

Implantação de uma arquitetura ciber-física baseada em agentes para um Smart Parking

Pedro W. Botelho¹, Gleifer V. Alves¹, André P. Borges¹

¹Departamento Acadêmico de Informática – Universidade Tecnológica Federal do Paraná
Rua Doutor Washington Subtil Chueire, 330 - 84017-220 – Ponta Grossa – PR – Brasil

pbotelho@alunos.utfpr.edu.br, {gleifer, apborges}@utfpr.edu.br

Abstract. *Smart Parking systems are being used to manage and allocate parking lots in several cities. These systems aim to reduce urban traffic by helping drivers to find a place to park. The inclusion of new technologies such as Cyber-Physical Systems and artificial intelligence techniques such as Multi-Agent Systems are considered a suitable alternative for solving this problem. In this context, the interconnection of the physical part with the agents is a fundamental issue for the correct functioning of the system. Thus, this paper presents the implementation of an agent-based cyber-physical architecture that focuses on connecting agents with physical controllers using IoT technologies.*

Resumo. *Sistemas para estacionamentos inteligentes estão sendo utilizados no gerenciamento e alocação de vagas em diversas cidades. Estes sistemas visam a redução do tráfego urbano auxiliando os motoristas na busca por vagas. A inclusão de novas tecnologias, como Sistemas Ciber-Físicos e técnicas de inteligência artificial, como Sistemas Multi-Agentes são consideradas uma alternativa considerável para a resolução deste problema. Neste contexto, a conexão da parte física com os agentes é uma questão fundamental para o correto funcionamento do sistema. Assim, este trabalho apresenta a implantação de uma arquitetura ciber-física baseada em agentes que tem como foco a conexão dos agentes com controladores físicos utilizando tecnologias de Internet das Coisas.*

Keywords: Agents; Smart Parking; Cyber-Physical Systems; Internet of Things

Palavras-chave: Agentes; Estacionamento Inteligente; Sistemas Ciber-Físicos; Internet das Coisas

1. Introdução

As Cidades Inteligentes usam tecnologias de comunicação e informação como sensores para coletar dados e então, usa-se estes dados para gerenciar seus recursos de forma eficiente [Caragliu et al. 2011]. Para gerenciar os dados de tráfego de maneira eficiente, deve-se também lidar com uma das causas de engarrafamento nas cidades, a busca por vagas de estacionamento [Lin et al. 2017].

A divergência entre a falta de espaço nas grandes cidades e o número de carros presentes nelas faz com que encontrar vagas seja um problema para os motoristas. Dirigir

à procura de uma vaga faz com que o trânsito se torne mais lento, causando congestionamentos, gasto de tempo e gasolina dos veículos e ainda emitindo no meio ambiente gases poluentes que não seriam emitidos se o carro estivesse estacionado [Lin et al. 2017].

Para tentar lidar com esse problema surgem os Estacionamentos Inteligentes. Estacionamento Inteligente é um sistema de estacionamento que auxilia os motoristas a encontrar vagas utilizando sensores que detectam se há ou não um veículo e então o direcionam para a vaga [Hassoune et al. 2016]. O objetivo desses sistemas é garantir que o motorista encontre uma vaga para estacionar da maneira mais rápida possível levando em consideração o local desejado. Assim, o motorista pode iniciar uma negociação para reservar uma vaga antes mesmo de entrar no veículo [Castro et al. 2017].

Por ser uma solução complexa, estacionamentos inteligentes precisam adotar tecnologias que atendam a demanda da implementação, como os Sistemas Ciber-Físicos (SCF) que oferecem comunicação entre elementos computacionais e entidades físicas utilizando plataformas de hardware [Kaithan and McCalley 2015]. Porém, esta alternativa não apresenta uma solução inteligente contendo gerenciamento de vagas e negociação, assim, é proposto utilizar agentes embarcados nestas entidades físicas. Um agente é um sistema computacional capaz de ações autônomas no ambiente que está situado visando atingir seus objetivos propostos [Wooldridge 2009]. Desta forma, é possível fazer com que agentes façam a negociação de uma vaga enquanto outros agentes gerenciam as vagas do estacionamento.

Em trabalhos relacionados foram desenvolvidos mecanismos de negociação entre agentes [Castro et al. 2017] [Ducheiko et al. 2018]. Esses trabalhos realizam simulações computacionais para avaliar sua aplicabilidade em um estacionamento inteligente. Logo, há necessidade de realizar tais simulações em ambientes físicos, o que possibilitará uma avaliação precisa para incorporar uma arquitetura baseada em agentes em um estacionamento inteligente.

Em [Sakurada et al. 2019], é justamente descrita uma arquitetura ciber-física para estacionamentos inteligentes, porém tal arquitetura considera um estacionamento para bicicletas. O trabalho descrito aqui propõe a implantação de uma arquitetura ciber-física baseada em agentes, para veículos. Tal arquitetura visa obter informações a respeito da vaga de estacionamento, para que a partir destas informações, seja possível ao agente interagir com o sistema de estacionamento. Assim, esta abordagem permitirá criar um meio de se realizar testes de protocolos de negociação em ambientes mais próximos de um estacionamento real, materializando assim os trabalhos anteriormente citados.

O restante do artigo está estruturado da seguinte forma. Na seção 2, é descrito o desenvolvimento da arquitetura, contendo uma maior descrição das tecnologias utilizadas e como cada uma delas atua dentro do sistema. Na seção 3, é descrito a implantação da arquitetura, como cada dispositivo foi programado para funcionar corretamente. E por fim, a seção 4 apresenta as considerações finais a respeito do desenvolvimento do trabalho e os trabalhos futuros.

2. Desenvolvimento da arquitetura

Em um SCF, a parte física é capaz de interagir com o mundo real por meio de sensores e a parte ciber é o processo de obter dados destes sensores [Jazdi 2014]. Assim, a arquitetura é baseada em [Sakurada et al. 2019] sendo a parte ciber constituída de placas Raspberry Pi contendo um agente embarcado e implementado no framework JADE para receber as informações da parte física e a parte física composta por módulos ESP-12e conectados a sensores de presença para monitorar o estado da vaga (por exemplo, livre ou ocupada).

A comunicação entre estes dois dispositivos dá-se-a através de um broker MQTT que também está embarcado no Raspberry Pi. Este broker segue o princípio *Publish-Subscribe*, que tem seu funcionamento por meio de tópicos. Um dispositivo publica em um tópico no broker e o mesmo replica esta mensagem a todos os dispositivos que estão inscritos neste tópico. Assim, é possível que dois dispositivos troquem informações, o que atende as requisições necessárias para o funcionamento do sistema.

A Figura 1 ilustra a arquitetura, que é composta por n setores distribuídos em um estacionamento e cada setor possui m vagas que são gerenciadas pelo agente responsável pelo setor. A parte da negociação ocorre em outra camada da arquitetura pois o presente artigo tem como escopo apenas o ambiente físico do estacionamento. Porém, basicamente o motorista faz a requisição de uma vaga e um Sistema Multi-Agente (SMA) é responsável por receber esta requisição, negociar a vaga com o motorista e atualizar uma base de dados que contém as informações sobre o motorista e a vaga, como o número do setor e da vaga, o id do motorista, horário da reserva, etc.

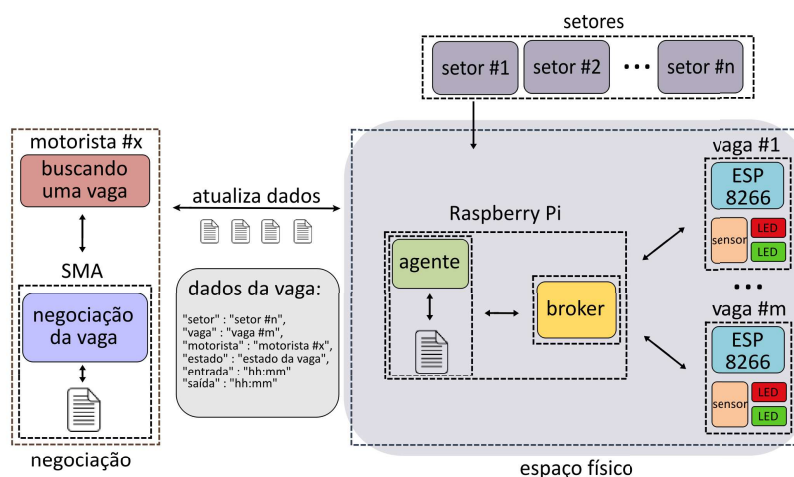


Figura 1. Implementação da comunicação ciber-física

O agente, desenvolvido utilizando o framework JADE e embarcado em um Raspberry Pi 3B+, é responsável por receber uma mensagem do controlador físico, verificar se a vaga está reservada consultando a base de dados e atualizar o estado da vaga. O controlador físico é composto por um microcontrolador ESP8266, um sensor ultrassônico de presença e dois LEDs, um verde e outro vermelho. Juntos, estes dispositivos são responsáveis por enviar ao agente uma mensagem através do broker contendo o status correspondente da vaga para que possa ser feita a verificação.

Cada vaga possui um ESP8266 conectado a um sensor de presença para medir a distância entre o dispositivo e um obstáculo a sua frente, no caso de estacionamentos fechados, a medida pode ser entre o chão e o teto. Quando um carro estaciona na vaga, o sensor medirá a distância dele até o carro, portanto, a distância anterior e a atual não é a mesma. Como resultado, é possível reconhecer que há um carro estacionado e então enviar uma mensagem para o agente. O mesmo procedimento é seguido quando o carro sai da vaga. Assim, o microcontrolador é programado para monitorar o sensor e enviar uma mensagem ao broker toda vez que o sensor detectar que a distância foi alterada.

Para que o sistema funcione corretamente, o agente correspondente do setor deve gerenciar as mensagens recebidas do broker e atualizar o estado da vaga, enquanto o SMA é o responsável pela negociação das vagas livres com o motorista. Como consequência, o agente é programado para receber as mensagens do controlador da vaga, consultar a base de dados para verificar o estado dela, e então enviar uma mensagem de volta ao controlador com qual LED deve acender-se. Especificamente, o sistema aqui proposto funciona da seguinte maneira:

1. O motorista requer uma vaga de estacionamento por meio de um aplicativo móvel ou de um sistema web;
2. O SMA recebe esta requisição, negocia uma vaga baseado nas vagas livres e atribui uma vaga ao motorista. Ex: vaga#15;
3. O SMA atualiza a base de dados mudando o estado da vaga e atualizando os demais campos. Assim, a vaga#15 não está mais livre e sim reservada;
4. Neste momento, o agente no Raspberry Pi recebe a informação que a base de dados foi atualizada, então o mesmo fica a espera de uma notificação;
5. O motorista estaciona na vaga#15;
6. O sensor no ESP8266 detecta que um carro estacionou na vaga#15 e envia uma mensagem notificando o agente;
7. O agente então deve verificar a base de dados daquela vaga: a vaga#15 está reservada para este horário?
8. Se a vaga está reservada, o agente atualiza o estado da vaga na base de dados de reservado para ocupado;
9. O agente envia uma mensagem para o ESP8266 que a vaga#15 agora está ocupada;
10. O LED vermelho deve ser aceso indicando que a vaga está ocupada.

Note que se a reserva não for confirmada (não existe reserva para aquela vaga na base de dados), o agente não vai publicar a confirmação da vaga e assim o ESP8266 não irá acendar o LED vermelho, mantendo o LED verde aceso. Portanto, assume-se que o motorista irá respeitar a política do sistema e apenas estacionar em uma vaga corretamente atribuída ao motorista.

3. Implantação da arquitetura

Como um primeiro experimento, foi desenvolvido um simples protótipo ciber-físico contendo duas vagas contendo um controlador físico em cada. Com isso, foi possível testar a troca de mensagens entre o agente e o controlador físico por meio do broker. Esta troca

de mensagens só é possível por meio de tópicos, assim, para receber as mensagens publicadas em um tópico, o dispositivo deve estar inscrito no mesmo. A Figura 2 ilustra os tópicos seguidos para a troca de mensagens e ainda em qual tópico cada dispositivo está inscrito e publica mensagens.

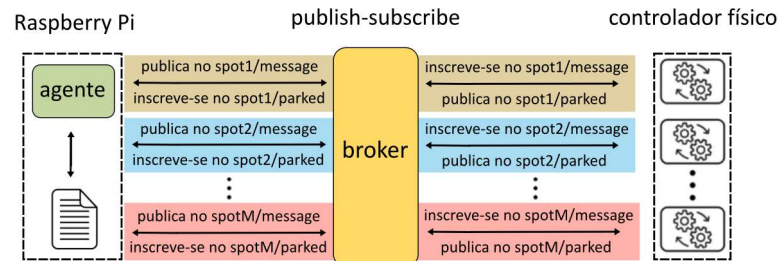


Figura 2. Tópicos do broker para o suporte de comunicação publish/subscribe

Neste experimento não é feito o uso do mecanismo de negociação, portanto, para simular um carro estacionado é necessário utilizar um objeto (representando o carro) dentro do raio de alcance do sensor ultrassônico. Dessa maneira, é possível determinar quando o controlador deve notificar o agente que um carro está estacionado. O agente implementado nos experimentos executa passos simples: recebe uma mensagem por meio do broker e envia de volta uma mensagem de acordo com as especificações abaixo.

O microcontrolador ESP8266 é capaz de conectar a Internet, portanto, pode ser programado para conectar-se no broker que está no Raspberry Pi por meio do endereço de IP. O sensor ultrassônico mede a distância pela duração em microsegundos de quando uma onda ultrassônica é emitida para quando ela bate no objeto a sua frente e volta. Assim, a distância deve ser ajustada para que haja uma distância padrão a que possa ser comparada com a distância atual. Deve-se também manter a distância anterior para evitar o envio mensagens repetidas para o broker. Portanto, quando um carro deixa a vaga, apenas uma mensagem é enviada, e isso é apenas possível comparando a distância anterior com a distância atual.

Quando a conexão com a Internet é estabelecida e a distância padrão é ajustada, o LED verde é acendido e a distância atual é obtida e comparada com a distância padrão. Se a distância atual é menor que a distância padrão significa que um carro está estacionado na vaga, portanto, os seguintes passos são seguidos:

- O ESP8266 publica no tópico 'spot1/parked' que um carro estacionou;
- O agente verifica a base de dados da vaga e atualiza o estado dela para ocupado;
- O agente então publica no tópico 'spot1/message' que a reserva está confirmada;
- O ESP8266 recebe a mensagem e acende o LED vermelho.

Quando o carro sai da vaga, uma nova mudança na distância é detectada pelo sensor, sendo necessário executar os seguintes passos:

- O ESP8266 publica no tópico 'spot1/parked' que um carro não está mais estacionado;
- O agente recebe a mensagem e atualiza o estado da vaga para livre;

- O agente então publica no tópico 'spot1/message' que o estado da vaga foi atualizado;
- O ESP8266 recebe a mensagem e acende o LED verde.

Se a vaga está livre, o sensor continua medindo a distância a cada cinco segundos até que uma mudança na distância é detectada. Quando isso ocorre, os passos anteriores são reproduzidos.

Neste contexto, um experimento foi realizado para identificar o correto funcionamento dos dispositivos escolhidos e também o processo de troca de mensagens entre o agente e o ESP8266 por meio do broker. Este experimento foi realizado durante quatro horas e os controladores físicos das duas vagas foram capaz de detectar o carro estacionado as 25 vezes que um objeto simulando o carro foi colocado a frente de cada, assim como foi capaz de enviar a mensagem correspondente ao agente 100 vezes, 50 informando que o carro estacionou e 50 que o carro saiu da vaga. Além disso, o agente apresentou o comportamento esperado gerenciando as 100 mensagens recebidas de forma correta e enviou de volta uma mensagem informando qual LED deveria ser aceso.

4. Conclusão

Este artigo apresenta a implantação de uma arquitetura baseada em agentes e SCF. A arquitetura possui como uma de suas características, a escalabilidade, visto que é possível replicar a estrutura em n setores e m vagas, dependendo da necessidade do ambiente de implantação. Outra característica diz respeito a modularização, já que facilmente módulos podem ser trocados de forma que seja implantado em outro ambiente, por exemplo um estacionamento inteligente de bicicletas.

Tendo como base os experimentos iniciais é possível evidenciar que o princípio *Publish-Subscribe* apresenta resultados satisfatórios na comunicação entre o agente e os controladores físicos. Além disso, o sensor ultrassônico é capaz de detectar um carro todas as vezes, enviando corretamente a mensagem para o agente o qual lidou satisfatoriamente com o gerenciamento das mensagens.

Como continuidade imediata deste trabalho, deseja-se implantar uma API REST para conectar a arquitetura com um SMA responsável pela negociação entre os agentes do sistema. A partir disto será implantando um simples protótipo de um estacionamento inteligente, onde serão executados diversos testes para evidenciar a aplicabilidade desta arquitetura em um cenário com componentes de um ambiente físico.

Referências

- Caragliu, A., Bo, C. D., and Nijkamp, P. (2011). Smart Cities in Europe. *Journal of Urban Technology*, 18(2):65–82.
- Castro, L. F. S. D., Alves, G. V., and Borges, A. P. (2017). Using trust degree for agents in order to assign spots in a Smart Parking. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, 6(2):45–55.
- Ducheiko, F. F., Alves, G. V., and Borges, A. P. (2018). Implantação de um modelo de raciocínio e protocolo de negociação para um estacionamento inteligente com mecanismo de negociação descentralizado. In *Revista Junior – ICCEE*, pages 25–32.

- Hassoune, K., Dachry, W., Moutaouakkil, F., and Medromi, H. (2016). Smart parking systems: A survey. In *2016 11th International Conference on Intelligent Systems: Theories and Applications (SITA)*, pages 1–6. IEEE.
- Jazdi, N. (2014). Cyber physical systems in the context of industry 4.0. *Proceedings of the IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR'14)*, pages 1–4.
- Kaithan, S. K. and McCalley, J. D. (2015). Design techniques and applications of cyberphysical systems: A survey. *IEEE Systems Journal*, pages 350–365.
- Lin, T., Rivano, H., and Le Mouël, F. (2017). A survey of smart parking solutions. *IEEE Transactions on Intelligent Transportation Systems*, 18(12):3229–3253.
- Sakurada, L., Barbosa, J., Leitão, P., Botelho, P. W., Alves, G. V., and Borges, A. P. (2019). Development of agent-based cps for smartparking systems. In *Aguardando publicação*.
- Wooldridge, M. (2009). *An introduction to multiagent systems*. John Wiley & Sons.