



Universidade do Minho

Departamento de Informática

Perfil Sistemas Inteligentes

Unidade Curricular de Agentes Inteligentes

Edição 2017/2018

Trabalho prático

Tema AGENTES E SISTEMAS MULTIAGENTE

Objetivos de aprendizagem

Com a realização deste trabalho prático pretende-se que os alunos:

- Conheçam os principais conceitos da computação baseada em Agentes;
- Identifiquem e caracterizem diferentes áreas de aplicação;
- Concebam uma arquitetura distribuída baseada em agentes para um dado problema a escolha;
- Conheçam as principais ferramentas oferecidas e/ou interconectam a plataforma Jade.

Enunciado

Este enunciado pretende ser o ponto de partida para a conceção e o desenvolvimento de um sistema multiagente utilizando o ambiente de desenvolvimento JADE, sendo ainda incentivado o uso de JADEX e JESS no desenvolvimento de agentes. Deverão usar o Agent UML para formalizar os protocolos de interação entre agentes. Para isso, será necessário o desenvolvimento de uma solução para o seguinte problema:

Conceber e desenvolver uma arquitetura distribuída para a monitorização de vários sensores virtuais de captura de localização GPS, representados por agentes.

Os **Sistemas de Partilha de Bicicletas (SPB)** permitem aos utilizadores alugar e devolver bicicletas para realizar viagens curtas. Os utilizadores alugam e devolvem as bicicletas em estações de bicicleta dedicadas, que normalmente encontram-se distanciadas a uma centena de metros entre cada uma das estações. Cada estação possui uma capacidade fixa, que determina o número de bicicletas que podem ser armazenadas. Apesar deste sistema ganhar recentemente grande popularidade como uma alternativa de transporte ecológico nas grandes cidades, sofre de um problema comum: o **Problema do Reequilíbrio de Partilha de Bicicletas (PRPB)**. O funcionamento de um **SPB** sem intervenção intencional resulta em desequilíbrios na gestão de ocupação da estação: enquanto algumas estações sofrem de falta de bicicletas, o que impede o aluguer nessas estações, outras sofrem de congestionamento, o que impede a devolução de bicicletas. Para garantir a alta satisfação dos utilizadores e um aumento de receitas ao operador, é necessária uma abordagem efetiva de reequilíbrio para manter um estado de sistema equilibrado.

Uma das abordagens de resolução deste problema baseia-se em oferecer incentivos aos utilizadores para convencerem para alterar as suas rotas de ciclismo e recomendar a devolução das bicicletas a estações selecionadas, garantindo o equilíbrio geral do sistema. O utilizador recebe o incentivo no seu smartphone e pode então decidir se quer aceitar. Se suficientes utilizadores forem persuadidos a escolher rotas de ciclismo benéficas ao sistema, os desequilíbrios consequentemente serão reduzidos.

Através de um processo de monitorização de agentes, o sistema apresenta conhecimento das posições dos utilizadores (GPS) e quando os alugueres e devoluções ocorrem nas estações.

Para especificar a proximidade dos utilizadores a uma estação, será necessário definir o conceito de **Área de Proximidade de uma Estação (APE)**. Um utilizador é lembrado da **APE** se este apresenta uma

bicicleta alugada e encontra-se ao alcance da estação. Este conceito está ilustrado na Figura 1, onde os utilizadores Alice, Bob e Carol apresentam-se ao alcance da estação s1, enquanto o Dave encontra-se fora desta situação. A *APE* é útil, pois os membros associados a uma estação são os utilizadores que podem, no futuro próximo, aumentar a ocupação de bicicletas dessa estação, uma informação que deve ser usada para a previsão da demanda. Além disso, os membros da *APE* são os principais destinatários potenciais de incentivos para devolver as bicicletas alugadas nessa estação.

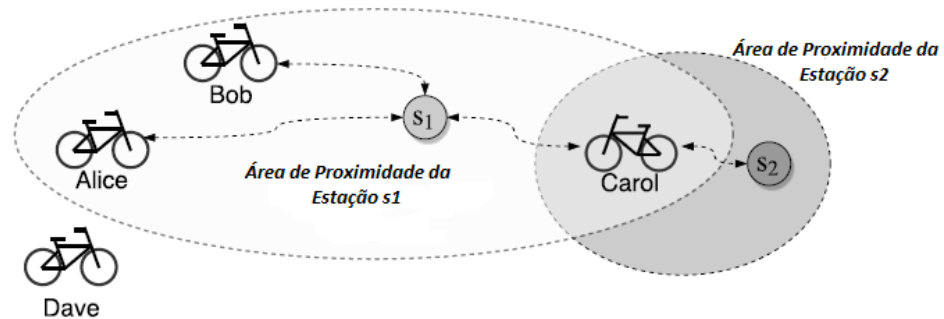


Figura 1: Áreas de proximidade de duas estações s1 e s2

A abordagem de reequilíbrio consiste em dois tipos de agentes, que apresentam o seguinte comportamento:

- **Agente de Estação (AE):** *AEs* representam as estações do *SPB* e escutam as mensagens dos *AUs*. Sempre que um usuário entra ou sai da sua *APE* (conforme indicado pela *AU* com uma mensagem de evento), a *AE* atualiza o seu conhecimento e associa os *AUs* que se encontrem dentro da sua área de proximidade. As *AEs* também monitorizam os alugueres e devoluções das bicicletas que ocorrem na estação. Com estes dados, os *AEs* determinam o seu estado, caracterizam o nível de preenchimento da estação: baixo, médio ou alto. Com base nessa informação, o agente infere se quer atrair bicicletas de usuários próximos e se deve oferecer um incentivo;

- **Agente Utilizador (AU):** *AUs* representam agentes associados a smartphones pessoais dos utilizadores. Os *AUs* usam a capacidade dos smartphones para monitorizar os dados do GPS fornecidos pelo sensor GPS do smartphone, como forma de influenciar a decisão de seleção da estação para a devolução da bicicleta do utilizador. Com base na posição e destino do utilizador, o agente determina (durante a movimentação do utilizador) a distância do utilizador para as estações de *SPB*, baseando-se na *APE* de cada estação. Quando a distância da viagem percorrida entre o agente e a estação de destino ultrapassar $\frac{3}{4}$ da distância do trajeto completo (desde o aluguer da bicicleta até à estação final), o *AU* começará a ser solicitado para entregar a bicicleta, de acordo com o esquema *APE* mencionado anteriormente e o estado da estação. O utilizador poderá aceitar ou rejeitar o pedido, de acordo com o incentivo definido (p.e. através da percentagem de desconto oferecido. Cabe ao *AU* decidir, através de um conjunto de regras definidos no agente, se deve ou não aceitar). Os *AUs* deverão registar-se no diretório de serviços (Directory Facilitator) ao serem inicializados, como forma dos *AEs* apresentarem a capacidade de comunicar com estes. Desta forma em todo o momento pode estar ativo um número desconhecido de agentes, cada um com um nome também desconhecido;

- **Agente Interface:** desenvolva um agente chamado Interface, com o qual o utilizador vai interagir com os *AEs* como forma de observar a gestão das bicicletas de cada uma. Esta observação pode ser textual (e.g. imprimir, a pedido, etc.) ou visual (desenvolver um interface gráfico que vai mostrando a evolução da temperatura a intervalos regulares). Sendo este o caso, considere o uso de uma API gráfica tal como a JFreeChart.

Para a resolução do problema, deve começar por analisar o código destes agentes de forma a perceber o seu funcionamento, nomeadamente em termos de comunicação e de gestão de recursos. Nesse sentido, desenvolva um esquema do seu protocolo de comunicação (i.e. a que mensagens responde, com que mensagens pode responder) e do algoritmo de distribuição de bicicletas entre as diferentes estações. Além disso, deverá ser necessário ter em conta a questão dos *AUs* aceitarem ou não o incentivo (de acordo com o desconto fornecido pelo *AE*, e com um conjunto definido de regras dentro do *AU*, deverá tomar uma decisão de aceitar ou recusar o incentivo.) Tendo em conta a quantidade de bicicletas apresentadas numa

determinada estação, quanto menor o nº de bicicletas maior será a percentagem de desconto oferecido ao AU.

Na 1ª parte deste trabalho, cada grupo deverá elaborar um estado da arte sobre os agentes e sua aplicação a domínios concretos, abordando as diferentes propriedades e vertentes. Deverão ainda conceber e modelar uma arquitetura distribuída baseada em agentes para o dado problema. Esta arquitetura servirá de ponto de partida para a 2ª parte do trabalho prático, no qual deverá ser entregue o código desenvolvido e o relatório associado.

Entrega

A data para a entrega final do relatório e apresentação das conclusões do trabalho é fixada no dia 26 de novembro de 2017. A sessão de apresentação decorrerá no período de aulas correspondente desta unidade curricular no dia 27 de novembro de 2017.

A entrega intermédia da 1ª parte do trabalho terá de ser efetuada até ao dia 13 de novembro 2017.

Cada grupo disporá de 10 minutos para a apresentação dos principais resultados alcançados.

Aconselha-se ainda a consulta:

- 1 Pokahr, A., Braubach, L., & Lamersdorf, W. (2005). Jadex: A BDI reasoning engine. *Multi-Agent Programming*, 149–174.
- 2 Balachandran, B. M. (2008). Developing intelligent agent applications with JADE and JESS. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems* (pp. 236–244).
- 3 Bauer, B., Müller, J. P., & Odell, J. (2001). Agent UML: A formalism for specifying multiagent software systems. *International Journal of Software Engineering and Knowledge Engineering*, 11(3), 207–230.
- 4 Bellifemine, F. L., Caire, G., & Greenwood, D. (2007). *Developing multi-agent systems with JADE* (Vol. 7). John Wiley & Sons.
- 5 Contardo, C., Morency, C., & Rousseau, L.-M. (2012). *Balancing a dynamic public bike-sharing system* (Vol. 4). Cirrelet Montreal.