

Ingénierie logicielle à base de composants

Didier DONSEZ (UJF/LIG/Adèle)
Lionel SEINTURIER (USTL/LIFL/Adam)

SCA : un modèle de composants pour le SOA

Constat : applications réparties (CORBA, .NET, Java EE, ...) actuelles

- souvent complexes, rigides, peu évolutives

Tendance SOA (Service Oriented Architecture)

- besoins identifiés
 - interfaces bien définies avec une sémantique lié au métier
 - protocoles de communication standardisés
 - recombinaison flexible de services pour améliorer la flexibilité du logiciel
- "vision" [Consortium Open SOA – www.osoa.org]
 - *A service is an abstraction that encapsulates a software function*
 - *Developers build services, use services and develop solutions that aggregate services*
 - *Composition of services into integrated solutions is a key activity*

SCA : un modèle de composants pour le SOA

Rappel SOA (Service Oriented Architecture)

- essentiellement des principes architecturaux (logiciels)
 - couplage faible
 - *Components integrate with other components without needing to know how other components are implemented*
 - flexibilité
 - *Components can easily be replaced by other components*
 - services
 - *Services can be easily invoked either synchronously or asynchronously*
 - composition
 - *Composition of solutions clearly described*
 - productivité
 - *Easier to integrate components to form composite application*
- souvent organisés autour des Web Services (mais pas nécessairement)
 - SOAP (HTTP, XML), WS-*, orchestration BPEL

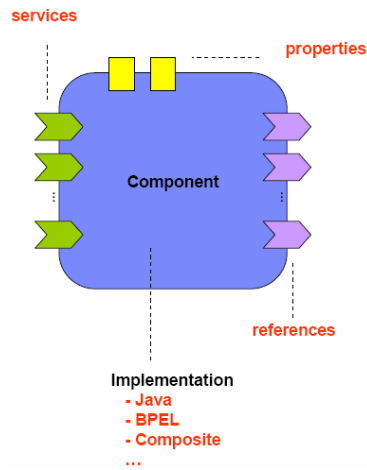
SCA : un modèle de composants pour le SOA

- Initiative : IBM, Oracle, BEA, SAP, Sun, TIBCO, ...
- But : structurer l'implémentation des SOA
- 1ères spécifications : 12/2005, v1 03/2007
- Implémentations : Apache Tuscany, Newton, fabric3

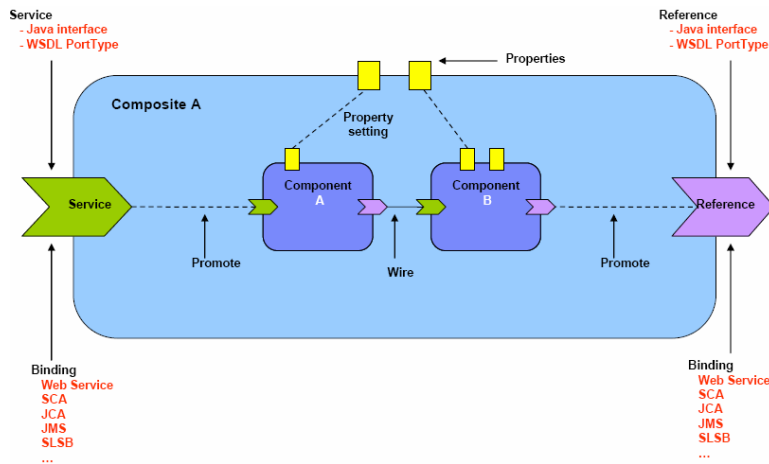
- Assembly model
 - how to define structure of composite applications
- Component implementations specifications
 - how to write business services in particular languages
 - Java, C++, PHP, Spring, BPEL, EJB SLSB, ...
- Binding specifications
 - how to access services
 - Web services, JMS, RMI-IIOP, REST, ...
- Policy framework
 - how to add infrastructure services
 - security, transactions, reliable messaging, ...

Composant SCA

- service/référence
- propriété
- Implémentation
 - Java, BPEL, JavaScript, Composite



Assemblage SCA



Assemblage SCA

■ service/référence

- type
 - langage de définition d'interface (IDL) : Java ou WSDL
- *binding* : association avec une technologie de communication
 - Web Service, RMI-IIOP, JMS, appel local intra-JVM

■ liens

- référence vers service : *wire*
- service/service ou référence/référence : lien de promotion (*promote*)

Modèle de programmation SCA

■ Assembly model

- ADL basé XML

■ Component implementations

- 1 ensemble de règles de programmation par langage
 - Java : utilise intensément annotations Java 5 (comme EJB3, ...)

■ Exemple

```
@Service(AccountService.class)
public class AccountServiceImpl implements AccountService {
    @Reference public AccountDataService accountDataService;
    @Reference public StockQuoteService stockQuoteService;
    @Property public String currency;
    public AccountReport getAccountReport(String s) { ... }
}
```

SCA Bilan

- *yet another component model*
 - une façon de revisiter les problématiques
 - de la répartition
 - des services pour les plates-formes *middleware*
 - des concepts des *frameworks* de composants
 - renforce l'idée de l'indépendance services/implémentations
- point original : indépendance interfaces/protocoles communications
- après CORBA, .NET
 - nouvelle tentative autour du "*middleware* universel" pour faire communiquer l'existant
- comme CORBA, .NET
 - met plus l'accent sur le modèle de programmation (*software engineering*) des applications que sur l'architecture de la plate-forme *middleware* (*product dependant*)
 - comment trouver la bonne abstraction pour l'ingénierie de la répartition