

Systèmes Multi-agents

Méthodes de conception

Claude Moulin

Université de Technologie de Compiègne

IA04

Sommaire

1 Introduction

2 Méthode JADE

Méthodologie de développement de software - 1

- Un processus, c'est-à-dire une séquence de phases et d'étapes qui guident le développeur dans la construction du système.
- Un ensemble de règles heuristiques qui aident le développeur à faire des choix pertinents.
- Un certain nombre d'objets, comme des diagrammes, schémas des documents qui représentent graphiquement ou textuellement un ou plusieurs modèles du système.

Méthodologie de développement de software - 2

- Une notation appropriée utilisée dans les représentations.
- Un ensemble de patterns qui peuvent être appliqués pour résoudre des situations communes.
- Un ou plusieurs outils qui
 - automatisent les phases identifiées dans le processus ;
 - assurent une consistance entre les modèles produits ;
 - mettent en évidence les problèmes provenant de choix de conception incorrect ;
 - qui peuvent générer du code, de la documentation.

Méthodologie agents

- Comment construire des agents capables d'actions autonomes et indépendantes de façon à exécuter les tâches qui leur sont déléguées ?
- Comment construire des agents capables d'interagir (coopérer, se coordonner, négocier) avec d'autres agents de façon à exécuter les tâches qui leur sont déléguées, et particulièrement lorsque d'autres agents ne partagent pas nécessairement les mêmes intérêts et buts ?

Précautions

- Il n'y a pas de preuve qu'un système développé avec une technique agents n'ait pu être développé avec des techniques non-agents.
- Les agents peuvent rendre plus simple la solution d'une classe de problèmes.
- Considérer naïvement tout comme un agent potentiel est inapproprié.
- Les agents n'ont pas besoin d'être eux-mêmes complexes pour générer des comportements complexes.
- Il est préférable de garder des protocoles simples.

Situation

- Développer un système d'agents à partir d'une approche expérimentale basée sur une étude informelle.
- Besoin d'une méthode de conception rigoureuse.
- Problème : beaucoup de méthodes développées pour un système particulier.

Exemple de méthodes

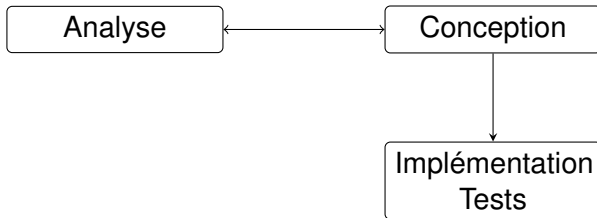
- AAIL (Australian AI Institute)
- GAIA
 - The Gaia Methodology for Agent-Oriented Analysis and Design, Michael Wooldridge, Nicholas R. Jennings, David Kinny, Journal of Autonomous Agents and Multi-Agent Systems, 2000.
- AUML
- Méthode pour le système JADE
 - A Methodology for the Analysis and Design of Multi-Agent Systems using JADE Magid Nikraz, Giovanni Caire, Parisa A. Bahri, (site JADE)

Sommaire

1 Introduction

2 Méthode JADE

Méthode



- La phase d'analyse est générale et indépendante de la plate-forme agent.
- Au contraire la phase de conception considère JADE comme plate-form d'implémentation et utilise directement les concepts de JADE.

1 Introduction

② Méthode JADE
Phase d'analyse
Phase de conception

Use cases

- Chaque use case présente un ou plusieurs scénarios qui montrent comment le système doit interagir avec l'utilisateur ou un autre système pour atteindre un des objectifs du système.
- Chaque use case capture le fonctionnement potentiel du système.
- Outil : il est possible d'utiliser UML pour décrire les besoins fonctionnels d'un système multi-agents.

Types d'agent (préliminaire)

- Identification des principaux types d'agents.
- Règles :
 - Ajouter un type d'agent par utilisateur et par dispositif.
 - Ajouter un type d'agent par ressource (incluant les systèmes propriétaires).
- Distinguer graphiquement les humains, les agents et les ressources.

Fonctionnalités des agents

- Pour chaque type d'agent, lister les principales fonctionnalités d'une manière informelle.
- Résultat : table des Fonctionnalités.
- Règles :
 - Dériver les fonctionnalités des use cases identifiés à l'étape 1.
 - Commencer par les agents dont les fonctionnalités sont les plus claires.
 - Cette phase correspond à la définition des rôles dans d'autres méthodes.

Identification des Accointances

- Décrire les interactions entre agents.
- Mise à jour du diagramme d'agents.

Raffinement des Agents

- L'ensemble des types d'agents initialement identifiés à l'étape 2 est raffiné par certaines considérations. Elles sont liées :
 - Ressources : de quelle source d'information les agents ont besoin pour accomplir leur fonctionnalités et comment, où et quand l'information est générée ou sauvegardée.
 - Découverte : comment les agents liés par accointance se découvrent les uns les autres.
 - Gestion : comment garder une trace des agents, de leur création, de leur arrêt.

Découverte des Agents

- Cas le plus simple : convention de nommage.
 - Problème : éviter les noms (duplication), le nommage basé sur des fonctions (ajout d'agents),
- Mécanisme de pages jaunes. La découverte d'agents est basée sur leurs caractéristiques, c'est-à-dire les services qu'ils fournissent.

Déploiement des Agents

- Produire le diagramme de déploiement des agents.
- Il indique les hôtes physiques sur lesquels sont situés les agents.

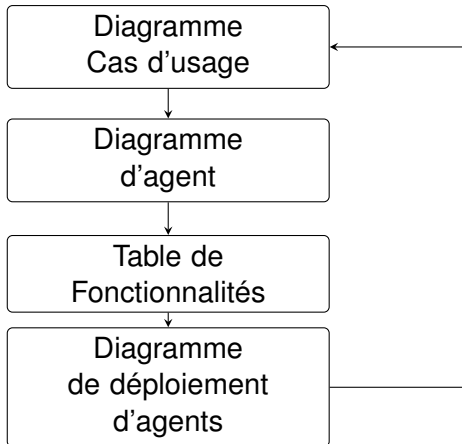
Résumé de la phase d'analyse - 1

- Etape 1 : Use Cases. Les besoins du système sont analysés.
 - résultat : diagramme créé sur ces fonctionnalités.
- Etape 2 : Identification initiale des types d'agent.
 - résultat : à partir d'un ensemble de règles, un diagramme initial du système multi-agent (diagramme d'agents).
- Etape 3 : Identification des Responsabilités des agents.
 - une table initiale des responsabilités, pour les agents dont les fonctions sont claires.

Résumé de la phase d'analyse - 2

- Etape 4 : Identification des Accointances. Les accointances évidentes entre agents sont identifiées,
 - mise à jour du diagramme d'agents et de la table des responsabilités.
- Etape 5 : Raffinement des Agents. Le diagramme d'agents et la table des responsabilités sont mis à jour en considérant le soutien, la découverte, la gestion et le monitoring des agents.
- Etape 6 : Information de déploiement des Agents.
 - résultat : diagramme de deployment des agents avec indication des stations ou dispositifs où les agents sont déployés.
- Itération sur les Etapes 1 à 6.

Phase d'analyse : résultats



Exemple : projet TATIN-PIC

Le projet consiste à concevoir et développer un dispositif informatique permettant d'instrumenter la phase de conception préliminaire dans des projets de mécanique.

Le projet comprend un dispositif utilisant :

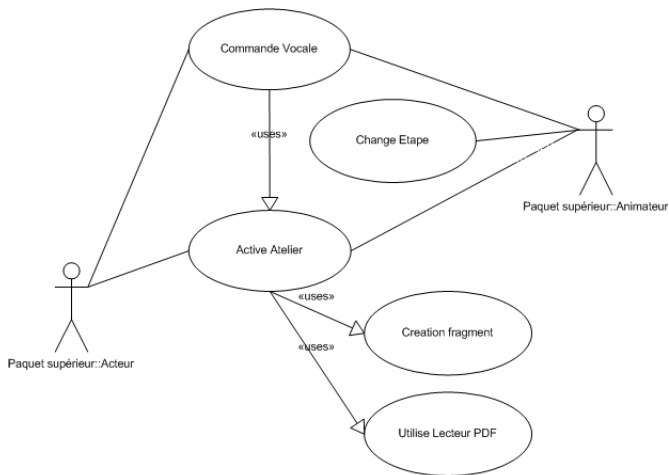
- une table graphique interactive, un tableau interactif,
- des boîtiers pour la gestion des interfaces vocales, et des tablettes pour faciliter l'écriture sur la table.

Le projet étudie la modélisation du processus de conception préliminaire.

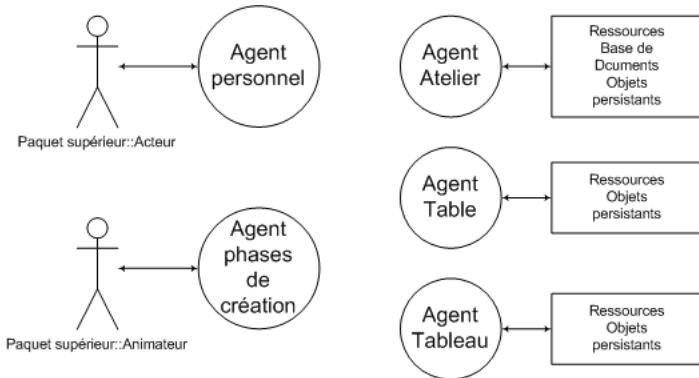
Tatin-PIC : Dispositif



Tatin-PIC : Use cases



Tatin-PIC : Diagramme des agents



Tatin-PIC : Diagramme des Fonctionnalités - 1

Type d'agent	Fonctionnalités
Agent personnel	<ul style="list-style-type: none">. reçoit les commandes vocales de l'utilisateur. transmet les commandes à l'atelier. reçoit les actions manuelles de l'utilisateur. contacte d'autres agents personnels. propose des idées à l'utilisateur

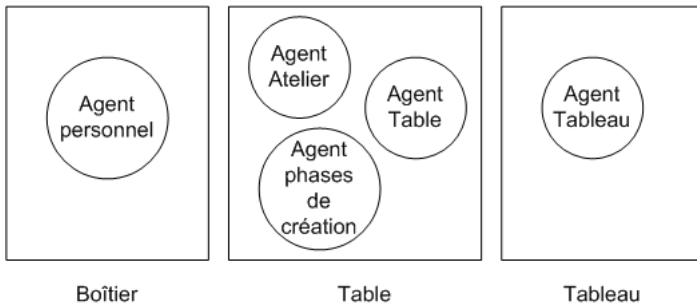
Tatin-PIC : Diagramme des Fonctionnalités - 2

Type d'agent	Fonctionnalités
Agent Atelier	<ul style="list-style-type: none">. reçoit les commandes vocales de l'utilisateur via l'agent personnel. transmet les actions de l'utilisateur à l'agent personnel. active les outils de l'atelier

Tatin-PIC : Diagramme des Fonctionnalités - 3

Type d'agent	Fonctionnalités
Agent Table	<ul style="list-style-type: none">. assure la persistance des objets. transfère le contenu de la table à l'agent tableau
Agent Tableau	<ul style="list-style-type: none">. dispose les objets dans un outil particulier. demande à la table certaines informations
Agent Phases de création	<ul style="list-style-type: none">. assure la bascule entre les phases de création. adapte la disposition des objets sur la table en fonction de la phase de création

Tatin-PIC : Réseau d'accointance



Sommaire

1 Introduction

2 Méthode JADE

Phase d'analyse
Phase de conception

Redéfinition des agents - 1

- Règle 1 : Eviter la duplication d'information. Si plusieurs agents partagent la majeure partie de l'information requise pour exécuter leurs tâches, ils peuvent être fusionnés en un seul.
- Règle 2 : Eviter la duplication de code pour accéder à des ressources. Si plusieurs agents ont besoin de la même ressource, ils peuvent être fusionnés.
- Règle 3 : Eviter d'éclater les agents et d'accroître le nombre d'agents.

Redéfinition des agents - 2

- Règle 4 : Situer les agents. Si deux tâches doivent agir sur deux stations différentes elles doivent être accomplies par deux agents différents.
- Règle 5 : Eviter les agents trop complexes.
- Règle 6 : Adopter l'approche traducteur pour accéder à des ressources externes.

Spécification des Interactions

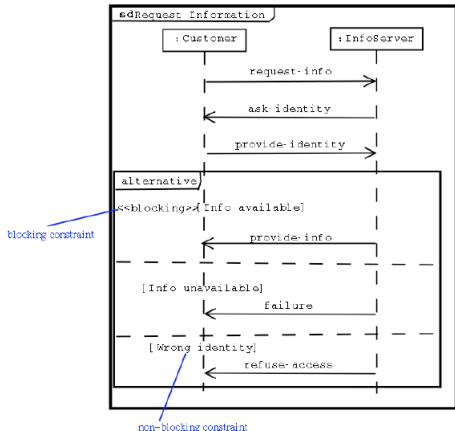
- Résultat : Table des interactions pour chaque agent.
- Chaque interaction a comme propriété :
 - un nom descriptif.
 - une fonctionnalité associée (voir table correspondante).
 - un protocole approprié.
 - le rôle joué par l'agent considéré (Initiateur, Destinataire)
 - le type de l'agent jouant le rôle complémentaire.
 - une condition de déclenchement.

Définition du protocole d'interaction et du formalisme

- Utiliser les protocoles d'interaction définis par la norme FIPA (voir cours sur la communication entre agents).
- Définir des protocoles ad-hoc et un formalisme propre.
 - Formalisme défini par AUML (<http://www.auml.org/>), valable pour des interactions simples.
 - Autre formalisme : graphe, réseau de Petri.

Exemple AUML

- Diagramme de séquence + symboles pour : chevauchement de messages, messages synchrones, asynchrones, contraintes de blocage

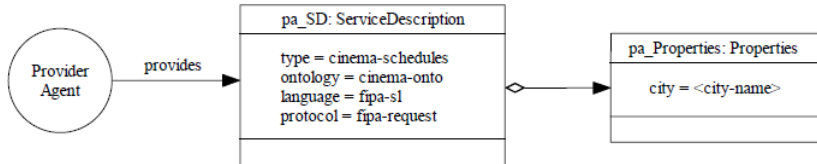


Patrons de Message

- Tous les rôles identifiés dans les précédents protocoles sont implémentés par des behaviours JADE.
- Définir des MessageTemplate appropriés pour la réception de messages, et les ajouter à la table d'interaction.
- Règles :
 - Utiliser des MessageTemplates basés sur le conversation ID.
 - Merger dans un même behaviour les rôles de répondeur basés sur une même combinaison de Performatif, Ontology et Language.
 - Eviter tout conflit.

Description des services

- Formalisation des descriptions des services dans les pages jaunes (DF).



Interactions entre Agents et Ressource

- Ressources Passives : modifiées uniquement par l'agent.
bases de données, fichier, sérialisation objet.
- Ressources Actives : modifiables par d'autres interactions.
 - L'agent est listener de la ressource,
 - Polling pour voir si des changements sont apparus (behaviour particulier).

Interactions Agents Utilisateur

- L'interface peut être considérée comme une sorte de ressource active
 - Interface Graphique Locale (événements Swing).
 - Interface Web créée par pages JSP (JADEGateway et GatewayAgent) pour communiquer avec la plate-forme agents côté serveur.

Behaviours internes des Agents

- Traduire les responsabilités (fonctionnalités) d'un agent par des behaviours.
- Règles :
 - Pour les interactions utilisant les protocoles FIPA, utiliser les classes de behaviour appropriées implémentées dans JADE.
 - Utiliser de préférence les différentes classes de Behaviour,
 - Utiliser les behaviours composites pour des tâches composées (séquence, alternative, FSM).

Définition d'une Ontologie

- Les entités échangées par les agents peuvent être simples (chaînes, nombres) ou être des structures définies par un nom et un ensemble de propriétés dont les valeurs peuvent être typées.
- Ces structures sont assimilables à des concepts d'une ontologie.
- Une parties des entités sont des actions qu'un agent peut entreprendre.

Sélection du Langage de Contenu

- Les contenus de messages sont des suites de caractères qui doivent être décodées et encodées puisque les agents travaillent sur des objets.
- JADE fournit des codecs pour deux langages : SL et LEAP
- Il est possible de créer ses propres codecs.

Résumé de la phase de conception

- Etape 1 : division, fusion, renommage des Agents selon des considérations de performance et de complexité.
- Etape 2 : Spécification des Interactions. Toutes les fonctionnalités de la Table des fonctionnalités, relatives à une relation d'acointance avec un autre agent sont considérées, et la table d'interaction est produite pour chaque type d'agent.
- Etape 3 : Définition du protocole d'interaction et du formalisme.

Résumé de la phase de conception

- Etape 4 : Patrons de Message. La table d'interaction est mise à jour pour spécifier les contenus des messages reçus par les behaviours concernés.
- Etape 5 : Description des services enregistrés en pages jaunes. Une forme en diagramme de classe sert de représentation.
- Etape 6 : Interactions entre Agents et Ressources. Les ressources du système sont identifiées sur la base du diagramme d'agents produit dans la phase d'analyse, ainsi que la façon avec laquelle les agents interagissent avec ces ressources.

Résumé de la phase de conception

- Etape 7 : Interactions Agents Utilisateur. Sur la base du diagramme d'agents produit dans la phase d'analyse, ces interactions sont identifiées et détaillées.
- Etape 8 : Behaviours internes des Agents. Sur la base de la table des fonctionnalités produite en la phase d'analyse, les fonctionnalités sont are mappées sur des behaviours. Différents types de fonctionnalités requièrent différents types de behaviours.
- Etape 9 : Défition d'une Ontologie. Une ontologie appropriée au domaine est spécifiée.
- Etape 10 : Sélection du Langage de Contenu.
- Itérer les étapes 1 à 10.

Phase de conception : résultats

