

Modélisation et implémentation des systèmes complexes

Mise en œuvre des systèmes multi-agents Exemples de plateformes

A. Cardon, C. Bertelle et D. Olivier

LIH - Laboratoire d'Informatique du Havre

DEA Informatique Théorique et Applications

Ecole Doctorale SPMI - Rouen/Le Havre

1. Swarm (www.swarm.org)
2. Madkit (www.madkit.org)
3. Zeus
(www.labs.bt.com/projects/agents/zeus)
4. Aperçu des autres environnements/plateformes

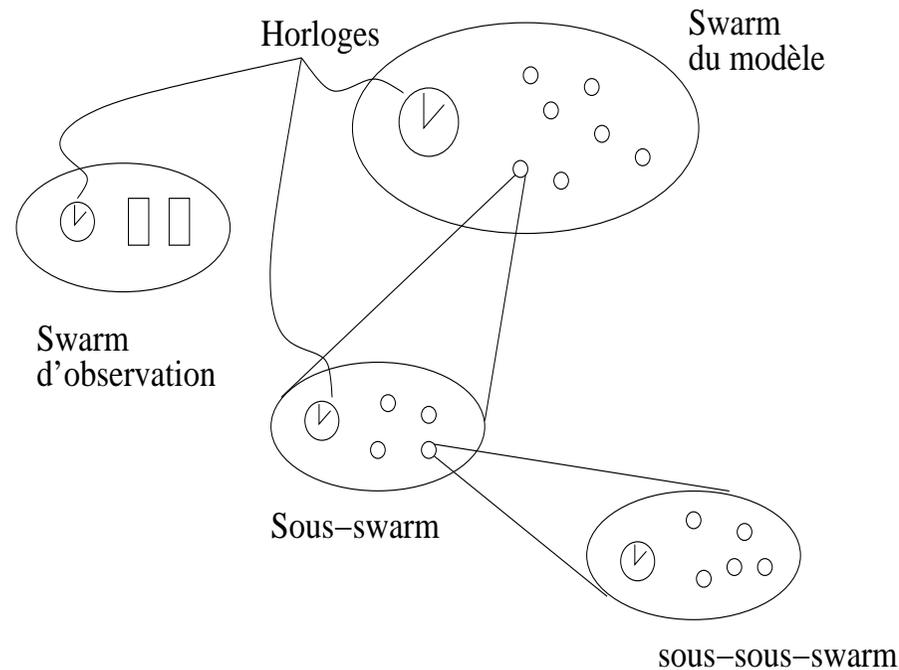
1. Swarm

- Swarm est une plate-forme générique, constituée d'une bibliothèque logicielle (en Objective C, avec une interface Java) qui permet de développer des simulations à base d'agents. Sa conception originale est due à C. Langton (artificial life) et s'est faite à l'Institut Santa Fe (nouveau Mexique).
- Objectif : définir une plate-forme de développement permettant de faire des "expérimentations informatiques" décentralisées à évènement discrets de systèmes complexes.

1. Création du monde virtuel : environnement artificiel spatial et temporel où évoluent des entités ;
2. Création d'agents d'observation qui sondent le monde virtuel précédent ;
3. Fonctionnement de l'ensemble (monde virtuel + observateurs) dans une simulation discrétisée en temps avec des horloges synchronisées.

Swarm - structures récursives

Possibilités de hiérarchies et de structures récursives.



De nombreuses applications développées, notamment à caractère écologique.

2. Madkit

- MadKit : écrit en JAVA et crée par O. Gutknecht & J. Ferber, depuis 1997 ;
- Permet l'utilisation de différents langages de communication (KQML, XML, ...) ;
- Une plate-forme générique sans spécification concernant le comportement des agents, mais proposant un modèle conceptuel d'organisation d'agents.

2.1 Modèle conceptuel Aalaadin

Basé sur 3 concepts principaux : *agents*, *groupes* et *rôles*.



Agent-Groupe-Rôle

- Agent : entité active, communicante, jouant un certain nombre de *rôles* dans un ou plusieurs *groupes* ;
- Groupe : aggrégation d'agents qui, en conjonction avec les *rôles*, permet l'implémentation d'un SMA ;
- Rôle : fonction ou service offert par un agent et qui est toujours local à un groupe.

2.2 Architecture

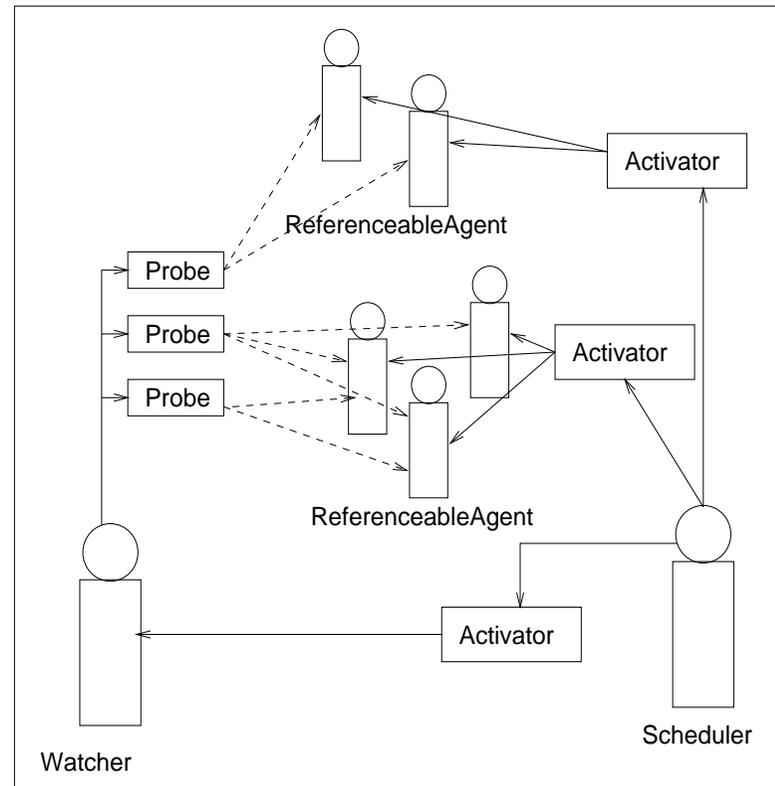
La plateforme est construite suivant les 3 principes :

- Un micro-noyau qui est très léger (gestion des groupes, rôles, cycle de vie des agents et messages)
- Une agentification des services : on utilise la notion d'agent de la plateforme elle-même pour effectuer un certain nombre de services du système (envoi de messages, ...). Ces agents ont des rôles dans des groupes spécifiques.
- Un modèle graphique componentiel, basé sur des JavaBeans.

2.3 Gestion de SMA de grande taille avec le “moteur synchrone”

Objectif : permettre la construction de SMA avec un grand nombre d'agents (des centaines), mais aussi la mise en place d'une synchronisation dans le fonctionnement des agents du SMA.

Architecture générale du moteur synchrone



Ses concepts s'appuient sur les notions :

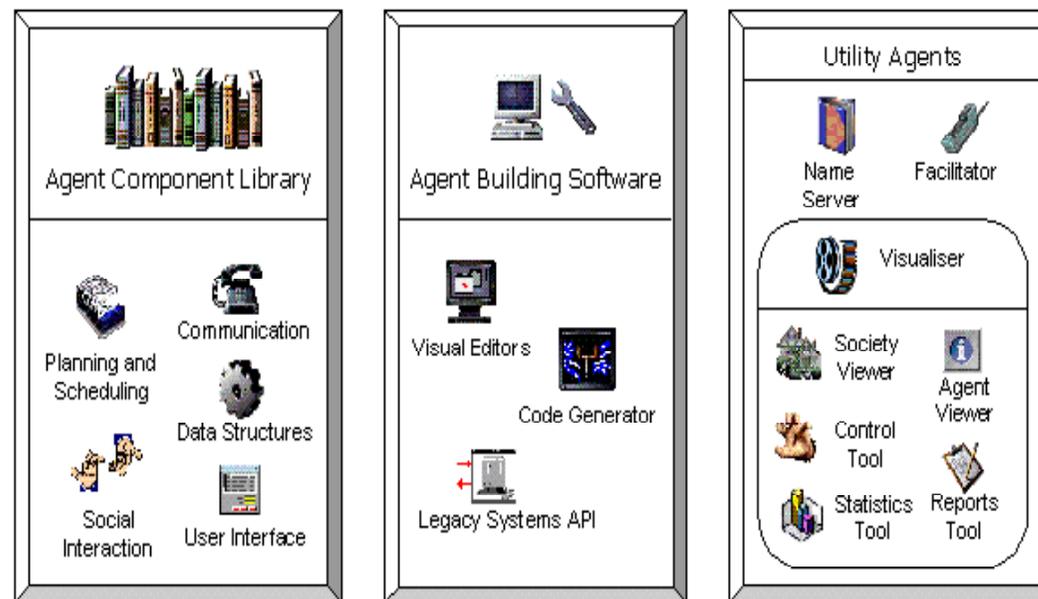
- d'agents ;
- de buts ;
- de tâches (que doit réaliser l'agent pour atteindre son but) ;
- et de faits (ce que l'agent considère comme vrai).

Les agents de Zeus

- Un agent possède une *définition* ;
- Il appartient à une *organisation* ;
- Il *perçoit* son environnement ;
- Il *modifie* son environnement ;
- Il possède un *cycle* de vie ;
- Il s'appuie sur un *protocole* d'interaction.

Fonctionnement/développement

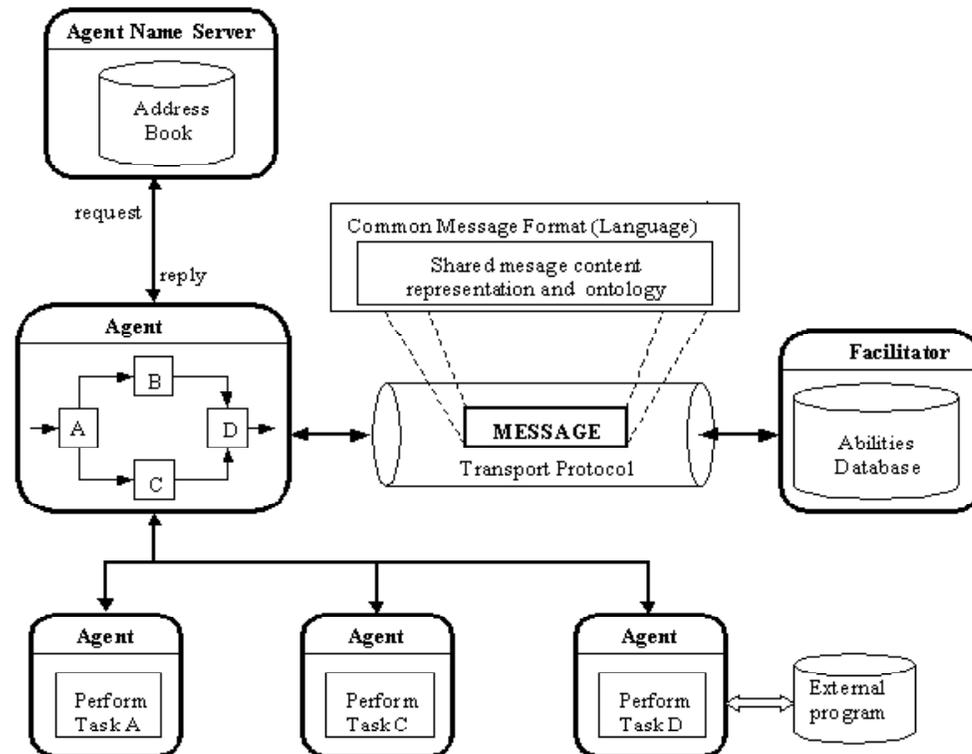
Un environnement graphique pour faciliter le développement et générant du code Java.



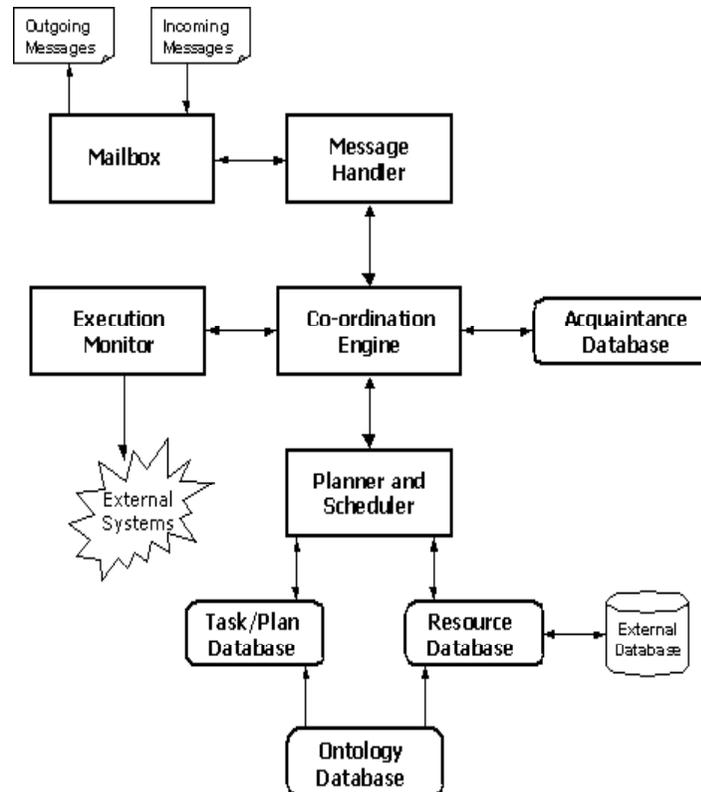
Fonctionnement/développement

- Librairie de composants agents : classes Java
 - Communications (KQML / FIPA ACL)
 - Ontologie : éditeur pour représentation des connaissances
 - Plannificateur de tâches
 - Comportement social : quand et comment interagir/négocier
- Software de construction d'agents (méthodologie, assistant, développement visuel, génération de code Java)
- Agents utilitaires (Serveur de noms, agence de service + visualisateur)

Zeus - Fonctionnement collaboratif



Zeus - Architecture d'un agent générique



4. Aperçu des autres environnements/plateformes

- Oris (ENIB - Brest) : C++, environnement dynamique, adapté à la réalité virtuelle.
- Lalo (CRIM - Montréal Canada) : C++, basé sur AGENT 0 de Shoham.
- Jade (Italie) : en Java, à la norme FIPA.
- MAML (Central European University) : langage de modélisation
- Agent Builder (Reticular systems inc.) est une suite logicielle commerciale en Java composée de deux modules : Toolkit pour définir les agents, des agences et les communications et le Run-Time system qui est un interpréteur du système.

4. Aperçu des autres environnements/plateformes

- Dima (Z. Guessoum - LIP6)
 - Smalltalk (1993), puis porté en Java ;
 - permet de modéliser des comportements réactifs ou cognitifs, à base d'ATN ;
 - comportement délibératif pouvant utiliser des bases de connaissances et des moteur d'inférence (base de méta-règles) ;
 - communications basées sur KQML.
- Environnements/Plateformes orientés agents mobiles : Aglets, Concordia, Grasshopper, Voyager, MOLE
- ...