

# GICOM : un atelier pour l'expérimentation des technologies de systèmes distribués d'entreprise

Fabienne Boyer\*+, Sébastien Chassande-Barrioz\*, Didier Donsez+,  
David Féliot\*, Sacha Krakowiak\*+

+Université Joseph Fourier, IMA

\*INRIA Rhône-Alpes, Projet Sardes

{prénom.nom}@imag.fr

Catégorie : Atelier

**Résumé.** Cet article présente un atelier destiné à l'enseignement pratique des technologies de systèmes répartis d'entreprise, utilisé depuis 5 ans au DESS de Génie Informatique de l'université Joseph Fourier, et depuis cette année dans la formation d'ingénieurs "Réseaux Informatiques et Communications Multimédia" de cette université. Cet atelier s'appuie sur la réalisation d'un serveur de commerce électronique en intégrant les principales techniques actuelles (Java, CORBA, *servlets*, EJB, services web). L'expérience montre que ce projet, par son caractère intégrateur et sa taille réaliste, contribue fortement à une bonne compréhension de l'architecture des systèmes répartis et à l'acquisition du savoir-faire nécessaire à la maîtrise des outils.

**Mots clés :** Atelier pour l'enseignement, J2EE, CORBA, Persistance, Transaction, Sécurité.

## 1 Motivations, objectifs et contexte

La mise en œuvre globale d'une application d'entreprise fait appel à de très nombreux outils et techniques relevant des systèmes distribués et des systèmes d'information, qui sont généralement enseignés dans les formations de 3<sup>ème</sup> cycle professionnalisées (DESS, Mastère, 3<sup>ème</sup> année d'école d'ingénieur en informatique). Ces techniques et ces outils évoluent très rapidement et présentent un haut degré de complexité. Les étudiants qui doivent concevoir et réaliser une application d'entreprise lors de leurs premières expériences professionnelles (stage, premier emploi) sont souvent désorientés face au nombre des techniques à maîtriser et au problème de leur intégration.

L'atelier GICOM répond à un double objectif :

- Faciliter la bonne compréhension des principes des systèmes répartis en leur donnant une application concrète.
- Favoriser l'acquisition de savoir-faire, par la maîtrise de l'usage des techniques et outils correspondant à l'état de l'art, et de leur intégration au sein d'un projet complexe.

Atteindre ces objectifs nécessite de mettre en œuvre un projet de taille réaliste, sur une période suffisamment longue, en intégrant des techniques variées. Le projet GICOM a été conçu dans cet esprit. Il comporte la réalisation d'une application de taille relativement importante (un serveur de commerce électronique) en intégrant un ensemble de techniques (programmation répartie, systèmes de bases de données, sécurité) et en utilisant les outils courants de la profession (J2EE, CORBA, EJB, *servlets*, etc.). Par sa nature, ce projet est facilement extensible pour prendre en compte de nouveaux besoins ou intégrer de nouveaux développements techniques (mobilité, web services B2B, etc.).

L'atelier GICOM est constitué d'un ensemble de plates-formes pré-installées, avec leur documentation, leurs scripts de configuration et de lancement ainsi que des scripts d'initialisation des bases de données des fournisseurs. Il est accompagné d'une documentation de 57 pages qui décrit l'architecture de l'application, les étapes de réalisation, la conception des services, les technologies à mettre en œuvre. Le texte complet (version 2001-2002) est disponible à partir de l'URL : <http://www-adele.imag.fr/~donsez/ujf/common/GICOM>.

Ce projet a initialement été développé depuis 1997-98 à l'université Joseph Fourier dans le cadre de l'option "Systèmes Répartis et Réseaux" du DESS de Génie Informatique, qui regroupe environ 25 étudiants. Depuis l'année 2001-2002, il a également été introduit dans l'option "Architectures de réseaux" de la formation d'ingénieurs RICM ("Réseaux Informatiques et Communications Multimédia") qui compte également 25 étudiants. Il s'agit donc d'un public spécialisé, dans des formations à vocation professionnelle.

## 2 Le projet de serveur de commerce électronique

L'objectif du projet est de concevoir et de réaliser une application distribuée, dont la mise en œuvre utilise deux principaux types de services sous-jacents :

- Services d'intégration et de coordination de composants existants (CORBA)
- Communication sur un réseau à grande distance via le World Wide Web (HTTP, *applets*, *servlets*)

L'application est un **serveur de commerce électronique** (appelé GICOM, pour DESS **GI-Commerce**) de type "galerie marchande", permettant à des clients de consulter et d'acheter des produits par voie électronique, au travers du World Wide Web. La suite de cette section décrit :

- les fonctions du serveur de commerce électronique GICOM (2.1),
- l'architecture globale proposée pour la réalisation du serveur (2.2),
- les étapes de réalisation (2.3),
- l'organisation pratique et le calendrier du projet (2.4).

### 2.1 Fonctions du serveur

Le serveur GICOM fournit des services à deux types de partenaires, les clients et les fournisseurs de produits. Les services fournis à un client sont les suivants :

- Consultation de catalogues de produits
- Remplissage d'un caddie
- Demande d'achat des produits présents dans le caddie

La suite des actions effectuées par un client donné entre le moment où il entre dans la galerie électronique et le moment où il en sort est appelée **session**. Le serveur de commerce est capable de traiter plusieurs sessions en parallèle. La terminaison d'une session peut être demandée par le client au travers de l'interface graphique du serveur, ou bien provoquée par le serveur lorsque la session atteint une durée maximale. Cela permet d'assurer que tout client qui oublie de se déconnecter du serveur le sera automatiquement au bout d'un temps fini.

Le serveur de commerce maintient une base d'informations dans laquelle sont décrits les fournisseurs de produits disponibles (nom, adresse, ...), ainsi que les produits disponibles (nom, prix, description, ...) dans leurs catalogues respectifs. L'arrivée d'un nouveau fournisseur implique que celui-ci s'enregistre auprès de l'administrateur de la base d'information du serveur. Une fois enregistré, et avec l'aide de l'administrateur de la base d'information, un fournisseur peut créer un nouveau catalogue, et par la suite modifier ceux existants.

Lorsque le client décide de valider ses achats, il émet un **ordre d'achat** vers le serveur de commerce, comprenant l'identification du client (nom, adresse, numéro de CB, etc.), ainsi que l'identification des produits achetés et de leur fournisseur. Le serveur procède alors ainsi :

En premier lieu, il effectue les transactions bancaires relatives aux produits achetés, en communiquant avec les organismes financiers concernés.

En second lieu, si le fournisseur est informatisé et accessible par le réseau, l'ordre d'achat le concernant lui est transmis (par exemple, par courrier électronique). L'ordre d'achat est dans tous les cas stocké dans une base locale d'ordres à envoyer aux fournisseurs par courrier papier.

Enfin, le serveur renvoie au client une clé lui permettant de prouver son achat en cas de litige. Comme pour les fournisseurs, cet envoi peut être fait à la fois par courrier électronique et par courrier papier.

Ces trois étapes nécessitent de respecter la cohérence du système comprenant le serveur de commerce, les banques, les fournisseurs et les clients. Il est donc nécessaire de garantir des propriétés d'atomicité, de sécurité, de protection (dans la pratique, en fonction du temps disponible, certains projets peuvent ne pas assurer certaines de ces garanties).

### 2.2 Architecture et Technologies

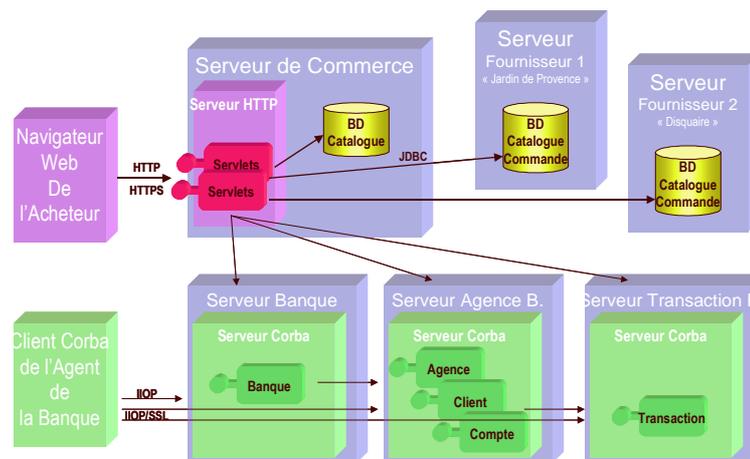
L'architecture globale du serveur comporte les parties suivantes (Figure 1) :

**Navigateur web** : il permet à un client de parcourir la galerie et de réaliser des achats. .

**Serveur de commerce** : il est décomposé en un ensemble de services (entrée d'un client dans la galerie, consultation d'un catalogue de produits, gestion d'un ordre d'achat, ...). Ces services sont réalisés par des **servlets**, ce qui permet d'optimiser les performances d'exécution du serveur et de conserver un contexte d'exécution (session) entre les services réalisés. La gestion des sessions clientes au niveau du serveur fait appel au mécanisme de *cookies* de HTTP.

**Serveurs fournisseurs** : ils gèrent en outre des données persistantes (catalogues de produits, commandes en cours, etc.), au niveau desquelles il est utile de pouvoir effectuer des accès associatifs. Ces données sont conservées dans une base de données de type **SQL**, à laquelle le serveur accède au travers d'un *driver JDBC*.

**Applications "banques"** : Les banques sont gérées par des serveurs **CORBA**, qui fournissent des fonctions de consultation et de modification de données bancaires (débits, crédits, etc.) que le serveur de commerce peut utiliser via l'interface d'accès à distance de CORBA. Les banques fournissent aux clients, sous la forme d'*applets*, une interface graphique qui permet à un client d'effectuer diverses opérations (consultation du solde, virement entre comptes, etc.) au travers du World Wide Web.



**Figure 1 : Architecture globale de l'atelier GICOM**

### 2.3 Les étapes de réalisation

L'application de commerce électronique est réalisée en 6 étapes décrites ci-après. Ce déroulement en étapes a plusieurs avantages : permettre une introduction progressive des différents outils et faciliter l'assimilation des notions introduites ; donner une certaine assurance aux étudiants en leur permettant de faire fonctionner assez tôt des morceaux autonomes d'application ; définir des étapes d'intégration, avec possibilité de retour en arrière. Le langage support choisi est Java, pour faciliter la portabilité de l'application et permettre l'utilisation de plates-formes technologiques J2EE [<http://java.sun.com/j2ee>] ne requérant pas de licences payantes : Apache et en particulier Jakarta/Tomcat pour HTTP, JDBC et MySQL pour les bases de données, JOnAS (ObjectWeb) pour les *Enterprise Java Beans* (EJB), ORBacus pour CORBA. Le projet fonctionne sur plates-formes Unix et Windows NT.

**Étape 1 - Serveur de commerce sans le paiement.** Cette étape consiste à réaliser le serveur de commerce électronique, sans se soucier de son interaction avec les applications bancaires. On s'intéresse ici à l'interface Web (*servlet*) du serveur de commerce qui permet à l'acheteur de parcourir les catalogues dans les bases de données des fournisseurs et de remplir d'achats son panier virtuel. Cette étape nécessite un caddie générique (indépendant de la structure du catalogue du fournisseur), la réalisation du suivi de session à l'aide des *cookies* et d'un *pool* de connexions JDBC vers le serveur de base de données.

**Étape 2 - Serveurs bancaires.** Cette étape consiste à réaliser une application de gestion d'une banque, qui fournit des services de consultation et de modification des données bancaires. Chaque banque est mise en œuvre par un ensemble de serveurs CORBA représentant le site central de la banque et les sites de ses agences. Chaque site d'agence gère des clients et leurs comptes sous forme d'objets CORBA manipulables à distance. Une application réalise les opérations courantes effectuées par un collaborateur de la banque ou par un client depuis un guichet : création et fermeture de compte, consultation du solde, virement. Cette application

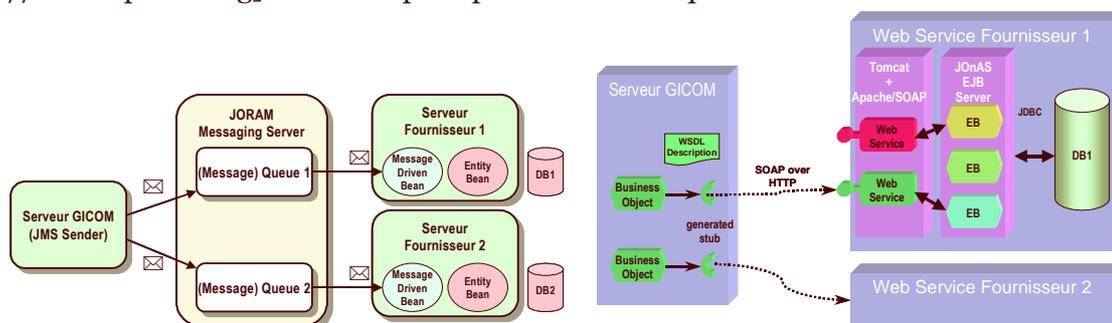
possède une interface graphique accessible par le Web, et mise en œuvre par une *applet*. Les opérations de virement peuvent être intégrées dans le serveur de commerce à cette étape.

Étape 3 - Persistance des informations bancaires. Dans l'étape 2, les objets représentant les informations bancaires (banque, agence, client et compte) ne sont pas persistants et disparaissent donc lors de l'arrêt de leur serveur. L'étape 3 consiste à rendre ces informations persistantes et accessibles lorsque le serveur est relancé. D'autre part, les mises à jour doivent être atomiques par rapport au média de sauvegarde. Pour cela, un Gestionnaire d'Objets Persistants (GOP) est mis en œuvre. Il récupère et sauvegarde les objets en les sérialisant/désérialisant vers/depus des fichiers, et gère également les références distantes persistantes. La sauvegarde est rendue atomique par des fichiers "ombres" pour éviter la corruption des fichiers (écriture incomplète) à la sauvegarde. Cette étape permet à l'étudiant de manipuler le *Portable Object Adapter* de CORBA.

Étape 4 – Fiabilisation des ordres d'achat. Cette étape réalise l'intégration des programmes réalisés dans les étapes 2 et 3, de manière à permettre à un client de l'application de commerce électronique de passer des ordres d'achat de produits. Le serveur de commerce accède aux banques pour débiter le compte d'un client et pour créditer les comptes des fournisseurs des produits achetés. Cet ordre de virement peut impliquer plusieurs serveurs simultanément. En cas de panne d'un de ces serveurs ou d'erreur (solde insuffisant sur le compte de l'acheteur par exemple) pendant cette opération de virement, le système doit revenir à un état antérieur cohérent. À cet effet, les opérations sont réalisées de manière transactionnelle. Les transactions gérant des ordres d'achat impliquent des serveurs distants (serveur de commerce et serveurs bancaires) et sont donc distribuées. Un Gestionnaire de Transactions Distribuées fiabilise leurs terminaisons selon le protocole de validation à deux phases. Les accès concurrents aux objets sont également contrôlés par le Gestionnaire d'Objets Persistants qui garantit la propriété de sérialisabilité en gérant la concurrence des accès selon la méthode de verrouillage à deux phases. De plus, la résolution des interblocages entre serveurs est effectuée en temporisant la durée de vie d'une transaction.

Étape 5 – Sécurisation des échanges. Cette étape vise à assurer la confidentialité et l'intégrité des données échangées entre les serveurs, ainsi que l'authentification des entités qui interagissent. La méthode utilisée pour garantir ces propriétés utilise la cryptographie à clé publique et à clé secrète. Le protocole SSL/TLS est mis en œuvre entre les navigateurs et le serveur de commerce (HTTPS) et entre le client CORBA et les différents objets CORBA (IIOP over SSL).

Étape 6 - Serveur du fournisseur. Le serveur du fournisseur est réalisé au moyen des *Enterprise Java Beans*. Cette étape fait l'objet de 2 options pour la transmission des commandes à préparer et expédier aux acheteurs du serveur de commerce vers les serveurs des fournisseurs. La première option (figure 2a) réalise la transmission asynchrone des commandes au moyen de la Messagerie Inter-Applicative (MOM) JORAM d'ObjectWeb, compatible avec l'interface normalisée JMS. La seconde option (figure 2b) réalise ces transmissions en se basant sur les protocoles de base des Web Services (SOAP, ...). Le *package* SOAP d'Apache [<http://www.apache.org>] est utilisé pour publier les *Enterprise Beans* du service.



**Figure 2 : L'envoi des commandes vers les fournisseurs en mode asynchrone (a) et avec les WebServices (b)**

## 2.4 Déroulement et évaluation

Les étudiants sont organisés en binômes. Chaque binôme travaille de manière indépendante, et des solutions parfois très différentes sont adoptées.

Le projet se déroule dans la seconde période de l'année, période où les étudiants sont relativement disponibles et ont acquis une certaine expérience grâce aux projets de la première période. Les étudiants sont répartis en deux groupes et le calendrier est le suivant.

- 11 séances de 3H, assistées par un enseignant, comprenant :
  - 5 séances communes aux 2 groupes, dans lesquelles sont présentés les principes des technologies utilisées,
  - 6 séances propres à chaque groupe, dans lesquelles un enseignant supervise la mise en œuvre de l'application, étape par étape.
- Un stage bloqué sur une semaine (5 journées), également assisté par un enseignant (en partie sur place, en partie en consultation à distance par courrier électronique). Cette période est consacrée à l'achèvement des modules, à leur intégration, à la mise au point, à la préparation de la démonstration, et à la rédaction de la documentation.

À l'issue du stage, une démonstration est demandée à chaque binôme. Cette démonstration doit permettre d'apprécier l'état d'avancement du projet, le respect des spécifications, et mettre en évidence les problèmes rencontrés et la démarche adoptée pour les résoudre.

### 3 Retours d'expérience

Le projet reçoit en général un accueil très favorable auprès des étudiants. Ceux-ci sont conscients de l'importance de la charge de travail qu'il implique, mais aussi du gain d'expérience et de savoir-faire qu'il leur apporte. Un projet intégré sur une longue période est également perçu plus favorablement qu'un ensemble de TP disjoints, et le défi technique proposé a un caractère stimulant certain. La durée du projet et la diversité des technologies oblige l'organisation en mini-équipe se répartissant les tâches. Enfin, la réalisation d'un tel projet et la connaissance pratique d'un important éventail d'outils est un élément non négligeable dans la présentation par les étudiants de leur savoir-faire à des futurs employeurs.

De leur côté, les enseignants constatent une forte implication des étudiants dans le projet, et une amélioration visible de leur savoir-faire. Le fait de maîtriser des outils professionnels augmente également leur confiance en eux. Certaines démonstrations atteignent chaque année un niveau de qualité remarquable.

Le projet a nécessité un important effort initial pour le développement de la plate-forme et de la documentation en ligne (le prototype initial a été réalisé au cours d'un projet d'élève ingénieur, suivi de 3 mois d'ingénieur à plein temps). Sa maintenance, son amélioration continue, et la mise à jour de sa documentation constituent encore une charge permanente non négligeable. Enfin, la bonne marche du projet nécessite un fort taux d'encadrement et une certaine disponibilité des enseignants pour l'assistance technique.

### 4 Conclusion et perspectives

Après 5 ans d'expérience, nous considérons que le projet a atteint les objectifs initialement fixés, et constitue un élément essentiel des formations où il est proposé. Grâce à sa structure modulaire, le projet a pu évoluer en permanence pour intégrer les nouveaux outils et standards.

Un tel projet représente un investissement important sur une longue durée, tant pour son développement initial que pour sa mise à jour, son suivi, et l'élaboration de sa documentation. Eu égard aux bénéfices constatés, nous estimons que cet investissement est largement justifié.

Les perspectives du projet GICOM sont multiples. Mentionnons entre autres :

- Introduction de la mobilité. Une expérience a déjà été faite en ce sens cette année (dans le cadre d'un TP annexe) pour le support de terminaux mobiles et contraintes WAP/WML et Java2 Micro Edition.
- Remplacement du service CORBA d'ORBacus par celui de la plate-forme Jonathan d'ObjectWeb. L'objectif est notamment de distribuer largement GICOM comme démonstrateur et tutoriel pour les technologies d'ObjectWeb [<http://www.objectweb.org>].

Le texte complet du projet (version 2001-2002) est disponible à partir de l'URL : <http://www-adele.imag.fr/~donsez/ujf/common/GICOM>