

MODELAGEM MULTIAGENTE PARA ARP COM JASON, JADE E CARTAGO

João Paulo da Silva Fonseca, Manufacturing Automated Planning Laboratory, jpsfonseca@doutorado.ufu.br
Bruno Luiz Pereira, Manufacturing Automated Planning Laboratory, bruno_luiz@meca.ufu.br
Pedro Ventura de Oliveira, Manufacturing Automated Planning Laboratory, ventura.ufu@gmail.com
José Jean-Paul Zanlucchi de Souza Tavares, Manufacturing Automated Planning Laboratory, jean.tavares@mecanica.ufu.br

Resumo. Os principais desafios referentes às aeronaves remotamente pilotadas estão relacionados ao voo colaborativo. Isto demanda novas alternativas de controle, tanto de estabilidade como de trajetória, e reforça a necessidade de um protocolo de comunicação bem definido para a troca de informações entre aeronaves. Devido às características de autonomia, habilidade social, reatividade e pró-atividade, o paradigma de agentes surge como uma alternativa interessante ao voo colaborativo. Este artigo apresenta o processo de modelagem de um sistema multiagente com aeronaves remotamente pilotadas. O modelo corresponde a um sistema de solicitação e entrega de produtos e foi concebido utilizando as plataformas Jason, Jade e Cartago. Resultados preliminares de simulação são apresentados, revelando um alto nível de abstração, controle distribuído e alto grau de autonomia e cooperação sem uma estrutura cliente servidor fixa.

Palavras chave: aeronave remotamente pilotada, sistemas multiagentes, jason, jade, cartago

1. INTRODUÇÃO

A aplicação de aeronaves remotamente pilotadas (ARP) tem se tornado assunto de interesse de pesquisadores e empresas devido a suas aplicações civis e militares. Suas principais atribuições são: tarefas de busca e resgate (Gan e Sukkarieh, 2011; Doherty e Rudol, 2007; Waharte e Trigoni, 2010), entregas (Bearth, 2013) reconhecimento de ambientes hostis (Tanner, 2007), inspeção e segurança (Irizarry et al., 2012) e na indústria cinematográfica (Weitzman, 2007).

Para Valanis (2007), os principais desafios referentes aos ARP estão relacionados ao voo colaborativo. Isso demanda novas tecnologias de controle, de comunicação e computacionais. Um paradigma relativamente recente que embarca novas tecnologias neste âmbito é o paradigma de agentes. Para Wooldridge e Jennings (1995) agentes inteligentes podem ser encarados como módulos de software independentes que apresentam quatro características chave: autonomia, habilidade social, reatividade e pró-atividade.

Um Sistema Multiagente (SMA), por sua vez, foi definido por Lesser et al. (2000) como sendo um sistema em que dois ou mais agentes interagem ou trabalham em conjunto de forma a desempenhar um determinado conjunto de tarefas ou satisfazer um conjunto de objetivos.

A programação de agentes é baseada na programação lógica (um paradigma de programação que faz uso da lógica matemática). A primeira linguagem de programação lógica foi a Planner (Hewitt, 1969). Em seguida, o Prolog (Kowalski, 1973) foi desenvolvido como uma simplificação do Planner que permitia a invocação orientada a padrões apenas a partir de objetivos.

AgentSpeak (L) é uma linguagem de programação baseada em linguagem de primeira ordem restrita com eventos e ações. O comportamento do agente (isto é, a sua interação com o ambiente) é ditado pelos programas escritos em AgentSpeak (L). AgentSpeak (L) é uma extensão de linguagem de programação lógica para se trabalhar com programação baseada em agentes. A extensão permite que os agentes coexistam em um ambiente baseado em lógica BDI (crenças, desejos e intenções): as crenças, desejos e intenções (lógica BDI) do agente não são explicitamente representados como fórmulas modais.

- o estado atual do agente, que é um modelo de si mesmo, o seu ambiente, e outros agentes, pode ser visto como o seu atual estado de crença;
- estados que o agente deseja alcançar, com base nos seus estímulos externos ou internos, podem ser vistos como desejos; e
- a adoção de planos para satisfazer tais estímulos podem ser vistos como intenções.

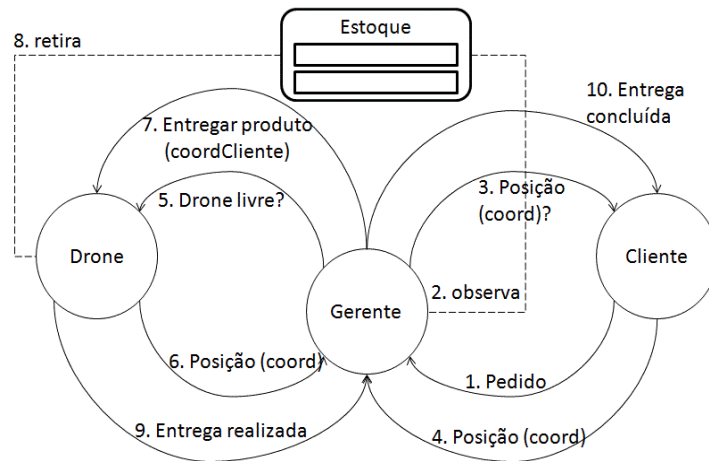


Figura 1. Modelo abstrato de interação entre agentes e artefatos

A linguagem de especificação consiste em um conjunto de crenças básicas (ou fatos, no sentido de lógica de programação) e um conjunto de planos. Os planos são “receitas” que permitem a decomposição hierárquica de objetivos, bem como a execução das ações.

Jason (Bordini et al., 2007) é um sistema de programação de agentes que utiliza uma extensão da linguagem AgentSpeak e implementa uma camada de comunicação de alto nível baseada em *speech-act* (disparo de planos de ações a partir da comunicação entre agentes).

Este trabalho tem como objetivo apresentar a modelagem de um SMA utilizando AgentSpeak em Jason para programar agentes, Jade como infraestrutura de comunicação entre agentes e CArtaGO para a utilização do paradigma de agentes e artefatos (A&A).

As demais seções deste texto estão assim divididas: a seção 2 apresenta a metodologia adotada, a seção 3 reserva os resultados de simulação atingidos e conclusões são ponderadas na seção 4.

2. METODOLOGIA

A metodologia é composta por análise de requisitos, modelo abstrato, implementação computacional e análise dos resultados. O problema em questão consiste em um sistema de solicitação e entrega de produtos. Alguns dos requisitos para a implementação do sistema são listados abaixo:

- existirão três tipos de agentes: cliente, gerente e drone e um artefato de estoque (produto);
- os agentes devem ter capacidade de comunicação entre si e acesso ao estoque;
- a implementação deve ser concebida de modo a permitir o disparo de plano de ações a partir da comunicação entre agentes.

O modelo abstrato deve representar o comportamento dos agentes e o seu interrelacionamento. No cenário proposto os agentes devem respeitar as seguintes premissas comportamentais:

- o cliente interessado em algum produto solicita o serviço para o gerente;
- o gerente deve verificar a capacidade do estoque e aceitar ou recusar o pedido;
- caso o pedido seja aceito o cliente informa sua coordenada geográfica para o gerente;
- neste momento o gerente cria uma nova ordem de entrega e solicita o serviço para um drone (que deve estar livre para realizar esta tarefa);
- o drone disponível aceita o pedido enviando sua coordenada geográfica para o cliente;
- o gerente transmite as informações referentes à entrega (tipo e quantidade e produto e a coordenada do cliente);
- o drone retira o produto do estoque e realiza a entrega;
- o drone informa o gerente sobre o fim da tarefa;
- o gerente conclui a transação com o cliente informando-o sobre o sucesso da operação.

Estas premissas estão representadas graficamente no modelo abstrato da Fig. (1).



Figura 2. Legenda utilizada para os modelos comportamentais dos agentes.

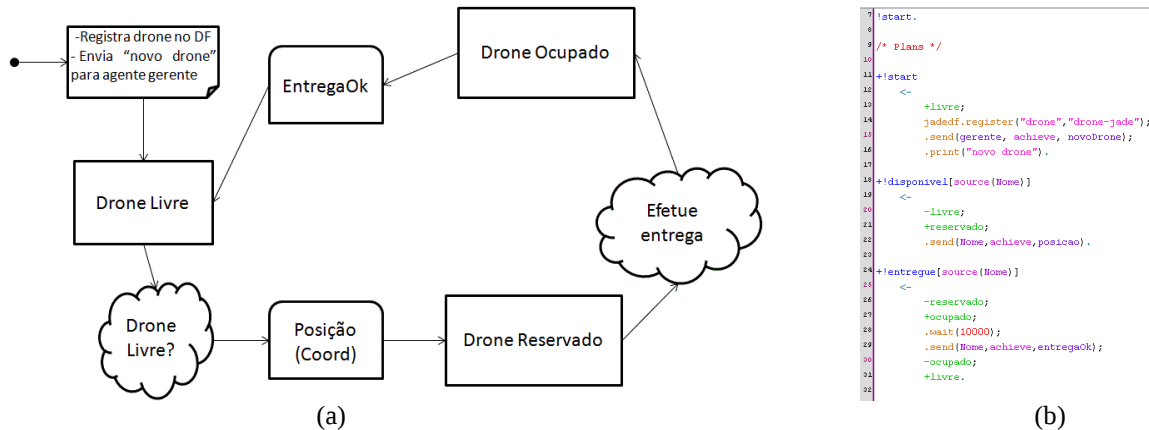


Figura 3. Modelagem para o agente drone (a) e implementação computacional parcial no Jason (b)

A implementação computacional dos agentes é baseada na utilização da plataforma Jason para a programação dos agentes, Jade como infraestrutura de comunicação e Cartago para utilização do paradigma de agentes e artefatos. Para isto, cada agente deve ser modelado individualmente e implementado utilizando em Jason. Os esquemas que representam os comportamentos dos agentes seguem a legenda apresentada na Fig. (2). A Fig. (3a) ilustra o modelo abstrato de comportamento e a Fig. (3b) parte do código para a implementação do “agente drone”.

3. RESULTADOS

Os agentes foram programados em Jason, Jade foi utilizado como infraestrutura de comunicação entre agentes e Cartago permitiu a implementação do artefato estoque.. Inicialmente estes agentes devem solicitar ao RMA (*remote management agent*) do Jade uma permissão para integrarem o container principal de agentes disponível na plataforma Jade. Uma vez que os agentes estão alocados no container principal do Jade, passam a trocar informações entre si, e o comportamento programado em Jason é estabelecido. O agente *Sniffer* do Jade apresenta o diagrama de comunicação entre os agentes, tal como ilustrado na Fig. (4).

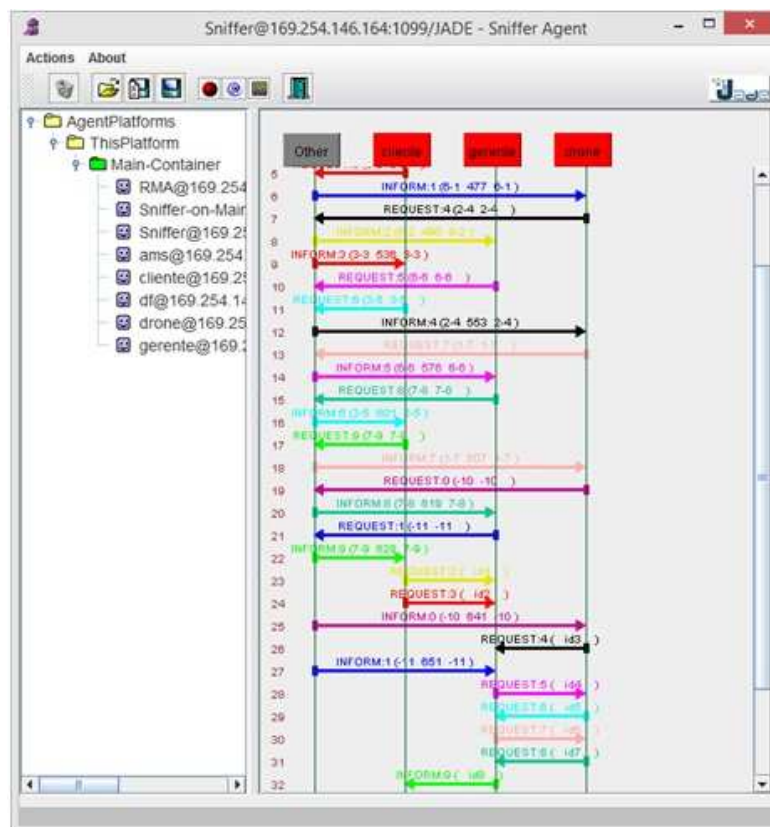


Figura 4. Diagrama de sequência de troca de mensagens entre agentes utilizando o agente *Sniffer* do Jade.

4. CONCLUSÃO

O paradigma de SMA substitui o controle centralizado por um comportamento distribuído. Isto permite atingir um alto grau de autonomia e cooperação sem uma estrutura cliente servidor fixa. É necessária uma interface entre o mundo abstrato (agentes Jason) e as entidades reais (sistemas embarcados) para que o modelo seja implementado na prática. Esta interface deverá relacionar crenças, eventos, objetivos e ações com percepções e execução dos sensores e atuadores. O estudo em questão apresenta um modelo simplificado e necessita de uma análise mais profunda para sua expansão. Futuramente estão previstos o aperfeiçoamento do modelo para atender o gerenciamento de energia e análise de eventos probabilísticos, o desenvolvimento de um ambiente que simule a navegação dos drones, a definição dos protocolos de comunicação e meios de transmissão de informações e a integração com o ambiente físico, ou seja, embarcar os agentes em microcontroladores e verificar o seu comportamento.

5. AGRADECIMENTOS

Os autores prestam sinceros agradecimentos ao Programa de Pós-Graduação em Engenharia Mecânica, à FEMEC, à UFU e às agências de fomento CAPES, CNPQ e FAPEMIG.

6. REFERÊNCIAS

- Bearth, D., 2013, "Amazon.com tests same-day delivery of small packages shuttled by drones", Northwestern University Transportation Library: American Trucking Associations ISSN: 0041-1558 p. 32.
- Bordini, R., Hubner, J, e Wooldridge, M., 2007, "Programming Multi-Agent Systems in AgentSpeak Using Jason". John Wiley and Sons, Ltd.
- Doherty, P.; Rudol, P., 2007, "A UAV Search and Rescue Scenario with Human Body Detection and Geolocalization". In: ORGUN, M.; THORNTON, J. AI 2007: Advances in Artificial Intelligence. Gold Coast, Australia: Springer Berlin / Heidelberg, Vol. 4830, pp. 1-13.
- Gan, S. K.; Sukkariéh, S., 2011, "Multi-UAV target search using explicit decentralized gradient-based negotiation", IEEE International Conference on Robotics and Automation (ICRA). Shangai, China, pp. 751-756.
- Hewitt, C., 1969, "PLANNER: A Language for Proving Theorems in Robots", In: Proceedings of the 1st international joint conference on Artificial intelligence, Vol. 1, San Francisco, USA, pp. 295-301.
- Irizarry, J., Gheisari, M., Walker, N., 2012, "Usability assessment of drone technology as safety inspection tools", Journal of Information Technology in Construction, Vol. 17, pp.194-212.
- Kowalski, R, "Predicate Logic as Programming Language", memo 70, University of Edinburgh, School of Artificial Intelligence, Nov. 1973. Also in Proceedings of IFIP 1974, Amsterdam: North Holland, Amsterdam, 1974, pp. 569--574.
- Lesser, V.; Horling, B.; Klassner, F.; Raja, A.; Zhang, S., 2000, "BIG: An Agent for Resource-Bounded Information Gathering and Decision Making", Artificial Intelligence Journal, Special Issue on Internet Information Agents Vol. 118, pp. 197-244.
- Tanner, H. "Switched UAV-UGV cooperation scheme for target detection, 2007, "IEEE International Conference on Robotics and Automation, pp. 3457-3462.
- Valavanis, K., 2007, "Advances in Unmanned Aerial Vehicles : State of the Art and the Road to Autonomy", Tampa, Florida, USA: Springer.
- Waharte, S.; Trigoni, N., 2010 "Supporting Search and Rescue Operations with UAVs", International Conference on Emerging Security Technologies, Canterbury, UK., pp. 142-147.
- Weitzman, G., 1007, "Law Enforcement Tools, Not Toys: Unmanned aerial vehicles are quickly becoming viable law enforcement tool", Airborne Law Enforcement Association (ALEA). Disponível em: http://alea.org/public/airbeat/back_issues/jul_aug_2007/Law%20Enforcement%20Tools.pdf
- Wooldridge, M., Jennings.N., 1995, "Intelligent Agents: Theory and Practice". In Knowledge Engineering Review, Vol. 10, Iss 2.

7. RESPONSABILIDADE PELAS INFORMAÇÕES

Os autores são os únicos responsáveis pelas informações incluídas neste trabalho.