

Universidade do Minho - Escola de Engenharia

Mestrado Integrado em Engenharia Informática

Agentes Inteligentes

Trabalho prático

Gestão do fluxo de tráfego aéreo

Carlos José Gomes Campos a74745

José Pedro Ferreira Oliveira a78806

Ludgero da Silva Diogo pg38417

12 de Novembro de 2018

Resumo Este relatório pretende descrever o desenho da arquitetura da solução e a implementação do sistema multiagente, desenvolvido no âmbito da unidade curricular de Agentes Inteligentes com o intuito de apresentar uma possível solução para o problema de gestão do fluxo do tráfego aéreo, utilizando técnicas de computação baseadas em Agentes.

1 Introdução

1.1 Contextualização

Este relatório surge no âmbito do trabalho prático da unidade curricular de Agentes Inteligentes, que pertence ao perfil de Sistemas Inteligentes do Mestrado Integrado em Engenharia Informática, da Universidade do Minho.

1.2 Caso de Estudo

O projeto tem por base a concepção de um sistema, desenvolvido num ambiente JADE (Java Agent Development Framework), para monitorizar uma rede de sensores de captura de localização GPS integrados num ambiente de controlo de tráfego aéreo.

Os intervenientes são as aeronaves e os aeroportos, que devem estabelecer comunicações para a negociação, com o objetivo de automatizar o processo de gestão de viagem das várias aeronaves, tendo em conta um conjunto de variáveis fulcrais que caracterizam este processo.

1.3 Estrutura do Relatório

Este projeto está dividido em duas fases, nesta primeira é apresentada a parte de modelação do projeto, utilizando a linguagem UML (Unified Modeling Language) adaptada ao novo paradigma de computação baseada em Agentes.

Na secção 2 serão explicados os diagramas de atividades de maneira a apresentar a primeira abordagem aos algoritmos usados pelos agentes face à necessidade de negociar/agir com outros elementos do sistema. Na secção 3 serão descritos os diagramas de sequência que ilustram as trocas de mensagens entre agentes nos seus diferentes estados.

De seguida, na secção 4 será apresentado o diagrama de estado para o agente aeronave e, por fim, na secção 5 será ilustrado o diagrama de classes idealizado para este sistema multiagente. O relatório termina com as conclusões na secção 6 onde é feita uma reflexão acerca do trabalho realizado bem como do trabalho futuro.

2 Diagramas de Atividades

Nestes diagramas são explicados todos os processos executados pelos agentes e que advêm da troca de mensagens entre si, pretende-se assim descrever com mais rigor as atividades internas dos agentes.

Para facilitar a compreensão do leitor, decidimos dividir esta parte da modelação em três diagramas de atividades, em que cada diagrama representa um estágio da viagem do avião, sendo eles a decolagem, o período de negociação entre aeronaves, e a aterragem.

2.1 Decolagem

Inicialmente, o Agente Aeronave verifica a condição meteorológica do aeroporto em que se encontra, e se existem pistas livres, e só prossegue caso estas sejam favoráveis ao início de viagem, caso isso se verifique, pede ao aeroporto destino autorização para decolar, sendo que o Agente Estação só dá luz verde se tiver lugar para estacionamento disponível e se as condições meteorológicas sejam suficientes para uma aterragem em segurança.

Caso alguma destas condições não se verifique, a estação destino manda a aeronave esperar e adiciona-a a uma fila de espera através do seu AID (Agent Identifier). Deste modo, quando existirem condições para aterrar vai ser dada autorização para iniciar viagem.

Antes de decolar o avião envia ao aeroporto destino, a informação relativa à viagem, como as coordenadas iniciais e a velocidade, para que posteriormente a estação destino possa calcular a prioridade de aterragem da aeronave.

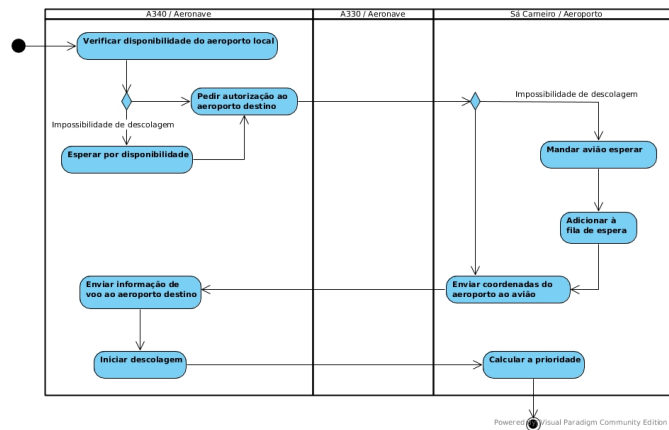


Figura 1. Diagrama de Atividades que descreve a decolagem

2.2 Negociação entre aeronaves

No decorrer da viagem, cada avião envia em broadcast as suas coordenadas para todos os outros aviões ativos. Simultaneamente, está constantemente pronto a receber as coordenadas dos outros aviões ativos, para efetuar o cálculo da distância entre ambos e assim descobrir se outros aviões estão na sua área de aproximação ou não.

Caso isto se confirme, o avião verifica se existe a possibilidade de cruzamento de rotas, assim verifica se a direção da rota do outro avião é a mesma que a sua e o sentido é o oposto, ou seja, verificar a existência de uma colisão frontal, neste caso o avião efetua um ligeiro desvio à direita.

Caso exista apenas cruzamento de rotas, o avião que estiver mais perto do destino diminui a velocidade, e o outro, que está mais longe do seu destino, aumenta. Se nenhuma destas condições for verdadeira, os aviões efetuam a viagem normal.

Se existir alguma mudança de rota/velocidade, o avião informa a estação destino que tal aconteceu enviando os novos parâmetros que sofreram alterações, para que esta possa recalcular a prioridade de chegada.

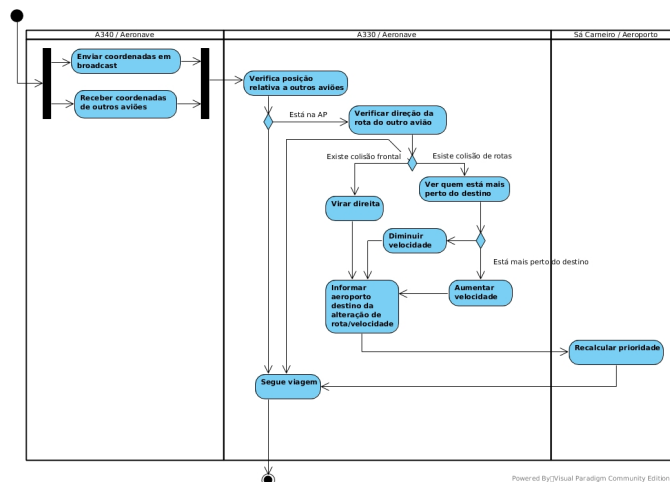


Figura 2. Diagrama de Atividades que descreve a negociação entre aeronaves

Este diagrama de atividades é visto apenas de uma perspectiva, ou seja, ambos os aviões fazem o cálculo da distância e tomam uma decisão que evita o choque, independentemente da decisão do outro, assim torna o sistema mais tolerante a falhas, como por exemplo o caso da troca de mensagens não ocorrer.

Essas decisões são independentes porque se ambos os aviões estiverem prestes a colidir de frente, ambos fazem um desvio para a sua direita, caso tenham apenas cruzamento de rotas, eles tomam uma decisão baseado em factos absolutos, como é a variável da distância a percorrer.

Esta negociação é um pouco individualizada, porque basicamente os aviões apenas recebem e enviam informação referente a localizações, e com base nesta tomam decisões que evitam colisões, mas não passam por um período de "discussão" para ver quem é que faz o quê.

2.3 Aterragem

O período de aterragem começa quando o Agente Aeronave entra na área de aproximação do Agente Estação destino. Quando isto acontece, a aeronave pede autorização para aterrar, de seguida o Agente Estação indica a velocidade que a aeronave deve seguir, tendo por base a lista de prioridades previamente calculada, de modo a evitar colisões entre aeronaves e facilitar a gestão dos recursos do aeroporto.

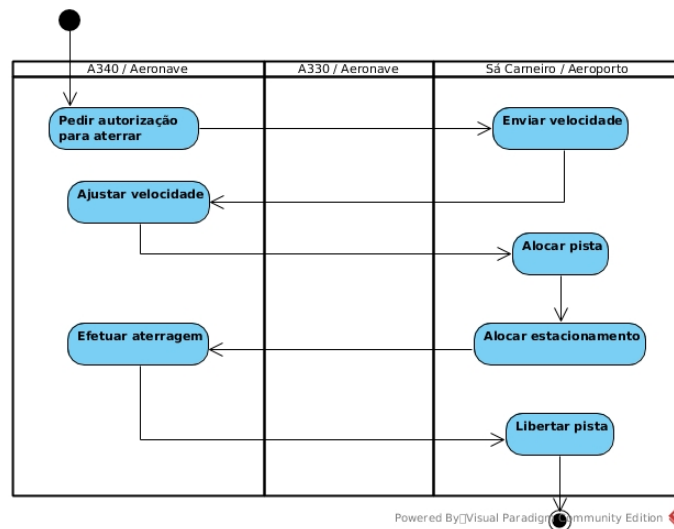


Figura 3. Diagrama de Atividades que descreve a aterragem

3 Diagramas de Sequência

Tal como nos diagramas de atividades, também foi feita uma divisão em três estágios do processo de viagem. Nestes diagramas, é ilustrado as mensagens que são trocadas entre agentes, mostrando o tipo de *performatives* (tipo de mensagem) utilizadas e uma breve referência ao seu conteúdo.

3.1 Descolagem

A primeira mensagem que o agente aeronave envia é um *request* para a aeroporto em que se encontra, de maneira a verificar se as condições meteorológicas são favoráveis ao início da viagem, bem como para ver se existem pistas disponíveis. Caso isso não se verifique, a estação põe a aeronave em espera, já que esta aguarda por uma resposta de confirmação.

Posteriormente é executado um processo semelhante com a estação destino, sendo que não existe dependência sobre o número de pistas livres, mas sim sobre o número de lugares de estacionamento disponíveis. Após ser confirmado o pedido à estação destino, o avião inicia a viagem.

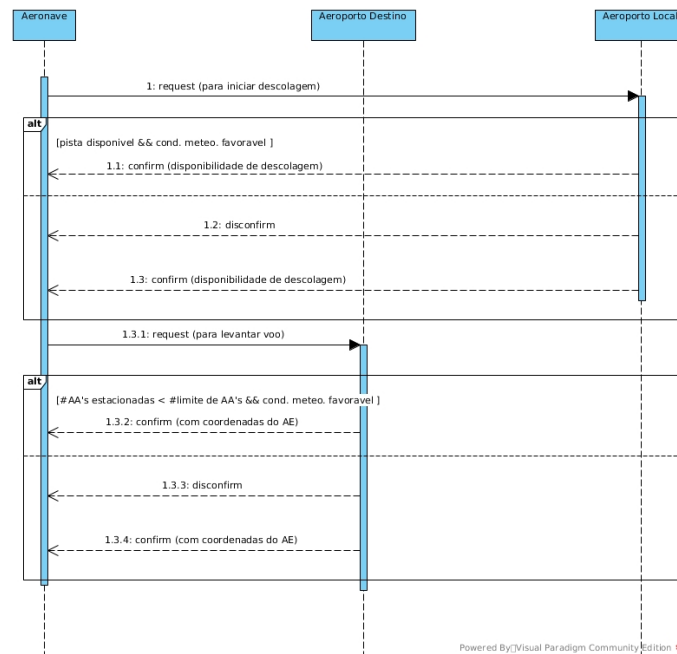


Figura 4. Diagrama de Sequência que mostra as mensagens trocadas na descolagem

3.2 Negociação entre aeronaves

Durante o voo, os Agentes Aeronave enviam constantemente as suas coordenadas a outros semelhantes que estejam ativos (utilizando performativos *Inform*) e recebem coordenadas dos mesmos. Sendo assim possível descobrir se estão numa rota de colisão, e dependendo do rumo dos aviões, tomar medidas evasivas tais como, abrandar, acelerar e virar. Posteriormente, caso seja verificado uma modificação de trajetória/velocidade, é enviada ao aeroporto destino as características da viagem que sofreram alteração no processo de negociação, para que seja recalculada a lista de prioridades de aterragem.

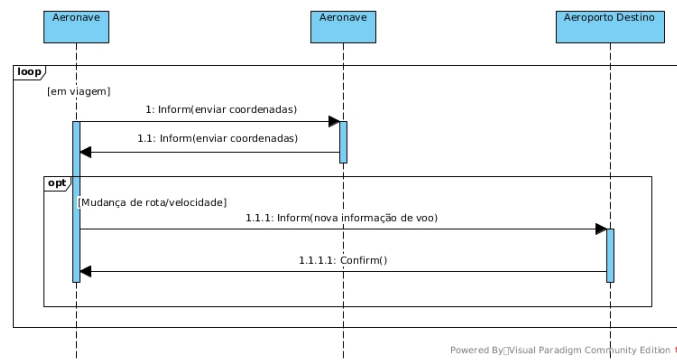


Figura 5. Diagrama de Sequência que mostra as mensagens trocadas no decorrer da viagem

3.3 Aterragem

Quando o avião entra na área de aproximação do aeroporto destino, envia um *request* para aterrizar, ao qual a estação responde com uma confirmação, em que o seu conteúdo possui a velocidade para o avião ajustar nesta fase final.

Quando o mesmo aterra e termina a sua viagem, informa a estação de tal acontecimento, estacionando posteriormente. A estação acaba por confirmar essa informação, libertando os recursos usados para a aterragem.

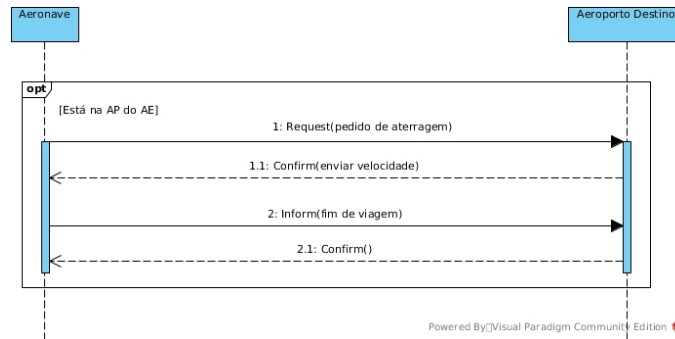


Figura 6. Diagrama de Sequência que mostra as mensagens trocadas na aterragem

4 Diagrama de Estados

4.1 Agente Aeronave

Este diagrama tem como objetivo representar os possíveis estados e as transições entre os mesmos que o Agente Aeronave pode ter ao longo do seu ciclo de vida no sistema. A inicialização de um Agente Aeronave passa pela associação a um aeroporto, sendo essa a sua posição inicial e aí permanece no estado *Parado* até que seja efetuada uma viagem.

Quando se encontra em viagem, o Agente Aeronave exibe dois comportamentos, no estado *Partilhar Posição* divulga a sua posição aos restantes AA's e quando for detetado um conflito o agente passa ao estado *Negociar com AA* onde procura chegar a um entendimento com o outro agente, voltando ao estado anterior quando a situação estiver resolvida. Em paralelo, este agente terá a possibilidade de comunicar com o aeroporto destino, de maneira a informar as eventuais alterações às características da viagem que tenham sido modificadas no processo de negociação de conflitos (velocidade, trajetória, etc.).

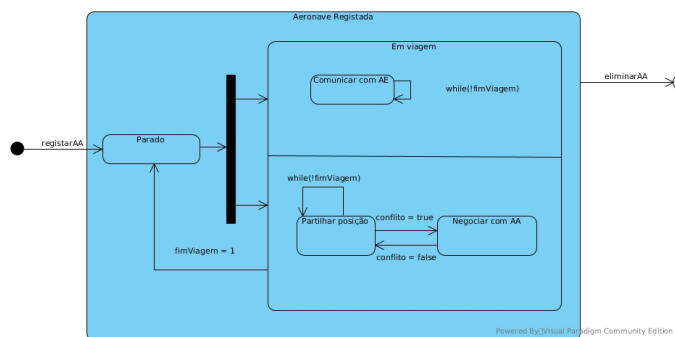


Figura 7. Diagrama de Estados de um Agente Aeronave

5 Diagrama de Classes

Pelo Diagrama de classes podemos observar como será a estrutura da solução que estamos a propor. Neste caso podemos ver que existem 3 classes que são a classe agente aeronave, que representa os aviões, a classe Agente Estação, que representa os aeroportos e a classe Agente Interface que mostra ao utilizador como os aviões se movimentam entre diferentes aeroportos. Estas três classes são especificações da classe Agent, que é uma classe disponibilizada pela biblioteca JADE.

É de notar que o Agente Estação possui uma classe interna, denominada por EstaçãoComp, que herda de uma outra classe da biblioteca JADE, a CyclicBehaviour, fazendo da classe EstacaoComp um comportamento cíclico, que vai executar a função action periodicamente. Neste mesmo cenário vão ser incluídas as classes EnviarCoordsComp e a ReceberCoordsComp, mas estas irão ser classes internas do Agente Aeronave.

Este último agente também vai possuir mais dois comportamentos, que são a DescolagemComp e a AterragemComp, que herdam da classe OneShotBehaviour, que faz com que estas executem a função action apenas uma vez.

Em cada classe, estão representadas as suas variáveis de instância e os seus métodos, sendo que neste caso, os agentes só têm variáveis de instância, e os comportamentos possuem apenas funções. O sinal que prefixa cada nome representa a sua exposição, private para o sinal '-' e public para o sinal '+', esse mesmo nome é precedido pelo seu tipo interno.

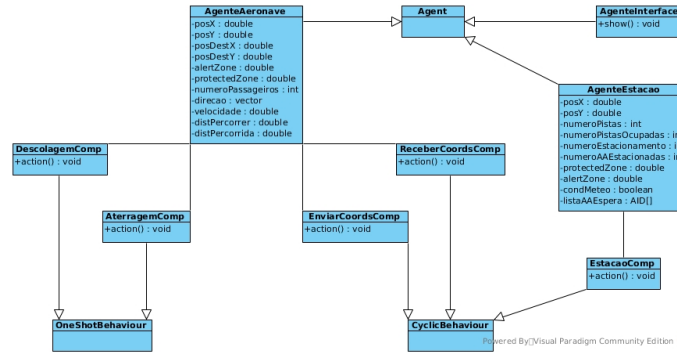


Figura 8. Diagrama de Classe

6 Conclusão

Após a realização desta primeira parte do projeto, o grupo ficou com uma ideia concreta do sistema de software a produzir na próxima fase, através da modelação UML foi possível representar com algum detalhe as estratégias para a resolução dos desafios apresentados.

No paradigma de programação direcionada aos agentes, existe uma variação da linguagem *UML*, que é o *AUML* (Agent Unified Modeling Language). Neste projeto não foi utilizada esta variação na íntegra, pois o grau de complexidade do sistema a modelar não justificava o uso.

É de realçar que esta fase é uma projeção do trabalho que iremos terminar na segunda fase, sendo que alguns destes diagramas poderão vir a ser refinados no futuro.

Das várias estratégias concebidas nesta solução, podemos destacar a preocupação em não sobrecarregar os Agentes Estação, optando pelo envio das coordenadas dos aviões apenas na descolagem, e caso haja um desvio na trajetória destes é reenviada a informação atualizada acerca dos parâmetros alterados. Esta é uma alternativa a uma comunicação contínua entre AA e AE, que com um elevado fluxo de aeronaves, poderia sobrecarregar o sistema computacional do aeroporto. Tendo isto em conta, e aliado ao facto de existir o cálculo das prioridades, mal o avião inicie viagem, as estações adquirem conhecimento suficiente para fazer a gestão atempada dos recursos disponíveis, evitando conflitos e atrasos.