

DESENVOLVIMENTO DE UM MODELO DE SMART HOME UTILIZANDO SISTEMAS MULTI-AGENTES

Estudantes BOLSISTAS CNPq / ICJ: Patrick Santos da Silva (Ensino Técnico)¹

Estudantes Colaboradores: Vinicius Souza de Jesus (Ensino Técnico)¹

Carlos Eduardo Pantoja¹, Nilson Mori Lazzarin¹, Rejane Cavalcante Sá²

pantoja@cefet-rj.br, nilsonmori@gmail.com, rejanecsa@gmail.com

¹ CEFET-RJ UNED MARIA DA GRAÇA
Rio de Janeiro – RJ

² INSTITUTO FEDERAL DE EDUCAÇÃO CIÊNCIA E TECNOLOGIA DO CEARÁ - CAMPUS FORTALEZA
Fortaleza – CE

Categoria: ARTIGO BÁSICO

Resumo: O controle e gerenciamento de aplicações embarcadas de todos os tipos, tais como Smart Home (domótica), através de Sistemas Multi-Agentes (SMA) apresenta duas faces: a possibilidade de raciocínio autônomo e pró-ativo, melhorando a qualidade das decisões com base nas percepções obtidas; e o aumento da dificuldade de implementação do raciocínio no baixo nível da aplicação. Neste trabalho é apresentado um modelo Smart Home gerenciada por SMA. Sua implementação foi possibilitada através do uso da arquitetura ARGO e do middleware Javino. Ao todo 6 controladores ATMEGA são gerenciados em diversos cenários que variam de 1 a 6 agentes. Por fim é avaliado o desempenho de duas diferentes estratégias de percepção do ambiente por parte dos agentes em cada um dos cenários.

Palavras Chaves: Domótica, Sistemas Multi-Agent, Smart Home, ARGO.

Abstract: *The control and management of embedded applications of all kind such as Smart Home (home automation) using Multi-agent Systems (MAS) present two facets: the chance of an autonomous and pro-active reasoning, improving decision making based on captured perceptions; and the increase of difficulty for implementing reasoning at low level application. In this work, it is presented a smart home model managed by a SMA. Its implementation was able by using the ARGO architecture and the Javino middleware. In all, 6 ATMEGA controllers are managed in several scenarios varying between 1 to 6 agents. After all, it is evaluated the performance of two different strategies for perceiving the real environment by the agents in each scenario.*

Keywords: *Home Automation, Multi-Agent Systems, Smart Home, ARGO.*

1 INTRODUÇÃO

Arduino é uma plataforma, baseada em hardware livre, facilitada para o desenvolvimento de protótipos, são capazes de receber informações de sensores e enviar comandos para atuadores. Uma placa Arduino é uma plataforma de computação ubíqua que pode ser programada para gerenciar as

entradas e saídas, conectadas a dispositivos externos (McRoberts, 2011).

Domótica é uma área que destina-se a implementação de automação, controle e gerenciamento de tarefas relacionadas ao ambiente residencial. Diversos trabalhos têm utilizado a plataforma Arduino no domínio da Domótica, com os mais diferentes objetivos, tais como (Bukman, 2016), (Núñez Guerrero, 2017), (Rodríguez Diego et al., 2017) e (Silva Junior, 2017).

Sistemas Multi-Agentes (SMA) são compostos de agentes autônomos situados em um ambiente (virtual ou real) e possuem uma capacidade de tomada de decisão baseados em suas percepções deste ambiente. Os agentes podem ser programados utilizando-se o framework o Jason, um interpretador da linguagem, orientada a agentes, AgentSpeak. Integrados com o ARGO, uma arquitetura que possibilita a comunicação de sensores e atuadores do Arduino com agentes SMA, os agentes podem ser capazes de realizar o raciocínio cognitivo e tomadas de decisão referente determinadas tarefas em ambiente real, para as quais foram projetados.

Neste trabalho é apresentada a implementação de um protótipo de Smart Home pró-ativa, autônoma, controlada através de agentes cognitivos. Seu controle e gerenciando é realizado através da integração de Sistemas Multi-Agentes e Arduino. A casa de seis cômodos conta com iluminação, sensores de luminosidade, portas com sensores magnéticos e servo motores, sensores de temperatura e campainha.

Durante os experimentos realizados na implementação, foi possível avaliar o desenvolvimento do SMA no gerenciamento do protótipo em duas diferentes formas de implementação, testes de performance foram executados levando em consideração parâmetros como o número de agentes e velocidade de raciocínio dos agentes, e filtragem de informação, afim de explorar diferentes estratégias de implementação acerca do sistema.

Este trabalho está estruturado da seguinte forma: na seção 2 serão apresentados os conceitos básicos; na seção 3 será mostrada a proposta do trabalho; na seção 4 serão dados detalhes sobre materiais e métodos; na seção 5 serão

apresentadas as propostas e discussões; na seção 6 será apresentada a conclusão e propostas futuras.

2 CONCEITOS BÁSICOS

2.1 Framework Jason

O framework Jason é um interpretador em Java do AgentSpeak para o desenvolvimento programação orientada a agentes utilizando o Belief-Desire-Intention (BDI).



Figura 1 - Logo do Framework Jason.

2.1.1 Belief-Desire-Intention

BDI é um modelo de software constituído de três premissas: crenças, desejos e intenções. i) Crenças; verdades agente acredita. Estas crenças podem ser adquiridas através da relação do agente com outros agentes ou através das percepções que o agente recebe de sensores (em um ambiente real) ou da observação de um ambiente virtual. ii) Desejos; motivação do agente para realizar determinado objetivo. Estes desejos são implementadas na criação do agente. iii) Intenções; ações que o agente deve executar. Estas ações são re fruto do comportamento do agente, baseado em suas crenças e desejos.

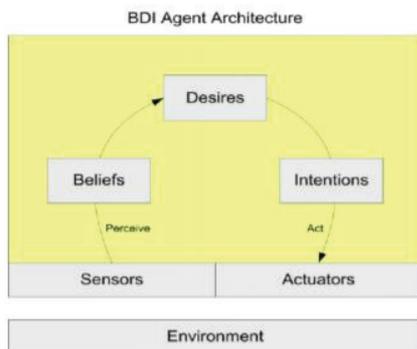


Figura 2 - Diagrama de uma arquitetura BDI genérica (Ghazouani et al., 2013).

2.1.2 Agent Speak

AgentSpeak é uma linguagem de programação voltada para a programação orientada a agentes, baseada em princípios do modelo BDI que permite ao agente construir um sistema de raciocínio, autônomo, pró-ativo e em tempo real para a execução de tarefas complexas.

2.1.3 Agente Jason

O agente Jason possui um clico de raciocínio construído através de informação recebida do ambiente de mensagem, recebida de outro agente ou de suas crenças e desejos. Na programação dos agentes Jason são definidos objetivos, intenções, crenças, planos e ações a serem executadas no ambiente, entretanto, um agente Jason comumente interage apenas em ambientes virtuais.

2.2 ARGO

ARGO é uma arquitetura customizada de agentes Jason que para possibilita a programação de agentes robóticos e ubíquos utilizando diversas plataformas de prototipação (e.g Arduino). Dessa forma, é possível a intermediação entre um agente SMA e um ambiente real, através de sensores e atuadores.

A aplicação do modelo BDI em ambientes reais pode gerar gargalos no processamento das percepções o que pode levar a retardo na tomada de decis este atraso a arquitetura ARGO conta com um sistema de filtros de percepções em tempo de execução.

2.3 Agente ARGO

Um agente ARGO é um agente tradicional (agente Jason) com características adicionais, como por exemplo, a capacidade realizar comunicação com o meio físico, percebê-lo e filtrar informações percebidas com o mundo real provêm do middleware Javino (Lazarin and Pantoja, 2015).

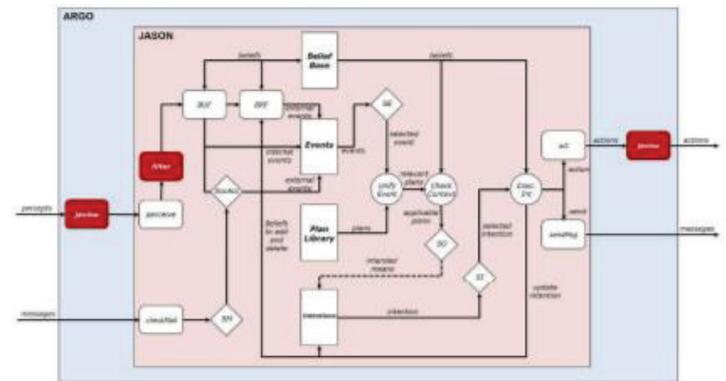


Figura 3 - Clico de raciocínio de um agente ARGO (Pantoja et al., 2016).

2.4 Javino

O Javino é um protocolo de comunicação criado para ser utilizado como middleware entre hardware (Arduino) e software (Java).

O Arduino possui uma comunicação serial síncrona, dessa forma, as informações são trocadas byte a byte entre o Arduino e o PC e cada byte é tratado individualmente. O Javino implementa uma camada de encapsulamento na transmissão desses bytes, sendo possível que hardware e software troquem mensagens de até 256bytes de uma única vez.

Este protocolo conta com uma técnica de validação de mensagem, baseada na verificação do tamanho da mensagem, possibilitando identificação de erro nos dados recebidos, fornecendo confiabilidade entre origem-destino. Uma mensagem Javino é composta por três campos: preâmbulo (2 Bytes), tamanho da mensagem (1Byte) e mensagem (até 256 Bytes) (de Jesus et al.).

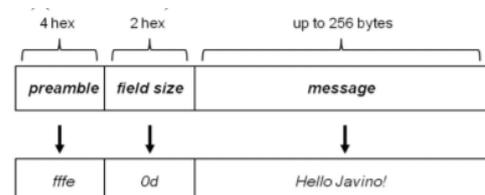


Figura 4 - Formato da mensagem Javino (Lazarin and Pantoja, 2015).

3 O TRABALHO PROPOSTO

Nesta seção será apresentada a metodologia de implementação do protótipo de Smart Home, onde foi utilizando o framework Jason e a arquitetura ARGO para programação dos agentes Jason e agentes ARGO; o protocolo Javino para comunicação entre os agentes e o hardware; e o Arduino para o controle dos sensores e atuadores. No protótipo de Smart Home diversos controladores realizam o sensoriamento de diversos dispositivos presentes nos cômodos da casa. Os dispositivos utilizados são sensores de temperatura e o de luminosidade, além de diversos atuadores como luzes de LED, servo motores para as portas e campainha.

Um SMA utilizando o framework Jason é responsável pelo gerenciamento cognitivo dos dispositivos conectados na estrutura física da casa. No SMA, os agentes ARGO são responsáveis pelo gerenciamento dos controladores enquanto que os agentes tradicionais do Jason são responsáveis por auxiliar na comunicação e pela colaboração na execução de determinadas tarefas. Um agente ARGO poderá gerenciar um ou mais dispositivos de acordo com a necessidade da tarefa, desde que não haja conflitos na utilização da porta serial.

O protótipo tem cômodos que são controlados individualmente um Arduino por cada cômodo. Para cada cômodo, também foi designado um agente ARGO responsável pelo gerenciamento cognitivo do Arduino ligado aos sensores e atuadores. Esses agentes, além de controlarem dispositivos de hardware, são capazes de filtrar percepções em tempo real que não são interessantes aquele agente, a fim de obter uma melhor performance de execução.

4 MATERIAIS E MÉTODOS

A metodologia de desenvolvimento do protótipo basicamente leva em consideração três camadas onde é necessária intervenção: a montagem do hardware; a programação da controladora e; a criação do SMA. Primeiramente toda a estrutura física deve ser preparada, onde sensores e atuadores devem ser conectados às controladoras no ambiente ou cômodo desejado. Em um segundo momento, todas as funções de acionamentos dos efetadores devem ser programadas nas controladoras em respostas a estímulos da porta serial e as percepções provenientes dos sensores devem ser preparadas levando em consideração o formato esperado pela linguagem do SMA (no caso o Jason).

As percepções são enviadas ao agente a cada vez que um agente ARGO realizam seu ciclo de raciocínio (através da chamada ao método `getPercepts`, que deve existir obrigatoriamente em toda controladora que precisa enviar percepções a agentes). Por fim, o SMA deve ser programado de forma independente ao hardware, levando em consideração apenas as ações que devem ser executadas de acionamento (e.g. `.act(ligarLedSala)`) e de seleção do dispositivo (e.g. `.port(com8)`).

Dessa forma, foi desenvolvido um protótipo levando em consideração a arquitetura de sensores e atuadores da Smart Home. O protótipo foi desenvolvido em madeira e foram utilizados seis Arduinos. Foram usados os sensores de luminosidade LDR e os sensores de temperatura LM35; diversos LED para cada cômodo; um botão para simular a campainha e um Buzzer e; a porta possui um sensor magnético (indicando se está aberta ou fechada) e um servo motor.

Além disso, os acionamentos para um aquecedor e um ar condicionado (para o gerenciamento da temperatura

ambiente) também são disponibilizados. A programação dos controladores foi feita baseado no acionamento dos LED, da porta, do ar-condicionado e do aquecedor. Os sensores são responsáveis pelas percepções que serão preparadas e enviadas aos agentes. O protótipo pode ser visto na Figura 2.



Figura 5 - Protótipo de Smart House.

5 RESULTADOS E DISCUSSÃO

A fim de testar a aplicabilidade e identificar as melhores estratégias de utilização do ARGO no domínio de Smart Homes, testes de performance foram executados realizando um circuito de ativação e desativação dos LED (o agente liga e desliga os LED de cômodo em cômodo).

No framework Jason, um evento é gerado para cada percepção recebida do ambiente. Esta característica pode gerar atrasos no raciocínio e na execução de ações em situações que necessitam de respostas rápidas. Para lidar com essa situação, um agente pode filtrar as percepções baseados no uso de filtros pré-definidos em tempo de design e/ou variar a utilização da ação interna de bloqueio e liberação das percepções do ambiente.

Assim, foram realizados testes e medido o tempo de execução do circuito de ativação nos 6 cômodos do protótipo, sendo controlados por 1 até 6 agentes no SMA usando duas estratégias diferentes.

5.1 Experimento 1

O experimento 1 levou em consideração, que os agentes abrissem o fluxo de percepções dos sensores no início da execução para ficar percebendo o ambiente continuamente. Ou seja, neste experimento não foram utilizando filtros de percepção.

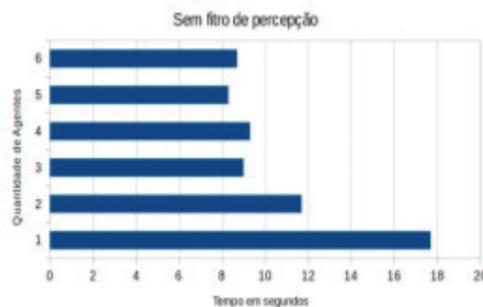


Figura 6 - Resultado no experimento 1.

5.2 Experimento 2

O Experimento 2 levou em consideração que os agentes abrissem e fechassem o fluxo de percepções no início do primeiro plano e no último ou quando houvesse mudança de

controladores (afim de atualizar as percepções provenientes do novo dispositivo).

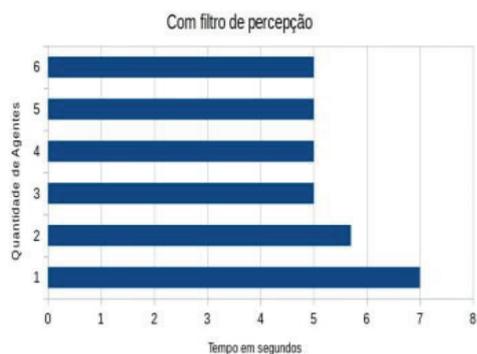


Figura 7 - Resultado no experimento 2.

5.3 Discussão

A abordagem do experimento 1 é a mais lenta, pois a quantidade de percepções processadas é maior do que a outra estratégia. A abordagem do experimento 2 é mais rápida que a primeira, contudo, dependendo da programação do agente, manter a percepção desatualizada por um longo período de tempo, pode fazer com que o agente raciocine com dados defasados.

Ao se analisar os resultados levando em consideração o quantitativo de agentes empregados no SMA, percebe-se que quando apenas um agente é responsável pelo controle dos dispositivos, o tempo de duração aumenta independente da estratégia adotada.

6 CONCLUSÕES

Este trabalho apresentou um modelo de uma Smart Home usando agentes cognitivos, desenvolvidos com o framework Jason e com agentes ARGO. Uma análise de duas diferentes estratégias de implementação da abordagem proposta também foi apresentada. A utilização de SMA para o controle de dispositivos de hardware não é uma tarefa simples e diversas abordagens tentam melhorar a performance na programação desses agentes.

O objetivo deste trabalho é evidenciar o potencial que a arquitetura possui para o desenvolvimento de SMA utilizando hardware de baixo custo e que também é possível a implementação de SMA para gerenciamento de dispositivos físicos utilizando o Jason em um tempo aceitável, gerenciando o ponto crítico da integração física de agentes BDI, que é o processamento das percepções vindas do ambiente.

Como trabalho futuro será utilizado um middleware para a internet das coisas que permitirá a comunicação e a integração de diversos dispositivos gerenciados por SMA em uma smart home. Também, a arquitetura da smart home será expandida no laboratório do grupo de pesquisa. Técnicas de comunicação e negociação para os agentes serão exploradas para identificar em quais situações a arquitetura proposta oferecerá melhores resultados.

REFERÊNCIAS BIBLIOGRÁFICAS

- Bordini, R. H., Hübner, J. F., and Wooldridge, M. (2007). *Programming Multi-Agent Systems in AgentSpeak using Jason*. John Wiley & Sons Ltd.
- Jesus, V.S., Samyn, L.M., Manoel, F.C.P., and Pantoja, C.E. *LuBras: Uma Arquitetura de um Dispositivo Eletrônico para a comunicação Libras-Língua Portuguesa Utilizando o Javino*.
- Ghazouani, M., Medromi, H., Boulaifdour, B., and Sayouti, A. (2013). A model for an Information security management system (ISMS Tool) based multi agent system.
- Junger, D., Guinelli, J. V., and Pantoja, C. E. (2016). An Analysis of Javino Middleware for Robotic Platforms Using Jason and JADE Frameworks. In 10th Software Agents, Environments and Applications School.
- Lazarin, N.M., and Pantoja, C.E. (2015). A robotic-agent platform for embedding software agents using raspberry pi and arduino boards. 9th Software Agents, Environments and Applications School.
- McRoberts, M. (2011). *Arduino básico*. São Paulo: Novatec.
- Pantoja, C.E., Stabile, M.F., Lazarin, N.M., and Sichman, J.S. (2016). ARGO: an extended jason architecture that facilitates embedded robotic agents programming. In *International Workshop on Engineering Multi-Agent Systems*, (Springer), pp. 136–155.
- Wooldridge, M. (2009). *An Introduction to MultiAgent Systems*. Wiley.
- Bukman, G. (2016). *Desenvolvimento de um protótipo para segurança residencial de baixo custo*.
- Núñez Guerrero, F.G. (2017). *Implementación de un prototipo de un sistema domótico basado en arduino para control, monitoreo y asistencia en hogares para personas de la tercera edad*. B.S. thesis. Quito. Universidad de las Américas.
- Rodríguez Diego, M., and others (2017). *Gestión domótica de una casa unifamiliar basada en Arduino*.
- Silva Junior, A.C. (2017). *Automação Residencial: Sistemas Residenciais Controlados via Smartphone*.