



Méthodes Informatiques Appliquées
à la Gestion des Entreprises

M2 MIAGE EVRY RAPPORT DE PROJET

TECHNOLOGIE SCA

Matière : Architecture orientée service
Enseignants : Boccon-Gibod, Godefroy
Étudiants : DIALLO Amadou Tidiane
GOLAB Barbara

IDENTIFICATION DU PROJET

Projet universitaire	N/A
Matière	Architecture orientée service
Publique visé	Professeurs, étudiants

	Nom	Fonction
Écrit par :	Golab Barbara	Étudiant M2 MIAGE
Écrit par :	Diallo Amadou T	Étudiant M2 MIAGE

Sommaire	Mots clés
Ce document présente l'initiative SCA	SOA, SCA, Tuscany, Eclipse, Java

Caractéristique du document

Nombre de pages	Nombre de figures	Langage
19	3	FR

GLOSSAIRE

sigle	Traduction
SOA	Architecture Orientée Service
SCA	Service Component Architecture
IDE	Environnement de développement intégré
STP	SOA Tools Platform
BPEL	Business Process Execution Language
BPMN	Business Process Modeling Notation
SOAP	Simple Object Access Protocol
HTTP	HyperText Transfer Protoco
RMI	Remote Method Invocation
OASIS	Organization for the Advancement of Structured Information Standards
JAVA	Langage de programmation objet
CORBA	Common Object Request Broker Architecture
JSONRPC	Protocole d'échange de données de RPC (Remote Procedure Call)

Sommaire

GLOSSAIRE	3
INTRODUCTION	5
I. PRESENTATION DE SCA.....	5
II. LES RUNTIMES SCA	6
III. ECLIPSE: SOA Tools Platform (STP)	7
IV. APACHE TUSCANY	8
V. CONTEXTE A ETUDIER	9
1. Gestion des réservations de vols.....	9
2. Gestion des réservations d'hôtels	11
3. Déroulement du projet	12
VI. MATURITE DE LA TECHNOLOGIE SCA.....	15
VII. COMPARAISON AVEC LES TECHNOLOGIES EXISTANTES.....	17
CONCLUSION.....	18
SOURCES	19

INTRODUCTION

Dans un contexte économique assez complexe, les entreprises informatiques tentent grâce à des technologies nouvelles, d'améliorer leur coût et leur rentabilité.

La technologie SOA (Architecture Orientée Service) se présente comme une alternative crédible permettant d'implémenter et de déployer des applications (ou services) pouvant être de natures différentes.

A cet effet, une initiative nommée SCA (Service Component Architecture) définit un ensemble de spécifications visant à simplifier la création et la composition de services (indépendamment de leur implémentation) dans le cadre d'Architectures Orientées Service (SOA).

Nous allons donc dans ce document, vous présenter grâce à un exemple concret, quelques possibilités qu'offre SCA. Et nous tenterons d'en étudier la maturité et l'intérêt qu'auront ou non les entreprises à l'adopter.

I. PRESENTATION DE SCA

SCA (Service Component Architecture) est une spécification d'implémentation des composants d'une architecture orientée services (SOA) proposant un modèle indépendant des langages de programmation et des protocoles de communication.

L'initiative SCA vise à généraliser l'approche orientée service des systèmes d'information et à découpler autant que possible les problématiques de conception d'application et les problématiques d'assemblage et de déploiement de composants.

SCA a été créé et lancé à la fin de 2005 par le consortium Open-SOA, qui a émis les spécifications 1.0 en mars 2007, et ensuite repris par le consortium OASIS qui depuis s'occupe de l'évolution de cette norme.

II. LES RUNTIMES SCA

Pour mettre en relation plusieurs applications pouvant être écrites dans des langages différents, SCA utilise un « runtime ».

En effet, le runtime est un moteur qui va être « uploader » au sein de l'IDE (environnement de développement) choisi afin que celui-ci crée les liens nécessaires.

Plusieurs initiatives tant commerciales qu'open source ont permis de créer et définir des runtimes. En voici quelques exemples :

- **Commerciales :** HydraSCA, IBM Websphere Process Server, RogueWave Hydra Suite, TIBCO ActiveMatrix Service Grid
- **Open source :** Apache Tuscany, Fabric3, SCOrWare, SOA PHP, Newton

Dans le domaine open source il existe aussi Eclipse SOA Tools Platform, qui permet de construire des frameworks et des outils permettant la conception, la configuration, l'assemblage, le déploiement, le suivi et la gestion des logiciels conçus autour d'une architecture orientée services.

Tous ces outils contribuent à l'amélioration sans cesse constante de l'utilisation de SCA. Nous allons par la suite vous présenter un runtime open source « Tuscany ».

III. ECLIPSE: SOA Tools Platform (STP)

Le but du projet STP est de fournir les outils nécessaires à la mise en œuvre d'architectures orientées services en se conformant aux travaux sur une nouvelle spécification SCA.

Le projet STP prévoit de fournir des outils destinés aux architectes et aux développeurs. Ces outils couvriront les différentes phases de la mise en place d'une architecture SOA : conception, configuration, assemblage, déploiement et supervision.

Le projet STP se divise en 5 sous projets :

- STP Core Framework (CF) : Ce sous-projet sert de base aux autres sous-projets STP. Il propose un framework permettant de manipuler les notions définies dans la spécification SCA.
- STP SOA System (SOAS) : SOAS a pour but de couvrir la phase de développement en proposant des outils pour assembler, tester, déboguer et exporter des services.
- STP Service Creation (SC) : SC se focalisera sur l'implémentation d'assistants pour la création et l'édition des interfaces des services.
- STP BPEL 2 Java (B2J) : B2J propose un générateur de code Java à partir de définitions de processus métier au format BPEL. Ce projet cible la phase déploiement, en plus de la conversion en classes Java d'un processus BPEL, le sous-projet B2J propose un moteur d'exécution sur lequel s'appuient les classes générées. B2J n'a pas pour objectif de proposer d'éditeur BPEL.
- STP BPMN : Le sous-projet BPMN propose des outils de conception visuelle de processus conforme à BPMN (Business Process Management Notation).

Ces sous projets sont pour la majorité d'entres-eux, toujours en développement. Néanmoins, il est possible de définir des composants SCA à partir d'Eclipse.

IV. APACHE TUSCANY

Apache Tuscany est un groupe de travail gérant les trois sous-projets relativement indépendants :

1. Tuscany SCA (Service Component Architecture)
2. Tuscany SDO (Service Data Object)
3. Tuscany DAS (Data Access Service)

Le sous-projet Tuscany SCA simplifie la tâche du développement des solutions SOA en fournissant une infrastructure globale basée sur la technologie SCA.

Les premiers travaux sur ce sous-projet de Tuscany ont commencé en 2006.

Tuscany SCA offre aux développeurs les avantages suivants :

- conforme aux spécifications définies par OASIS Open CSA
- deux environnements d'exécutions : Java et C++
- large éventail de bindings : SOAP/HTTP, RMI, EJB, JSONRPC, Atom, HTTP, RSS, DWR, Notification, JMS, AJAX
- diverses implémentations de composants : Java, C++, BPEL, Spring, JavaScript, Groovy, Ruby, Python, XQuery
- support de JAXB, SDO et AXIOM
- compatibilité avec les différentes plateformes : Tomcat, Geronimo, Eclipse, WebSphere, WebLogic
- légèreté du runtime qui peut fonctionner de façon autonome ou avec d'autres serveurs d'application
- architecture modulaire qui permet d'intégrer différentes technologies
- intégration avec les technologies web 2.0

V. CONTEXTE A ETUDIER

Afin d'étudier la technologie SCA sur un cas concret, nous avons choisi de l'implémenter dans un système de réservation de vols et d'hôtels. Ce système sera accessible via une interface, dans laquelle l'utilisateur pourra récupérer les informations recherchées concernant son voyage et ensuite (ou simultanément) faire une réservation s'il le souhaite. Cette interface communiquera avec deux services distincts :

- gestion de réservations de vols
- gestion de réservations d'hôtels.

Chacun de ces services seront d'abord gérés indépendamment, puis simultanément en cas de besoin.

1. *Gestion des réservations de vols*

L'une des applications à exploiter concerne la gestion de réservations de vols. On suppose dans notre cas, qu'elle ne concerne qu'une compagnie aérienne (pas de possibilités d'avoir des offres d'autres compagnies) et qu'il n'y a pas de possibilité que des vols se chevauchent, cela veut dire qu'à une heure donnée, il ne peut y avoir qu'un seul vol au départ ou à l'arrivée. De plus, pour faire simple, on ne gère pas les escales éventuelles.

La gestion des vols d'une compagnie aérienne suppose d'avoir un certain nombre d'informations :

- pays de départ
- ville de départ
- pays d'arrivée
- ville d'arrivée
- date de départ

- heure de départ
- date d'arrivée
- heure d'arrivée
- nombre de passagers

Chacune de ces informations va être exploitée afin d'optimiser au mieux les résultats possibles. Voici un modèle conceptuel de données représentant l'application de gestion de vols :

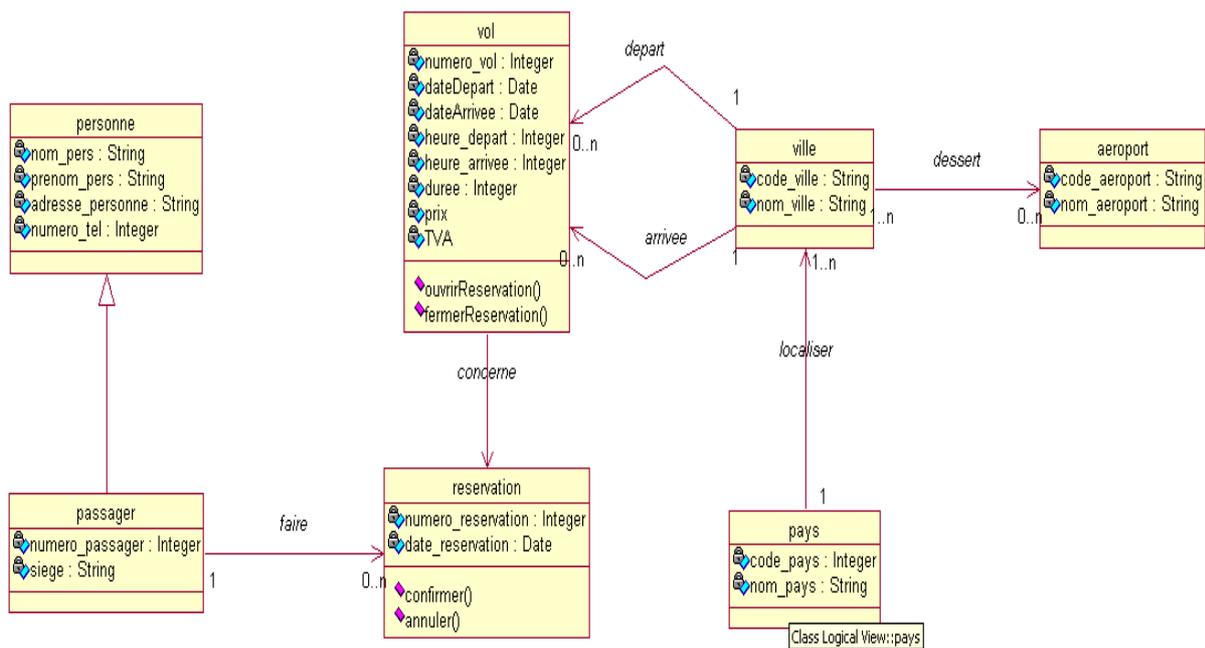


Figure 1 : diagramme de classes de la réservation de vol

2. Gestion des réservations d'hôtels

La deuxième application concerne la gestion des réservations d'hôtel. L'application gère au plus simple les réservations de chambres. En effet, par défaut, une chambre est réservée pour deux personnes au maximum. Dans le cas d'une réservation pour une famille de 3, il est impératif de prendre 2 chambres.

La prise des clés se fait toujours à la même heure (c'est-à-dire 12h). Donc même-ci le client arrive plus tard, la réservation du jour commence à 12h. De la même façon, l'heure de départ est toujours la même (c'est-à-dire 12h). Les chambres sont occupées par tranche de 24h (de 12h à 12h le lendemain).

Par soucis de simplicité, il n'y a pas d'options possibles et tous les hôtels proposés sont de niveau équivalent (par défaut 3 étoiles).

Voici le diagramme de classe représentant l'application :

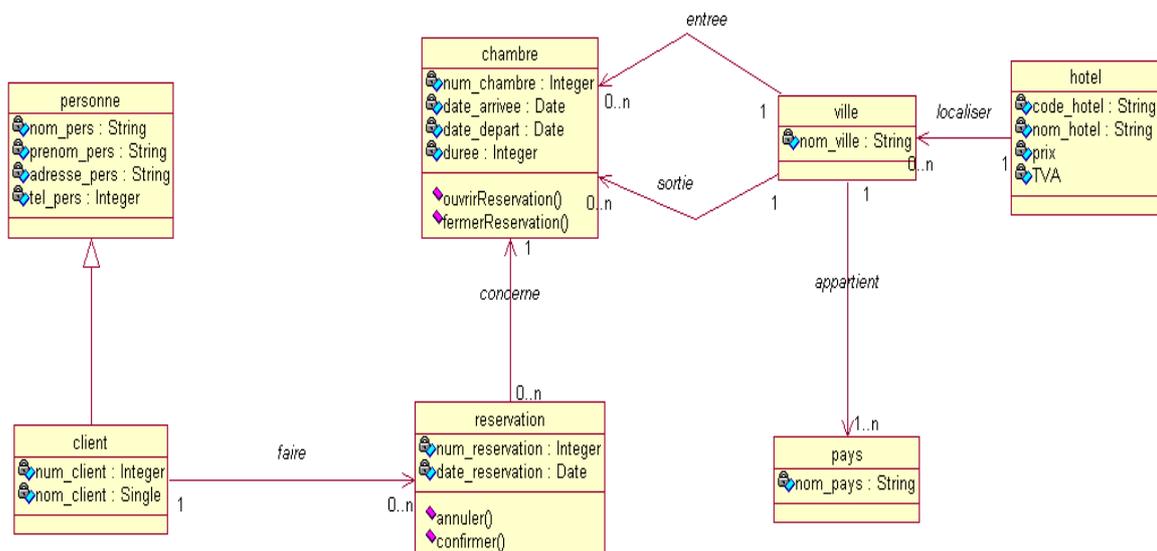


Figure 2 : diagramme de classes de la réservation d'hôtel

3. *Déroulement du projet*

Le projet consiste à tester sur un exemple donné (ici les réservations de vols et d'hôtels, soit séparément soit « simultanément ») un runtime permettant de faire du SCA. Le runtime choisi pour ce projet est « Tuscany ». Ce choix est motivé par le fait qu'il est gratuit et que de la documentation existe déjà dessus.

La première partie du projet est la création du composite de notre système de réservation.

La deuxième partie du projet est l'implémentation en JAVA des différentes applications de réservations de vols et d'hôtels et leur connexion au composite.

La troisième partie du projet est l'implémentation du composant qui va communiquer avec les deux applications. Ce composant est réalisé en HTML et JavaScript.

La quatrième partie consiste à relier nos applications grâce à « Eclipse » et « Tuscany ».

La dernière partie correspond à définir l'apport de SCA dans la communication entre différentes applications par rapport aux technologies qui existent déjà (sans runtime).

Voici un schéma SCA de l'exemple réalisé :

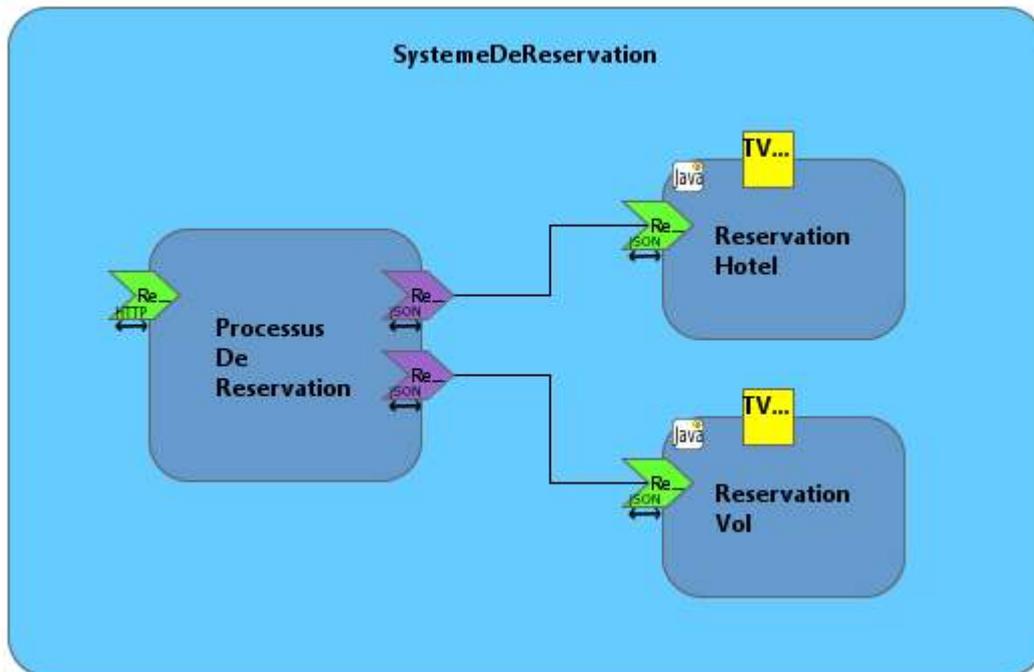


Figure 3 : composite de réservation

Comme on peut voir sur le schéma dessus, notre système est composé de trois composants :

- Processus de réservation :

Le processus de réservation est un composant qui va jouer le rôle d'interface entre l'utilisateur et les différentes applications. Ce composant fournit des informations vers l'extérieur à travers le protocole HTTP. Il possède deux « références » vers les deux autres applications qui interagissent avec les services de ces applications en utilisant le protocole JSONRPC. Ce composant va être implémenté en HTML et JavaScript.

- Réservation Hôtel :

Le processus de réservation d'hôtel est un composant indépendant qui fournit un service permettant de récupérer les informations sur les hôtels. Ce composant interagit avec le composant de processus de réservation (précédemment présenté) en utilisant le protocole JSONRPC. Le composant est implémenté en JAVA et possède une propriété concernant la TVA sur les hôtels.

- Réservation Vol

Le processus de réservation de vol est un composant indépendant qui fournit un service permettant de récupérer les informations sur les vols. Ce composant interagit avec le composant de processus de réservation (précédemment présenté) en utilisant le protocole JSONRPC. Le composant est implémenté en JAVA et possède une propriété concernant la TVA sur les vols.

Nous avons choisi le protocole JSONRPC dans notre projet, car c'est est un format léger d'échange de données. Il est basé sur un sous-ensemble du langage de programmation JavaScript. JSON est un format texte complètement indépendant de tout langage.

VI. MATURETE DE LA TECHNOLOGIE SCA

Pour étudier la maturité de cette technologie, nous allons prendre en compte trois principaux facteurs :

- Acteurs qui font les spécifications,
- Acteurs qui produisent la technologie
- Acteurs qui l'utilisent

Tout d'abord il faut noter que c'est OASIS (Organization for the Advancement of Structured Information Standards) qui s'occupe de la spécification de SCA. C'est une organisation de référence, qui collabore avec au moins 4000 participants représentant souvent les sociétés les plus importantes dans le domaine de nouvelles technologies.

Comme nous l'avons mentionné auparavant, il existe plusieurs runtimes SCA, certains sont payants et d'autres gratuits. L'avantage de la technologie open source est sa robustesse, et son adaptabilité aux besoins de clients. Tuscany, qui est sous licence de Apache est l'une d'entre elle. La première version de son runtime est sortie peu après le lancement de la technologie SCA, et depuis n'arrête pas d'évoluer.

Aujourd'hui, les entreprises sont à la recherche de nouvelles technologies qui leur garantiraient la flexibilité dans le déploiement de leurs processus métiers, qui deviennent de plus en plus complexes. Cela leurs permet de faire des économies car les technologies sont de plus en plus couteuses. SCA fournit justement un modèle simple qui permet de gérer les applications métiers de façon plus générale, en simplifiant leur développement.

En prenant en compte le niveau d'expertise et le temps d'existence de la technologie, SCA a-t-elle vraiment atteint la maturité ? Nous pensons qu'au travers de notre étude, ce n'est pas encore le cas, car même si la technologie a bien évoluée, elle reste encore assez jeune, et en constant développement. Il est en effet assez difficile de trouver des informations d'installation et de déploiement, fournies sur les différents

runtimes, notamment ceux qui sont open source. Nous avons eu des difficultés à installer et tester des runtimes tels que « Fabric3 » ou « SCOrWare », du fait de la pauvreté de leur documentation et de la multitude de composants à intégrer, rendant difficile la configuration. Concernant les produits commerciaux, leurs prix ne nous permettaient pas de les tester. Nous avons pu constater que des produits tels que « Websphere » d'IBM, sont en plein développement et sont même considérés comme stratégiques. Ces produits commerciaux bénéficient donc d'équipes dédiées à leur développement, ce qui leur permet aujourd'hui d'atteindre une certaine maturité, notamment en ce qui concerne l'installation et le déploiement.

VII. COMPARAISON AVEC LES TECHNOLOGIES EXISTANTES

Aujourd'hui, plusieurs entreprises se trouvent confrontées à une problématique concernant la coordination de leurs applications informatiques et de leurs systèmes d'information en réseau. En informatique, le middleware aussi appelé intergiciel est un logiciel intermédiaire qui permet aux systèmes d'application de communiquer entre eux. Trouver une solution à cette problématique est devenu un élément stratégique pour les entreprises.

Plusieurs initiatives telles que CORBA, DOTNET, J2EE et autres ont déjà été mis en œuvre pour apporter des solutions. Aujourd'hui on se tourne vers les Architectures Orientées Services pour pallier aux carences de ces technologies.

SCA est donc un ensemble de spécifications visant à simplifier la création et la composition de services (indépendamment de leur implémentation) dans le cadre d'Architectures Orientées Service (SOA). De ce fait, il y a plusieurs recherches sur le sujet qui promettent énormément.

CONCLUSION

SCA est une technologie qui présente sans doute du potentiel. Dans le domaine de SOA, l'hétérogénéité d'applications s'échangeant les informations entre elles, fait de SCA une solution intéressante.

Malgré son développement encore en cours, cette technologie présente des possibilités intéressantes et permet d'apporter des solutions dans la problématique de l'association de services implémentés dans des langages différents.

Il est vrai qu'aujourd'hui, SCA n'a pas encore atteint son niveau de maturité et a encore du mal à s'imposer. Néanmoins, les informations trouvées sur internet, nous permettent d'affirmer que la technologie SCA a un très grand intérêt pour les entreprises, dans leur vision des architectures informatiques du futur.

SOURCES

<http://blog.xebia.fr/2007/04/11/introduction-a-sca-service-component-architecture/>

http://www.progress.com/progress_software/worldwide_sites/fr/products/docs/sonic_model_0015.pdf

<http://www.eclipsetotale.com/articles/Eclipse-SOA.html>