

Desenvolvimento de Interface Gráfica para Gerenciamento de um Smart Parking

Felipe Felix Ducheiko
Universidade Tecnológica Federal do Paraná
Av. Monteiro Lobato, s/n - km 04
84016-210 - Ponta Grossa - PR - Brasil
felipeducheiko@alunos.utfpr.edu.br

Gleifer Vaz Alves
Universidade Tecnológica Federal do Paraná
Av. Monteiro Lobato, s/n - km 04
84016-210 - Ponta Grossa - PR - Brasil
gleifer@utfpr.edu.br

RESUMO

O projeto MAPS (*MultiAgent Parking System*) é idealizado com objetivo de desenvolver um Sistema Multiagente para alocação de vagas em um estacionamento inteligente, onde um agente *manager* é responsável pela alocação de vagas aos agentes *drivers*. O artigo aqui apresentado faz parte do projeto MAPS e tem como objetivo implementar uma interface gráfica, a qual poderá auxiliar o controle e gestão de um estacionamento inteligente.

Palavras-chave

Sistema Multiagente; Cidade Inteligente; Estacionamento Inteligente; *Framework* JaCaMo; Interface Gráfica.

ABSTRACT

The MAPS project (MultiAgent Parking System) is designed with the objective of developing a Multi-Agent System for allocation of parking spots in a smart parking. Where an agent manager is responsible for assigning the parking spot to the agent drivers. Here we aim to build a graphic interface in order to help the management of a smart parking and it is part of the MAPS project.

Keywords

Multi-Agent system; Smart city; Smart Parking; JaCaMo framework; graphic interface.

1. INTRODUÇÃO

Frequentemente novas tecnologias são desenvolvidas ou aprimoradas a fim de solucionar problemas enfrentados pela população. Cidade Inteligente (*Smart City*) refere-se ao uso de novas tecnologias na tentativa de solucionar os problemas das cidades a fim de melhorar a qualidade de vida de seus habitantes.

A concepção de Cidade Inteligente surgiu durante a última década com a fusão de várias ideias, tendo o intuito de

melhorar a eficiência e a competitividade das cidades, criando novas maneiras para solucionar problemas. A essência do conceito é integrar as tecnologias que até agora têm sido desenvolvidas separadamente umas das outras, mas que tem ligações claras em seu funcionamento e podem ser desenvolvidas de forma integrada [1].

Um dos principais problemas enfrentados atualmente pelas cidades é o de mobilidade urbana. Estima-se que em Nova York cerca de 40% do tráfego é gerado por carros a procura de vagas de estacionamento, este fato ocasiona um agravamento dos congestionamentos e por conseguinte a emissão de poluentes [6].

Quando se percebe que a grande demanda de vagas de estacionamentos não está sendo satisfeita, normalmente provem a noção de que a solução é um aumento quantitativo do número de vagas. Porém, nem sempre essa é a solução mais sensata, pois a utilização das mesmas vagas de modo mais inteligente pode solucionar ou amenizar o problema. Um *Smart Parking* (Estacionamento Inteligente) faz uso de dispositivos de *hardware*, capazes de detectar o nível de ocupação dos estacionamentos, e de *softwares* integrados para gerir a atribuição desses espaços de maneira a otimizar sua utilização [7].

Várias abordagens de pesquisa em *Smart Parking* estão sendo realizadas para solucionar problemas de mobilidade. Estas abordagens podem ser categorizadas conforme as tecnologias empregadas em sua implementação, algumas das categorias são: *Automated Parking* (foco em mecanismos computadorizados), *E-Parking* (utiliza Internet ou SMS) e *Agent Based Guiding System* (ABGS), esta última é caracterizada pela utilização de Sistemas Multiagentes na implementação de *Smart Parkings* [8].

Sistemas Multiagentes (SMAs) são definidos como sistemas compostos de vários elementos computacionais que realizam interações, sendo tais elementos conhecidos como agentes. Esses sistemas possuem duas características importantes: primeiramente são, ao menos em certa medida, capazes de ações autônomas e em segundo lugar têm a capacidade de interagir uns com os outros de maneira análoga às interações sociais humanas. Além disso, os agentes estão envolvidos em um ambiente onde eles podem ter organização, comunicação, entre outros aspectos [9].

Com o propósito de aplicar métodos e técnicas de SMA, na criação de uma solução para alocação de vagas e gerenciamento de um *Smart Parking*, foi concebido o projeto MAPS (*MultiAgent Parking System*). No desenvolvimento do MAPS é usado o *framework* JaCaMo, o qual possui três componentes principais: *Jason* (programação de agen-

tes autônomos), *Cartago* (programação de artefatos do ambiente) e *Moise* (programação da organização do SMA) [4].

Na versão inicial do MAPS era usado apenas o console padrão do JaCaMo, o qual oferece uma interface de texto com informações básicas a respeito dos agentes. Porém, a interface de usuário deve ser clara, intuitiva e de fácil compreensão para a uma efetiva utilização do sistema, visto que usuários têm um grande potencial para fazer interpretações inesperadas de elementos da interface e para executar o seu trabalho de forma diferente do aguardado [5]. A partir disso, notou-se a necessidade de criar uma interface gráfica mais intuitiva e com informações adicionais sobre o *Smart Parking*.

Especificamente, o artigo aqui apresentado tem como objetivo principal apresentar o desenvolvimento de uma interface gráfica para o projeto MAPS, esta interface é voltada para auxiliar o controle e gestão do *Smart Parking*. A implementação deste novo recurso visa também futuras expansões do projeto. Como, por exemplo, a aplicação de diferentes algoritmos para alocação de vagas, onde o uso da interface poderá facilitar a visualização (e comparação) dos resultados gerados pelos algoritmos implementados.

2. PROJETO MAPS

O projeto MAPS é desenvolvido no GPAS (Grupo de Pesquisa em Agentes de Software - UTFPR - PG) com a meta principal de elaborar soluções para Estacionamentos Inteligentes. Para implementar o *Smart Parking* foram criados dois tipos de agentes (implementados em *Jason*): os agentes *drivers* que interagem e utilizam o Sistema Multiagente e o agente *manager* que é responsável por informar, alocar e gerenciar as vagas do estacionamento [2].

O sistema também é composto por dois artefatos (implementados no *Cartago*): o artefato *control* que é responsável pelo gerenciamento dos motoristas na fila e o artefato *gate* que é responsável pela cancela. A alocação de vagas é realizada conforme o grau de confiança (*degree of trust*, ou apenas *trust*) de cada motorista, visto que o grau de confiança e a reputação são temas centrais para a interação efetiva em um SMA aberto em que os agentes entram e saem do sistema [3]. O grau de confiança para cada agente *driver* é incrementado a cada momento que o *driver* usa o estacionamento. Logo, aquele *driver* que utiliza o estacionamento com maior frequência, terá um maior grau.

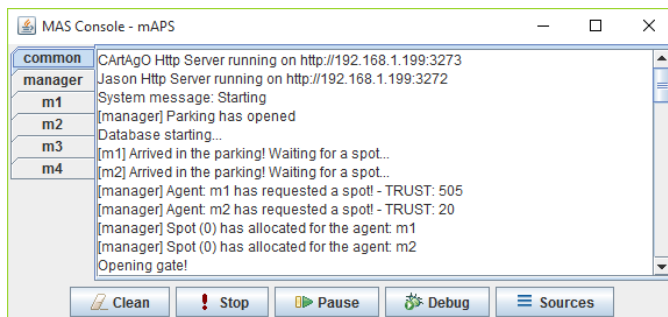


Figura 1: Console do *framework* JaCaMo

Atualmente o MAPS conta apenas com um console, disponibilizado pelo próprio *framework* JaCaMo, que pode ser visualizado na Figura 1. Esse console tem como objetivo exibir a execução dos diferentes agentes do sistema, mas são

apenas informações das ações dos agentes, por exemplo, qual agente conseguiu uma determinada vaga. Todavia, não há informações a respeito da quantidade de vagas já ocupadas no estacionamento, por exemplo. Assim, identificou-se a necessidade de desenvolver uma interface gráfica que atende-se tais requisitos. A partir desta nova interface acredita-se que será possível realizar não apenas o gerenciamento do *Smart Parking*, mas também a visualização de simulações do SMA.

3. IMPLEMENTAÇÃO DA INTERFACE

Para o desenvolvimento de uma interface é necessário levar em consideração a usabilidade, que consiste na utilização de métodos que têm o objetivo de facilitar o uso de um sistema. O desenvolvimento desta interface gráfica se deu pautado nos seguintes requisitos de usabilidade: (i) facilidade de aprendizagem: o sistema deve ser facilmente assimilado pelo usuário; (ii) eficiência: o sistema deve ser eficiente para que o usuário possa atingir uma boa produtividade; (iii) facilidade de memorização: o sistema deve ser facilmente memorizado, para que depois de algum tempo sem o utilizar, o usuário se recorde como usá-lo; (iv) segurança: o sistema deve prever erros, evitar que os usuários os cometam e, se o cometerem, permitir fácil recuperação ao estado anterior; e (v) satisfação: o sistema deve ser usado de uma forma agradável, para que os usuários fiquem satisfeitos com a sua utilização [5].

A linguagem Java foi escolhida para implementar a interface gráfica, devido a facilidade de realizar sua comunicação com o *framework* JaCaMo. A interface gráfica desenvolvida também é facilmente extensível, devido às futuras alterações nas versões do sistema.

Na tela inicial da interface gráfica, que pode ser visualizada na Figura 2, o usuário já possui informações sobre o nível de ocupação do *smart parking* e sobre a situação da fila de espera. Referente ao nível de ocupação do *smart parking* o usuário pode verificar o número de motoristas na fila e o tempo médio de espera, quanto a situação da fila de espera o usuário pode verificar a quantidade total de vagas e o número de vagas livres e em uso.

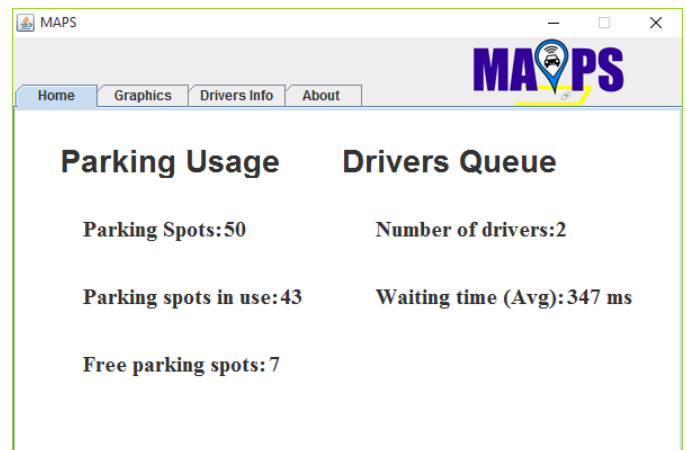


Figura 2: Aba Home

Na aba *graphics*, que pode ser visualizada na Figura 3, o usuário possui informações sobre o nível de ocupação do *smart parking*, através de um gráfico de barra (na Figura 3 mais de 3/4 (três quartos) das vagas estão ocupadas) e sobre a situação da fila de espera, através de um gráfico de linha

que relaciona o horário e quantidade de motoristas na fila (na Figura 3 o valor máximo de *drivers* na fila de espera foi de 5). Estas informações são apresentadas através de gráficos simples para que sejam facilmente verificadas.

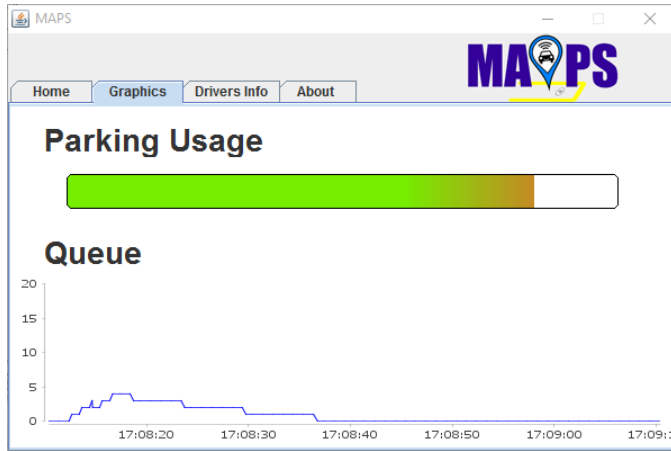


Figura 3: Aba Graphics

Na Figura 4 tem-se a aba *drivers info* que traz os valores máximo, mínimo e médio do tempo de permanência (*Parking time*) e também do grau de confiança (*Trust Degree*) dos motoristas que utilizam o *smart parking*. Essas informações são úteis para melhorias futuras, por exemplo se o valor médio do *trust* estiver alto pode ser um indício de que novos motoristas teriam dificuldade de conseguir vagas. Logo, essas informações podem ajudar o agente *Manager* a calibrar os valores para assegurar o bom funcionamento do SMA.

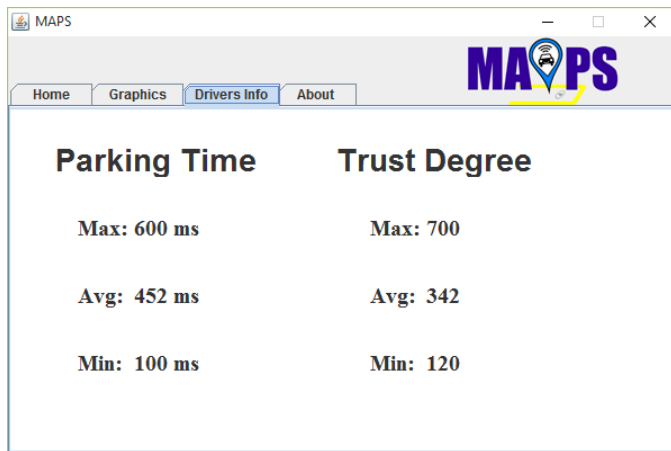


Figura 4: Aba Drivers Info

Para incorporar a interface desenvolvida em Java ao sistema MAPS foi criado o artefato *GUInterface*, utilizando o *Cartago*. Este artefato é responsável pela criação da interface gráfica, nele se encontram todas as operações responsáveis pela manipulação da mesma.

4. CONCLUSÃO

Este trabalho apresenta a implementação de uma interface gráfica voltada ao controle e gestão do MAPS. Isto é

uma evolução para o projeto, pois além de ajudar na administração do *Smart Parking* pode auxiliar na visualização e comparação de resultados gerados por simulações, o que é útil para testar novas extensões do projeto, como novos algoritmos para distribuição de vagas e gerenciamento da fila de espera dos agentes *drivers*. Com a interface gráfica é possível perceber facilmente se os *drivers* estão permanecendo pouco tempo no estacionamento, ou seja, se há grande rotatividade de vagas. Se isso ocorre, o especialista (responsável pela administração do *Smart Parking*) pode tentar alternativas, como reorganizar as vagas do estacionamento de forma a buscar um melhor uso dos recursos, no caso as vagas. Outra contribuição é a possibilidade de futuramente adaptar esta interface para a versão embarcada do projeto MAPS, assim conseguindo um acompanhamento do funcionamento do *Smart Parking* em diferentes plataformas.

O emprego do *framework* JaCaMo facilitou a implantação dos recursos aqui propostos, visto que o *framework* faz uso de artefatos em Cartago. Por sua vez o código usado em Cartago é basicamente a linguagem Java, isso facilitou a conexão da interface (implementada em Java) com o projeto MAPS.

Em relação aos trabalhos futuros é possível destacar os seguintes: (i) Avaliação de uso da interface; (ii) Criação de interface para dispositivos móveis. Até o momento a interface (aqui apresentada) não foi avaliada usando técnicas de usabilidade de interface. Logo, é necessário realizar a devida avaliação de uso, para aplicar eventuais alterações e ajustes. Além disso, uma das extensões do projeto MAPS diz respeito a criação de aplicativos para dispositivos móveis. E, portanto, será necessário criar uma interface para tal finalidade.

5. REFERÊNCIAS

- [1] M. Batty, K. Axhausen, G. Fosca, A. Pozdnoukhov, A. Bazzani, M. Wachowicz, G. Ouzounis, and Y. Portugali. *Smart Cities of the Future*. 2012.
- [2] L. F. S. d. Castro, G. V. Alves, and A. P. Borges. Utilização de grau de confiança entre agentes para alocação de vagas em um Smart Parking. 10^o Workshop-Escola de Sistemas de Agentes, seus Ambientes e aplicações. Maceió - AL. mai. 2016. 2016.
- [3] T. D. Huynh, N. R. Jennings, and N. R. Shadbolt. An integrated trust and reputation model for open multi-agent systems. 2006.
- [4] JACAMO. The JaCaMo approach. Disponível em <http://jacamo.sourceforge.net/?page_id=40>, 2011.
- [5] N. Jakob. *Usability Engineering*. 1993.
- [6] A. Koster, F. Koch, and A. L. Bazzan. Incentivising Crowdsourced Parking Solutions. 2014.
- [7] D. D. Nocera, C. D. Napoli, and S. Rossi. A Social-Aware Smart Parking Application. 2014.
- [8] G. Revathi and V. R. S. Dhulipala. *Smart Parking Systems and Sensors: A Survey*. 2012.
- [9] M. Wooldridge. *An Introduction to MultiAgent Systems*. J. Wiley, New York, 2nd edition, 2009.