

# Programação orientada a agentes em sistemas embarcados

Orientador: Jomi Fred Hübner  
Co-orientador: Maiquel de Brito

## 1 Motivação e Problemática

Recentes avanços em tecnologias diversas têm possibilitado a criação de sistemas em que programas de computador embarcados nos mais diversos dispositivos, combinados com sensores e atuadores, atuam sobre o ambiente físico. Sistemas com esta característica são frequentemente referenciados na literatura como *cyber-physical systems* (CPS) [11, 3].

CPS podem ser complexos, reunindo uma grande quantidade de dispositivos que atuam de maneira interconectada [5, 3]. Exemplos deste tipo de sistema são *smart grids*, veículos autônomos, sistemas de transporte inteligente, *smart cities* etc [12]. Considerando a complexidade destes sistemas, é desjável que os dispositivos que compõem CPS apresentem características como (i) *autonomia*, para atuar sem a necessidade de um operador humano; (ii) *habilidades sociais*, para colaborar com outros dispositivos na solução de problemas; (iii) *capacidade de adaptação e aprendizado*, para aprimorar sua atuação à medida que participam do sistema; e (iv) *proatividade*, para atuar guiados por objetivos a serem atingidos e não por meras reações ao estado atual do sistema [3, 1, 9].

*Agentes* são uma metáfora adequada para o desenvolvimento de aplicações com as características anteriormente mencionadas [8]. Um agente pode ser definido como um sistema computacional situado em um ambiente e que é capaz de atuar de forma autônoma neste ambiente para atingir seus objetivos [15]. Além da autonomia, agentes possuem, por definição, as demais características citadas anteriormente como relevantes na construção de CPS: habilidades sociais, capacidade de aprender e proatividade [14]. Assim, tem-se indicativos de que a programação orientada a agentes (ou AOP - *agent oriented programming*) é um paradigma adequado à implementação de sistemas embarcados que requeiram autonomia, habilidades sociais, capacidade de aprender e proatividade. Em especial, linguagens AOP baseadas na arquitetura BDI possuem construtores de alto nível, como planos, objetivos etc, que facilitam – em comparação com linguagens de programação tradicionais – o desenvolvimento de sistemas que permanecem em contínua execução, reagindo a eventos, revisando o curso de ações previstas em função de mudanças no ambiente, agindo de forma proativa para atingir objetivos de longo prazo, e incorporando aspectos sociais que permitem a interação e colaboração entre os agentes.

Ferramentas para o desenvolvimento de agentes BDI têm sido, tradicionalmente, concebidas considerando ambientes com abundância de recursos compu-

tacionais (capacidade de processamento, memória etc), tais como computadores pessoais [8]. Por outro lado, os dispositivos que compõem CPS geralmente dispõem de recursos computacionais mais limitados. Esta diferença dificulta a execução de agentes BDI embarcados nestes dispositivos. Assim, o *problema* considerado nesta proposta é o *desenvolvimento de agentes BDI para sistemas embarcados*, considerando os recursos computacionais limitados dos microcontroladores empregados neste tipo de sistema.

Diversos trabalhos têm empregado diferentes abordagens para contornar as dificuldades encontradas para embarcar agentes em dispositivos com recursos computacionais limitados [2, 4, 7, 13, 10, 6]. Estas diferentes propostas, no entanto, são concebidas como soluções *ad hoc* aplicadas a problemas específicos e não necessariamente replicáveis. Torna-se necessário, assim, um estudo que conduza a uma solução de propósito geral para que agentes BDI possam ser embarcados em dispositivos com recursos computacionais limitados, característicos dos sistemas embarcados que compõem CPS.

## 2 Objetivo

O principal objetivo desta proposta é conceber meios para que agentes BDI sejam executados em dispositivos com recursos computacionais limitados. Mais especificamente, pretende-se conceber meios para traduzir especificações de agentes BDI feitas em linguagens AOP de alto nível para linguagens executáveis em microcontroladores.

## 3 Etapas do projeto

O desenvolvimento do projeto passará pelas seguintes etapas (não necessariamente feitas em sequência):

1. estudo sobre agentes, sistemas multiagente e programação multiagente;
2. estudo sobre CPS e sistemas embarcados;
3. estudo sobre trabalhos relacionados;
4. avaliação do estado da arte;
5. definição do microcontrolador a ser utilizado no projeto (por exemplo, o microcontrolador do kilobot);
6. projeto e implementação do mecanismo de tradução de uma linguagem BDI (por exemplo, Jason) para uma linguagem executável no microcontrolador definido na etapa 5 (por exemplo, linguagem C);
7. definição de um cenário de aplicação que requeira o desenvolvimento de um programa de computador a ser embarcado em um microcontrolador;
8. para o cenário escolhido:
  - (a) implementação de uma solução em que o programa a ser embarcado seja desenvolvido usando ferramentas e modelos de programação tradicionais;

- (b) implementação da solução através do desenvolvimento de um agente BDI, cujo código será traduzido e embarcado em um microcontrolador.
- 9. definição de critérios de avaliação;
- 10. avaliação das soluções;
- 11. redação de artigos;
- 12. redação da dissertação.

## 4 Requisitos do candidato

- Facilidade de leitura (e escrita) em inglês;
- Habilidade de programação (estruturada, orientada a objetos e lógica);
- Experiência (ou interesse) com sistemas embarcados;
- Interesse pelo estudo e desenvolvimento de projetos integradores (uso de vários produtos/software, feitos em outros projetos acadêmicos, sem muita documentação, etc.);
- Trabalho em equipe.

## Referências

- [1] Ali Aliyuda. Towards the design of cyber-physical system via multi-agent system technology. *International Journal of Scientific & Engineering Research*, 7(10), 2016.
- [2] Reydson Schuenck Barros, Victor Hugo Heringer, Carlos Eduardo Pantoja, Nilson Mori Lazarin, and Leonardo Machado de Moraes. An agent-oriented ground vehicle's automation using jason framework. In Béatrice Duval, H. Jaap van den Herik, Stéphane Loiseau, and Joaquim Filipe, editors, *ICAART 2014 - Proceedings of the 6th International Conference on Agents and Artificial Intelligence, Volume 2, ESEO, Angers, Loire Valley, France, 6-8 March, 2014*, pages 261–266. SciTePress, 2014.
- [3] Manfred Broy, María Victoria Cengarle, and Eva Geisberger. Cyber-physical systems: Imminent challenges. In Radu Calinescu and David Garlan, editors, *Large-Scale Complex IT Systems. Development, Operation and Management*, pages 1–28, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
- [4] A. Calce, P. Mojiri Forooshani, Andrew Speers, K. Watters, T. Young, and Michael R. M. Jenkin. Autonomous aquatic agents. In Joaquim Filipe and Ana L. N. Fred, editors, *ICAART 2013 - Proceedings of the 5th International Conference on Agents and Artificial Intelligence, Volume 1, Barcelona, Spain, 15-18 February, 2013*, pages 372–375. SciTePress, 2013.

- [5] Sebastian Engell, Radoslav Paulen, Michel A. Reniers, Christian Sonntag, and Haydn Thompson. Core research and innovation areas in cyber-physical systems of systems - initial findings of the cpsos project. In Christian Berger and Mohammad Reza Mousavi, editors, *Cyber Physical Systems. Design, Modeling, and Evaluation - 5th International Workshop, CyPhy 2015, Amsterdam, The Netherlands, October 8, 2015, Proceedings*, volume 9361 of *Lecture Notes in Computer Science*, pages 40–55. Springer, 2015.
- [6] Chien-Liang Fok, Gruia-Catalin Roman, and Chenyang Lu. Agilla: A mobile agent middleware for self-adaptive wireless sensor networks. *TAAS*, 4(3):16:1–16:26, 2009.
- [7] Nilson Mori Lazarin and Carlos Eduardo Pantoja. A Robotic-agent Platform for Embedding Software Agents Using Raspberry Pi and Arduino Boards. In Balduino Fonseca, Viviane Torres da Silva, and Ricardo Choren, editors, *Anais do IX Workshop-Escola de Sistemas de Agentes, seus Ambientes e Aplicações – IX WESAAC*, pages 13–30, 2015.
- [8] Gregory M. P. O’Hare, Rem W. Collier, Mauro Dragone, Michael J. O’Grady, Conor Muldoon, and Alcides de J. Montoya. Embedding agents within ambient intelligent applications. In Tibor Bosse, editor, *Agents and Ambient Intelligence - Achievements and Challenges in the Intersection of Agent Technology and Ambient Intelligence*, volume 12 of *Ambient Intelligence and Smart Environments*, pages 119–133. IOS Press, 2012.
- [9] Carlos Pantoja, Vinicius Souza de Jesus, Fabian Manoel, and José Viterbo. A heterogeneous architecture for integrating multi-agent systems in AmI systems (s). In *Proceedings of the 30th International Conference on Software Engineering and Knowledge Engineering*. KSI Research Inc. and Knowledge Systems Institute Graduate School, 2018.
- [10] S. R. R. D. Purusothaman, R. Rajesh, K. K. Bajaj, and V. Vijayaraghavan. Implementation of arduino-based multi-agent system for rural indian microgrids. In *2013 IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia)*, pages 1–5, 2013.
- [11] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic. Cyber-physical systems: The next computing revolution. In *Design Automation Conference*, pages 731–736, June 2010.
- [12] I. Stojmenovic. Machine-to-machine communications with in-network data aggregation, processing, and actuation for large-scale cyber-physical systems. *IEEE Internet of Things Journal*, 1(2):122–128, April 2014.
- [13] Gabriel Villarrubia, Juan F. De Paz, Javier Bajo, and Juan M. Corchado. Ambient agents: Embedded agents for remote control and monitoring using the PANGEA platform. *Sensors*, 14(8):13955–13979, 2014.
- [14] M. Wooldridge and N.R. Jennings. Intelligent agents: Theory and practice. *Knowledge engineering review*, 10(2):115–152, 1995.
- [15] Michael Wooldridge. *An introduction to multiagent systems*, volume 30. Oxford University Press New York, 1997.