

Explorando o Transporte de Agentes Cognitivos entre Sistemas Multi-Agentes Distintos

Vinicius Souza de Jesus
CEFET/RJ
Av. Maracanã, 229 - Maracanã
Rio de Janeiro, Brasil
souza.vdj@gmail.com

Carlos Eduardo Pantoja
CEFET/RJ
R. Miguel Ângelo, 96 - Maria da Graça
Rio de Janeiro, Brasil
pantoja@cefet-rj.br

ABSTRACT

This work aims to transfer cognitive agents between different multi-agent systems programmed in Jason, taking into account the relationship established between the target SMA and the agent, through protocols. The protocols are based on concepts derived from biology, more precisely from the area of ecological relations. This work will focus on the Predator protocol, which is a protocol where agents have as main objective the preservation of knowledge and crucial data, such as in a mission, the hardware where the leading agent is located is reached and their knowledge are key to the success of the mission, then the same self transfers to other hardware, thus preserving the integrity of their knowledge. The implementation proposal will be made by extending the characteristics of a standard agent of the Jason framework to create a mobile agent, and to make the connection between the different SMAs, the ContextNet middleware will be used. Preliminary tests were conducted to make the work feasibility study, for example, sending the source code of a cognitive agent in Jason using ContextNet.

Keywords

Mobile agents; Open MAS; Open Environment

RESUMO

Este trabalho tem como objetivo a transferência de agentes cognitivos entre sistemas multi-agentes distintos programados em Jason, levando em consideração a relação estabelecida entre o SMA de destino e o agente, através de protocolos. Os protocolos são baseados em conceitos oriundos da biologia, mais precisamente da área de relações ecológicas. Este trabalho, terá foco no protocolo Predador, que é um protocolo onde os agentes têm como principal objetivo a preservação do conhecimento e dados cruciais, como em uma missão, o *hardware* onde está situado o agente líder é atingido e seus conhecimentos são chave para o sucesso da missão, então o mesmo se auto transfere para outro *hardware*,

assim preservando a integridade de seus conhecimentos. A proposta de implementação será feita estendendo as características de um agente padrão do *framework* Jason para criar um agente móvel, e para fazer a conexão entre os diferentes SMAs, será utilizado o *middleware* ContextNet. Testes preliminares foram realizados visando fazer o estudo de viabilidade do trabalho, por exemplo, enviando o código fonte de um agente cognitivo em Jason utilizando o ContextNet.

Palavras-chave

Agentes móveis; SMA aberto; Ambiente aberto

1. INTRODUÇÃO

Sistemas multi-agentes (SMA) são sistemas compostos de agentes cognitivos capazes de raciocinar, tomar decisões e interagir com outros agentes [11]. Além disso, os SMA podem ser abertos ou fechados: em um SMA aberto [4] os agentes podem mover-se para outros sistemas e interagir com outros agentes, os agentes com essa capacidade de transcender seu SMA são denominados agentes móveis. Já nos fechados, os agentes podem somente interagir com os agentes do mesmo sistema, ficando fixo dentro do seu SMA.

Para o desenvolvimento de SMA, existe o Jason [2], que é um *framework* com um interpretador para a linguagem AgentSpeak em Java, que implementa o modelo Belief-Desire-Intention (BDI) [3]. Além disso, existe uma extensão de agentes para o Jason, chamada de ARGO [8], que é uma arquitetura customizada que permite programar agentes cognitivos capazes de controlar e interagir com microcontroladores. Estes agentes podem ser programados como agentes robóticos caso utilizem *hardware*, contudo, caso este *hardware* seja danificado, os agentes do SMA embarcado ficam impossibilitados de deixar o SMA e migrar para um outro sistema pois são agentes fixos. Muitas das vezes, o conhecimento desses agentes podem ser essenciais a uma determinada missão e, neste caso, podem ser perdido juntamente com a deterioração do *hardware*.

A Internet of Things (IoT) ou Internet das Coisas [1] é um conceito para interligar dispositivos eletrônicos utilizados no dia-a-dia, afim de realizar o controle pervasivo de ambientes automatizados. Dado estas características, este conceito pode ser aplicado, por exemplo, na criação de uma rede de SMA abertos. O ContextNet é um *middleware* para IoT que tem o objetivo de proporcionar a comunicação entre dispositivos ou nós móveis visando atender atividades de contexto suportando milhares de conexões simultâneas [6]. Com isso, determinada classe de agentes cognitivos podem utilizar des-

tas características para se moverem para outros SMA, como uma espécie de agentes migradores, tornando possível não só a interação entre agentes de diferentes SMA, mas como também, a movimentação de agentes entre diferentes SMA.

O objetivo deste trabalho é propor mecanismos para movimentação de agentes cognitivos entre SMA distintos programados em Jason, levando em consideração os tipos de relação de um agente móvel com o novo SMA para onde este migrará. Os tipos de relação serão inspiradas nas relações ecológicas da biologia [10]. Para isso, foram propostos protocolos visando possibilitar o transporte de agentes capazes de interagir com os demais agentes do sistema. Essa interação pode ocorrer de diversas formas como: o agente móvel migrar com o objetivo de adquirir ou transmitir conhecimento (Mutualismo); migrar para fazer parte do sistema (Inquilinismo); migrar para dominar (Predatismo).

O trabalho está estruturado da seguinte maneira: na seção 2, os trabalhos relacionados serão discutidos; na seção 3, o referencial teórico será apresentado; na seção 4, o protocolo de transferência de agentes será desmistificado; na seção 5, a conclusão e trabalhos futuros serão apresentados; e por fim na seção 6, as referências bibliográficas serão apresentadas.

2. TRABALHOS RELACIONADOS

Nesta seção serão apresentados as limitações na literatura de agentes móveis e dois trabalhos com foco em SMA aberto, agentes móveis e relação entre agentes. Os trabalhos encontrados na literatura de agentes móveis são puramente teóricos, ou seja, sem implementação, e em sua maioria, com foco no comportamento de um SMA ao receber um agente externo, visando a auto preservação e segurança do sistema.

A arquitetura abstrata chamada THOMAS [7] provê a solução de problemas e limitações na programação de agentes, com a possibilidade programar SMA abertos. Porém um agente não se move de um SMA para outro, somente se relaciona com agentes de outros SMA em um ambiente aberto e no tratamento de negociação entre agentes cognitivos em um ambiente aberto [9], não é levado em consideração a origem dos agentes que entram neste ambiente, e situações como um agente se movendo para outro SMA não são exploradas. Todavia, o protocolo de transferência de agentes proposto visa a aplicação em ambientes reais ou simulados com questões de relacionamento entre agentes de diferentes SMA discutidas e analisadas por meio da proposta dos protocolos.

3. REFERENCIAL TEÓRICO

Nesta seção, serão apresentados os principais conceitos necessários para um melhor entendimento da proposta deste trabalho, como SMA aberto, ambiente aberto, agentes móveis e o *middleware* ContextNet.

3.1 SMA Aberto e Agentes Móveis

No conceito de SMA aberto [4], um agente é capaz de interagir com entidades externas ao sistema, que se manifestam na forma de agentes móveis, de mensagens, entre outras. Para um SMA ser dito aberto, o mesmo não necessita ter agentes móveis ou agentes comunicadores, necessita somente de um mecanismo destinado a interação com entidades externas ao sistema.

Já os agentes móveis são capazes de transcender seu SMA, podendo mover-se para outro SMA. Eles também são ca-

pazes de interagir com agentes de outros SMA e também transferir-se para um ambiente aberto, onde agentes de diferentes SMA podem interagir e trocar informações. Estes agentes têm a liberdade de escolher ficar em qualquer SMA (desde que o SMA permita), sendo assim, o mesmo pode acumular conhecimentos e aprender mais se comparado a um agente que não tem a habilidade de se mover. Além disso, o agente móvel pode ter uma expectativa de vida útil maior, pois em caso de algum perigo, este agente é capaz de mover-se para outro SMA, assim, preservando da sua integridade cognitiva.

3.2 ContextNet

O ContextNet é um *middleware* que visa aplicações colaborativas abrangentes de pequena e grande escala, como monitoramento *on-line* ou coordenação de atividades de entidades móveis e compartilhamento de informações através de redes sociais [6]. Essas entidades podem ser usuários de dispositivos portáteis, como *smartphones*, veículos ou robôs móveis autônomos. No ContextNet, as capacidades de comunicação e de distribuição de contexto são implementadas no *Scalable Data Distribution Layer* (SDDL), enquanto que outros serviços e extensões são construídos como módulos de *software* no topo desta camada de distribuição. Juntos, esses módulos de *software* formam o *middleware* ContextNet.

Além disso, pode-se observar o desempenho do *middleware* ContextNet trabalhando no controle do fluxo de milhares de nós conectados ao mesmo tempo, em uma situação de rastreamento em tempo real de milhares de nós móveis para obter o posicionamento, por exemplo, de uma carga sendo transportada [5]. Características como o processamento em tempo real e suporte a milhares de conexões simultâneas foram pontos que justificam a escolha do *middleware* ContextNet como ferramenta de comunicação entre os SMA nesta proposta de trabalho.

4. PROTOCOLO DE TRANSFERÊNCIA DE AGENTES

Nesta seção será explicado a metodologia dos protocolos de transferência de agentes, bem como as possíveis relações do agente móvel com um SMA e os testes realizados até o presente momento do trabalho como forma de avaliar o estudo de viabilidade do projeto.

4.1 Metodologia

A metodologia para a criação dos protocolos para a transferência de agentes foi baseada na possibilidade de criação e programação de agentes capazes de se moverem entre SMA distintos, estabelecendo a relação que este agente terá com o novo SMA. Ao se programar um agente móvel em um SMA, este deve ser capaz de permitir e possibilitar a movimentação desse agente para outro SMA utilizando um canal de troca de informações para poder se transportar. Para este canal será utilizado o ContextNet. A transferência de um agente móvel para outros SMA pode ser vista na Figura 1.

A transferência de agentes pode ser dividido em três etapas. Na primeira etapa acontece a clonagem do agente e relação adotada, o protocolo acessa todas as informações do agente, como a base de crenças, desejos e intenções, assim como, a biblioteca de planos do agente para a clonagem. Na segunda etapa ocorre o envio do clone, o protocolo envia o clone do agente para outro SMA e aguarda confirmação de

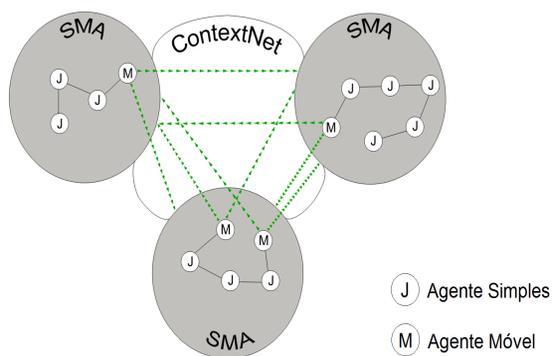


Figure 1: Transferência de agentes móveis.

chegada e se o SMA de destino é capaz de recebê-lo, pois o agente pode ter conhecimentos complexos demais para o SMA. Após a confirmação, inicia-se a etapa 3 onde há a destruição do agente de origem, o SMA de origem exclui o agente e todas as informações como forma de segurança.

Os agentes móveis poderão ser programados para utilizar diversos protocolos, que definem o propósito do agente móvel ao mover-se para outro SMA. Estes agentes têm como principal objetivo a auto preservação, como por exemplo, um agente situado em um *hardware* condenado e para preservar seus conhecimentos este realiza uma auto transmissão para outro SMA. Com essa transferência, o agente pode interagir de diversas formas com o sistema e os agentes que estiverem presentes nele, podendo exercer uma relação de mutualismo, de inquilinismo, ou até mesmo de predatismo visando uma invasão e conquista.

Quando um SMA utiliza do protocolo de transferência de agente com o intuito (Predatismo), detalhes como o *hardware* em questão devem ser levados em consideração, por exemplo, o agente conquistador ou predador tem como origem um SMA aplicado em um protótipo de carro e move-se para o protótipo de um avião, o SMA de origem não é capaz de controlar o novo *hardware*, como consequência, não atingirá o propósito de auto preservação. Uma maneira de solucionar essa questão é o SMA conquistador ficar vivendo de maneira colaborativa com o SMA conquistado ou presa (inquilinismo), e se depois de um tempo de convivência o SMA predador aprender a controlar o novo *hardware*, pode ou não remover o SMA conquistado, contudo, para que ocorra a transferência do agente predador, o SMA presa deve permitir a entrada deste agente. De certa forma, o SMA presa, ao permitir, entende que o conhecimento do SMA predador é mais significativo para a organização de SMA em um ambiente inteligente do que o próprio conhecimento.

4.2 Testes

O primeiro teste foi realizado para verificar se era possível, através do *middleware* ContextNet, mover um arquivo na extensão .asl (extensão de arquivo onde é armazenado o código fonte de um agente em Jason). Para isso, foram criados dois programas desenvolvidos em JAVA, um programa com a finalidade de enviar o arquivo e o outro de receber, para interligar os programas, utilizou-se o ContextNet.

Ao enviar o arquivo asl de um agente para outro SMA, significa que estará enviando somente seu código fonte de

criação, com as crenças, desejos e intenções iniciais, portanto todo o conhecimento adquirido durante a execução do agente não será enviado. Além disso, esses conhecimentos podem ser adquiridos por interações com outros agentes e/ou com ambiente e atualização da sua base de crenças. Portanto, é necessário construir um arquivo asl a partir da base de crenças, planos, desejos e intenções atuais do agente para poder transferi-lo, preservando seu conhecimento. Além do mais, ao reproduzir a biblioteca de planos torna-se possível preservar o comportamento e o processo para o raciocínio cognitivo do agente. O próximo teste será transferir um agente cognitivo entre SMAs e *hardwares* distintos, porém será utilizado *hardwares* similares, neste caso, protótipos de carros autônomos com a simples finalidade de detectar e evitar obstáculos. Os veículos são compostos de sensores e atuadores, interligados a um microcontrolador que, por sua vez, pode ser controlado por agentes ARGO situado em um SMA aplicado na Raspberry PI embarcada nos protótipos. O protótipo dos carros pode ser visto na Figura 2.

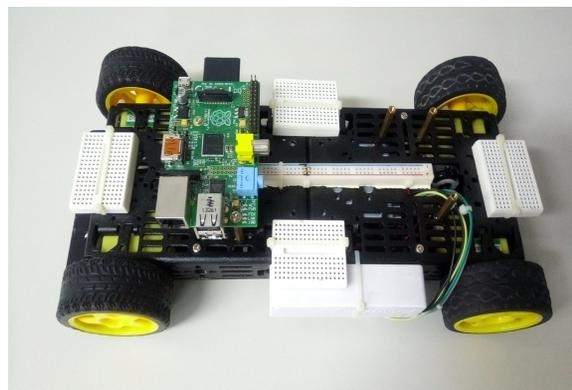


Figure 2: Protótipo dos carros autônomos

O teste relacionado aos protótipos não somente avalia o transporte de um agente móvel entre os diferentes protótipos e sistemas, mas também, o comportamento de um SMA situado em um *hardware* condenado. Através disso, observar o acionamento do protocolo de transferência de agentes com a relação predatismo, onde o agente móvel situado no *hardware* danificado irá montar um código de construção do SMA e de todos os agentes, para enviá-lo a outro SMA com o intuito de dominar o novo *hardware*. Após dominar o novo *hardware*, o agente predador constrói por completo o SMA predador e apaga qualquer rastro de seus conhecimentos no antigo *hardware*. Na Figura 3 pode ser visto o fluxo de transferência do agente móvel entre os protótipos em três etapas, o estado inicial, a transferência e o estado final.

4.3 Resultados e Discussões

Quando o protocolo de transferência de agentes é ativado com a relação predatória, o SMA de origem deve ou não ser excluído do *hardware* de origem? Se sim, antes ou depois de realizar a transferência para o novo *hardware*? Pois pode haver rejeição do SMA de destino ou até mesmo quebra na conexão entre os SMAs. Acreditamos que o SMA de origem deve ser excluído do *hardware* de origem, para preservação da privacidade e segurança, porém somente após realizar a transferência completa para o novo *hardware*.

Outra questão deve ser levada em consideração, quando

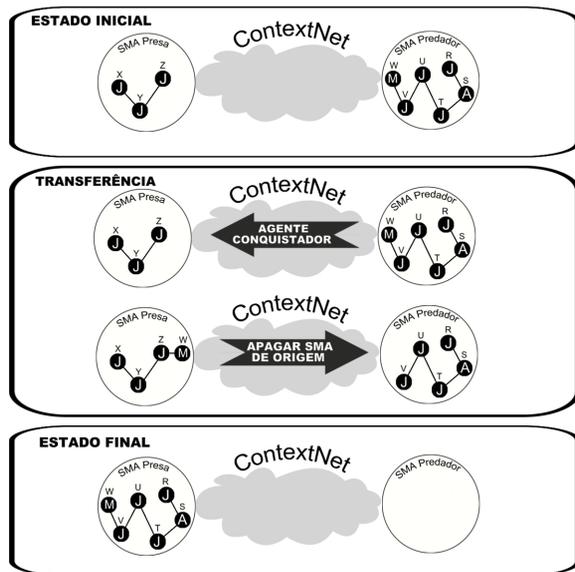


Figure 3: Movimentação de agentes nos protótipos

se trata do protocolo de transferência de agentes utilizando a relação de Predatismo, como, por exemplo, um SMA pode ser composto de diversos agentes e estes podem ser de diferentes tipos (ARGO, padrão, móvel, entre outros), porém somente o móvel pode se transferir para outro SMA, então como preservar a inteligência de todo o SMA de origem? Para responder esta pergunta chegou-se a conclusão de que o agente móvel, quando invadir o SMA a ser conquistado, deve ter em sua arquitetura a capacidade para construir não somente seu próprio estado mental no novo SMA, mas como também, dos outros agentes do SMA conquistador.

Na proposta dos protocolos para adquirir conhecimento, os pontos em relação à confiabilidade e segurança podem ser explorado de uma maneira diferenciada, por exemplo, um agente móvel chegando a um novo sistema e tentar transmitir conhecimento, será que os agentes tradicionais irão acreditar nas informações recebidas? Ou se os agentes tradicionais enviarem informações, o agente móvel deverá confiar e recebê-las? Se sim, será que após adquirir esses novos conhecimentos, o mesmo poderá voltar para o SMA de origem? A fim de sanar estas questões, concluiu-se que, a proposta dos protocolos de transferência de agentes ainda não deve entrar no mérito de métodos de filtragem de informações e gerência de grau de confiabilidade, porém a decisão de permanecer ou voltar para o SMA de origem será uma decisão tomada pelo próprio agente.

5. CONCLUSÃO

A proposta de desenvolvimento foi criar protocolos de transferência de agentes, que por sua vez, podem ser divididos de acordo com a relação que o agente móvel deva ter com o outro SMA. No atual estágio de desenvolvimento, pôde ser observado que o foco está sendo no protocolo de transferência de agentes com a relação de predatismo, visto que é o protocolo mais simples de ser implementado e não é necessário lidar com questões de negociação entre agentes de diferentes SMA. Além disso, tem o ideal de preservar os aspectos cognitivos da inteligência do SMA predador. Os

resultados iniciais são promissores tanto levando em consideração a implementação, mas como também, a interligação dos conceitos derivados da biologia com os conceitos de SMAs abertos, agentes móveis e ambientes abertos.

Para trabalhos futuros deseja-se transformar esse projeto dos protocolos de transferência de agentes em uma extensão para ser incorporado ao *framework* Jason, possibilitando a programação de agentes móveis. Pretende-se também, aplicar em um projeto de uma sala inteligente em escala real, que está sendo desenvolvido pelos autores com diversos SMA distintos e agentes móveis transitando entre eles.

6. REFERÊNCIAS

- [1] J. P. B. Andrade, M. Oliveira, E. J. T. Gonçalves, and M. E. F. Maia. Uma Abordagem com Sistemas Multiagentes para Controle Autônomo de Casas Inteligentes. In *XIII Encontro Nacional de Inteligência Artificial e Computacional (ENIAC)*, 2016.
- [2] R. H. Bordini, J. F. Hübner, and M. Wooldridge. *Programming Multi-Agent Systems in AgentSpeak using Jason*. John Wiley & Sons Ltd, 2007.
- [3] M. E. Bratman. *Intention, Plans and Practical Reasoning*. Cambridge Press, 1987.
- [4] M. S. Chebout, F. Mokhati, M. Badri, and M. C. Babahenini. Towards preventive control for open mas - an aspect-based approach. In *Proceedings of the 13th International Conference on Informatics in Control, Automation and Robotics - Volume 1: ICINCO*, pages 269–274. INSTICC, SciTePress, 2016.
- [5] L. David, R. Vasconcelos, L. Alves, R. André, G. Baptista, and M. Endler. A communication middleware for scalable real-time mobile collaboration. In *Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2012 IEEE 21st International Workshop on*, pages 54–59. IEEE, 2012.
- [6] M. Endler, G. Baptista, L. Silva, R. Vasconcelos, M. Malcher, V. Pantoja, V. Pinheiro, and J. Viterbo. Contextnet: context reasoning and sharing middleware for large-scale pervasive collaboration and social networking. In *Proceedings of the Workshop on Posters and Demos Track*, page 2. ACM, 2011.
- [7] S. Ossowski, V. Julián, J. Bajo, H. Billhardt, V. Botti, and J. Corchado. Open mas for real world applications: An abstract architecture proposal. In *Proc. XII Conference of the Spanish Association for Artificial Intelligence (CAEPIA)*, volume 2, pages 151–160, 2007.
- [8] C. E. Pantoja, M. F. Stabile Jr, N. M. Lazarin, and J. S. Sichman. ARGO: A customized jason architecture for programming embedded robotic agents. *Fourth International Workshop on Engineering Multi-Agent Systems (EMAS 2016)*, 2016.
- [9] V. Sanchez-Anguix, R. Aydogan, V. Julian, and C. M. Jonker. Analysis of intra-team strategies for teams negotiating against competitor, matchers, and conceders. In *The 5th International Workshop on Agent-based Complex Automated Negotiations (ACAN 2012)*, pages 1–8, 2012.
- [10] M. Tissot-Squalli. *Interações ecológicas & biodiversidade*. Unijui, 2009.
- [11] M. Wooldridge. *An Introduction to Multi-Agent Systems*. Wiley, 2009.