

Protocolo de Interação Entre SMA Embarcados Bio-Inspirado na Relação de Predatismo

Vinicius Souza de Jesus¹, Fabian Cesar P. B. Manoel¹, Carlos Eduardo Pantoja^{1,2}

¹Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (CEFET/RJ)
Av. Maracanã, 229 - Maracanã - 20271-110 - Rio de Janeiro - RJ - Brazil

²Universidade Federal Fluminense (UFF)
Niterói - Rio de Janeiro - RJ - Brazil.

{souza.vdj, fabiancpbm}@gmail.com, pantoja@cefet-rj.br

Abstract. Approaches inspired by biological concepts are commonly found in the area of Artificial Intelligence and help to solve some problems related to the interaction between entities in societies and groups. Thus, the objective of this work is to present a protocol of interaction between open MAS and embedded agents in robotic platforms based on a model adapted from the ecological relations of predation. This protocol aims to preserve the integrity of its knowledge when the hardware of a platform is damaged, by transferring all the agents and knowledge obtained to a known MAS and embarked on another similar platform. The proposed protocol will be implemented using the Jason framework and a proof of concept in a real environment will be presented with two prototypes of land vehicles assuming each the role of predator and prey in the relationship.

Resumo. Abordagens inspiradas em conceitos biológicos são comumente encontradas na área da Inteligência Artificial e contribuem para auxiliar na resolução de alguns problemas relacionados com a interação entre entidades em sociedades e grupos. Sendo assim, o objetivo deste trabalho é apresentar um protocolo de interação entre agentes integrantes de SMA abertos e embarcados em plataformas robóticas baseado em um modelo adaptado da relação ecológica de predatismo. Este protocolo tem como objetivo preservar a integridade de seus conhecimentos quando o hardware de uma plataforma estiver danificado, através da transferência de todos os agentes e conhecimentos obtidos para um SMA conhecido e embarcado em outra plataforma similar. O protocolo proposto será implementado utilizando o framework Jason e uma prova de conceito em um ambiente real será apresentada com dois protótipos de veículos terrestres assumindo cada um o papel de predador e presa na relação.

1. Introdução

Na biologia, o conceito de relações ecológicas [Begon et al. 2005] classifica as interações que ocorrem entre os seres vivos de acordo com a posição que cada ser vivo assume na relação. A relação ecológica de predatismo, por exemplo, é estabelecida quando um ser vivo por meio de seus instintos de sobrevivência domina outro ser vivo para se manter vivo. Similarmente, agentes inteligentes são entidades autônomas capazes de interagir com outros agentes em um Sistema Multi-Agente (SMA) [Huynh et al. 2006]. Nos SMA,

a composição dos agentes integrantes do sistema, a forma como eles interagem, e a possibilidade deles poderem migrar de sistema permitem classificar estes SMA como abertos ou fechados.

Um SMA fechado é caracterizado por seus agentes serem restritos a um sistema de origem e não poderem se locomover para outros sistemas e, normalmente, estes agentes somente interagem com agentes dentro de seu sistema ou esfera de influência [Wooldridge 2009]. Já um SMA aberto [Huynh et al. 2006] é caracterizado pela capacidade dos agentes poderem migrar de um sistema para outro, permitindo que a forma de interação exceda a esfera de influência de origem de um agente. A capacidade de um agente se mover de um sistema a outro depende da existência de agentes móveis por meio de um ambiente aberto. Um agente móvel [Chen et al. 2009] é um agente que possui a capacidade de se transferir de um SMA para outro e interagir com outros agentes em um ambiente compartilhando artefatos e recursos comum a eles.

Diversas técnicas e soluções bio-inspiradas utilizam a abordagem de agentes para a resolução de problemas relacionados com a interação entre entidades em sociedades e grupos [Zeghida et al. 2018]. Existem trabalhos que são inspirados pelos conceitos de biologia para elaborar técnicas e algoritmos que buscam aumentar o nível de interação e comunicação entre sistemas que estão em um mesmo ambiente [Günay et al. 2015]. Por exemplo, [Ferri et al. 2006] apresenta um algoritmo bio-inspirado para SMA para identificar concentrações de gás em um ambiente; [Zedadra et al. 2016] propõe uma situação de busca e exploração de recursos naturais em que um SMA aberto precisa trabalhar em conjunto com outro e executar sub-tarefas simples para concluir uma tarefa complexa comum a todos os SMA. Embora estes trabalhos considerem SMA abertos e aplicações bio-inspiradas, eles interagem somente com um ambiente aberto onde agentes de diferentes SMA podem se conectar e trocar conhecimentos. Sendo assim, estes trabalhos não levam em consideração o fato de um agente móvel poder se transferir de um SMA para outro e, com isso, não focam na interação entre agentes de diferentes SMA e caso ocorra algum dano na parte física onde está aplicado o SMA destes trabalhos, seus agentes não são capazes de se transferir para outro SMA e preservar a integridade de seus conhecimentos.

Portanto, o objetivo deste trabalho é apresentar um protocolo de transferência de agentes baseado em um modelo adaptado da relação ecológica de predatismo, onde será possível uma plataforma robótica controlada por um SMA embarcado assumir o controle de outra plataforma através da movimentação de todos os agentes de um SMA para o SMA da plataforma de destino. O protocolo de transferência de agentes proposto tem como objetivo preservar a integridade dos conhecimentos dos agentes de um SMA embarcado em situações onde o hardware de sua plataforma robótica estiver comprometida. Sendo assim, o SMA ativa o protocolo para transferir todos os agentes e seus respectivos conhecimentos para um outro SMA embarcado em uma plataforma robótica similar, assumindo assim, o controle desta plataforma. O protocolo é considerado um predatismo adaptado uma vez que o SMA de origem não realiza uma predação escolhendo uma presa aleatória, pois necessita conhecer e ter a permissão do SMA de destino para poder assumir o controle da plataforma.

A implementação dos SMA, o protocolo utilizará o *framework* Jason [Bordini et al. 2007], que possui um interpretador de uma linguagem orientada

a agentes *AgentSpeak* [Rao 1996] e implementa o modelo *Belief-Desire-Intention* (BDI) [Bratman 1987] para programação de agentes cognitivos. O Jason foi escolhido por ser um *framework* de código aberto, com uma interface facilitadora para a programação de SMA. Para permitir SMA abertos e a comunicação entre agentes de SMA distintos, o Jason foi estendido para permitir a integração de um *middleware* [Endler et al. 2011] para a *IoT* e serviços de contexto, que visa aplicações colaborativas, coordenação de atividades entre entidades móveis e compartilhamento de informações. Este *middleware* permite trabalhar com a conexão de milhares de nós móveis conectados ao mesmo tempo [David et al. 2012] e garante escalabilidade.

Como prova de conceito, foram desenvolvidos dois protótipos de veículos terrestres com SMA embarcados para promover testes em ambientes reais. Um dos protótipos é considerado o líder e contém planos, crenças e intenções importantes para uma hipotética missão coletiva. Contudo, é simulado um dano induzido a esta plataforma considerado irreparável e o protocolo de predatismo é ativado. O SMA como um todo migra para uma plataforma de hardware similar previamente conhecida, tomando o seu controle. Testes foram realizados para poder medir a velocidade de acionamento do protocolo de transferência de agentes e verificar se houve a preservação de integridade dos conhecimentos dos agentes durante a transmissão assim como se houve a devida tomada de controle.

O trabalho está estruturado da seguinte maneira: na seção 2, o referencial teórico é apresentado; na seção 3, os trabalhos relacionados são discutidos; na seção 4, a metodologia utilizada é explorada e apresentada; na seção 5, a avaliação experimental é discutida; na seção 6 é levantada as discussões do trabalho; na seção 7, as considerações finais são mostradas e; por fim, as referências são apresentadas.

2. Referencial teórico

Esta seção tem o objetivo de apresentar e descrever os principais conceitos utilizados no trabalho para prover uma melhor compreensão do protocolo de transferência de agentes proposto.

O predatismo é uma relação ecológica da biologia em que um ser vivo, denominado predador, caça outro ser vivo, denominado presa. Esta ação é motivada pelo instinto de sobrevivência da espécie predadora, que captura e mata a espécie de nível inferior da cadeia alimentar, e com isso, se fortalece. No contexto geral, o predatismo é uma relação necessária para manter o equilíbrio do ambiente ecológico. Na IA, SMA agrupam entidades chamadas de agentes para realizarem atividades coletivamente e atingirem seus objetivos. Alguns destes sistemas podem se manter fechados; já em outros cenários, pode haver mais de um SMA interagindo entre si para melhorar a capacidade de resolver problemas.

Os SMA fechados [Wooldridge 2009] são SMA onde seus agentes ficam restritos a se relacionar somente com os agentes de seu SMA para realizar um objetivo comum. Ou seja, em um SMA fechado, os agentes possuem restrições para adquirir novos conhecimentos já que são capazes de interagir somente com os agentes do próprio SMA. Sendo assim, quando são implementados em uma plataforma robótica, estes agentes ficam restritos a esta plataforma. Além disso, caso a plataforma robótica sofra danos físicos, o SMA fechado aplicado ficará impossibilitado de preservar os conhecimentos de seus agentes, que serão perdidos. Já os SMA abertos [Huynh et al. 2006] são SMA que possuem a

capacidade de interagir com agentes de outros sistemas por meio de agentes móveis ou através de um ambiente aberto onde agentes de diferentes sistemas podem interagir e trocar conhecimentos. Com isso, um SMA aberto possui a capacidade de transferir seus agentes para outros SMA ou para um ambiente aberto. Quando aplicado em uma plataforma robótica, este SMA pode preservar os conhecimentos de seus agentes mesmo que esta plataforma sofra danos físicos, pois a transferência dos seus agentes pode acontecer a qualquer momento para um ambiente aberto ou para outro SMA.

Para o desenvolvimento de SMA, este trabalho está utilizando o *framework* Jason [Bordini et al. 2007], que além de ser interpretado pela linguagem *AgentSpeak* [Rao 1996], possui implementação do modelo BDI [Bratman 1987] para agentes cognitivos e uma interface facilitadora para a programação de SMA. Além disso, o *framework* Jason possui uma arquitetura customizada de agentes chamada ARGO [Pantoja et al. 2016] que permite programar agentes cognitivos com a capacidade de se comunicar e controlar plataformas robóticas. Com a utilização do *framework* Jason em conjunto com agentes ARGO, é possível realizar somente aplicações com SMA fechados e, portanto, caso a plataforma robótica fosse danificada, os conhecimentos dos agentes seriam perdidos. Com isso, em [Pantoja et al. 2018] foi proposta uma arquitetura customizada de agentes com a capacidade de se comunicar com agentes de outros SMA chamada de *Communicator*, que visa permitir a programação de SMA abertos utilizando o *framework* Jason e os agentes ARGO.

Para os agentes *Communicator* se comunicarem com outros SMA, estes SMA abertos utilizam o *middleware* ContextNet criado para IoT. O ContextNet é capaz de oferecer um ambiente para aplicações colaborativas entre dispositivos móveis, permitindo que eles compartilhem informação ao estar conectados em uma rede. Por ser um *middleware* feito para abordagens em IoT, o ContextNet permite a conexão de múltiplos nós móveis ao mesmo tempo em uma rede de forma estável [David et al. 2012]. No entanto, a utilização dos agentes *Communicator* nesta conexão possibilita a troca de conhecimento dos agentes de um SMA apenas entre agentes de outros SMA. Mesmo assim, caso a plataforma robótica de um SMA sofresse algum tipo de dano físico, os agentes deste sistema ainda não poderiam se transferir para outro SMA para preservar a integridade de seus conhecimentos.

Com a capacidade do *middleware* ContextNet de prover a troca de informações de forma estável, foi possível também obter um canal de movimentação de agentes. Com isso, tornou-se possível a criação de SMA abertos e, conseqüentemente, a proposta do protocolo de transferência de agentes. Este protocolo consiste em desenvolver SMA abertos com a capacidade não só de se comunicar com agentes de diferentes SMA, mas também de transferir agentes entre eles, formando assim uma rede de movimentação de agentes e seus conhecimentos fornecida pelo *middleware* ContextNet.

3. O Protocolo de Predatismo

Esta seção tem como objetivo apresentar o protocolo inspirado nos conceitos de predatismo da biologia, mostrando sua ideia, onde e porque é indicado, como funciona e quais são os detalhes da implementação que o compõe. Além disso, será apresentada a prova do conceito utilizando prototipagem em um ambiente real e serão discutidos os pontos fortes e fracos deste trabalho.

Assim como as relações ecológicas são necessárias para garantir o equilíbrio e a evolução dos seres vivos do ambiente biológico, os SMA abertos também podem se relacionar entre si para garantir a integridade de um sistema. Um ponto ainda não muito explorado neste cenário para garantir tal integridade é a possibilidade de preservar os conhecimentos de um SMA importante que está prestes a ser destruído. Sendo assim, este trabalho se inspira na relação ecológica de predatismo para criar um protocolo que visa explorar a interação entre dois SMA com o intuito de garantir a integridade do sistema.

O protocolo de predatismo envolve dois SMA, onde um deles é o predador e o outro é a presa. O SMA predador é aquele que possui conhecimentos valiosos para atingir os objetivos do grupo, porém, controla um hardware que passou a apresentar inconsistências. Ao ser identificado o problema físico, este SMA realiza a transferência de todos os agentes para outro destino fisicamente semelhante e mais seguro. Já o SMA presa é aquele que ainda não possui conhecimentos significativos sobre as situações que o cerca no ambiente real, mas por controlar um hardware consistente e semelhante ao hardware do predador, ele serve como o destino dos agentes que foram transferidos pelo SMA predador. Após o processo de predatismo, o SMA predador passará a controlar o hardware que pertencia ao SMA da presa e este deixará de existir.

Para que o processo de transferência seja iniciado, é necessário que haja um agente responsável por realizar percepções sobre as condições do ambiente e identificar se as anomalias causam danos ao hardware. Em caso positivo, o agente responsável pela comunicação entre SMA deverá avaliar, de acordo com suas crenças sobre a relação de seu SMA com os demais da rede, se o plano de transferência pode ser iniciado. No momento em que a transferência começa, é feita a clonagem de todos os agentes do sistema e suas características, como crenças iniciais e adquiridas, planos e os objetivos. Após isso, o SMA de origem localiza o SMA de destino, e este recebe os clones criados na origem. Quando os agentes clonados chegarem no SMA de destino, haverá uma tentativa de iniciá-los e depois uma mensagem será enviada para o SMA de origem informando sobre a recepção destes agentes. Caso os agentes clonados do SMA de origem tenham sido inicializados no SMA de destino, os agentes originais do SMA de origem e aqueles que já pertenciam ao SMA de destino antes da interação serão finalizados e apagados. Ao apagar os agentes originais do SMA de origem, garante-se que de fato ocorreu uma transferência de agentes, e não apenas uma cópia; além disso, ao apagar os agentes originais do SMA de destino, garante-se que não haverá interferência nos planos do predador, garantindo que este predador de fato possua controle sobre o hardware. Por outro lado, os agentes clonados serão deletados do SMA de destino caso não tenham sido inicializados corretamente. Essa sequência de operações da interação, que está representada na Figura 1 através do diagrama de sequência da AUML, pode ocorrer quantas vezes forem necessárias enquanto o problema de *hardware* não interferir nas atividades do SMA.

Este protocolo foi feito para ser aplicado nos problemas que demandam o uso de mais de um SMA trabalhando em conjunto onde o foco da aplicação preza o conhecimento, e com isso, as interações entre SMA colaboram para que os objetivos da implementação não sejam comprometidos. Além disso, o fato do protocolo predador permitir que um SMA se transfira para outro destino cria uma oportunidade para este SMA de obter conhecimentos mais variados, dependendo do ambiente externo que o SMA da presa se encontrava.

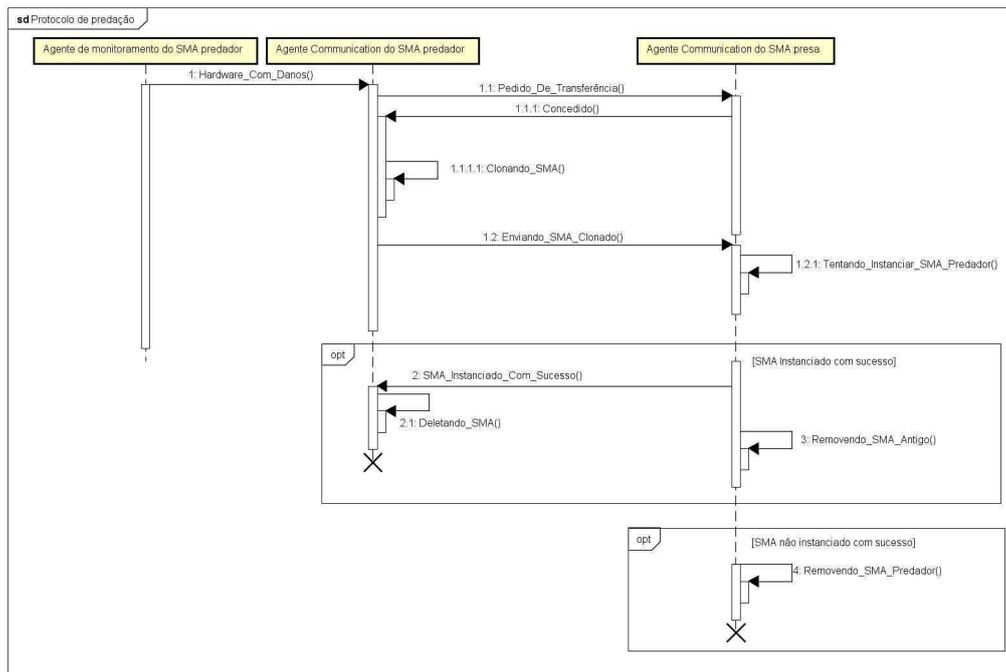


Figura 1. Diagrama de sequência da AUML do processo de interação do protocolo de predatismo entre o predador e a presa.

3.1. A implementação do Protocolo de Predatismo

O protocolo de transferência proposto está embasado na relação de predatismo que ocorre em um determinado ecossistema. Para fornecer este ecossistema, é preciso criar uma infraestrutura que permita que agentes de SMA abertos se comuniquem entre si. Para implementar essa infraestrutura será utilizado a *middleware* ContextNet, que oferece conexão a nível de servidor utilizando uma *middleware* para camada de distribuição de dados escalável (SDDL) [David et al. 2012] e que utiliza o padrão de assinante para para endereçar aplicativos em tempo real e sistemas embarcados estendendo o serviço de distribuição de dados (DDS) padrão do OMG [Pardo-Castellote 2003]. Com isso, o ContextNet permite conectar múltiplos dispositivos tratando os problemas de reconexão e escalabilidade.

Para implementar um SMA que seja capaz de trocar informações nessa infraestrutura de comunicação, é necessário desenvolver um tipo de agente customizado para ter condições de executar esta tarefa. Para criar SMA com essa característica, será utilizado o *framework* Jason que, além de ser uma plataforma que já é bem explorada, também oferece arquiteturas customizáveis para exploração.

O agente responsável por realizar trocas de mensagem entre os SMA através da arquitetura de rede fornecida pelo ContextNet é o agente *Communicator*. Ao entrar nessa rede, o agente *Communicator* assume o papel de cliente, além de possuir uma identificação única que o distingue de outras entidades dentro da mesma rede. Através deste tipo de agente, será possível customizar um padrão de troca de mensagens para que possuam como conteúdo as informações de todos os agentes de um SMA e assim, possibilitar a transferência de agentes de um SMA para outro. Ao embutir o protocolo de transferência na implementação do agente *Communicator*, o SMA ganha autonomia para fazer transferência entre SMA sempre que um plano é executado.

Para adaptar o protocolo de predatismo no agente *Communicator* é necessário alterar o seu ciclo de raciocínio. Inicialmente, o ciclo de raciocínio do agente tradicional era capaz de receber percepções e mensagens vindas de outros agentes do SMA, que eram passadas para a base de crenças do agente e eram mapeados como um evento externo. Cada evento adicionado neste mapa dispara um ação de verificação dos planos da biblioteca de planos. Caso haja um plano a ser executado após a chegada do novo evento externo, ele será designado como uma intenção e com isso, o agente executa o plano, que pode desencadear em uma ação ou uma mensagem a outro agente do SMA. O agente *Communicator* foi resultado da customização deste ciclo de raciocínio, que passou a poder receber e enviar mensagens vindas de outro agente aberto que esteja dentro da rede ContextNet. Para implementar o protocolo de predatismo, o agente *Communicator* foi customizado para realizar três operações que irão variar de acordo com o papel do agente *Communicator* na relação e de acordo com o momento em que cada ação precisa ser tomada. A primeira operação é realizada no agente *Communicator* do SMA predador, que, após fazer a solicitação para efetuar a transferência e ter essa permissão concedida, realiza a clonagem de todo o SMA e este é enviado para o SMA presa. A segunda operação é realizada no agente *Communicator* do SMA presa, que é a substituição de todo SMA da presa pelo SMA do predador. A terceira operação ocorre no agente *Communicator* do SMA predador, que é a remoção do SMA após ter recebido a mensagem que indique que o SMA já foi criado com sucesso no destino.

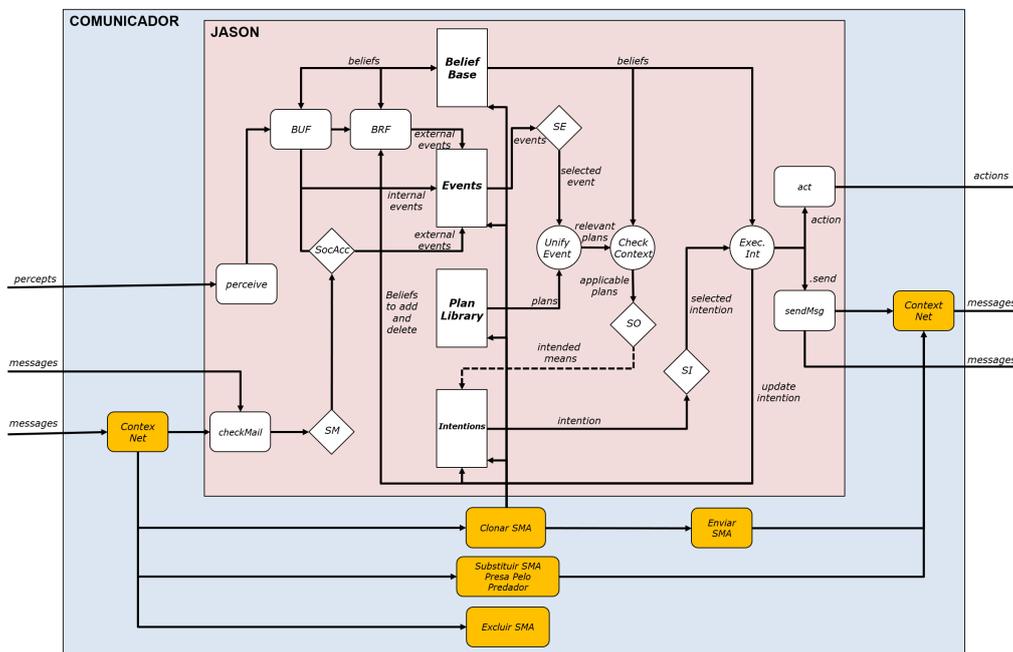


Figura 2. Ciclo de raciocínio do agente *Communicator* com o protocolo de predatismo.

3.2. Prova de Conceito

Como prova de conceito, foram criados dois protótipos de veículos terrestres utilizando plataformas robóticas e cada um deles tem embarcado um SMA programado utilizando o protocolo proposto nesse trabalho. Esses SMA são programados de forma diferente —

para que um contenha mais informações do que o outro — a fim de que seja feita a devida transferência de conhecimento no momento da ativação do protocolo. Os veículos possuem hardware idênticos: são compostos por três sensores: de temperatura, luminosidade e um de distância; dois atuadores: os motores das duas rodas traseiras; um controlador *Arduino*; e uma *Raspberry* onde fica situado o SMA embarcado. Na Figura 3 a seguir é possível observar os protótipos dos veículos terrestres.

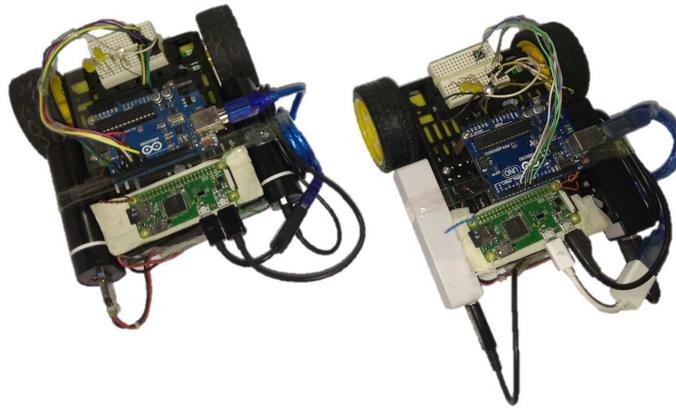


Figura 3. Protótipo dos Veículos Terrestres.

Os sensores dos protótipos denotam a forma de percepção do ambiente para o SMA. Através destas percepções, os agentes do SMA são capazes de identificar riscos de danos físicos para a plataforma robótica. Para isso, a temperatura ambiente de um dos protótipos foi elevada para que o SMA interprete que a plataforma robótica está tendo superaquecimento e ativar o protocolo de transferência de agentes. Na figura 4 é apresentado o código fonte do agente remetente que é responsável por ativar o protocolo de transferência de agentes, porém, para a ativação, este agente necessita receber a crença de que o veículo está superaquecendo do agente que monitora o ambiente.

```
!start.
+!start : true <-
    .print("Sou o remetente e vou enviar uma mensagem ao destinatário.");
    .sendOut("788b2b22-baa6-4c61-b1bb-01cfff1f5f878", tell, oi).
+!ola : true <-
    .print("Sou o remetente e recebi uma resposta do destinatário").
+superAquecimento <-
    .moveOut("788b2b22-baa6-4c61-b1bb-01cfff1f5f878", predator).
```

Figura 4. Código fonte do agente remetente.

Esse teste tem o objetivo de verificar o grau de confiabilidade do processo de transferência envolvido na ativação do protocolo e na verificação da integridade dos conhecimentos dos agentes após o fim do processo. A aferição da velocidade com que a transferência dos agentes também é realizada. Para isso, os testes foram realizados alterando a quantidade de agentes do SMA predador. Com isso, o SMA predador que está embarcado na plataforma robótica chegou a possuir 10, 30, 50 e 100 agentes.

Com isso, foram realizados 10 repetições para cada uma das variações de quantidade de agentes — somando no total 40 testes — e concluiu-se que o protocolo foi ativado

e finalizado corretamente em 100% dos casos, garantindo a preservação e a integridade dos conhecimentos do SMA predador. Essa verificação foi constatada pois o veículo de origem era capaz de perceber o ambiente de forma simultânea enquanto que o veículo que foi predado não possui essa habilidade. Ao fim do processo, o hardware do veículo de origem ficou inativo e o veículo predado começou realizar as funções que inicialmente não faziam parte de sua programação.

3.3. Discussões

Os benefícios do uso dos protocolos de predatismo giram em torno da preservação do conhecimento dos agentes, da possibilidade de transportar agentes para outros SMA e possibilitar que os agentes adquiram e transmitam novos conhecimentos por meio de interações com agentes de outros SMA. Além disso, a implementação do protocolo de predatismo é bastante abrangente, levando em consideração aplicações de SMA em ambientes físicos ou simulados, lidando com questões de segurança do SMA e agentes maliciosos através da implementação de conceitos de relações ecológicas.

Levando em consideração a segurança do SMA, os protocolos de predatismo lida com questões de segurança através da aplicação de conceitos de relações ecológicas em todos os casos de transporte de agentes. Além disso, para que a transferência de agentes seja concluída, o SMA de destino deve garantir que a transferência seja realizada somente com o seu consentimento. Já levando em conta agentes maliciosos, as implementações de segurança precisam ser aprimoradas com um critério mais rigoroso de aceitação da transferência, uma vez que o predatismo substitui completamente o SMA presa pelo SMA predador e a falha nessa segurança pode por em risco todo o sistema onde o SMA interage.

Outra questão relacionada à implementação do protocolo de predatismo gira em torno do agente *Communicator*. A responsabilidade deste agente de transferir SMA poderia ter sido atribuída a outra entidade não agentificada. Porém, a vantagem de utilizar o agente *Communicator* para este fim está na autonomia que o SMA ganha para executar este protocolo. Por outro, essa autonomia permitirá ao agente, por exemplo, decidir por deixar de se relacionar com o SMA danificado para reiniciar uma outra relação com um sistema íntegro. Neste caso, a implementação do protocolo não deve permitir que este evento interrompa uma relação caso ela já tenha sido iniciada; caso contrário, as intenções do agente *Communicator* devem ser executadas normalmente.

Ainda sobre agentes *Communicator*, quando o SMA predador domina a SMA presa, o agente *Communicator* do SMA predador é clonado e transferido junto com os demais agentes e em seguida, o SMA de origem é apagado e o antigo SMA da presa também é. No entanto, caso seja necessário reativar o protocolo de predatismo, o agente *Communicator* tentaria novamente dominar a presa que foi inicialmente definida para este agente *Communicator*, porém este agente já não existe mais e um erro ocorreria. Portanto, antes que o *Communicator* do SMA predador seja instanciado e o agente comunicador da SMA presa seja excluído, o agente *Communicator* do SMA predador deve remover a referência que identifica o SMA presa que deixará de existir nas redes de SMA e adicionar outra referência com a ajuda do próprio agente *Communicator* do SMA presa.

Considerando os agentes ARGO no protocolo de predatismo, há a necessidade de atualizar a interface de comunicação entre o agente ARGO e o controlador do *hardware* onde o SMA presa está embarcado, porque os agentes ARGO precisam conhecer o iden-

tificador da porta de comunicação à qual o controlador está conectado. Para resolver esse problema, antes que os agentes ARGO do SMA predador sejam inicializados, é necessário realizar a atualização dos seus identificadores de porta com a ajuda dos agentes ARGO do SMA de presa antes que eles sejam excluídos.

4. Trabalhos Relacionados

Alguns trabalhos, como [Ferri et al. 2006], já utilizam as técnicas bio-inspiradas em Sistemas Multi-Agentes para tarefas de localização em mundo real, nas quais é necessário acessar o *hardware* implementado. Além disso, novas técnicas e metodologias surgiram, tais como [Paes et al. 2005, Zeghida et al. 2018], e, tendo elas ou não inspiração em biologia, todas visam melhorar a forma de trabalhar com Sistemas Multi-Agente abertos e agentes móveis. Consequentemente, trabalhos como [Zedadra et al. 2016] aparecem como uma tentativa de assegurar que diferentes agentes interajam e cooperem entre si em um dado ambiente simulado.

Em [Ferri et al. 2006], é apresentado um algoritmo inspirado em biologia, para um sistema multi-agente robótico cooperativo no qual existe um computador central que localiza todos os robôs e armazena os dados coletados. Sua aplicação é focada na identificação de fontes de gás em um ambiente real, onde cada robô possui sensores e atuadores que servem como uma interface interativa entre este ambiente e o Sistema Multi-Agente. Além disso, o algoritmo de cooperação do robô é biologicamente inspirado no comportamento *Bombus mori* e é dividido em duas fases: pesquisa individual de gás de alta concentração feita por cada robô de forma independente e uma pesquisa cooperativa que envolve todo o enxame de robôs. No entanto, implementações em ambientes reais estão sujeitas a vários comportamentos que podem ocorrer e que podem colocar em risco a integridade do sistema. Se um robô fosse danificado, a eficiência do sistema ficaria comprometida. Na pior das hipóteses, se o computador central estiver danificado, todo o sistema parará de funcionar.

Em [Zedadra et al. 2016], o assunto discutido é a robótica cooperativa e a exploração de recursos naturais, que é uma tarefa complexa, dividida em pequenas subáreas e caracteriza o ato de busca e captura de alimento em um local de armazenamento particular. Este trabalho apresenta um algoritmo de exploração de recursos naturais bio-inspirado envolvendo agentes simples que são capazes de realizar tarefas complexas em grupo.

Esses trabalhos reforçam a importância de explorar sistemas abertos como uma alternativa para cooperar e interligar diferentes sistemas de um ambiente. A necessidade de adotar os paradigmas biológicos nas aplicações de SMA também pode ser vista em [Zeghida et al. 2018], pois, apesar dos desafios relacionados à sobrevivência dos seres, os organismos biológicos têm capacidade de evolução, autocorreção e controle. No entanto, as inspirações biológicas estudadas no SMA estão limitadas à cooperação e interação entre estes SMA por meio de troca de mensagens. Além disso, de todas as técnicas que foram estudadas, uma maneira de preservar a integridade dos sistemas em situações críticas ainda não foi explorada. Os trabalhos relacionados mostram que as técnicas bio-inspiradas auxiliam na implementação de sistemas abertos para aumentar a eficiência de algoritmos colaborativos. Porém, as técnicas bio-inspiradas também podem auxiliar na formação de um sistema que preserve a existência de um SMA assim como a biologia é capaz de fazer com os seres vivos dentro do ecossistema.

Em [Jesus et al. 2018], foram apresentadas ideias de protocolos inspirados em relações ecológicas da biologia, como o predatismo, mutualismo e inquilinismo. Porém, este trabalho relacionado apresentou brevemente as definições e o funcionamento dos protocolos. Sendo assim, este trabalho focará em explorar as ideias, os detalhes de implementação e as aplicações do protocolo de predatismo. A escolha por este protocolo foi feita devido ao fato de que as implementações que devem ser feitas para o seu funcionamento podem ser reaproveitadas, com algumas adaptações, para os demais protocolos.

5. Considerações Finais

Este trabalho apresentou um protocolo de interação entre SMA embarcados inspirado na relação ecológica de predação da biologia, mostrando sua ideia com o objetivo de preservação de conhecimento e principalmente, sua implementação, onde foi reaproveitada a arquitetura do agente *Communicator* para customizar o seu ciclo de raciocínio e assim, permitir transportar todos os agentes de um SMA predador para a presa. Como prova do conceito, dois protótipos de veículos terrestres foram criados utilizando plataformas robóticas que possuem, cada uma, um SMA embarcado. Na prova do conceito, os dois protótipos possuíam o *hardware* semelhante, porém, um SMA possuía mais conhecimento do que outro. Com este cenário, foi possível mostrar a situação em que o hardware de um veículo era danificado e então, este aplicava o protocolo de predação no SMA que estava embarcado no veículo com o hardware que apresentava boas condições.

Como trabalhos futuros serão tratados os problemas que envolvem a conversão de portas de comunicação com a parte física onde o SMA está embarcado, garantindo que o novo sistema conseguirá se instalar por completo no novo *hardware* de forma automática. Também será implementado o mecanismo que reajusta a lista de SMA disponíveis para predação sempre que o protocolo for utilizado dentro da rede do ContextNet, evitando assim, erros de transferência. Além disso, é visado o ajuste do mecanismo de comunicação para reforçar a segurança quanto aos agentes suspeitos durante a transferência do SMA. Com isso, o processo de transferência passará a ser mais rigoroso, porém, a predação ocorrerá de forma segura e sem prejudicar na conclusão dos objetivos do sistema.

Referências

- Begon, M., Townsend, C. R., and Harper, J. L. (2005). *Ecology: from individuals to ecosystems*, volume 51. Freshwater Biology - FRESHWATER BIOL.
- Bordini, R. H., Hübner, J. F., and Wooldridge, M. (2007). *Programming Multi-Agent Systems in AgentSpeak using Jason*. John Wiley & Sons Ltd.
- Bratman, M. E. (1987). *Intention, Plans and Practical Reasoning*. Cambridge Press.
- Chen, B., Cheng, H. H., and Palen, J. (2009). Integrating mobile agent technology with multi-agent systems for distributed traffic detection and management systems. *Transportation Research Part C: Emerging Technologies*, 17(1):1–10.
- David, L., Vasconcelos, R., Alves, L., André, R., Baptista, G., and Endler, M. (2012). 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.

- Endler, M., Baptista, G., Silva, L., Vasconcelos, R., Malcher, M., Pantoja, V., Pinheiro, V., and Viterbo, J. (2011). 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.
- Ferri, G., Caselli, E., Mattoli, V., Mondini, A., Mazzolai, B., and Dario, P. (2006). A biologically-inspired algorithm implemented on a new highly flexible multi-agent platform for gas source localization. In *Biomedical Robotics and Biomechanics, 2006. BioRob 2006. The First IEEE/RAS-EMBS International Conference on*, pages 573–578. IEEE.
- Günay, A., Winikoff, M., and Yolum, P. (2015). Dynamically generated commitment protocols in open systems. *Autonomous Agents and Multi-Agent Systems*, 29(2):192–229.
- Huynh, T. D., Jennings, N. R., and Shadbolt, N. R. (2006). An integrated trust and reputation model for open multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 13(2):119–154.
- Jesus, V. S., Manoel, F. C. P. B., Pantoja, C. E., and Viterbo, J. (2018). Transporte de agentes cognitivos entre sma distintos inspirado nos princípios de relações ecológicas. *Workshop-Escola de Sistemas de Agentes, seus Ambientes e aplicações — XII WESAAC*, pages 179–187.
- Paes, R., Carvalho, G. d., Lucena, C. d., Alencar, P., Almeida, H. d., and Silva, V. d. (2005). Specifying laws in open multi-agent systems. *Agents, Norms and Institutions for Regulated Multi-agent Systems (ANIREM), AAMAS2005*.
- Pantoja, C. E., Jesus, V. S., Manoel, F. C. P. B., and Viterbo, J. (2018). A heterogeneous architecture for integrating multi-agent systems in ami systems. *The Thirtieth International Conference on Software Engineering and Knowledge Engineering (SEKE 2018)*.
- Pantoja, C. E., Stabile Jr, M. F., Lazarin, N. M., and Sichman, J. S. (2016). Argo: A customized jason architecture for programming embedded robotic agents. *Fourth International Workshop on Engineering Multi-Agent Systems (EMAS 2016)*.
- Pardo-Castellote, G. (2003). Omg data-distribution service: Architectural overview. In *Distributed Computing Systems Workshops, 2003. Proceedings. 23rd International Conference on*, pages 200–206. IEEE.
- Rao, A. S. (1996). AgentSpeak(L): BDI agents speak out in a logical computable language. In de Velde, W. V. and Perram, J. W., editors, *Proceedings of the 7th European workshop on Modelling autonomous agents in a multi-agent world (MAAMAW'96)*, volume 1038 of *Lecture Notes in Artificial Intelligence*, pages 42–55, USA. Springer-Verlag.
- Wooldridge, M. (2009). *An introduction to multiagent systems*. John Wiley & Sons.
- Zedadra, O., Seridi, H., Jouandeau, N., and Fortino, G. (2016). A cooperative switching algorithm for multi-agent foraging. *Engineering Applications of Artificial Intelligence*, 50:302–319.
- Zeghida, D., Meslati, D., and Bounour, N. (2018). Bio-ir-m: A multi-paradigm modelling for bio-inspired multi-agent systems. *Informatica*, 42(3).