

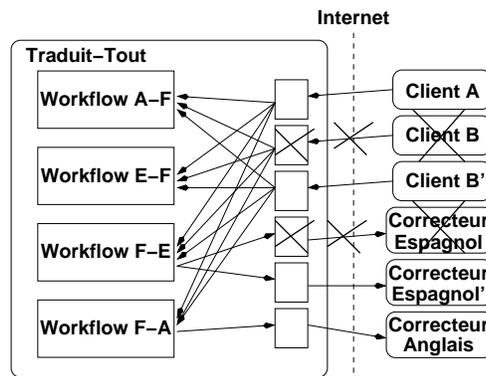
rendre un service natif (une application existante)  
accessible par d'autres services Web, quelque soit :

- son modèle de données
- son format de messages reçus et émis
- son protocole de communication

Tutoriel INFORSID, 3 juin 2003, © M.-C. Fauvet – p.23/4

## Encapsulation de services (2)

*Exemple :*



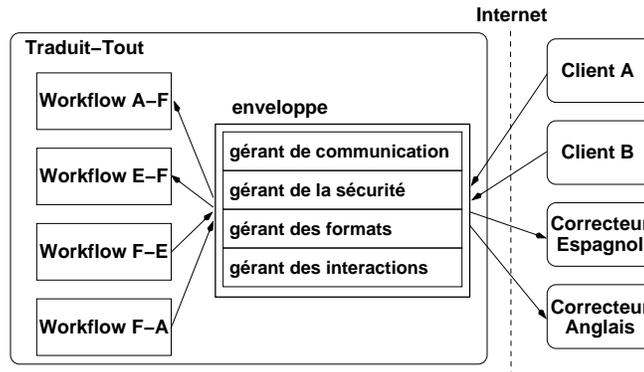
Définir et faire évoluer :

- Les formats de documents
- Les protocoles d'interaction
- Le maintien de l'intégrité, de la confidentialité et du non rejet

Tutoriel INFORSID, 3 juin 2003, © M.-C. Fauvet – p.24/4

## Encapsulation de services (3)

*Solution :*



Tutoriel INFORSID, 3 juin 2003, © M.-C. Fauvet – p.25/4

## Encapsulation de services (4)

*Implantation :*

- Gérant de communication :
  - Supporte divers protocoles de communication (HTML/HTTP, SOAP/HTTP, Java RMI, CORBA IIOP, DCOM, ..)
  - Assure la traduction automatique des messages (en SOAP par exemple)
- Gérant de la sécurité :
  - Offre les mécanismes d'authentification, de chiffrement et d'audit (par exemple HTTPS, SSL).

Tutoriel INFORSID, 3 juin 2003, © M.-C. Fauvet – p.26/4

## Encapsulation de services (5)

*Implantation (suite) :*

- Gérant du contenu :
  - Fournit les interfaces correspondant aux divers formats de messages.
  - Convertit les messages en entrée et en sortie.
  - Technologies : XML, RDF.

## Encapsulation de services (6)

*Implantation (suite) :*

- Gérant des interactions : se comporte comme :
  - un contrôleur : dispatche chaque message (en entrée et en sortie), selon la tâche de l'interaction vers la fonction associée au service (en entrée) ou au partenaire cible (en sortie).
  - un traducteur : traduit la séquence des tâches correspondant à un certain protocole d'interaction vers une autre correspondant à un autre protocole d'interaction.
  - des solutions émergentes : Sayal et al. 2001, Kuno et al. 2001.

## Composition de services (1)

*Problème posé :*

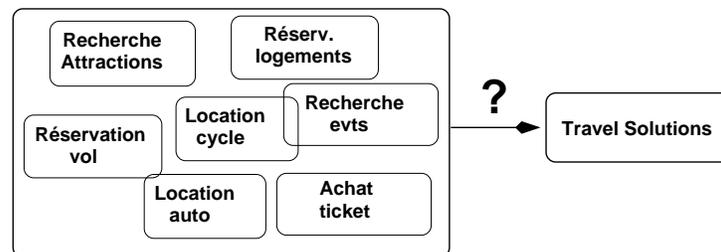
identifier les services nécessaires à la réalisation d'une composition et spécifier leurs interactions à un *haut niveau d'abstraction*, pour permettre :

- une intégration rapide et à un faible coût
- le passage à l'échelle

Tutoriel INFORSID, 3 juin 2003, © M.-C. Fauvet – p.29/4

## Composition de services (2)

*Exemple :*



Tutoriel INFORSID, 3 juin 2003, © M.-C. Fauvet – p.30/4

## Composition de services (3)

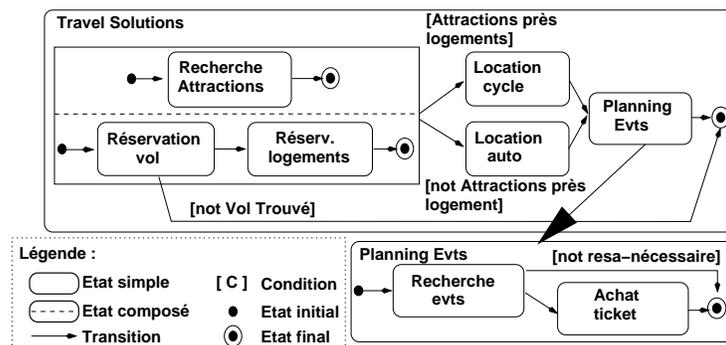
*Solution :*

- Distinguer services élémentaires et services composites
- Service composites : définis récursivement comme des agrégations de services (élémentaires ou composites).
- Spécifier les flots de contrôles et de données.

Tutoriel INFORSID, 3 juin 2003, © M.-C. Fauvet – p.31/4

## Composition de services (4)

*Implantation (1) :* diagrammes d'états pour décrire la composition et le flot de contrôle (Harel et al. 1996, Benatallah et al. 2002c)



Tutoriel INFORSID, 3 juin 2003, © M.-C. Fauvet – p.32/4

## Composition de services (5)

*Implantation (2) : d'autres approches*

- Architectures à base de flots de tâches (*workflows*)  
(voir BPEL4WS, BPML, ebXML, WfMC 1996, WSFL, XLang, Yang et al. 2000)
- Architectures à base de composants  
(Dogac 1998, Brodie 2000, Cugola et al 2000, .Net, J2EE)

## Exécution de services (1)

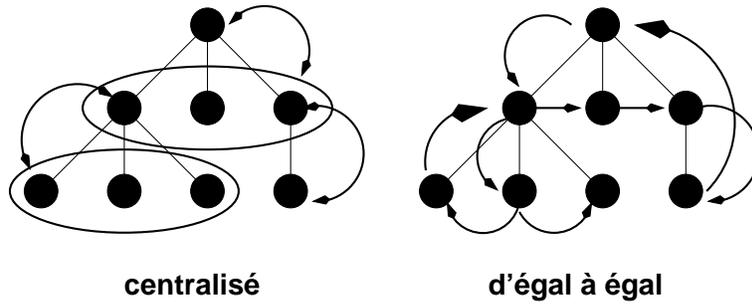
*Problème posé :*

étant donnée une composition, dériver un schéma d'exécution selon le mode d'orchestration choisi (centralisé ou d'égal-à-égal).

- Quand un composant doit-il être exécuté ?
- Que faire lorsque l'exécution d'un composant est terminée ?

## Exécution de services (2)

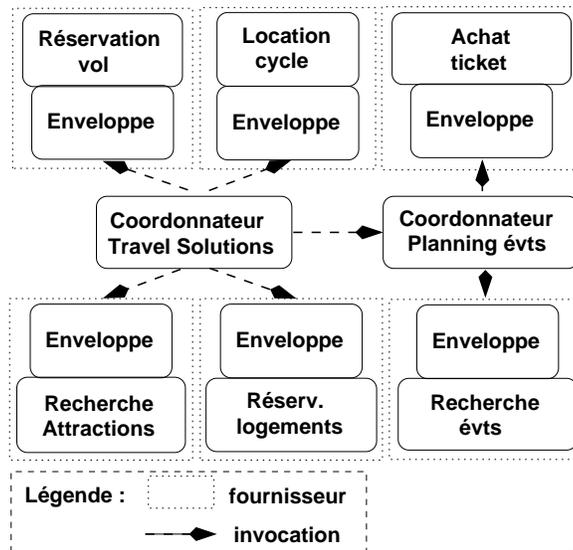
Modes d'orchestration distribuée : centralisé vs. d'égal-à-égal



Tutoriel INFORSID, 3 juin 2003, © M.-C. Fauvet – p.35/4

## Exécution de services (3)

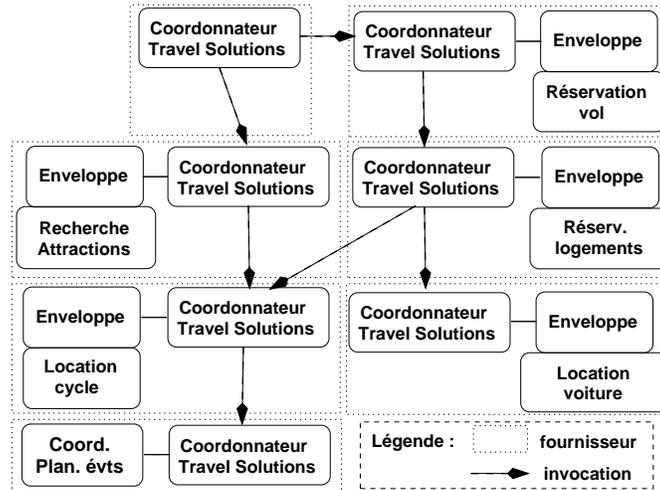
Solution : exécution centralisée



Tutoriel INFORSID, 3 juin 2003, © M.-C. Fauvet – p.36/4

## Exécution de services (4)

*Solution* : exécution d'égal-à-égal



Tutoriel INFORSID, 3 juin 2003, © M.-C. Fauvet – p.37/4

## Exécution de services (5)

*Implantations (1)* : exécution centralisée

- ADEPT (Jennings et al. 2000) : système multi-agents qui supporte l'exécution de flots de tâches inter-organisations.
- EFlow (Casati et al. 2001), CMI (Schuster et al. 2000) : systèmes capables d'exécuter et de coordonner des services Web.

Tutoriel INFORSID, 3 juin 2003, © M.-C. Fauvet – p.38/4

## Exécution de services (6)

*Implantations (2) : exécution d'égal-à-égal*

- SELF-SERV (Benatallah et al. 2002c) : système capable d'exécuter et de coordonner des services Web.
- CPM (Chen et al. 2001) : supporte l'exécution de processus au travers de collaborations entre des moteurs d'exécution de flots de tâches.

## Découverte de services (1)

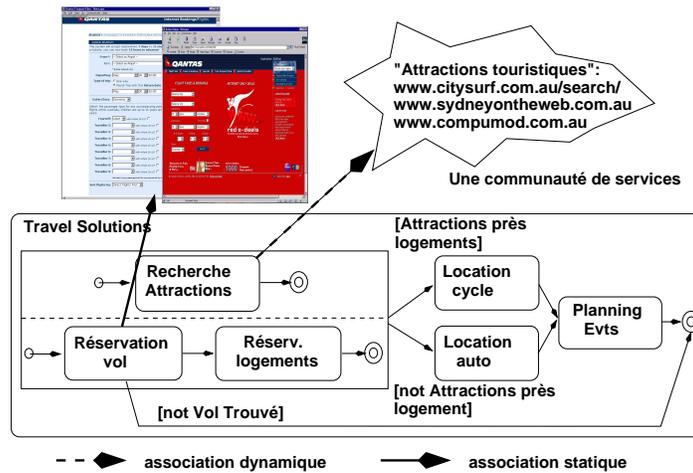
*Problème posé :*

choisir un service Web dans un ensemble vaste et volatile.

- statiquement à la spécification de la composition.
- dynamiquement à l'exécution.

## Découverte de services (2)

Exemple :



Tutoriel INFORSID, 3 juin 2003, © M.-C. Fauvet – p.41/4

## Découverte de services (3)

Solution :

- Décrire les services Web en termes de méta-données.
- S'appuyer sur un langage interprétable par une machine.
- Se donner des métriques pour comparer les offres :
  - Coût
  - Qualité
  - Confiance
  - etc.

Tutoriel INFORSID, 3 juin 2003, © M.-C. Fauvet – p.42/4

## Découverte de services (4)

*Implantation* : association statique

- Descriptions syntaxiques  
UDDI, WSDL, ebXML (Bussler 2001)
- Approches basées sur la notion d'ontologie  
DAML+OIL, DAML-S (Cardoso et al. 2002)

## 4 Conclusion

## En résumé...

- Quelques repères sur les technologies sous-jacentes
- Mise en évidence de phases dans la conception des services Web
- Formalisation de chaque phase par le biais d'un patron de conception (voir Gamma et al. 1995, Benatallah 2002b)

## Conclusion

- Profusion des solutions
- Nécessité d'identifier et d'unifier les concepts
- Définir le bon niveau d'abstraction (tout XML ?)

## D'autres aspects

- Négociation (Jennings et al. 2000) :
  - Négocier et établir un contrat
  - Concilier et masquer ses contraintes et ses préférences
  - Faire respecter le contrat
  - Manuel, plus ou moins automatisé
- Surveillance (Fauvet et al. 2002) :
  - Collecter et stocker les traces des exécutions de services (distribuées chez les fournisseurs)
  - En amont d'outils d'analyse évaluer les requêtes sur les traces (traitées de manière coopérative par les fournisseurs)
- Adaptabilité : aux utilisateurs, aux structures d'accueil.

Tutoriel INFORSID, 3 juin 2003, © M.-C. Fauvet – p.47/4

## Des questions ?



Tutoriel INFORSID, 3 juin 2003, © M.-C. Fauvet – p.48/4

### Références et glossaire

- Benatallah et al. 2002a : *E-commerce enabling technologies*, Notes de cours UNSW Sydney, Australia. Pearson SprintPrint. ISBN 4 74009 418 2.
- Benatallah et al. 2002b : *Towards Patterns of Web Services Composition*, Dans ouvrage *Patterns and Skeletons for Parallel and Distributed Programming*, Springer Verlag, London. ISBN 1-85233-506-8.
- Benatallah et al. 2002c : *Declarative Composition and Peer-to-Peer Provisioning of Dynamic Web Services*, Proceedings of ICDE'02 Conference (IEEE), San Jose, California.
- Bouguettaya 2003 : *Towards a Web Service Management System*,  
Voir <http://www.nvc.cs.vt.edu/eceg>
- BPEL4WS (Business Process Execution Language for Web Services, réunion de WSFL, WSCL, XLang), recommandation du W3C.  
Voir <http://www.ibm.com/developerworks/library/ws-bpel/>.
- BPML (Business Process Modeling Language) :  
voir <http://www.bpml.org/bpml.esp>

48-1

- Brodie 2000 : *The B2B E-commerce revolution and its technology requirements*, Présenté au Workshop "National Science Foundation's Information and Data Management", Chicago, Illinois. GTE Corporation.
- Bussler 2001 : *B2B Protocol Standards and their Role in Semantic B2B Integration Engines*, Bulletin of the Technical Committee on Data Engineering, vol. 24, nb. 1.
- Cardoso et al. 2002 : *Semantic Web Services and Processes: Semantic Composition and Quality of Service*, Tutoriel à la Conférence Fédérée "On the move to meaningful internet computing and ubiquitous computer". Irvine, CA.  
Voir : cliquer ICI  
adresse : <http://lstdis.cs.uga.edu/lib/presentations/SWSP-tutorial-resource.htm>
- Casati et al. 2001 : *Dynamic and Adaptive Composition of E-Services*, Information Systems, vol. 26, nb. 3, Elsevier Science.
- Chen et al. 2001 : *Inter-Enterprise Collaborative Business Process Management*, Proceedings of ICDE'01 Conference (IEEE), Heidelberg, Germany.
- CORBA/IIOP (Common Object Request Broker Architecture/Internet Inter-Orb Protocol) :  
Voir <http://www.corba.org/>

48-2

- Cugola et al. 2000 : *Support for Software Federations: the PIE Platform*, EWSPT 2000 Conference, Springer Verlag, LNCS 1780.
- DAML (DARPA Agent Markup Language), DAML-S, DAML-OIL :  
Voir <http://www.daml.org/>
- DCOM (Distributed Component Object Model) :  
voir <http://www.microsoft.com/com/tech/DCOM.asp>
- Dogac 1998 : *Special Issue on Electronic Commerce*. ACM Record Vol.27, Nb.4.
- ebXML (Electronic Business using eXtensible Markup Language) :  
voir <http://www.ebxml.org/>
- Fauvet et al. 2002 : *Collecting and Querying Distributed Traces of Composite Service Executions*, Proceedings of the 10th International Conference CoopIS02, Irvine, CA.
- Fremantle et al. 2002 : *Enterprise Services*, Communications of the ACM. Vol.45, Nb.10. octobre.
- Gamma et al. 1995 : *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, Reading MA, USA

48-3

- Gardarin 2002 : *XML. Des bases de données aux services Web*, Dunod, Paris. ISBN 2 10 006933 0.
- Harel et al. 1996 : *The STATEMATE Semantics of Statecharts*, ACM Transactions on Software Engineering and Methodology, vol.5, nb.4.
- HTTP (HyperText Transfert Protocol) : au dessus de TCP/IP.
- J2EE (Java 2 Platform, Enterprise Edition) :  
voir <http://java.sun.com/j2ee/>
- Java RMI (Java Remote Method Invocation) :  
voir <http://java.sun.com/products/jdk/rmi/index.html>
- Jennings et al. 2000 : *Autonomous Agents for Business Process Management*, Journal of Applied Artificial Intelligence, vol. 14, nb. 2.
- Kuno et al. 2001 : *Conversations + Interfaces = Business Logic* Proc. of the 2nd Workshop on Technologies for E-Services (TES), Rome. Italie.
- .Net (dot Net, Microsoft) :  
voir <http://www.microsoft.com/net/>
- RDF (Resource Description Framework) :  
voir <http://www.w3.org/RDF/>

48-4

- Sayal et al. 2001 : *Integrating Workflow Management Systems with Business-to-Business Interaction Standards*, HP Labs report HPL-2001-167.
- SMTP (Simple Mail Transfer Protocol).
- SOAP (Simple Object Access Protocol) :  
voir <http://www.w3.org/TR/SOAP/>
- SSL (Secure Sockets Layer) :  
voir <http://www.apache-ssl.org/>
- TCP/IP (Transmission Control Protocol/Internet Protocol).
- UDDI (Universal Description Discovery and Integration Standard) :  
voir <http://www.uddi.org/>
- W3C (World Wide Web Consortium) :  
voir <http://www.w3.org/>
- WfMC 1996 : *WorkFlow Management Coalition, Terminology and glossary*
- WSCL (HP Web Services Conversation Language) :  
voir <http://www.w3.org/TR/wscl10/>
- WSDL (Web Services Description Language) :  
voir <http://www.w3.org/TR/wsdl>

- WSFL (IBM Web Services Flow Language) :  
voir <http://www.ebpml.org/wsflhtm>
- XLang (Microsoft extension of WSDL) : v  
oir <http://www.ebpml.org/xlang.htm>
- XML (eXtensible Markup Language) :  
voir <http://www.w3.org/XML/>
- Yang et al., 2000 : *Interoperation support for electronic business*, Communication of the ACM, Vol. 43, Nb. 6.