

Gestão de Serviços de Táxi

Trabalho 3 (T03)



Mestrado Integrado em Engenharia Informática e Computação

Agentes e Inteligência Artificial Distribuída

EIC0033 1 Semestre, Ano letivo 2015/2016

Eduardo Fernandes – 200803951 (ei12130@fe.up.pt)

José Pinto – 201203811 (ei12164@fe.up.pt)

5 de Novembro 2015

Índice

Enunciado.....	3
Descrição do cenário.....	3
Objetivos do trabalho.....	3
Resultados esperados e forma de avaliação.....	3
Plataforma.....	4
<i>Repast Symphony</i> + <i>SAJaS</i> + <i>JADE</i>	4
<i>Repast Symphony</i>	4
<i>SAJaS</i>	4
<i>JADE</i>	4
Características principais.....	5
Funcionalidades relevantes para o trabalho.....	5
Especificação.....	6
Identificação e caracterização dos agentes.....	6
Arquitetura.....	6
Comportamentos.....	6
Protocolos de interação.....	7
Faseamento do projeto.....	9
Recursos.....	10
Bibliografia.....	10
Software.....	10

Enunciado

Descrição do cenário

O cenário proposto para este projeto consiste na criação de um sistema de agentes que aproximem um serviço de gestão de táxis.

Neste cenário existem três tipos de agentes, o agente táxi, o agente passageiro e o agente central de táxis. Cada um tem os seus objetivos.

No caso do agente táxi o seu principal objetivo é transportar passageiros e tentar lucrar o mais possível com isso (desperdiçar o menos possível de combustível e estar o menor tempo possível inativo). Para isso tem de tentar ocupar os seus lugares o melhor que conseguir, com isto em mente o táxi irá transportar vários passageiros cujo os pontos de origem e chegada sejam idênticos. No entanto durante as deslocações o táxi consome combustível, e sempre que necessário tem de abastecer no postos de gasolina existentes.

O agente passageiro tem como principal objetivo chegar ao destino pretendido o mais depressa possível. Para isso tem de se deslocar ao ponto de paragem de táxis mais perto da sua localização e daí utilizar o serviço de táxis. Deve então escolher o primeiro táxi da fila. Contudo pode não haver táxi disponível, neste caso é necessário solicitar ao sistema um táxi.

O agente central de táxis gere os pedidos de táxis por parte de passageiros nos pontos de paragem dos táxis.

Objetivos do trabalho

Desenvolver um programa que simule o serviço de gestão de táxis e seus utilizadores (neste caso os táxis e os passageiros).

Pretende-se que ambos os agentes sejam totalmente autónomos e que consigam obter o melhor comportamento face as necessidades dos mesmos.

Existe também o objetivo de conseguir utilizar o melhor possível a plataforma utilizada e garantir a consistência de resultados, sendo para isso necessário existir uma comunicação sem falhas entre os agentes.

Resultados esperados e forma de avaliação

Com este trabalho espera se obter um programa capaz de simular corretamente o serviço de gestão de táxis, tal como o comportamento dos agentes envolvidos na simulação.

Para avaliar os resultados iram ser utilizadas várias métricas para analisar o desempenho dos vários comportamentos. Para tal irá ser contabilizado o tempo de esperar dos passageiros, o tamanho das filas de espera dos passageiros, o tempo de espera dos táxis, a taxa de ocupação dos táxis e os lucros dos táxis.

Plataforma

Repast Symphony + SAJaS + JADE

O *Repast Symphony* é um sistema de modelação e simulação de agentes rico em funcionalidades. Utilizando o *SAJaS* é possível integrar a biblioteca *JADE* no *Repast Symphony* obtendo assim uma plataforma solida de modelação de sistemas multi agentes.

Repast Symphony

O *Repast Symphony* é uma plataforma interativa e fácil de aprender baseado em *Java* que permite a criação de modelos de simulação. Utilizando o *Repast Symphony* deixa de ser necessário criar um ambiente de simulação de raiz, obtendo muito mais funcionalidades do que se fosse necessário implementar uma nova plataforma. Com isto ganha-se tempo para melhor implementar os agentes que são o foco deste trabalho.

SAJaS

O *SAJaS* permite integrar no *Repast Symphony* a plataforma *JADE*. Assim é possível utilizar as facilidades de sistema multi-agente presentes no *JADE* com o ambiente de simulação do *Repast Symphony*.

JADE

A plataforma *JADE* é uma plataforma que ajuda ao desenvolvimento e implementação de agentes que seguem as normas definidas pela *FIPA (Foundation for Intelligent Physical Agents)* para a criação de sistemas multi-agente.

Para tal o *JADE* disponibiliza um serviço de gestão de agentes, o *AMS (Agent Management System)* que lida com os acessos e utilizações dos agentes. Mantém também uma base de dados de agentes atribuindo um *Agent ID* a cada agente que é adicionado a plataforma.

O *JADE* ainda possui um serviço de páginas amarelas, o *DF (Directory Facilitator)*, onde os agentes podem registar os seus serviços e procurar serviços fornecidos por outros agentes do mesmo sistema.

Para a comunicação entre agentes o *JADE* disponibiliza o *MTS (Message Transport Service)* que gere a transmissão de mensagens entre agentes. As mensagens são criadas de acordo com o *ACL (Agent Communication Language)*.

Características principais

A plataforma do *Repast Symphony* tem como principais características:

- Ambiente gráfico de simulação
- Agregação de estatísticas
- Possibilidade de facilmente correr a simulação com parâmetros diferentes e de fazer *stepping* à mesma

A biblioteca *SAJaS* tem como característica principal integrar o *JADE* no *Repast Symphony* de maneira transparente. Facilita também o registo dos agentes e da gestão do *container* dos mesmos.

A *framework JADE* tem como principais características:

- Facilidade de programação dos agentes, visto ser feito em *Java*
- A possibilidade de distribuir os agentes por várias máquinas para simulações mais pesadas
- A utilização de mensagens estruturadas (Mensagens *ACL*)
- Facilidade de descobrimento de agentes (Serviço de páginas amarelas *DF*)
- Facilidade da definição de diversos tipos comportamentos

Funcionalidades relevantes para o trabalho

Praticamente todas as características principais são de extrema importância para este projeto, excluindo as funcionalidades nativas de agentes do *Repast Symphony* que não iram ser utilizadas pois serão fornecidas pelo *JADE* em conjunto com o *SAJaS*.

Assim em resumo as funcionalidades mais importantes são: a facilidade de criação de agentes e comportamentos combinada com a facilidade de comunicação entre agentes do *JADE*, a componente de simulação do *Repast Symphony* e a facilidade da integração do *JADE* com o *Repast Symphony* através da biblioteca *SAJaS*.

Especificação

Identificação e caracterização dos agentes

Arquitetura

A arquitetura do sistema é dividida em duas camadas. A camada da gestão da simulação e a camada dos agentes.

Na camada de simulação temos a componente de carregamento do mapa utilizado pelos táxis e passageiros, a inicialização de todos os componentes do sistema (componente gráfica, inicialização do *JADE* e criação de agentes).

Na camada dos agentes temos os agentes e os seus respetivos comportamentos. Separando por hierarquia temos:

- Agente
 - Central de Táxis
 - Agente Móvel Genérico
 - Agente Táxi
 - Agente Passageiro
- Comportamento
 - Comportamentos Táxi
 - Comportamentos Passageiro

Comportamentos

O Táxi tem os seguintes comportamentos definidos:

- Possui um ponto de paragem, o qual pode ser fixo ou ser trocado pela central de táxis se for mais conveniente (ponto de maior afluência de passageiros por exemplo)
- Pode em transito recolher um passageiro se está livre, senão solicita um táxi a central de táxis
- Se estiver a ficar sem combustível dirige-se a bomba de gasolina mais próxima

O Passageiro tem os seguintes comportamentos definidos:

- Desloca-se a pé para o ponto de paragem de táxi mais próximo.
- Quando está num ponto de paragem de táxis entra no primeiro táxi da fila, se não existir um táxi disponível na fila é solicitado um táxi a central dos táxis

Protocolos de interação

A interação entre os diversos agentes irá ser feita recorrendo as mensagens *ACL* como escrito anteriormente. Existem varias mensagens possíveis:

De: Central De Táxis → Destino: Táxi	
Tipo: Pedido Formato: Localização	Descrição: Pede a um táxi para comparecer numa determinada paragem de táxis para transportar um ou mais passageiros.
De: Táxi → Destino: Central De Táxis	
Tipo: Resposta Formato: Booleano	Descrição: Responde se pode ou não responder ao pedido.

De: Táxi → Destino: Central De Táxis	
Tipo: Pedido Formato: Localização	Descrição: Pede à central de táxis para enviar um táxi para uma determinada paragem de táxis para transportar um ou mais passageiros.

De: Passageiro → Destino: Central de Táxis	
Tipo: Pedido Formato: Localização	Descrição: Pede à central de táxis para enviar um táxi para uma determinada paragem de táxis.

De: Passageiro → Destino: Táxi	
Tipo: Pedido Formato: Localização	Descrição: Pede a um táxi para efetuar o transporte para o destino pretendido.
De: Táxi → Destino: Passageiro	
Tipo: Resposta Formato: Booleano	Descrição: Responde afirmativamente se poder efetuar a viagem ao destino.

De: Táxi -> Destino: Passageiro	
Tipo: Pedido Formato: Localização	Descrição: Pede aos passageiros da sua paragem de táxis quais os que partilham o destino do(s) actual(is) passageiros presentes no táxi.
De: Passageiro -> Destino: Táxi	
Tipo: Resposta Formato: Booleano	Descrição: Responde afirmativamente ou não dependendo se partilha o destino recebido.

De: Táxi -> Destino: Central de Táxis	
Tipo: Pedido Formato: Localização	Descrição: Pede a central qual é o seu destino depois de entregar os passageiros.
De: Central de Táxis -> Destino: Táxi	
Tipo: Pedido Formato: Localização	Descrição: Envia o destino ao Táxi.

Faseamento do projeto

De acordo com as necessidades do projeto foi definido um planeamento algo simples para facilitar o desenvolvimento do mesmo.

1. Familiarização com as 3 ferramentas a utilizar.
2. Integração das 3 ferramentas e testes iniciais.
3. Criação do esqueleto do programa (Agentes, Comportamentos e classes auxiliares).
4. Fazer a ligação dos agentes a componente gráfica do *Repast Symphony*.
5. Desenvolver os agentes.
6. Implementar a comunicação entre agentes recorrendo ao *MTS* do *JADE*.
7. Detecção de agentes pelos próprios (Ex: O agente Passageiro comunicar com o agente Táxi)
8. Testar programa.
9. Adicionar parâmetros de input para o utilizador manipular a simulação.
10. Agrupamento de resultados e estatísticas para avaliar o desempenho do sistema criado.

Neste momento a atual tarefa é a 5, tendo assim as tarefas anteriores terem sido concluídas com sucesso. Concluimos que ainda falta desenvolver cerca de 60% do projeto para este ser dado como completo.

Recursos

Bibliografia

- *JADE v4.3.3 API Documentation*, <http://jade.tilab.com/doc/api/>.
- *Simple API for JADE-based Simulations (SAJaS) Documentation*, <https://web.fe.up.pt/~hlc/doku.php?id=sajas>
- *Repast Suite Documentation*, <http://repast.sourceforge.net/docs.php>

Software

- *Java Software Development Toolkit 8, Oracle*, <http://www.oracle.com/technetwork/java/index.html>
- *Eclipse Mars integrado com Repast Symphony, SAJaS e JADE*
 - *Repast Symphony 2.3.1, The Repast Development Team*, <http://repast.sourceforge.net/>
 - *SAJaS 0.91b, Henrique Lopes Cardoso*, <http://paginas.fe.up.pt/~hlc/doku.php?id=start>
 - *JAVA Agent DEvelopment Framework, Telecom Italia Lab*, <http://jade.tilab.com/>