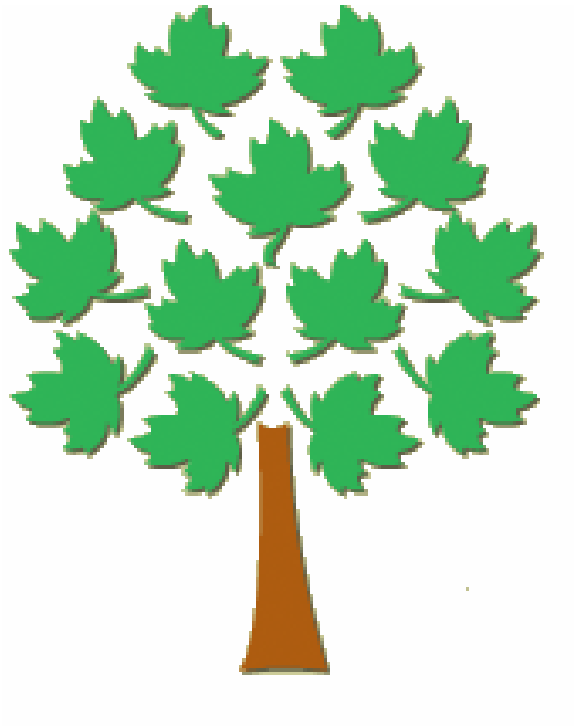
Cours de Systèmes Intelligents
M. Pierre Gançarski
pierre.gancarski@dpt-info.u-strasbg.fr

Dossier rendu par :
Romary Fabien
fabienromary@ifrance.com



JATLiteBean

Agent Intelligent

Année scolaire 2003/2004

Introduction

Depuis toujours en informatique, l'un des soucis majeur a été de pouvoir réutiliser l'existant. Avec les méthodologies objets, il est déjà possible de réutiliser des morceaux de code source pour divers types d'applications. Les applications mises en place sont souvent complexes, et il est difficile, voire impossible d'en concevoir un arrêt total. Les systèmes basés sur des agents intelligents donnent une autonomie à chaque agent, qui prit de façon unitaire peuvent s'adapter à un nouvel environnement et prendre les décisions pour intervenir.

JatLiteBean que nous allons étudier dans ce dossier, est basé sur JatLite Agent Toolkit. Pour cette raison nous allons accorder la première partie à JatLite. Ensuite nous étudierons le principe et le fonctionnement de JatLiteBean, puis les domaines d'applications, les avantages et inconvénients, pour finir par un exemple d'application.

1 JatLite

JatLite Agent Toolkit, projet mené par l'université de StandFord (USA). JatLite signifie Java Agent Template.
JatLite, n'est pas vraiment un projet, mais plutôt un ensemble de classe Java permettant de développer un système multi agent.

JatLite est doté de possibilités spécifiques de communication et d'interaction. Il est ainsi possible de créer rapidement des applications utilisant des agents intelligents par le biais d'Internet.
Aucune théorie ou technique des agents autonomes n'est imposé, la construction des agents intelligents est laissée au développeur.

1.1 Agent Message Router (AMR) infrastructure

JATLite agents
- Solution portable, sur tous type d'OS supportant Java (machine virtuel)
- authentification par nom d'utilisateur et mot de passe (login/password), connection/déconnexion d'internet, envoi et réception de messages
- port standard des couches TCP/IP pour la communication

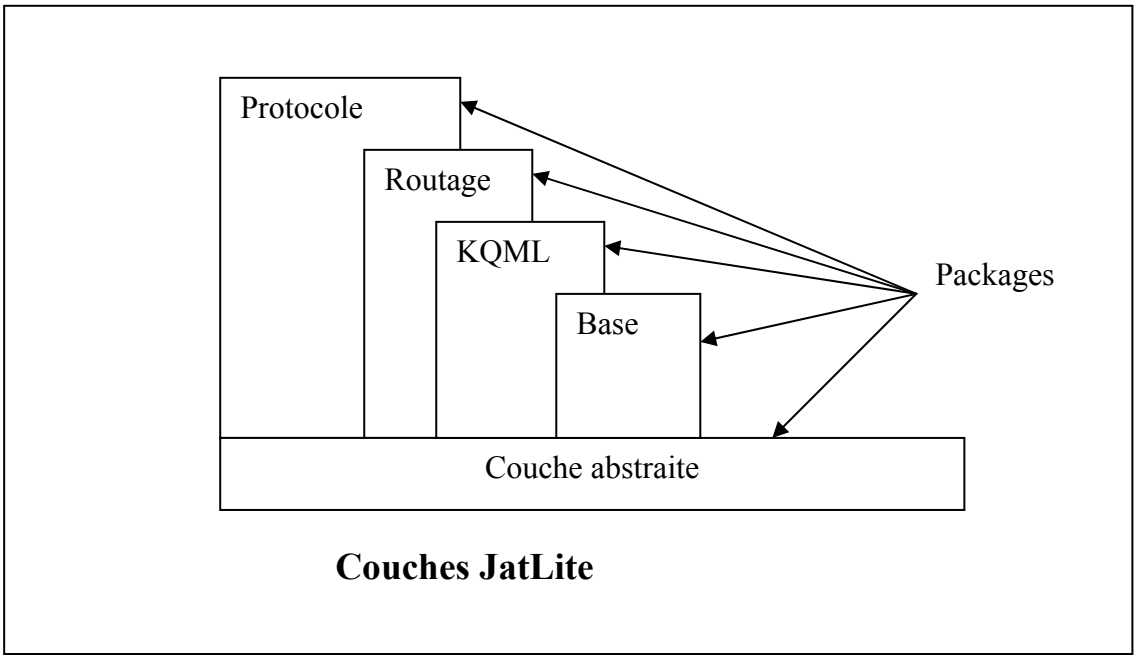
Message Router
- programme Java indépendant tournant sur une machine connectée à Internet
- communication entre les agents (routage des messages), et communication des agents via (internet) IP
- les agents sont conçus comme des applets qui tourne sur un serveur HTTP

1.2 Architecture

L'architecture est organisée sous une forme de hiérarchie de couche.

Diverses classes Java sont prédéfinies sous forme de couche, afin de faciliter la création d'agent, d'où la notion de « Template », dans l'acronyme JatLite (Java Agent Template).

Construction d'un agent intelligent à l'aide de « Template » :



Couches JatLite

Source : d'après [LIEN_JAVASTANDFORD]

Couches de JATLite :

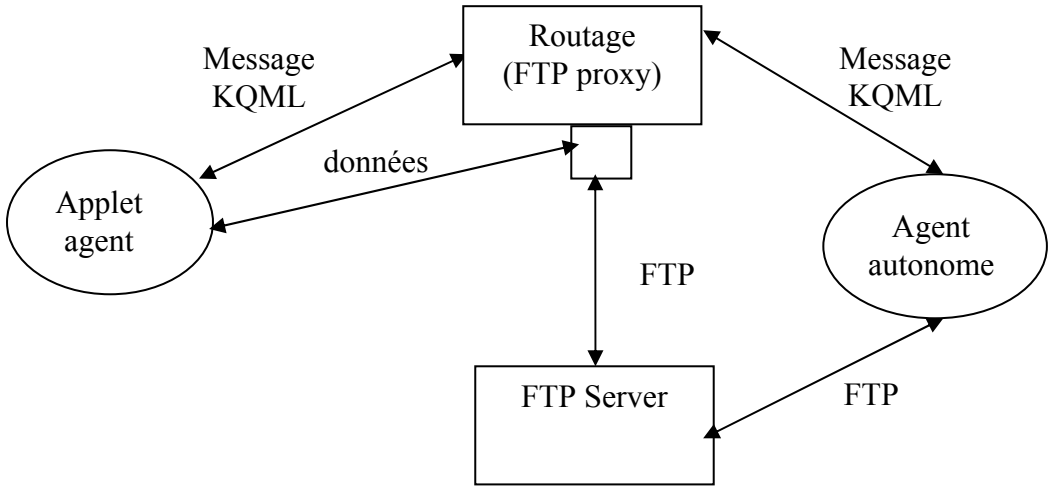
1) Couche abstraite (couche de haut niveau)
- Les classes abstraites sont nécessaires pour l'exécution de JATLite

2) Couche de base
- communication TCP/IP
- pas de restriction des messages du langage ou du protocole

3) KQML
- restriction du langage : stockage et analyse des messages KQML

4) Couche de routage
- Message asynchrone (file d'attente) au lieu de message synchrone: tous les agents reçoivent et envoient les messages via le routeur
- Surmonte les problèmes de sécurité des applets
- deux paquets principaux:
    - RouterLayer.AgentClient: le paquet client des agents
    - RouterLayer.Router: le paquet de routage

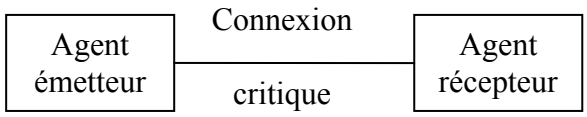
5) Couche Protocole
- support des services internet standard : SMTP, FTP
- les agents autonomes peuvent utiliser les serveurs SMTP et FTP sans le service de routage par proxy
- les applets utilisant des agents doivent utiliser le routage par proxy pour se connecter aux serveurs SMTP ou FTP sur les machines distantes.



1.2.1 Modélisation du routeur

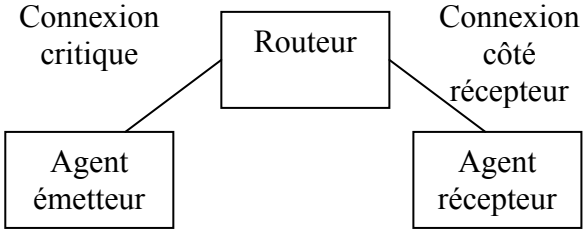
- message synchrone VS message asynchrone (file d'attente)

Message Synchrone



- des problèmes se posent lorsque la connexion ne peut être établit

Message Asynchrone (file d'attente)

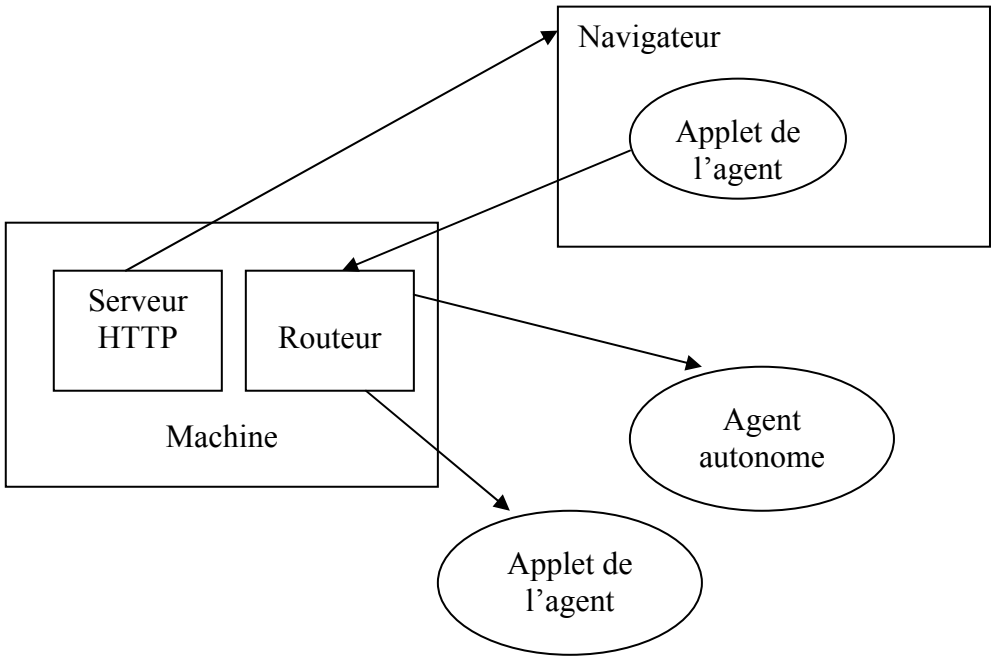


- file d'attente : principe de sauvegarde
- mécanisme de routage : envoyer le message quand la connexion côté récepteur est établie (récepteur connecté au routeur).

- Service de nom des agents (Agent Name Service, ANS)

- Le routeur est un ANServeur, lequel maintient le changement dynamique des adresses des agents.
- Cela n'est pas nécessaire pour l'analyse des messages, mais un agent peut demander au routeur l'adresse des autres agents.

- Applet de communication des agents



- problème du scaling

- beaucoup de message envoyé au routeur peuvent affecté ses performances
- solution : utiliser autant de routeur que nécessaire

1.2.2 Caractéristiques du routeur

- mise en file d'attente des messages

Les messages sont mis en file d'attente dans le système de fichier et retrouvé ou effacé selon les requêtes de l'agent

- Routage des messages

Les messages sont redirigé au bon récepteur lorsque cela est possible

- Service de nom d'agent
- Enregistrement

Seul les agents enregistrés peuvent utilisés le routage

- Sécurité

La sécurité test les fonctionnalités en utilisant les noms et mot de passe de l'agent

- Liste des agents

C'est un message pour connaître tous les agents enregistrés et leurs statuts de connexion.

- Déconnexion

- message de déconnexion
- déconnexion accidentelle (sans message)

Agents autonomes :

1) Le routeur test la connexion de l'agent à intervalle régulier
2) Si le test de connexion échoue plusieurs fois, l'agent est automatiquement dévalidé

3) Le temps d'écoulement entre deux tests et le nombre maximum d'échec sont mis en paramètre.

- Réservation de message

Les messages peuvent être réservé afin d'être reçu à une heure spécifique et à un endroit précis.

1.3 Spécificités de JatLiteBean

JatLiteBean encapsule et étend les fonctionnalités de JatLite. Il s'agit d'un composant JAVA (JavaBean), développé par l'Université d'Otago (Nouvelle Zélande) à l'aide du langage Java de Sun. JatLiteBean est un système multi agents (SMA).

Ce composant a son propre thread d'exécution, contrairement à d'autres agents qui repose sur des graphes à états finis (Zeus, Bond, FIPA OS, … .).

1.3.1 JavaBean

Comme indiquer sur [LIEN_JAVABEAN], les composants JavaBeans sont réutilisables, et permette ainsi une optimisation des codes sources.
Les composants JavaBeans coopère avec les Active X. Cela signifie que l'on peut inclure un composant JavaBean dans n'importe quel application pouvant coopérer avec de l'ActiveX (Internet Explorer, Visual Basic, Microsoft Word, Lotus Notes, …).
Cela signifie, pour notre étude, que l'on peut implémenter JatLiteBean dans une application déjà existante.

1.3.2 Fonctionnalités

Tel que décrite dans [MCK98] les fonctionnalités supplémentaires de JatLiteBean sont les suivantes :
- une interface améliorée avec les fonctionnalités communicante de JatLite, à savoir KQML avec ses méthodes d'analyse, de réception et d'émission.
- une architecture évolutive pour la gestion des messages et des processus géré dans des processus mémoire séparé (notion de « thread »).
- fonctions pratiques pour l'analyse du contenu de message KQML

2 Domaines d'application

JatLite a été conçu dès le départ pour concevoir des applications autour d'internet. Que cela soit pour des applications basé sur des pages web, ou alors pour des applications installé, reposant sur les protocoles internet. Evidemment la version JatLiteBean a été conçue dans le même but, avec en plus la souplesse d'un composant JavaBean.

Tous les types d'application sont imaginables. JatLite n'est pas un produit spécifique à un domaine particulier, mais plutôt un produit de base pour concevoir tout type d'application basé sur les systèmes multi agent.

3 Avantage/inconvénient

3.1 Avantages

- JatLite

L'efficacité d'un système multiagent dépend de ses moyens de communication. JatLiteBean est basé sur JatLite, l'un des pionniers dans le domaine des systèmes multiagents. Plusieurs SMA repose sur JatLite (NetSA, …). Cela assure une pérennité dans la maintenance et l'évolution des systèmes actuellement mis en place.

- KQML

KQML est un véritable standard dans la communication inter agent.

- Java de Sun

Le choix du langage Java pour le développement de la plateforme JatLiteBean permet d'assurer une portabilités sur toutes les plateformes.

- licence GNU

Il s'agit d'un produit gratuit, disponible et utilisable sans limitations de durée. Les sources sont disponibles et modifiable. Ce qui a donné lieu à diverse version de JatLite (JatLiteBean est une modification du code source original)

- ActiveX

Possibilité d'implémenter le composant JatLiteBean dans une application déjà existante.

- Message asynchrone (file d'attente)

Contrairement aux messages synchrones, les files d'attentes permettent de délivrer les messages lorsque l'agent récepteur se connecte.

3.2 Inconvénients

- support

Le support de JatLite est quasiment inexistant. Il n'y a pas de communauté d'utilisateurs suffisamment importante. Etant donné que le produit est gratuit, il n'y a pas de support « éditeur ».

- documentation

La quasi-totalité des documentations sont en langue anglaise. Les différents groupes d'utilisateurs sont principalement des chercheurs, qui se réunisse dans un but commun « l'état de l'art des SMA ». Cela implique que l'usage dans le cadre de développement d'application de JatLite n'est pour l'instant que réservé à une certaine élite.

4 Exemples : Projet Baghera

4.1 Présentation

Ce projet se définit comme un environnement informatique d'apprentissage et d'enseignement à distance de la preuve en géométrie. Il a été élaboré par l'équipe MAGMA du laboratoire LEIBNIZ. Le laboratoire LEIBNIZ se trouve à l'université de Grenoble. L'équipe MAGMA est spécialisée dans la conception de SMA. Pour obtenir plus d'informations sur cette équipe de chercheur dynamique, il suffit de consulter le site [LIEN_MAGMA].

Le projet Baghera possède un site web qui lui est dédié à l'adresse [LIEN_BAGHERA]
L'objet du projet Baguera tel que décrite par l'équipe est de « Développer les principes théoriques et méthodologiques de la conception d'un environnement d'apprentissage de la résolution de problèmes et de la preuve en géométrie dans une démarche mettant les besoins

et les difficultés de l'élève au centre du processus de conception et de modélisation informatique ».
Le projet a été conçu pour fonctionner par le biais d'Internet. A terme, il a pour but d'aider les personnes ayant des difficultés à ce déplacer (handicapés), ou bien qui ne sont plus dans le système scolaire.

4.2 L'existant

Le projet Baghera s'appuie sur plusieurs pôles de compétence et programmes existants. L'ambition du projet est de mettre en œuvre un système afin de faire coopérer des compétences hétérogènes.

- Rendu graphique
Le rendu graphique des formes géométrique s'appuie sur cabri-géomètre, télé-cabri et cabri-euclide.
Au départ les produits « cabri » ont été développés par les équipes de recherches Grenobloise (laboratoire Leibniz), puis lorsqu'il ont atteint une certaine maturité, il ont été développé et commercialisé par la société CABRILOG.

- **ATINF** (ATelier d'INFérence)
Il s'agit de techniques de raisonnement, d'outils de vérification et de construction de preuve en géométrie.
Ces recherches sont effectuées également à Grenoble, par une équipe du laboratoire Leibniz.

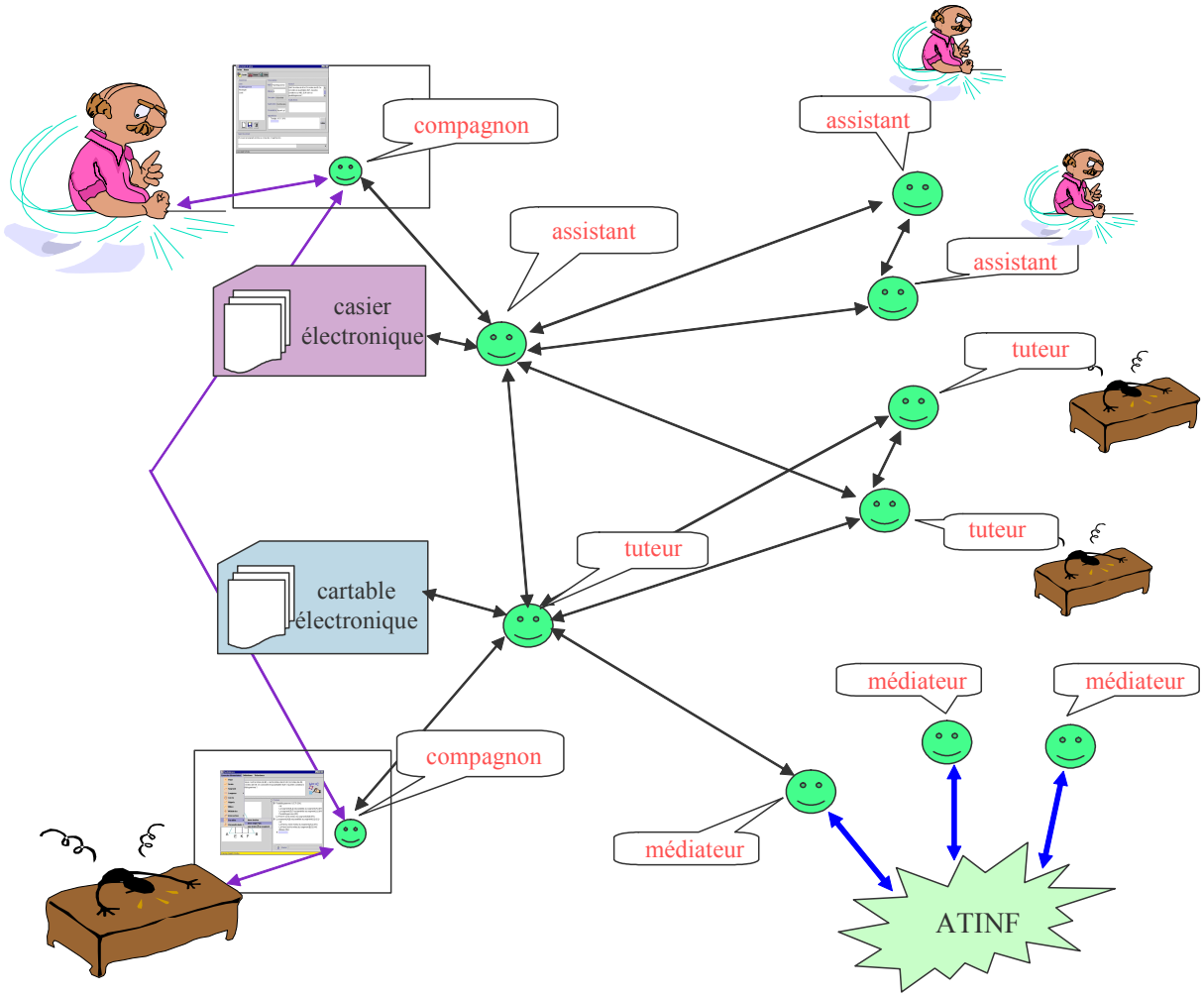
- **MAGMA**
L'équipe est spécialisée dans l'analyse et la conception de systèmes multi agents.

4.3 Les agents

Les agents qui ont été identifiés sont :
- l'agent "CompagnonElève" (interface entre l'élève et le système)
- l'agent "tuteur" (gestion d'un cartable électronique)
- l'agent "médiateur" (traitements de la production de l'élève en vue de son analyse par le résolveur) pour chaque élève
- le "Compagnon Enseignant" et l'agent "assistant" (gestion du casier de l'enseignant) pour chaque enseignant.

Le rôle de chacun de ces agents est décrit précisément dans [BER00]. Tous les agents disposent de moyens de communication de haut niveau, de capacités de raisonnement et de prise de décisions. Le type de communication mis en place dans Baghera s'appuie sur la notion d'actes de langage et est conforme au standard FIPA-ACL

4.4 Schéma des interactions



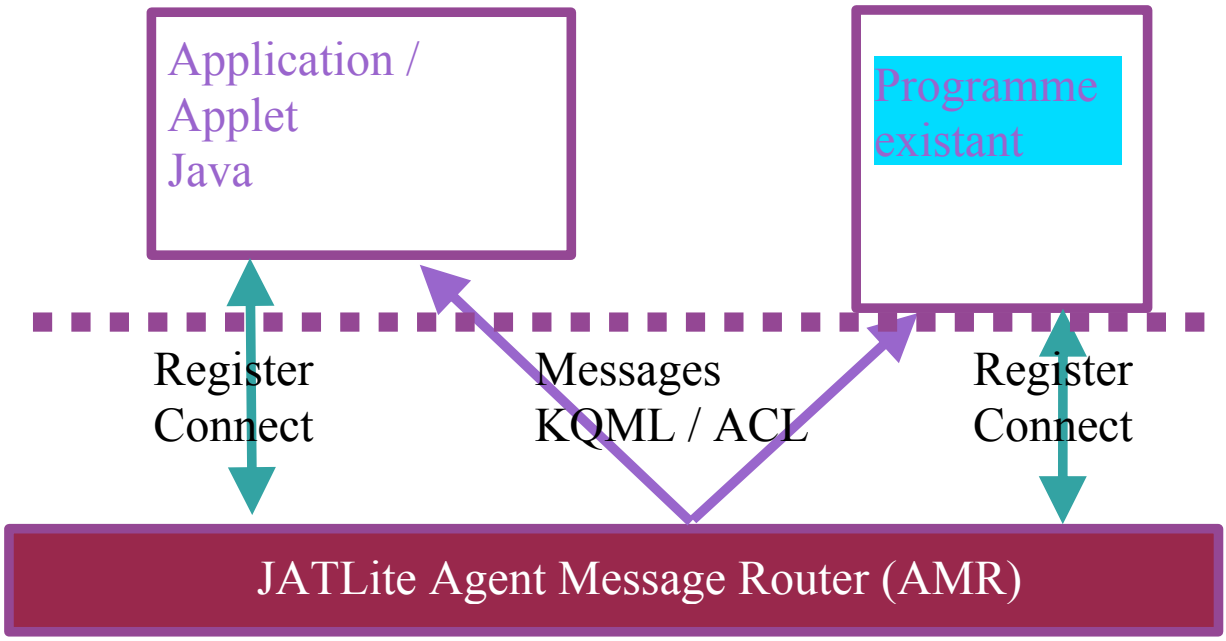
Ce schéma montre l'interaction des agents avec les différents acteurs. On remarquera notamment la couche analyse des formes géométrique noté ATINF.

4.5 Implémentation des interactions

JATLite a été retenu car il répondait efficacement aux critères de l'application à mettre en place :

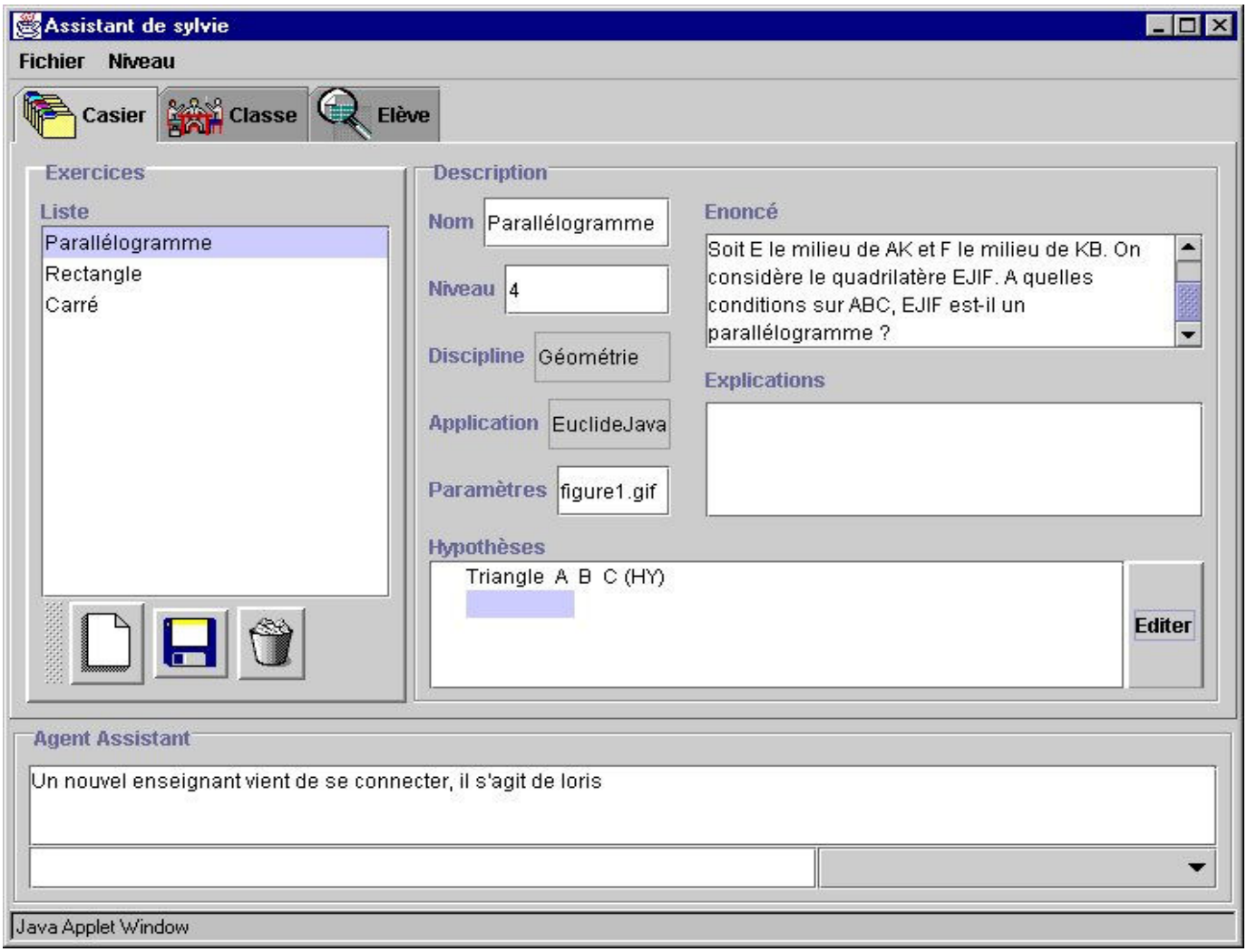
- couche de communication « agent »
- pas de modèle d'agent imposé
- Java – Applications / Applets – support Internet
- basé sur un « Agent Message Router » qui gère la connexion/déconnexion des agents, le routage des messages entre agents, le stockage des messages si nécessaire
- Langage KQML conforme à ACL

4.6 Infrastructure de l'application



Ce schéma montre comment JatLite a pu s'intégrer à l'existant

4.7 Copie d'écran de l'application



Exemple d'écran sur un poste élève

Conclusion

Nous venons de voir dans ce dossier que JatLite est un ensemble de méthodes et de classes permettant la construction de SMA.
Le projet Baghera est un bel exemple d'une application mise en place et maintenu par une équipe dynamique et disposant de moyen important. L'avantage de JatLite sur d'autres SMA, est sa capacité à être implémenté dans des programmes existant.
Néanmoins, malgré des caractéristiques technique impressionnante (cf Agent Message Router), on peux se demandé pourquoi JatLite et les SMA ne connaisse pas actuellement une grande envolé ?
La raison, nous l'avons invoqué dans le chapitre des inconvénients, c'est le manque de support et de documentations. De nombreuses archive disponible sur internet n'ont pas été mise à jour depuis l'an 2000. Certain site officiel sont indisponible depuis plusieurs mois (par exemple le [LIEN_JAVASTANDFORD] qui n'est pas totalement disponible). Tout ceci ne permet pas un déploiement efficace dans les communautés de développeurs traditionnelles. Cela a été également ma plus grande difficulté dans la réalisation de ce dossier : le manque d'informations synthétiques. Il manque une vision d'ensemble (une sorte d'«how to »).

Actuellement les technologies de type « peer to peer », se développent à grande vitesse. Le client-serveur sur internet commence à trouver ses limites à cause du nombre croissant de connexion.
Dans ce contexte, les SMA sont une véritable aubaine pour le développement future d'application intelligente et communicante. On pense sur internet, aux paniers d'achats électronique. L'exemple du projet Baghera pour les personnes en manque d'autonomie nous réserve de formidable application des technologies basées sur les systèmes multi agent. Au niveau pratique, les différents fichiers sources fournis sur [LIEN_JATLITEBEAN] s'utilisent facilement avec le compilateur Java de Borland. Comme nous l'avons vu, JatLiteBean, est un ensemble de classes Java prête à l'emploi. Dans l'exemple de Baghera, chaque agent a été implémenté comme une classe avec diverses méthodes afin que chaque agent puisse communiquer.

L'un des derniers ouvrage sur les SMA [JPB01], explique que sur internet, les SMA qui repose sur Java, on beaucoup d'avenir.

C'est pour toutes ces raisons que je pense que JatLite (ou ses dérivées) est promis à un fort développement au cours des prochaines années. La nature est dotée d'une forme d'intelligence ; lorsque l'on coupe une branche à un arbre, d'autres petites branches pousse, avec autant de feuille. Si depuis si longtemps la nature sais faire de tel choses, alors pourquoi les méthodes intelligentes en informatique ne trouveraient t'elles pas leur place ?

Bibiographie

Ouvrage & Documentation

[FER95] Jacques Ferber, Les systèmes multi-agents, Vers une intelligence collective, InterEditions, 1995, ISBN : 2-7296-0572-X
[ALP97] Alper Caglayan et Colin Harrison, Les agents, Applications bureautiques, Internet et Intranet, InterEditions, 1997, ISBN : 2-225-83146-7
[MCK98] MCKINLAY B., "JATLiteBean documentation", Université d'Otago, Dunedin, Nouvelle Zélande, avril 1998.

[HEN99] Java Agent Template, Lite, JatLite.pdf, Henrique Lopes, Cardoso, Février 99
[JPB01] Principes et architecture des systèmes multi-agents, Jean Pierre Briot, Yves Demazeau, Lavoisier, 2001
[BER00] Bergia Loris. (2000) *Conception et réalisation d'une plate-forme logicielle pour l'apprentissage et l'enseignement à distance*. Mémoire de diplôme d'ingénieur CNAM.

Internet

[LIEN_JATLITEBEAN] Site officiel de l'université de Standford
http://kmi.open.ac.uk/people/emanuela/JATLiteBean/
[LIEN_JAVABEAN] Site officiel du java de Sun http://java.sun.com
[LIEN_JAVASTANDFORD] Quelques exemples d'utilisation de JatLite
http://java.standford.edu
[LIEN_MAGMA] Equipe Magma du laboratoire LEIBNIZ http://www-leibniz.imag.fr/MAGMA/
[LIEN_BAGHERA] Le site du projet baghera http://www-baghera.imag.fr

Table des matières