

# Utilização de simulador para representação de estratégias de um agente inteligente no desvio de obstáculos

Vinicius Custodio  
Universidade Tecnológica Federal do Paraná  
Ponta Grossa, Brasil  
viniciuscustodio@alunos.utfpr.edu.br

Gleifer Vaz Alves  
Universidade Tecnológica Federal do Paraná  
Ponta Grossa, Brasil  
gleifer@utfpr.edu.br

## RESUMO

Os veículos autônomos são uma revolução para a indústria automotiva e estão cada vez mais próximos da realidade. A área de veículos autônomos só tende a crescer nos próximos anos e vem para modificar e ampliar a estrutura da sociedade em termos de como os veículos são utilizados e como são as interações com os usuários. O desenvolvimento requer a junção de diversas áreas, uma delas é a de agentes inteligentes. Existem diversas formas de se implementar um agente inteligente, uma delas é o framework chamado JaCaMo. E para que o agente consiga atuar será utilizado um ambiente gráfico, para esse fim o Simbad é utilizado, um simulador que permite a construção de ambientes. O ambiente definido para esse trabalho possui somente obstáculos estáticos, que estão definidos dentro de faixas, e o agente transita entre as faixas para desviar desses obstáculos. O simulador é responsável por fornecer as informações ao agente, de forma que esse seja capaz de controlar a sua representação dentro do simulador. Para que as informações fossem trocadas, um protocolo de comunicação foi definido, e permite que o agente receba informações sobre o ambiente e envie as ações que devem ser realizadas ao simulador.

## ABSTRACT

Autonomous vehicles are a revolution for the automotive industry and is already happening. The area of autonomous vehicles only tends to grow in the coming years and will modify and broaden the structure of society in terms of how vehicles are used and how are the interactions with users. Its development requires the combination of several fields, one of which is intelligent agents. There are several ways to implement a intelligent agent, one of them is a framework called JaCaMo. And for the agent to be able to act a graphic environment will be used, for this purpose Simbad was chosen. The environment defined for this work has only static obstacles, which are defined within ranges, and the agent transits between the tracks to avoid obstacles. The simulator is responsible for providing the information to the agent,

so he can analyze the situation and take some action when needed. In order for information to be exchanged, a communication protocol has been defined, and allows the agent to receive information about the environment and send the actions that must be performed to the simulator.

## Palavras-chave

Agentes inteligentes; Veículos autônomos; Simulação; Protocolo de comunicação

## 1. INTRODUÇÃO

A sociedade sempre busca aprimoramentos para os processos que são realizados diariamente. Uma tecnologia que está sendo alterada é a dos veículos. Nos últimos anos a indústria automobilística vem buscando otimizar o trânsito das cidades é facilitar a vida dos cidadãos através da criação dos veículos autônomos [9].

Os veículos atualmente sofrem de diversos problemas devido ao aumento do número de veículos nas cidades, esse aumento é ocasionado pela grande migração de pessoas dos ambientes rurais para os centros urbanos [5]. Alguns dos problemas que podem ser citados são: (i) o aumento do tráfego de veículos, resultando em congestionamentos [4]; (ii) dos acidentes no trânsito [6]; (iii) e da poluição emitida pelo grande número de veículos.

A criação dos veículos autônomos procura solucionar os problemas citados. A habilidade de se auto-dirigir permite que os veículos sejam utilizados em seu máximo, segundo Cliffe [2] atualmente os veículos são utilizados em somente 5% de sua capacidade, significando um desperdício de 95% do tempo útil do veículo. Com sistemas autônomos guiando os veículos, esperasse que esse seja possível utilizar ao menos 75% da capacidade de cada veículo através do compartilhamento de veículos.

O acréscimo da utilização dos veículos será reflexo dos novos modelos que surgiram com o mercado dos veículos autônomos. Um desses é o compartilhamento de veículos, funcionando de forma semelhante a um táxi. Ao ser requisitado o veículo se dirige ao passageiro e durante o percurso ele pode realizar outras atividades, aumentando assim suas horas produtivas que antes eram gastas dirigindo [9].

Para que um veículo possua capacidades autônomas, sua arquitetura deve compreender em dois níveis, o nível baixo correspondendo aos sensores e atuadores responsável por reconhecimento de objetos, mapeamento, navegação, direção, etc [8]. O nível alto corresponde aos componentes responsáveis muitas vezes pela tomada de decisão, inferindo quais são os sensores e atuadores que devem ser utilizados.

O estudo dessa arquitetura requer uma forma de representação, para os componentes de baixo e alto nível. Este trabalho foca na apresentação de um simulador para representar os componentes de baixo nível, e de um agente como componente de alto nível que realiza a tomada de decisão. O simulador escolhido para esse fim é o Simbad [7], e o framework para a implementação do agente é o JaCaMo [1].

O Simbad [7], permite a criação de ambientes e a adição de recursos como sensores e câmeras que podem ser acessados durante a simulação para a coleta de dados. O Simbad é um simulador graficamente simples, que é implementado em Java o que viabiliza a conexão com o framework JaCaMo. Mesmo simples, ele possui as seguintes funcionalidades como: controle sobre a velocidade, sensores para localização de obstáculos e câmeras para visualização do ambiente do ponto de vista do agente.

O JaCaMo é um framework que implementa agentes inteligentes utilizando a arquitetura BDI. Nessa arquitetura o agente é modelado através de crenças, desejos e intenções. As crenças são os conhecimentos do ambiente, essas informações são mapeadas a partir de percepções que o agente recebe. Os desejos são os objetivos que o agente procurar alcançar e as intenções são os desejos que o agente está buscando alcançar no momento, ou seja, existe um plano que está sendo executado para que o desejo seja satisfeito [1].

Com essas ferramentas, ambiente e agente foram implementados. O agente aqui definido, foi implementado para conseguir escolher entre as manobras definidas, dado um conjunto de percepções que o ambiente repassava ao agente. O agente possui duas manobras, manobra à direita ou à esquerda, para desviar dos obstáculos estáticos definidos no ambiente. Inicialmente a manobra possui uma rota definida, porém quando existem obstáculos dentro da rota inicialmente, essa rota pode ser alterada para que o agente possivelmente consiga desviar do obstáculo.

## 2. AGENTES E FRAMEWORK JACAMO

Um agente é definido como um sistema de computador que é capaz de realizar ações autônomas [11]. Neste trabalho o agente será responsável por realizar a tomada de decisão, e será implementado utilizando o framework JaCaMo [1], composto pelo Jason (implementa o agente), Cartago (implementa o ambiente) e Moise (implementa a organização dos agentes) e implementa a arquitetura de um agente racional. Esse tipo de agente é capaz de tomar a iniciativa, planejar e agir racionalmente e não baseados em eventos ocorridos no ambiente [10].

Neste trabalho não será utilizado o componente Moise pertencente ao JaCaMo. Para se trabalhar com a organização dos agentes é necessário pelo menos dois agentes. O fim deste é a criação de um agente que consiga se comportar dentro do cenário proposto, que considera um agente isolado no ambiente, sem interações com outros agentes.

## 3. SIMBAD

O Simbad é um simulador *open source* com fins científicos e educacionais. Focado principalmente para o desenvolvimento e estudo de Inteligência Artificial, Aprendizado de Máquina, Robôs e Agentes Autônomos [7].

O simulador é dividido em três partes: (i) responsável pela construção do ambiente e definição de quais serão as posições dos obstáculos; (ii) responsável por captar as informações do

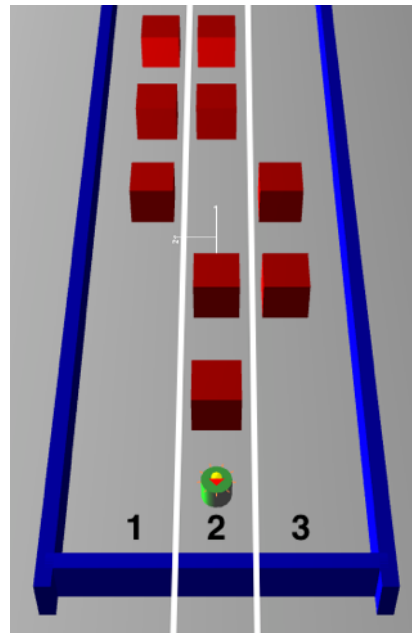


Figure 1: Exemplo de ambiente definido para esse trabalho

ambiente e repassar ao agente; (iii) receber as ações e então atuar no ambiente. Na Figura 1 tem-se um exemplo de um ambiente.

Para que as informações fossem trocadas entre o agente (JaCaMo) e o simulador (Simbad), foi necessário decidir um meio para que as mensagens fossem trocadas entre esses. Um protocolo de comunicação foi definido, criando assim um padrão de como as mensagens devem ser enviadas.

## 4. CONFIGURAÇÕES: AGENTE E AMBIENTE

Antes de criar as mensagens, foi necessário definir regras, que permitissem a criação de um grupo definido de ações. Para definir esse conjunto de decisões, o ambiente foi delimitado de acordo com as seguintes normas:

- Todo obstáculo é um cubo de 1m;
- O ambiente é organizado em 3 faixas;
- Cada obstáculo se localiza em somente uma das faixas;
- O veículo sempre se locomove pela faixa central;
- E uma manobra só é completada quando o agente retorna a faixa central;

Com essas regras o seguinte grupo de manobras foi definido:

- Desvio à esquerda: Manobra executada quando o agente deseja se locomover para à esquerda dele;
- Desvio à direita: Manobra executada quando o agente deseja se locomover para à direita dele;
- Retornar para direita: Manobra executada quando o percurso à esquerda está bloqueado, nesse caso tentar ir pela direita;

- Retornar para esquerda: Manobra executada quando o percurso à direita está bloqueado, nesse caso tentar ir pela esquerda;
- Manter a direita: Manobra executada quando o agente deseja permanecer à direita;
- Manter a esquerda: Manobra executada quando o agente deseja permanecer à esquerda.

## 5. INTERLIGAÇÃO: JACAMO E SIMBAD

A interligação foi realizada através do meio TCP, pensando em possíveis testes futuros onde o agente e a simulação não se encontrassem na mesma rede. As seguintes mensagens podem ser enviadas do agente para o simulador, cada mensagem corresponde a uma ação que deve ser executada dentro da simulação, quando essa é enviada.

- 0: Requisição para iniciar o simulador;
- 2: Informa que uma manobra a esquerda é necessária;
- 4: Informa que uma manobra a direita é necessária;
- 6: Informa que o veículo deve parar de se movimentar;
- 8: Informa que o veículo deve desfazer os passos atuais e tentar realizar outra manobra (caso 1 - esteja realizando a manobra pela direita, deve tentar proceder pela esquerda; caso 2 - quando está realizando a manobra pela esquerda, tentar proceder pela direita).
- 10: Informa que veículo deve desfazer os passos da manobra atual e tentar realizar as manobras oposta (direita -> esquerda, esquerda -> direita);
- 12: Informa que veículo deve desfazer o último passo realizado e continuar com a manobra.

E as seguintes mensagens podem ser enviadas do simulador para o agente. Essas mensagens correspondem a informações da simulação que serão utilizadas para que o agente consiga definir qual ação será executada.

- 1#x\_pos#y\_pos#z\_pos: Informa a posição do veículo na simulação;
- 3#x\_pos#y\_pos#z\_pos: Informa a posição do obstáculo;
- 5#obstacle\_left#obstacle\_center#obstacle\_right: Informa em quais posições existem obstáculos (três posições possíveis: esquerda, centro e direita);
- 7#maneuver#direction#distance\_to\_hit#actual\_x#actual\_y#actual\_z#last\_x#last\_y#last\_z: Informa a manobra atual, direção do veículo, posição atual (coordenadas x, y, z) e a última posição que o agente fez uma curva (x, y, z);
- 9#maneuver#direction#distance\_to\_hit#turn\_point#0actual\_x#actual\_y#actual\_z#last\_x#last\_y#last\_z: Informa a manobra atual, direção do veículo, próxima posição que o veículo pretende fazer uma curva, posição atual (coordenadas x, y, z) e a última posição que o veículo fez uma curva (x, y, z).

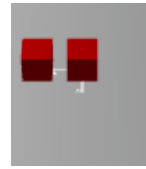


Figure 2: Cenário 1

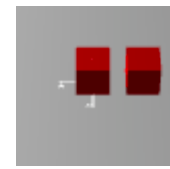


Figure 3: Cenário 2

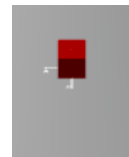


Figure 4: Cenário 3

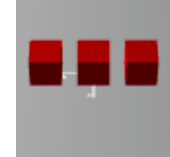


Figure 5: Cenário 4

Através dessas mensagens, é possível que o simulador repasse informações ao agente e vice-versa. Uma vantagem dessa implementação também é que quando for necessário incrementar tanto o simulador quando agente é possível adicionar novas mensagens ao protocolos para que esses continuem a se comunicar.

## 6. SIMULAÇÃO

Através das mensagens trocadas o simulador consegue interagir com o ambiente de acordo com as decisões do agente. Considerando a mensagem 5, que define se existe um obstáculo na posição, tem-se a seguintes possibilidades:

- true#true#false: Figura 2, para esse cenário a manobra a ser executada pelo agente é a direita, pois é a faixa desocupada.
- false#true#true: Figura 3, cenário oposto ao primeiro de forma que a manobra executada pelo agente é a esquerda, pois é a faixa desocupada.
- false#true#false: Figura 4, como ambas as faixas laterais estão desocupadas o agente sempre escolha a manobra à esquerda.
- true#true#true: Figura 5, nenhuma faixa está disponível e nenhum manobra evitará os obstáculos por isso o agente deixa de se locomover.

Para cada um desses cenários, a simulação possui a manobra implementada que é executada de acordo com a ordem do agente e que corresponde a uma rota que fará com que a representação do agente desvie do obstáculo.

## 7. CONCLUSÃO

A criação do protocolo de comunicação permite que dois componentes distintos se comuniquem utilizando o padrão de mensagens definido. Sendo assim para trabalho futuros, além do incremento dos próprios componentes, novas formas de implementação do agente e do simulador podem ser exploradas. Os únicos requisitos é que esses permitam o envio de mensagens através da rede e que ambos os componentes obedeçam as mensagens definidas.

A escolha de ambas ferramentas para o trabalho foi devido a suas implementações em Java, uma linguagem que possui mais adeptos em relação há outras. Ambas ferramentas não foram utilizadas em seus potenciais máximo. O agente ainda possui muito espaço para expansão, permitindo novos planos para tratar cenários mais complexos. O simulador, também ainda possui recursos a serem explorados como mais sensores e/ou câmeras que estão disponíveis, para capturar uma mais informações do ambiente.

Alem disso a interligação, que é feita através do protocolo permite a (i) substituição do simulador, para um mais realista. Sendo possível a reutilização da classe de comunicação e do agente, necessitando somente adaptar o simulador para que ele repasse as informações através do protocolo definido; (ii) substituição do agente, criando de um agente em outra linguagem, como por exemplo através do Gwendolen [3], que é uma linguagem para criação de agentes com a arquitetura BDI e que permite a verificação formal dos planos do agente.

Uma vantagem da criação do protocolo é a flexibilidade em relação quais serão os componentes que estarão utilizando o mesmo. Porém, esse flexibilidade vem ao custo de compreensão, quando novas mensagens começarem a surgir, com quantidades cada vez maiores de informação complica a utilização da interligação

Para a melhoria do interligação, a alteração do estilo das mensagens simples para uma configuração REST permitindo assim uma melhor organização das informações, pois este utiliza um padrão definido, como por exemplo JSON. JSON é uma forma de serializar as informações de objetos à pares com valores atribuídos e vetores e facilitando assim o próprio entendimento ao receber uma mensagem.

## 8. REFERÊNCIAS

- [1] O. Boissier, R. H. Bordini, J. F. Hübner, A. Ricci, and A. Santi. Multi-agent oriented programming with jacamo. *Sci. Comput. Program.*, 78(6):747–761, 2013.
- [2] M. Cliff. Driverless cars — the route to more than smart cities, 2016. Accessed, 22 May 2017.
- [3] L. A. Dennis and B. Farwer. *Gwendolen: A BDI Language for Verifiable Agents*. University of Aberdeen, 2008.
- [4] D. J. Fagnant and K. M. Kockelman. Preparing a nation for autonomous vehicles: Opportunities, barriers and policy recommendations. *Eno Center for Transportation*, 2, 2013.
- [5] U. Nations. *Department of Economic and Social Affairs: Population Division. World Urbanization Prospects: The 2014 Revision*. United Nations, 2015.
- [6] M. Peden, R. Scurfield, D. Sleet, D. Mohan, A. A. Hyder, E. Jarawan, and C. Mathers. *World report on road traffic injury prevention*. World Health Organization, 2004.
- [7] Simbad. Disponível em: <http://simbad.sourceforge.net>, 2017. Acesso em: 22 jan. 2017.
- [8] S. Ulbrich, A. Reschka, J. Rieken, S. Ernst, G. Bagschik, F. Dierkes, M. Nolte, and M. Maurer. Towards a functional system architecture for automated vehicles. *CoRR*, abs/1703.08557, 2017.
- [9] R. Wallace and G. Silber. Self-driving cars: The next revolution. 2012. Acesso em: 23 ago, 2017.
- [10] M. Wooldridge and N. R. Jennings. *Agent Theories, Architectures, and Languages: A survey*. 1994.
- [11] M. J. Wooldridge. *An Introduction to MultiAgent Systems (2. ed.)*. Wiley, 2009.