



Universidade do Porto
Faculdade de Engenharia
FEUP

Recolha de lixo numa cidade

Relatório Final

Agentes e Inteligência Artificial Distribuída

4º Ano do Mestrado Integrado em Engenharia Informática e Computação

Elementos do Grupo:

Georgina Esteves – ei10010@fe.up.pt

Nuno Cruz – ei10082@fe.up.pt

Vasco Gonçalves – ei10054@fe.up.pt

Dezembro de 2014

Índice

1 Enunciado	3
1.1 Descrição do cenário	3
1.2 Objetivo	4
2 Especificação	5
2.1 Agentes.....	5
2.1.1 <i>Truck Agent</i>	6
2.1.2 <i>World Agent</i>	8
2.1 Protocolos de interação	9
3 Desenvolvimento.....	12
3.1 Plataforma/Ferramenta	12
3.2 Diagrama de classes	14
4 Experiências.....	18
4.1 Testes.....	18
4.1.1 Tempo de execução: 1 minuto	18
4.1.2 Tempo de execução: 3 minutos	21
5 Conclusões.....	26
5.1 Melhoramentos	26
6 Recursos	27
6.1 Bibliografia.....	27
6.2 Software	27

1 Enunciado

1.1 Descrição do cenário

Um dos problemas na gestão de uma área metropolitana é a elevada produção de resíduos urbanos. A sua recolha é fundamental para o bem-estar das populações.

Para além disso, atualmente não é aceitável que todos os resíduos sejam tratados do mesmo modo. A reciclagem é muito importante dado que são reaproveitadas matérias-primas que não são ilimitadas no nosso planeta.

Recentemente, o município da Maia arrancou com o projeto Ecoponto em Casa, em que cada habitação dispõe de contentores próprios para Papel, Vidro e Embalagens para além dos habituais resíduos indiferenciados.

Isto vem complicar ainda mais o processo de recolha de resíduos dado que existem camiões específicos para cada tipo. Este município resolveu o problema atribuindo diferentes dias para a recolha dos diferentes tipos de resíduos.

Esta solução, apesar de funcional, não é ótima, pois pode causar a acumulação de resíduos de determinados tipos.

Assim, propõe-se a comunicação entre camiões de diferentes tipos utilizando agentes para melhor gerir esta situação e melhorar a eficiência de todo o processo de recolha de lixo.

1.2 Objetivo

O trabalho teve como objetivo desenvolver um sistema distribuído multiagente para recolha de lixo numa cidade, sendo essa recolha realizada por vários camiões que atuam de forma cooperativa.

O programa simula a recolha de lixo realizada por vários camiões (agentes) numa cidade. Existem quatro tipos diferentes de lixo: papel, vidro, embalagens e lixo indiferenciado.

O mapa da cidade é constituído por duas superfícies: estrada ou relva. As estradas e cruzamentos servem como referência para a locomoção dos diferentes camiões. A superfície relva estabelece as posições em que é possível colocar os diferentes contentores e depósitos.

O mapa pode ser definido através de um ficheiro de texto, existindo já mapas pré-definidos para efeitos de teste. Nesse ficheiro podem ser especificadas todas as ruas e interseções, a localização de contentores e tudo o que é referente aos camiões: nome, capacidade, posição inicial e tipo do camião.

Os contentores têm uma capacidade máxima, e um nível de ocupação atual (dinâmica, atualizada em tempo de execução através de um modelo simples de simulação) que os camiões são capazes de aferir quando passam por eles. Na situação de um contentor entrar no raio de visão do camião: se forem do mesmo tipo pergunta ao agente *World* qual o conteúdo do contentor, e se tiver espaço disponível, despeja o conteúdo do mesmo; nos outros casos, informa todos os camiões que estão registados com o mesmo tipo do contentor (no caso de ser o tipo a que ele pertence, retira-se a ele próprio da lista) e transmite uma mensagem com a localização do contentor.

2 Especificação

2.1 Agentes

Na Figura 1 encontra-se uma representação desenvolvida com o intuito de ser possível a identificação de todos os agentes e do seu tipo (lixo comum, papel, embalagens ou vidro), dos contentores desses tipos e do depósito. É também facilmente distinguível o caminho que os agentes podem percorrer (asfalto), e os locais onde podem ser colocados os contentores e os depósitos (relva).

Características do nosso sistema:

- Uniformidade: Homogéneo
- Granularidade: Grão fino
- Controlo: Hierárquico



Figura 1 – Interface Gráfica

2.1.1 *Truck Agent*

Os agentes *Truck* representam os caminhões de recolha de lixo. Têm como principais características a delimitação do limite de visão, um tipo de lixo associado e uma capacidade de recolha máxima.

Estes possuem uma rota que devem percorrer (esta pode ser posteriormente alterada, de acordo com notificações de outros agentes) definida por *pointsToVisit*, sendo calculado através do algoritmo *Dijkstra* o caminho mais curto para todos os seus objetivos. Nessa rota, devem recolher o lixo dos contentores adjacentes que tiverem o mesmo tipo que o seu. Caso o contentor adjacente não seja do seu tipo ou ele não tenha espaço livre suficiente para esvaziar o contentor, envia então uma mensagem a todos os outros caminhões desse tipo, deixando ao cargo deles a decisão sobre quem vai recolher aquele contentor.

Essa decisão processa-se da seguinte forma:

1. Após receber uma mensagem *INFORM_OTHER_TRUCKS* contendo a posição do contentor em questão, cada *Truck* calcula a sua distância mínima ao contentor através do algoritmo *Dijkstra* e envia-a para os outros numa mensagem *INFORM_DISTANCE*.
2. Após todos terem recebido mensagens *INFORM_DISTANCE* dos outros agentes ou após dar *timeout*, aquele que tiver o valor mínimo de distância (estiver mais perto) adiciona o contentor aos *pointsToVisit* e, opcionalmente, à sua rota.

Descrição "PAGE" dos Agentes *Truck*

- Perceção: própria localização, tipo de camião, rota
- Ações: informar outros agentes *Truck*, pedidos de informação ao agente *World*, informar o agente *World*, despejar o contentor, esvaziar o seu depósito
- Objetivos: recolha do lixo do seu tipo através do caminho mais curto
- Ambiente: mapa com estradas, interseções e relva

Análise interna

- Tempo de vida: Permanente
- Nível de Cognição: Maioritariamente deliberativo
- Implementação: Procedimental
- Mobilidade: Móvel
- Adaptabilidade: Ensinável por outros agentes

Análise externa

- Localização: Local
- Autonomia Social: Independente, ajustando a rota por indicação de outros agentes
- Sociabilidade: Membro
- Colaboração: Cooperativo,
- Interação: Direta com Agentes

2.1.2 *World Agent*

O agente mundo é o que possui o conhecimento de todo o mapa. É ele que informa os agentes *Truck* de todos os pedidos relacionados com os elementos do mapa.

Descrição "PAGE" do Agente *World*:

- Percepção: todo o mapa e seus constituintes
- Ações: envio de informações aos agentes *Truck*, geração de lixo nos contentores

Análise interna

- Tempo de vida: Efêmero
- Nível de Cognição: Deliberativo
- Implementação: Procedimental
- Mobilidade: Estacionário
- Adaptabilidade: Autodidata

Análise externa

- Localização: Local
- Autonomia Social: Independente
- Sociabilidade: Responsável,
- Colaboração: Cooperativo
- Interação: Direta com Agentes e Ambiente

2.1 Protocolos de interação

De forma a garantir que as operações ocorrem de acordo com o desejado, foi necessário definir um protocolo de interação (troca de mensagens). A Tabela 1 enuncia e descreve cada uma das mensagens trocadas, o seu conteúdo e uma breve descrição das mesmas. Na Figura 2 e na Figura 3 encontram-se diagramas de interação UML, para ser mais perceptível em que situação ocorrem as mensagens e as sequências das mesmas.

Id	Nome	Conteúdo	Descrição
1	REQUEST_CONTAINER_CAPACITY	X + Y (do contentor)	Pedido de informação de um <i>Truck</i> ao <i>World</i> acerca do conteúdo de um <i>Container</i>
2	REQUEST_MOVE	Nome <i>Truck</i> + X + Y (para onde que mover) + Direção	Pedido de um <i>Truck</i> ao <i>World</i> para se deslocar
3	INFORM_OTHER_TRUCKS	X + Y (do contentor)	<i>Truck</i> informa o posicionamento de um <i>Container</i> aos agentes <i>Truck</i> desse tipo
4	INFORM_EMPTIED_CONTAINER	X + Y (do contentor)	<i>Truck</i> informa o <i>World</i> que o <i>Container</i> foi despejado
5	INFORM_CONTAINER_CAPACITY	X + Y (do contentor) + <i>UsedCapacity</i>	<i>World</i> informa um <i>Truck</i> da capacidade do <i>Container</i>
6	CONFIRM_REFUSE_MOVE	CONFIRM or REFUSE	<i>World</i> informa um <i>Truck</i> se pode ou não deslocar-se
7	GOT_INFORM_EVENT	--	Confirmação de mensagem INFORM
8	INFORM_DISTANCE	X + Y (do container) + Distância	<i>Truck</i> informa os outros elementos <i>Truck</i> da sua distância mínima ao <i>container</i> (x,y)
9	INFORM_CREATE_TRUCK	<i>TruckName</i> + <i>Capacity</i> + X + Y (posição inicial) + <i>TruckType</i>	<i>Truck</i> remoto informa o mundo da sua criação
10	INFORM_EMPTIED_TRUCK	<i>TruckName</i>	<i>Truck</i> remoto informa mundo que esvaziou num depósito
11	INFORM_GOING_TO_DEPOSIT	<i>TruckName</i>	<i>Truck</i> remoto informa mundo que está a deslocar-se a um depósito
12	INFORM_CURRENT_DESTINATION	<i>TruckName</i> + X + Y (do destino atual)	<i>Truck</i> remoto informa mundo do seu destino atual

Tabela 1 - Informação das mensagens do protocolo de interação

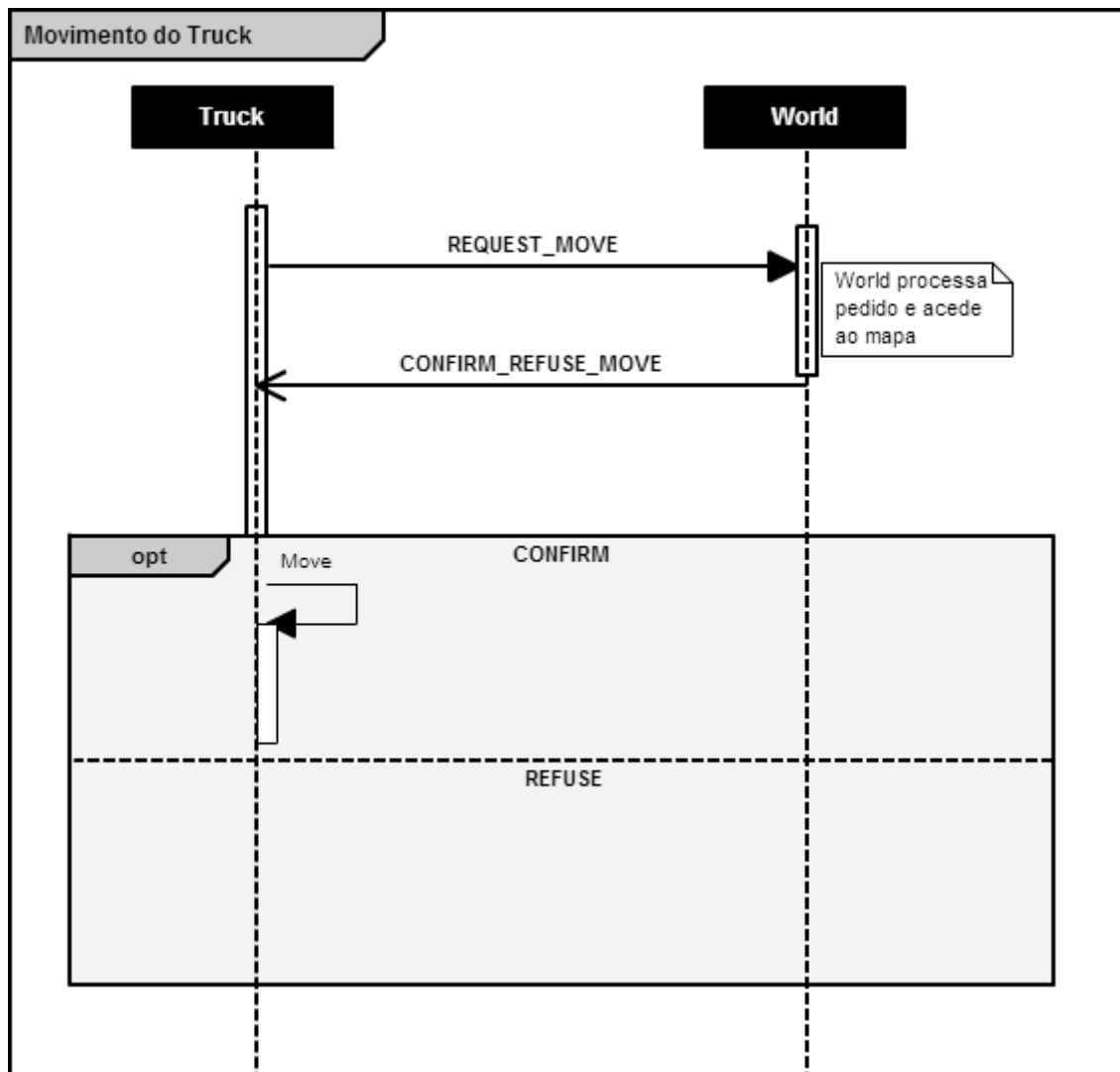


Figura 2 - Diagrama de interação UML do movimento de um agente Truck

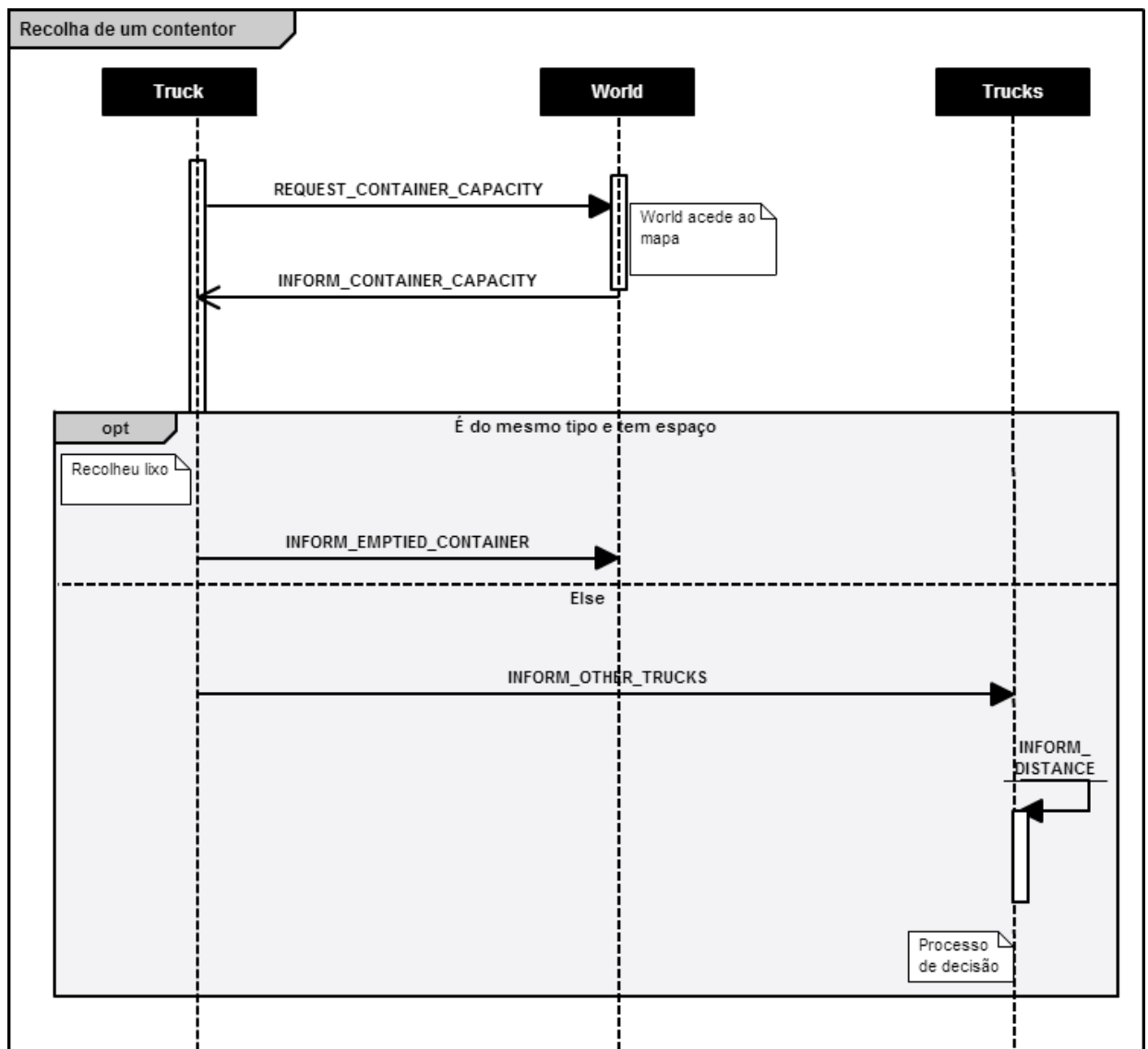


Figura 3 - Diagrama de interação UML de recolha de lixo de um agente Truck

3 Desenvolvimento

3.1 Plataforma/Ferramenta

A aplicação foi desenvolvida em Java, utilizando o IDE Eclipse. Para a implementação dos agentes foi utilizada a plataforma JADE.

O JADE (*Java Agent DEvelopment Framework*) é uma *framework* totalmente implementada na linguagem Java, que simplifica a implementação de sistemas multiagente através de um *middleware* que segue as especificações FIPA e, através de um conjunto de ferramentas gráficas, auxilia as fases de desenvolvimento e *debug*.

Um sistema baseado em JADE pode ser distribuído por diversas máquinas (que não precisam de correr o mesmo sistema operativo) e a sua configuração pode ser controlada através de uma interface gráfica remota. A configuração pode inclusive ser alterada em *run-time*, movendo agentes de uma máquina para outra, conforme for preciso.

Para além da abstração de agentes, o JADE providencia modelos de execução de tarefas e de composição, comunicação *peer-to-peer* entre agentes baseada nos paradigmas de mensagens assíncronas, um sistema de diretoria de agentes e outras funcionalidades que facilitam o desenvolvimento de um sistema distribuído multiagente.

Na figura 3 encontra-se a *Graphical User Interface* (GUI) para controlo e monitorização do estado dos agentes. É também possível através da GUI criar e começar a execução de um agente.

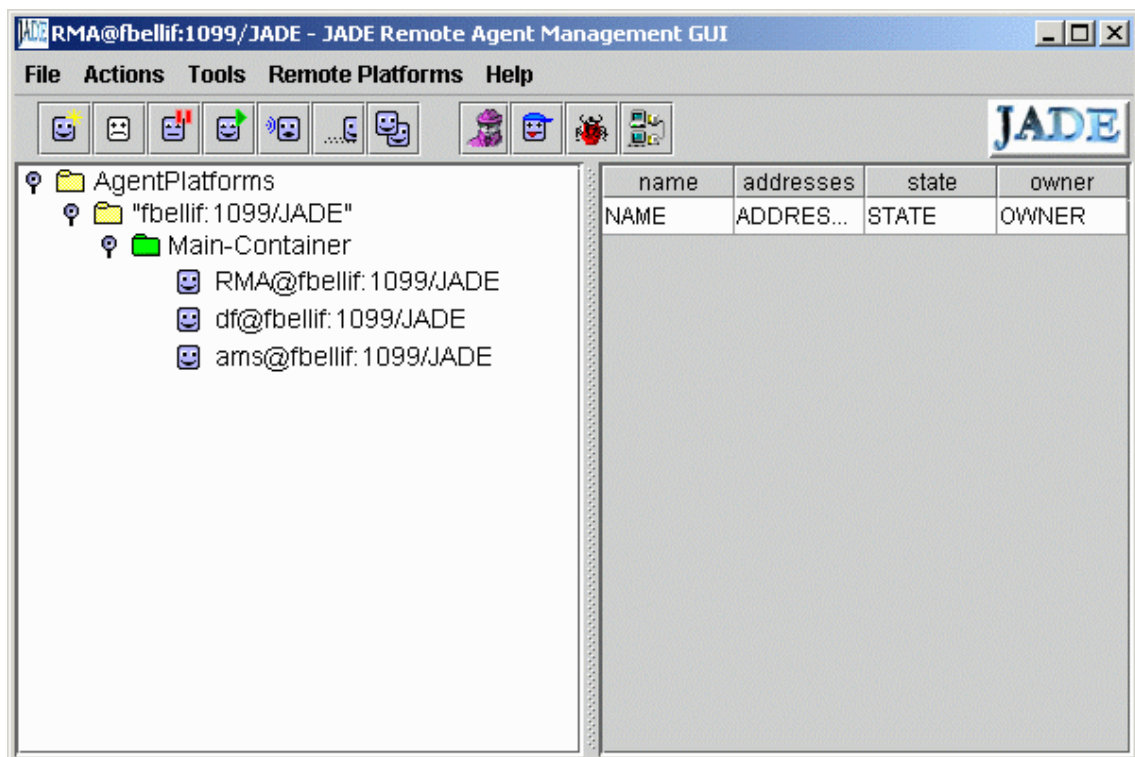
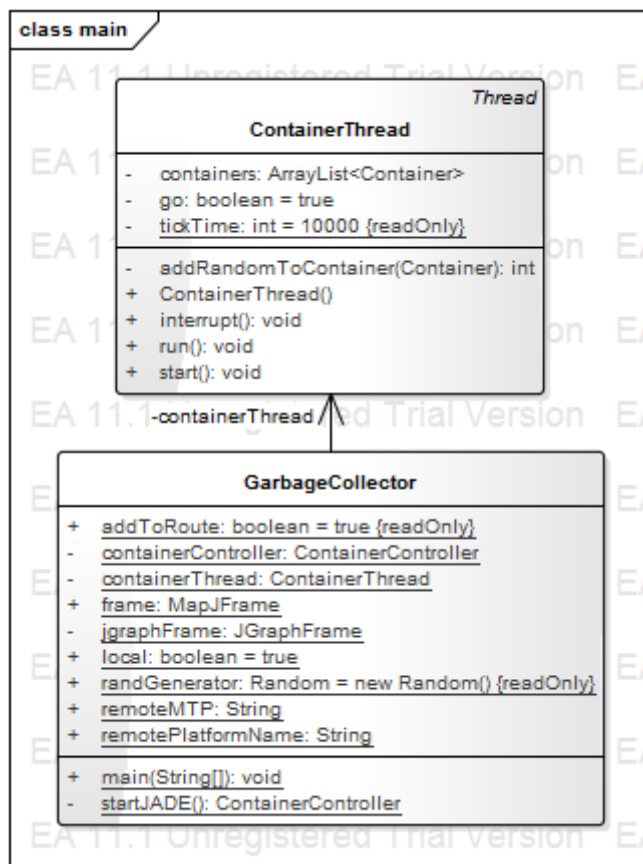
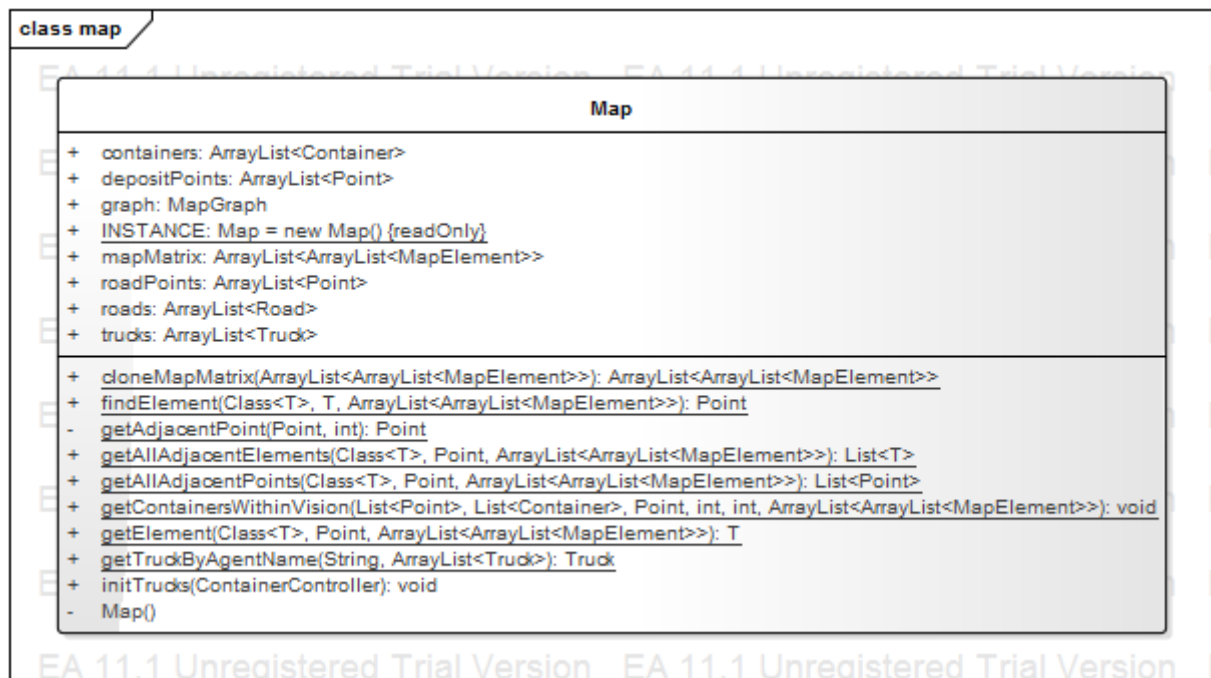
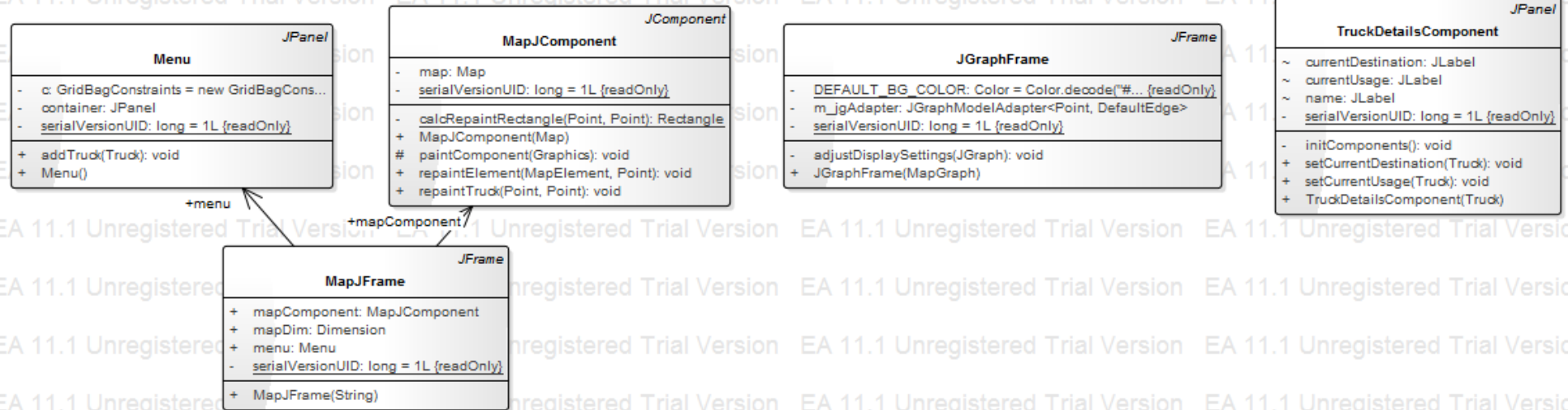


Fig. 3 - Interface RMA (Gestão de Agentes Remota) do JADE

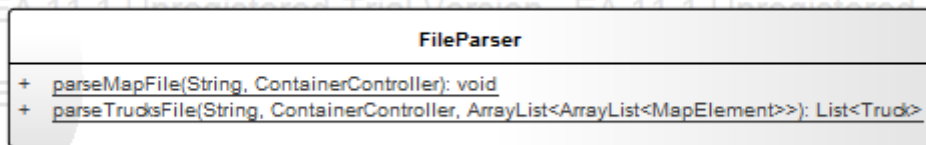
3.2 Diagrama de classes



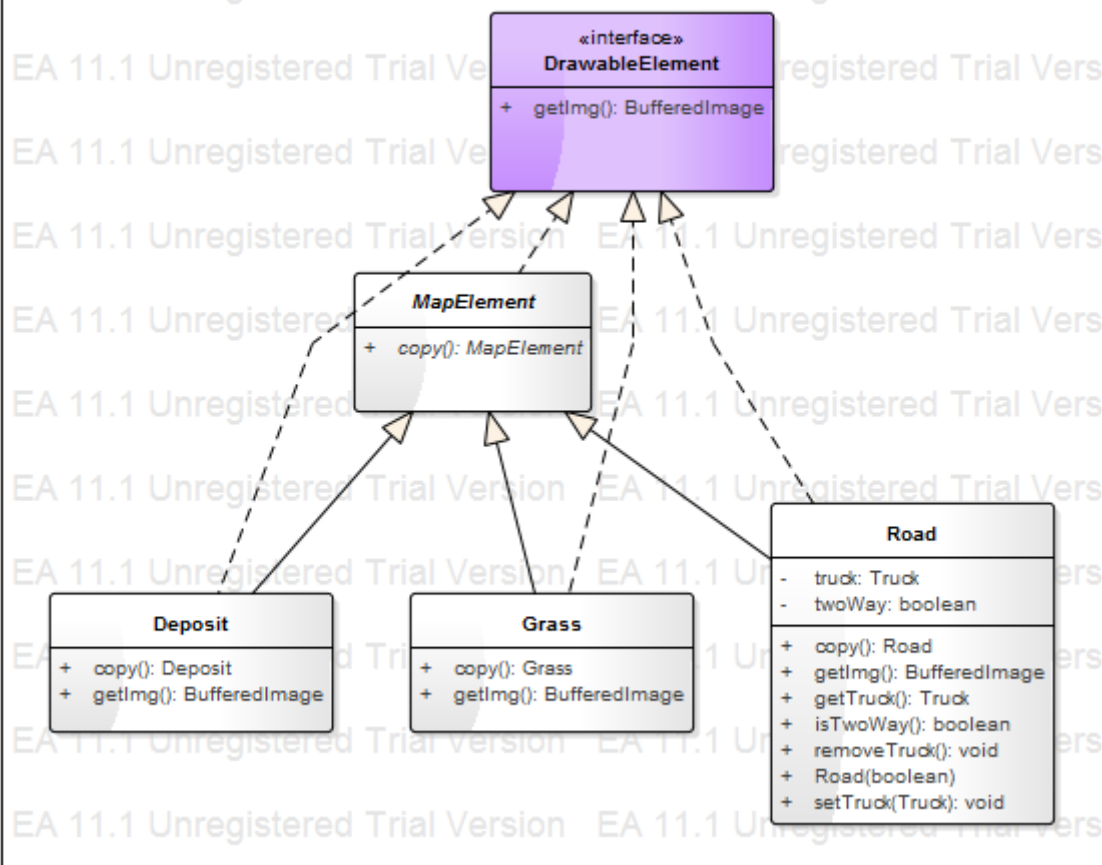
class gui



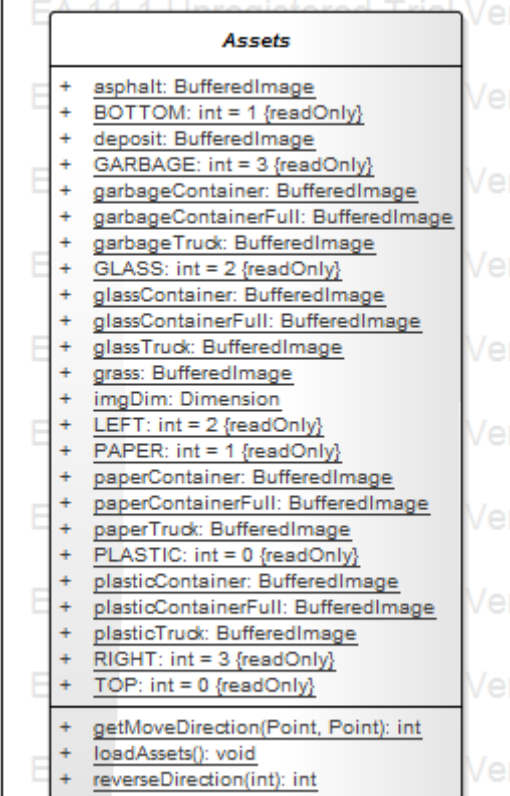
class files



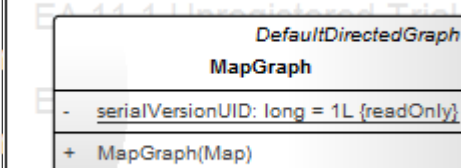
class elements



class assets



class algorithms



class agents

TruckAgent	Agent
# event: Event + GOT_INFORM_EVENT: int = 7 {readOnly} + INFORM_CREATE_TRUCK: int = 9 {readOnly} + INFORM_CURRENT_DESTINATION: int = 12 {readOnly} + INFORM_DISTANCE: int = 8 {readOnly} + INFORM_EMPTIED_CONTAINER: int = 4 {readOnly} + INFORM_EMPTIED_TRUCK: int = 10 {readOnly} + INFORM_GOING_TO_DEPOSIT: int = 11 {readOnly} + INFORM_OTHER_TRUCKS: int = 3 {readOnly} + REQUEST_CONTAINER_CAPACITY: int = 1 {readOnly} + REQUEST_MOVE: int = 2 {readOnly} - serialVersionUID: long = 1L {readOnly} - truckInformThread: TruckInform - worldAgent: AID - findTrucksByType(String, ACLMessage): boolean # setup(): void # takeDown(): void	

WorldAgent	Agent
+ CONFIRM_REFUSE_MOVE: int = 6 {readOnly} + INFORM_CONTAINER_CAPACITY: int = 5 {readOnly} - serialVersionUID: long = 1L {readOnly} # setup(): void # takeDown(): void	

4 Experiências

Existem diversas formas de determinar todo o processo de recolha de lixo. De forma a conseguir avaliar o impacto de cada decisão tomada na execução do projeto e de ser possível a avaliação de uma medida, foram adotadas as seguintes métricas:

- Média de consumo de um camião, ou seja, lixo recolhido
- Média da quantidade máxima de lixo nos contentores

Através de pequenas alterações no funcionamento do projeto e da análise destes valores previamente enunciados, será possível determinar quais as estratégias mais eficientes.

4.1 Testes

4.1.1 Tempo de execução: 1 minuto

4.1.1.1 Adição permanente de novos contentores à rota, variando o raio de visão

Parâmetros do teste: raio de visão é 1.

	Contentores*	Contentores Esvaziados	Idas ao depósito	Total de lixo recolhido
Agente 1	3	6	3	9
Agente 2	6	5	1	16
Agente 3	4	11	2	17
Agente 4	1	3	2	4

*adicionados por informação vinda de outros agentes

Média de lixo máximo que ficou nos contentores: 2,58.

Parâmetros do teste: raio de visão é 5.

	Contentores*	Contentores Esvaziados	Idas ao depósito	Total de lixo recolhido
Agente 1	3	8	3	10
Agente 2	6	6	1	11
Agente 3	5	4	2	10
Agente 4	3	4	2	10

*adicionados por informação vinda de outros agentes

Média de lixo máximo que ficou nos contentores: 2,2.

Parâmetros do teste: raio de visão é 10.

	Contentores*	Contentores Esvaziados	Idas ao depósito	Total de lixo recolhido
Agente 1	3	6	3	6
Agente 2	4	5	0	8
Agente 3	5	11	2	16
Agente 4	3	4	3	11

*adicionados por informação vinda de outros agentes

Média de lixo máximo que ficou nos contentores: 2.

Análise de Resultados: Com o aumento do raio de visão conseguimos verificar que o número de contentores esvaziados aumenta e apesar de o total de lixo recolhido permanecer aproximadamente o mesmo, a média de lixo máximo que fica nos contentores diminui.

4.1.1.2 Adição de novos contentores temporária, variando o raio de visão

Parâmetros do teste: raio de visão é 1.

	Contentores*	Contentores Esvaziados	Idas ao depósito	Total de lixo recolhido
Agente 1	0	3	4	3
Agente 2	0	7	1	11
Agente 3	0	11	2	16
Agente 4	0	3	2	6

*adicionados por informação vinda de outros agentes

Média de lixo máximo que ficou nos contentores: 2,375.

Parâmetros do teste: raio de visão é 5.

	Contentores*	Contentores Esvaziados	Idas ao depósito	Total de lixo recolhido
Agente 1	0	6	1	8
Agente 2	0	7	1	11
Agente 3	0	9	1	16
Agente 4	0	3	0	4

*adicionados por informação vinda de outros agentes

Média de lixo máximo que ficou nos contentores: 2,3.

Parâmetros do teste: raio de visão é 10.

	Contentores*	Contentores Esvaziados	Idas ao depósito	Total de lixo recolhido
Agente 1	0	7	4	11
Agente 2	0	7	0	8
Agente 3	0	11	1	3
Agente 4	0	7	0	9

*adicionados por informação vinda de outros agentes

Média de lixo máximo que ficou nos contentores: 2,1.

Análise de Resultados: Comparando estas tabelas com as do subcapítulo anterior, verificamos que com a adição de novos contentores permanentemente à rota dos camiões, o número de contentores esvaziados diminui ligeiramente, devido ao camião ter que passar a percorrer um caminho maior em cada rotina.

4.1.2 Tempo de execução: 3 minutos

4.1.2.1 Adição permanente de novos contentores à rota, variando o raio de visão

Parâmetros do teste: raio de visão é 1.

	Contentores*	Contentores Esvaziados	Idas ao depósito	Total de lixo recolhido
Agente 1	3	23	7	39
Agente 2	6	17	4	39
Agente 3	4	27	7	57
Agente 4	1	9	10	17

*adicionados por informação vinda de outros agentes

Média de lixo máximo que ficou nos contentores: 4.5.

Parâmetros do teste: raio de visão é 5.

	Contentores*	Contentores Esvaziados	Idas ao depósito	Total de lixo recolhido
Agente 1	3	25	8	34
Agente 2	6	16	5	42
Agente 3	5	31	8	63
Agente 4	3	9	9	27

*adicionados por informação vinda de outros agentes

Média de lixo máximo que ficou nos contentores: 4.54.

Parâmetros do teste: raio de visão é 10.

	Contentores*	Contentores Esvaziados	Idas ao depósito	Total de lixo recolhido
Agente 1	3	24	8	35
Agente 2	4	16	3	34
Agente 3	5	29	11	61
Agente 4	3	10	8	26

*adicionados por informação vinda de outros agentes

Média de lixo máximo que ficou nos contentores: 4.375.

Análise de Resultados: Com uma amostra de tempo maior, conseguimos também verificar que o aumento do raio de visão aumenta o número de contentores esvaziados, apesar de o total de lixo recolhido permanecer aproximadamente o mesmo.

4.1.2.2 Adição temporária de novos contentores, variando o raio de visão

Parâmetros do teste: raio de visão é 1.

	Contentores*	Contentores Esvaziados	Idas ao depósito	Total de lixo recolhido
Agente 1	0	15	6	20
Agente 2	0	15	5	33
Agente 3	0	29	7	52
Agente 4	0	5	9	11

*adicionados por informação vinda de outros agentes

Média de lixo máximo que ficou nos contentores: 5.2.

Parâmetros do teste: raio de visão é 5.

	Contentores*	Contentores Esvaziados	Idas ao depósito	Total de lixo recolhido
Agente 1	0	17	2	20
Agente 2	0	28	4	46
Agente 3	0	31	8	52
Agente 4	0	11	5	20

*adicionados por informação vinda de outros agentes

Média de lixo máximo que ficou nos contentores: 4.5.

Parâmetros do teste: raio de visão é **10**.

	Contentores*	Contentores Esvaziados	Idas ao depósito	Total de lixo recolhido
Agente 1	0	18	10	33
Agente 2	0	22	4	37
Agente 3	0	28	4	43
Agente 4	0	15	4	29

*adicionados por informação vinda de outros agentes

Média de lixo máximo que ficou nos contentores: 4.54.

Análise de Resultados: Com uma amostra de tempo maior, as conclusões mantêm-se.

4.1.2.3 Variando a condição para um camião ir ao depósito esvaziar

Parâmetros do teste: condição para esvaziar: quando está **50%** cheio. (**adicionar** contentores vindos de outros agentes à rota, raio de visão é **1**).

	Contentores*	Contentores Esvaziados	Idas ao depósito	Total de lixo recolhido
Agente 1	3	19	7	39
Agente 2	6	19	7	54
Agente 3	4	35	9	59
Agente 4	1	7	9	14

*adicionados por informação vinda de outros agentes

Média de lixo máximo que ficou nos contentores: 4.75.

Parâmetros do teste: condição para esvaziar: quando está **75%** cheio (**adicionar** contentores vindos de outros agentes à rota, raio de visão é **1**).

	Contentores*	Contentores Esvaziados	Idas ao depósito	Total de lixo recolhido
Agente 1	3	19	7	35
Agente 2	6	20	4	48
Agente 3	4	18	11	55
Agente 4	1	9	10	14

*adicionados por informação vinda de outros agentes

Média de lixo máximo que ficou nos contentores: 4.33.

Análise de Resultados: Conseguimos verificar que com uma condição mais baixa para o camião ir ao depósito de lixo esvaziar, não há tanto tempo perdido, e consegue-se esvaziar mais contentores e consequentemente recolher mais lixo.

5 Conclusões

Conforme os resultados das experiências, pudemos verificar que a eficiência da soma de lixo apanhado e das idas ao depósito aumenta quando introduzido um maior valor de visão para os agentes, quando estes não alteram a sua rota pré-definida, e quando a condição para ir esvaziar o seu lixo atual é mais baixa.

5.1 Melhoramentos

Apesar de considerarmos que este se trata de um trabalho já com alguma complexidade, vemos que existe espaço para expandir e melhorar. Alguns desses pontos seriam, por exemplo, a introdução da variante *tempo* na simulação, sendo que ao longo de um dia todos os fatores alterariam segundo uma função definida (nº de camiões a trabalhar, quantidade de lixo produzido, etc.). Outro ponto a melhorar seria a inclusão de uma interface gráfica para escolher as diferentes variações do nosso problema, sendo que de momento estas apenas podem ser feitas alterando *flags* no código.

Finalmente, a implementação dos camiões remotos ficou com uma lacuna, em que estes recebem *inform*s dos camiões na instância principal do programa mas não conseguem enviar, pois os últimos não estão registados no seu *Directory Facilitator*.

6 Recursos

6.1 Bibliografia

- **Agentes e Inteligência Artificial Distribuída,**
<http://paginas.fe.up.pt/~eol/AIAD/aiad1415.html>
- **Developing multi-agent systems with Jade,**
http://homepages.abdn.ac.uk/w.w.vasconcelos/pages/teaching/CS4027/abdn.only/jade_book.pdf

6.2 Software

- **Eclipse IDE,** *<https://www.eclipse.org/downloads/>*
- **Jade RMA Interface,** *<http://jade.tilab.com/>*

O trabalho foi distribuído de forma igualitária por todos os elementos do grupo