WEIT 2023: Minicurso

Model Checking na Verificação Formal de Sistemas Autônomos

Gleifer Vaz Alves (gleifer@utfpr.edu.br) - UTFPR (Universidade Tecnológica Federal do Paraná) Campus Ponta Grossa.

11 de outubro de 2023 - **WEIT 2023**

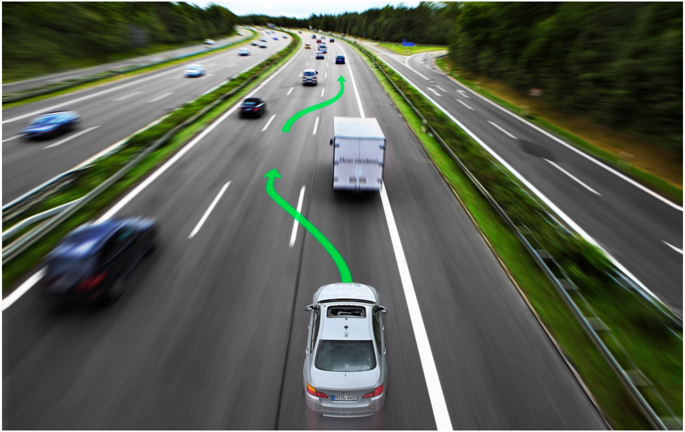
Sumário

Introdução

Desafios

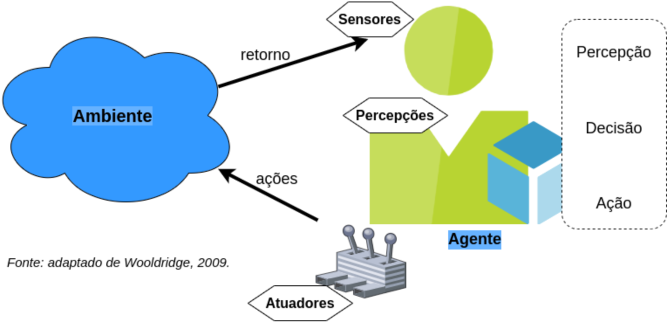
- O que é um sistema autônomo?
- Como assegurar que um sistema autônomo funciona corretamente?
- Uso de técnicas de Verificação Formal pode oferecer uma abordagem adequada e robusta para verificar se o comportamento do sistema autônomo ocorre como esperado.


nuTonomy
nuTonomy
RD
6096 A



Agentes e Sistemas Autônomos

- Agentes e Sistemas Multiagentes são utilizadas como uma abstração adequada para representar sistemas autônomos e seu respectivo comportamento.



Fonte: adaptado de Wooldridge, 2009.

https://www.mdpi.com/2224-2708/10/3/41

Journal of
Sensor and
Actuator Networks

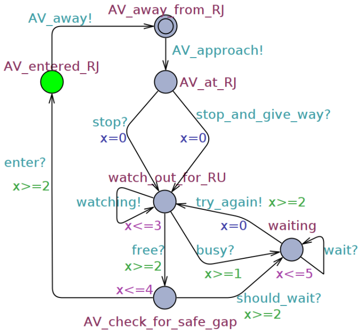
an Open Access Journal by MDPI

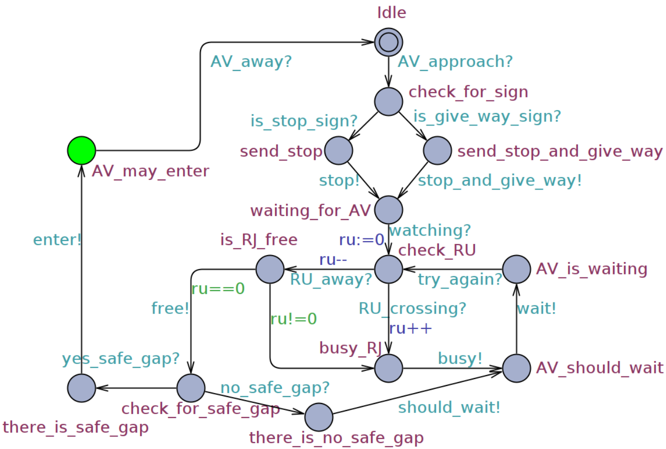
A Double-Level Model Checking Approach for an Agent-Based Autonomous Vehicle and Road Junction Regulations

Gleifer Vaz Alves; Louise Dennis; Michael Fisher

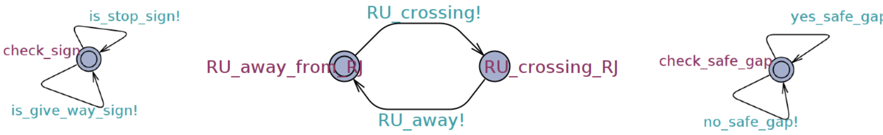
J. Sens. Actuator Netw. 2021, Volume 10, Issue 3, 41

- Utilização de Model Checking em tempo de design e de execução.

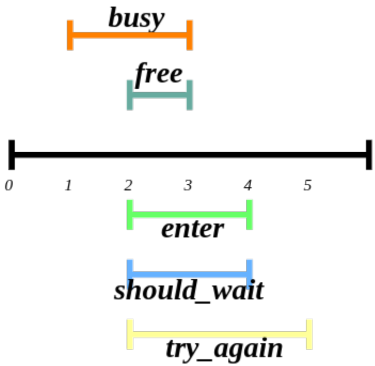


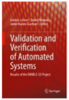

Idle
AV_away?
AV_approach?
check_for_sign
is_stop_sign?
is_give_way_sign?
send_stop
send_stop_and_give_way
AV_may_enter
stop!
stop_and_give_way!
waiting_for_AV
watching?
enter!
is_RJ_free
ru:=0
check_RU
ru--
AV_is_waiting
RU_away?
try_again?
ru==0
free!
ru!=0
RU_crossing?
wait!
ru++
busy_RJ
busy!
yes_safe_gap?
AV_should_wait
no_safe_gap?
check_for_safe_gap
should_wait!
there_is_safe_gap
there_is_no_safe_gap

Aplicações


is_stop_sign!
check_sign
is_give_way_sign!
RU_crossing!
RU_away_from_RJ
RU_crossing_RJ
RU_away!
check_safe_gap
no_safe_gap!

Aplicações





Validation and Verification of Automated Systems pp 105-117 | Cite as

Reliable Decision-Making in Autonomous Vehicles

Authors Authors and affiliations

Gleifer Vaz Alves, Louise Dennis, Lucas Fernandes, Michael Fisher

Chapter

First Online: 11 November 2019

Abstract

The use of Autonomous Vehicles (AVs) on our streets is soon to be a reality; increasingly, interacting with such AVs will be part of our daily routine. However, we will certainly need to assure the reliable behaviour of an AV, especially when some unexpected scenarios (e.g. harsh environments, obstacles, emergencies) are taken into account. In this article we use an *intelligent agent* approach to capture the high-level decision-making process within an AV and then use formal verification techniques to automatically, and strongly, analyse the required behaviours. Specifically, we use the MCAPL framework, wherein our core agent is implemented using the GWENDOLEN agent programming language, and to which we can apply model checking via the AJPF model checker. By performing such formal verification on our agent, we are able to prove that the AV's decision-making process, embedded within the GWENDOLEN agent plans, matches our requirements. As examples, we will verify (formal) properties in order to determine whether the agent behaves in a reliable manner through three different levels of emergency displayed in a simple urban traffic environment.

Work partially supported by EPSRC projects EP/L024845 and EP/M027309.

https://link.springer.com/article/10.1007/s10462-023-10596-z

Published: 22 September 2023

A middleware for providing communicability to Embedded MAS based on the lack of connectivity

Vinicius Souza de Jesus, Nilson Mori Lazarin, Carlos Eduardo Pantoja, Fabian César Pereira Brandão Manoel, Gleifer Vaz Alves & José Viterbo

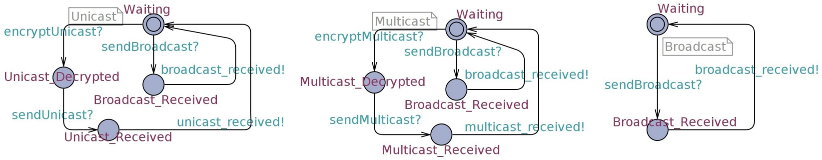
Artificial Intelligence Review (2023) | Cite this article

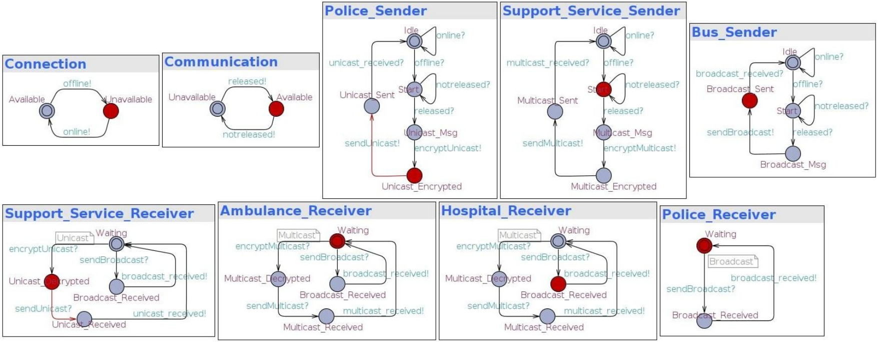
Abstract

An Embedded multi-agent system (Embedded MAS) is an embedded cognitive system based on agents cooperating to control hardware devices. These agents are autonomous and proactive entities capable of decision-making and can constantly acquire new knowledge via interaction with other agents and the environment. Since the interaction between agents is relevant for acquiring new knowledge, issues such as the communicability and mobility of agents from different Embedded MAS must be highlighted. The classification of a MAS as Open or Closed only considers the mobility of agents, but communicability also needs to be considered. For this, we extend the notion of openness in these systems to consider the existence of Totally Closed and Limited Open MAS, to consider agents from an Embedded MAS

Aplicações

- Utilização do UPPAAL na verificação formal de protocolo comunicação entre agentes embarcados.




Connection
offline!
Available
Unavailable
online!
Communication
released!
Unavailable
Available
notreleased!
Police_Sender
Idle
online?
unicast_received?
offline?
Unicast_Sent
Start
notreleased?
released?
Unicast_Msg
sendUnicast!
encryptUnicast!
Unicast_Encrypted
Support_Service_Sender
Idle
online?
multicast_received?
offline?
Start
notreleased?
Multicast_Sent
released?
Multicast_Msg
sendMulticast!
encryptMulticast!
Multicast_Encrypted
Bus_Sender
Idle
online?
broadcast_received?
offline?
Broadcast_Sent
Start
notreleased?
sendBroadcast!
released?
Broadcast_Msg
Support_Service_Receiver
Unicast
Waiting
encryptUnicast?
sendBroadcast?
Unicast_Decrypted
broadcast_received!
Broadcast_Received
sendUnicast?
unicast_received!
Unicast_Received
Ambulance_Receiver
Multicast
Waiting
encryptMulticast?
sendBroadcast?
Multicast_Decrypted
broadcast_received!
Broadcast_Received
sendMulticast?
multicast_received!
Multicast_Received
Hospital_Receiver
Multicast
Waiting
encryptMulticast?
sendBroadcast?
Multicast_Decrypted
broadcast_received!
Broadcast_Received
sendMulticast?
multicast_received!
Multicast_Received
Police_Receiver
Waiting
Broadcast
broadcast_received!
sendBroadcast?
Broadcast_Received

https://c3.furg.br/weit/

Modelagem formal de abordagens éticas para comportamento de agentes

João Vicente Markovicz[1], **Gleifer Vaz Alves**[1]

[1]Departamento Acadêmico de Informática
Universidade Tecnológica Federal do Paraná (UTFPR)
Ponta Grossa – PR – Brasil

joao.080603@alunos.utfpr.edu.br gleifer@utfpr.edu.br

Abstract. *This paper formally models three ethical approaches (deontology, consequentialism and virtues ethics) in agents' decision-making. For that, we have used the UPPAAL model checker for creating (timed) automata and verifying properties related to the agent's behaviour.*

Guia de Instalação

Guia de Instalação do UPPAAL

- Acessar o guia no repositório:

  `https://github.com/laca-is/uppaal`

Conceitos Básicos

Conceitos relacionados

- Autômato Temporal.
- Lógica Temporal.

  LTL, CTL, TCTL.

Utilização do UPPAAL

- Desenvolvido em parceria entre as universidades:
- **Upp**sala (Suécia).
- **Aal**borg (Dinamarca).
- Existem algumas extensões da ferramenta para finalidades específicas:
  - **Cora** para otimização.
  - **Tron** para testes de sistemas de tempo-real.
  - **Cover** para testes de otimização.
  - **Tiga** para Jogos.
  - **Port** para componentes temporais.
  - **SMC** para Model Checking Estatístico.

Weights

The weight over branch is a constant non-negative integer expressions denoting the probabilistic likely-hood of the branch being executed. The probability of a particular branch is determined as a ratio of its weight over the sum of weights of all branches emanating from the same branch node.

The weights are used in probabilistic and statistical model checking .

Example

Select: `i : int[0,3]`
Update: `gen_data(i)`
Weight: `w[i]`

Visão Geral da Ferramenta

- Editor de Modelos
- Simulação de Modelos
- Verificação de Propriedades
- Declaração de Variáveis
  - linguagem com sintaxe similar a C, C++.
- Ainda é possível utilizar a ferramenta diretamente pelo terminal usando:

```
verifyta
```

Programação: funções

```
Function        ::= [Type] [ID] '('
[Parameters] ')' Block
Block           ::= '{'
LocalDeclaration* Statement* '}'
LocalDeclation  ::= TypeDeclaration |
VariableDeclaration
Statement       ::= Block
                 | ';'
                 |  [Expression] ';'
                 |  ForLoop
                 |  Iteration
                 |  WhileLoop
                 |  DoWhileLoop
                 |  IfStatement
                 |  ReturnStatement

ForLoop         ::= 'for' '('
[Expression] ';' [Expression] ';'
[Expression] ')' Statement
Iteration       ::= 'for' '(' [ID]
':' [Type] ')' Statement
WhileLoop       ::= 'while' '('
[Expression] ')' Statement
DoWhile         ::= 'do' Statement
'while' '(' [Expression] ')' ';'
IfStatment      ::= 'if' '('
[Expression] ')' Statement [ 'else'
Statement ]
ReturnStatement ::= 'return' [
[Expression] ] ';'
```

https://docs.uppaal.org/language-reference/system-description/declarations/functions/

Especificação de Propriedades

```
SymbolicQuery ::=
         'A[]' Expression Subjection
       | 'E<>' Expression Subjection
       | 'E[]' Expression Subjection
       | 'A<>' Expression Subjection
       | Expression --> Expression
Subjection

       | 'sup' ':' List Subjection
       | 'sup' '{' Expression '}' ':'
List Subjection

       | 'inf' ':' List Subjection
       | 'inf' '{' Expression '}' ':'
List Subjection

List ::= Expression | Expression ','
List

Subjection ::=
         // empty for no subjection
       | under StrategyName
```

https://docs.uppaal.org/language-reference/requirements-specification/

Exemplos de Sistemas Autônomos

- Os exemplos dessa seção encontram-se no seguinte repositório:
  `https://github.com/laca-is/uppaal`

Verificação Formal de Sistemas Autônomos

Especificação usando lógica temporal

- Os exemplos dessa seção encontram-se no seguinte repositório: `https://github.com/laca-is/uppaal`

Referências

- UPPAAL website
  http://www.uppaal.org/
- Frits Vaandrager
  Chapter 1
  A First Introduction to Uppaal

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CAMPUS PONTA GROSSA

Model Checking na Verificação Formal de Sistemas Autônomos

Gleifer Vaz Alves

- gleifer@utfpr.edu.br
- https://sites.google.com/view/gleifer
- **LaCA - Intelligent Systems**
  https://laca-is.github.io/

- 11 de outubro de 2023 - WEIT (VII Workshop-Escola de Informática Teórica).

Obrigado! Dúvidas?

- gleifer@utfpr.edu.br
- `https://sites.google.com/view/gleifer`
- **LaCA - Intelligent Systems**
  `https://laca-is.github.io/`