

Teste em Sistemas MultiAgentes

Orientador: [Jomi F. Hübner](#)

Motivação & Problemática

Sistemas multiagentes (SMA) estão sendo explorados em diversas vertentes nas últimas duas décadas. Seja em definição de metodologias (BRINKKEMPER, 1996) (BERNON et al., 2005), exploração de sua organização ou na definição mais complexa de seu funcionamento com o paradigma BDI (BORDINI, HÜBNER, 2006). Porém, pouca atenção tem sido dada em como eles podem ser **testados** (CERNUZZI, 2005).

Dentre os poucos projetos que exploraram este tema, a tese de Zhang (ZHANG et al., 2013) explorou a criação de um algoritmo para criar casos de uso automatizados, explorando diversos tipos de entrada de dados, para a criação de testes unitários. Coelho (Coelho et al., 2006) explorou testes unitários criando uma metodologia de como se pensar para sua criação, explicado abaixo, utilizando a abordagem de criar um agent *mock*. Já a tese de Nguyen (NGUYEN, 2008) focou na criação de uma metodologia de desenvolvimento focada em teste e objetivos (*Goal-Oriented Software Testing*) e na criação de técnicas de teste baseadas em limites do sistema (*constraints*), ontologia e requerimentos. Porém, todos esses projetos foram desenvolvidos com a utilização de linguagens orientadas a objetos com *frameworks* orientados a agente (como o JADE) (BELLIFEMINE et al.) e não linguagens específicas para agentes BDI (como o Jason, 2APL, entre outros) (BORDINI, HÜBNER, 2006). A estrutura BDI é uma arquitetura popularmente aceita que caracteriza um agente pelas suas crenças, desejos e intenções (*Beliefs, Desires, Intentions*).

Testes são difíceis e consomem tempo. Frequentemente mais de 50% do custo de desenvolvimento é gasto em testes (KIT, FINZI, 1995). Ao mesmo tempo, testar é crítico para assegurar a qualidade e reduzir gastos de problemas de software. Segundo a NIST (*National Institute of Standards and Technology*) (NIST, 2002) cerca de \$59.5 bilhões de dólares são gastos anualmente para corrigir problemas de software e mais de um terço poderia ser economizado se melhores testes fossem feitos. Testes em sistemas multi-agentes é uma tarefa desafiadora porque estes sistemas são distribuídos, autônomos e deliberativos. Há problemas relacionados a interoperabilidade, comunicação e coordenação, que são funcionalidades difíceis de criar e programar (BERGENTI et al., 2004), e também de testar.

Objetivo

O objetivo desta dissertação é desenvolver modelos, mecanismos e ferramentas para testar SMA programados em linguagens baseadas em BDI.

Este projeto tem como intenção se apropriar do que já foi estudado e aplicar para uma linguagem BDI uma análise da melhor forma de se testar os agentes criados com este paradigma. A proposta é criar estruturas para se testar agentes que interagem com o ambiente, organização e outros agentes.

Etapas do projeto

Os passos para o desenvolvimento da dissertação são:

1. Estudo aprofundado sobre testes automatizados
2. Estudo aprofundado de programação para agentes
3. Criação de sistemas exemplo para realização de casos de teste, especialmente voltados para aplicações na área de *robótica*
4. Elaboração da proposta inicial para testes destes sistemas
5. Formalização do método de geração de testes pelo projetista
6. Implementação do código e elaboração de uma ferramenta
7. Avaliação dos resultados

Requisitos do candidato

- Facilidade de leitura (e escrita) em inglês
- Habilidade de leitura de textos com formalização matemática e/ou lógica
- Habilidade de programação (estruturada, orientada a objetos e lógica)
- 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

Bibliografia Relacionada

S. BRINKKEMPER. *Method engineering: Engineering of information systems development methods and tools*. Information and Software Technology, 38(4):275–280, 1996.

BERNON, C., CAMPS, V., GLEIZES, M., PICARD, G. *Engineering adaptive multi-agent systems: The adelfe methodology*. In P. Giorgini and B. Henderson-Sellers, editors, *Agent-Oriented Methodologies*, chapter VII. Idea Group Publishing, 2005

BORDINI, R. H., & HÜBNER, J. F. (2006). *BDI agent programming in AgentSpeak using Jason*. Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 3900 LNAI, 143–164.

PADGHAM, L., ZHANG, Z., THANGARAJAH, J., MILLER, T. (2013). *Model-based test oracle generation for automated unit testing of agent systems*. *IEEE Transactions on Software Engineering*, 39(9), 1230–1244.

COELHO, R., KULESZA, U., VON STAA, A., LUCENA, C. (2006). *Unit testing in multi-agent systems using mock agents and aspects*. *Proceedings of the 2006 International Workshop on Software Engineering for Large-Scale Multi-Agent Systems - SELMAS '06*, 83

NGUYEN, C. (2008). *Testing Techniques for Software Agents. Technologies*, C. University of Trento. Bellifemine, F., Romoli, V. G. R., Rimassa, G., & Poggi, A. (n.d.).

JADE – A FIPA-compliant agent framework.

Occello, M., Baeijs C., Demazeau, Y., Koning J. (2004). *MASK: and AEIO toolbox to design and build multi-agent systems*. IOS press.

OMICINI, A., RICCI, A., & VIROLI, M. (2008). *Artifacts in the A&A meta-model for multi-agent systems*. *Autonomous Agents and Multi-Agent Systems*, 17(3), 432–456.

DIGNUM, V., DIGNUM, F. (2001). *Modelling agent societies: Co-ordination frameworks and institutions*. In Pavel Brazdil and Alípio Jorge, editors, *Proceedings of the 10th Portuguese Conference on Artificial Intelligence (EPIA'01)*, LNAI 2258, pages 191–204, Berlin. Springer.

KANER, C., FALK, J., NGUYEN, H. (1999). *Testing Computer Software, Second Edition*. John Wiley & Sons, Inc., New York, NY, USA.

CERNUZZI, L., COSENTINO, M., ZAMBONELLI, F. *Process Models for Agent-based Development*. *Journal of Engineering Applications of Artificial Intelligence*, v. 18 n. 2, p. 205-222, March 2005.

KIT E., FINZI, E. (1995). *Software Testing in The Real World: Improving The Process*. ACM Press/Addison-Wesley Publishing Co.

NIST. (2002). *The Economic Impacts of Inadequate Infrastructure for Software Testing*. Final report, National Institute of Standards and Technology.

WOOLDRIDGE, M. 2002. *An Introduction to Multiagent Systems*. Chichester, UK: John Wiley and Sons.

GEORGEFF, M. P. INGRAND, F. F. (1989), *Decision-making in an embedded reasoning system*, International Joint Conference on Artificial Intelligence.

TAVETER, K., STERLING, L. (2008). *An expressway from agent-oriented models to prototype systems*. In M. Luck and L. Padgham (eds.), *Agent-Oriented Software Engineering VIII: The 8th International Workshop on Agent-Oriented Software Engineering, AOSE 2007*, Honolulu, HI, May 14, Revised Selected Papers. Springer-Verlag.

EYTANI, Y., TZOREF, R., UR, S. (2008). *Experience with a Concurrency Bugs Benchmark*. In ICSTW '08: Proceedings of the 2008 IEEE International Conference on Software Testing Verification and Validation Workshop. IEEE Computer Society.

RUNESON, P. (2006). *A Survey of Unit Testing Practices*. IEEE Software.

MYERS, G. J., SANDLER, C., BADGETT, T., and THOMAS, T. M. (2004). *The Art of Software Testing*, Second Edition. Wiley.

BECK, K. (2003). *Test-Driven Development: By Example*. Addison-Wesley Professional.

MACKINNON, T., FREEMAN, S., CRAIG, P. (2001). *EndoTesting: Unit Testing with Mock Objects*. In: EXTREME PROGRAMMING EXAMINED, Beck, K. Addison Wesley.