Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Towards Increased Road Safety: Real-Time Decision Making for Driverless City Vehicles

Andrei Furda and Ljubo Vlacic
Intelligent Control Systems Laboratory (ICSL)
Institute of Integrated and Intelligent Systems
Griffith University, Brisbane, Australia
a.furda@griffith.edu.au
l.vlacic@griffith.edu.au

***Abstract*—This work elaborates on the topic of decision making for driverless city vehicles, particularly focusing on the aspects on how to develop a reliable approach which meets the requirements of safe city traffic. Decision making in this context refers to the problem of identifying the most appropriate driving maneuver to be performed in a given traffic situation. The overall decision making problem is decomposed into two consecutive stages. The first stage is safety-crucial, representing the decision regarding the set of feasible driving maneuvers. The second stage represents the decision regarding the most appropriate driving maneuver from the set of feasible ones. The developed decision making approach has been implemented in C++ and initially tested in a 3D simulation environment and, thereafter, in real-world experiments. The real-world experiments also included the integration of wireless communication between vehicles.**

***Index Terms*—driverless city vehicles, decision making**

I. INTRODUCTION

Driverless city vehicles for civilian, non-military applications are not likely to gain a wide public acceptance unless they prove to be safer than conventional human-driven vehicles. Therefore, road safety is the highest objective in the effort of developing such vehicles, as is the case for any transportation system.

The correctness of driverless vehicles's control software is one of the crucial requirements to ensure safety. As a hard real-time (mission-critical) system, correctness refers not only to functional correctness, but also includes temporal correctness, i.e. the control software needs to produce correct results within specified time intervals. In order to achieve this objective, the design and implementation of the control software for driverless vehicles needs to be focused on enabling the proof of correctness through verification and validation procedures.

Figure 1 shows a simplified structure of the driverless vehicle control software. It consists of the following software modules: the World Model, the Real-Time Decision Making & Driving Maneuver Control, and the Vehicle Interface.

The World Model represents the driverless vehicle's view of its road environment. This software module merges a priori given information (e.g. known intersections loaded from an XML file) with traffic features which are perceived during the vehicle's movement (e.g. dynamic obstacles). The data contained in the World Model is constantly updated in real-time. The main purpose of the World Model is to provide other modules, such as the *Real-Time Decision Making & Driving Maneuver Control* module, with accurate information about the vehicle's surrounding environment.

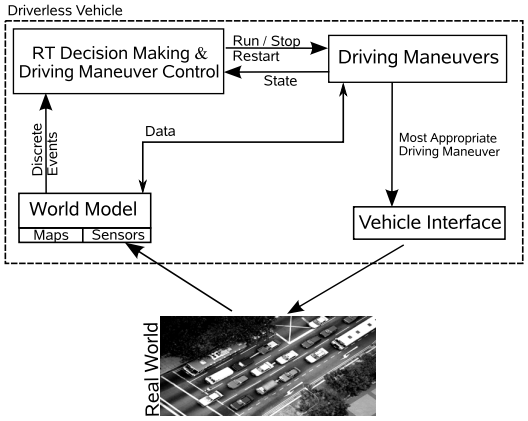

Fig. 1. Simplified view of the driverless vehicle control software architecture.

The Real-Time Decision Making & Driving Maneuver Control module represents the system's brain. Based on the information received from the World Model, this module makes real-time decisions about the activation of the most appropriate driving maneuver. Each driving maneuver is a closed-loop control algorithm, able to maneuver the vehicle in a specific traffic situation. This module directs its output to the *Vehicle Interface* module.

Within the driverless vehicle's control software, the Real-Time Decision Making & Driving Maneuver Control module plays a major safety-critical role, and therefore requires ensured operational correctness. In order to be able to ensure (i.e. to prove) the correctness of this module, testing alone is not sufficient [1].

The topic of real-time decision making for driverless city vehicles in the context of the road safety is a relevant research topic, which requires further research. This work addresses this topic, and the remainder of this paper is structured as follows. Section II elaborates on the problem definition and decomposition, Sections III and IV present solutions for the decision stages 1 and 2, respectively. Section V presents relevant implementation aspects and test results, and Section VI concludes this work.

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

II. DEFINITIONS AND APPROACH

Decision making in this context refers to the problem of identifying the most appropriate driving maneuver to be performed under the given road traffic circumstances.

The currently proposed solutions to the real-time decision making problem for driverless city vehicles are not sufficient to successfully deal with the high complexity of real-world, non-simplified city traffic conditions. This work aims at closing this gap by addressing this topic in a systematic way, proposing a solution which enables civilian driverless city vehicles to deal with real-world city traffic.

A. Solution Requirements, Decision Making Input and Output

1) Decision Making Requirements: The specification of requirements is a fundamental part of the development cycle of any complex system, in order to enable its verification and validation [2].

Regarding the development of the real-time decision making module, the specification of requirements is crucial. Without a specification for this module, the implementation, and eventually human lives will depend on the software developers' interpretation of requirements, without the possibility to prove the functional correctness.

A discussion on how to develop a correct, unambiguous and complete specification for the decision making module for driverless city vehicles is beyond the scope of this work. However, without the loss of generality, it can be assumed that such a decision making specification requires the decision making module to respond to certain traffic conditions in a defined way, i.e. certain traffic conditions imply the execution of certain driving maneuvers.

The occurrence of any traffic condition is modeled in this work by a set of discrete events. These discrete events are sent to the decision making module in the form of World Model Events. Depending on a defined set of discrete events, the specification defines which driving maneuvers can be performed safely. Table I shows examples of such discrete events. A detailed and complete specification would probably lead to a table containing thousands of such discrete events and driving maneuvers.

Consequently, the decision making module is modeled as an event-driven system [3]. Having in mind the complexity of city traffic, a complete specification for a typical decision making module will consist of a large number of defined inputs (i.e. discrete events), and a large number of outputs (i.e. driving maneuvers), which are logically interrelated in a complex way.

TABLE I
EXAMPLES OF DISCRETE EVENTS OF RELEVANCE.

Discrete Event	Explanation
w_1: stopped vehicle in front	Stopped vehicle was detected in front.
w_2: oncoming lane free	The oncoming lane is obstacle free.
w_3: pedestrian in front	Pedestrian crossing the road.
⋮	⋮

2) Input 1 - The World Model: The *World Model* represents the driverless vehicle's view of its road environment. Therefore, it represents an input to the decision making module. The World Model merges a priori given information (e.g. road infrastructure) with traffic features which are perceived during the vehicle's movement (e.g. dynamic obstacles). The data contained in the World Model is constantly updated in real-time and provided in two forms: as World Model Events, and additionally in the form of an object-oriented structure.

A *World Model Event* is defined in this work as a discrete event implemented as a boolean variable. A World Model Event represents the availability or state of certain information generated by the World Model.

The purpose of World Model Events is to notify other software modules about the state of certain predefined conditions, which are relevant for the execution of driving maneuvers. Such conditions can be, for instance, related to traffic situations (e.g. "pedestrian in front of the vehicle"), but might as well be related to the availability of information (e.g. "GPS localization available").

The state of all defined World Model Events is provided as a k-tuple:

$$WM_{events} = (w_1, w_2, ..., w_k) : w_l \in \{true, false\}, \quad (1)$$

where each element w_l $(l = 1, 2, .., k)$ represents a discrete event.

The k-tuple WM_{events} is updated cyclicly, and the World Model notifies other software modules about the occurrence of certain conditions as defined for each event.

In addition to World Model Events, full access is provided to all data stored in the World Model, through an object-oriented data structure. This data structure contains more detailed information, such as exact distances to obstacles. However, a detailed discussion about the World Model structure is beyond the scope of this work.

3) Input 2 - The Route Planner: In many situations, the planned route has a significant impact on decision making. For instance overtaking a slower vehicle just before a planned turn may not be adequate. Consequently, the route planner represents the second input to the decision making module.

This work assumes that a route planner provides the decision making module with a minimum amount of information about the planned route. Without the loss of generality, the provided route planner information can be defined as one element of the set $D_{route} = \{forward\ straight, forward\ right, forward\ left, turn\ around\}$, where each element indicates the future travel direction. If required, this set can be further extended.

4) Output - The Driving Maneuvers: The decision making module decides about the most appropriate driving maneuver. Therefore, the output of the decision making module is defined as one element of a driving maneuver set. In order to enable the decision making algorithm to deal with different types of driving maneuvers, each performing a different driving task, it is crucial to define a general structure for driving maneuvers.

Driving maneuvers are defined in this work as closed-loop control algorithms, each capable of maneuvering the driverless vehicle over a time period or distance. All driving maneuvers are structured in a common way. Their operational behaviors are modeled as deterministic finite automata [3], [4].

The states, input symbols and transition function are as follows (Figure 2):

- a *start state* q_0,
- a set of *Run states* $Q^{run} = \{q_1^r, q_2^r, ..., q_n^r\} \subset Q$,
- two final states $\{q_F, q_E\} = F$,
- a set of input symbols Σ, which consists of at least the symbols: $Run, Stop, Restart, Error$, and
- the state transition function $\delta : Q \times \Sigma \rightarrow Q$.

The *start state* q_0 is the *waiting* or *idle* state, in which the automaton is waiting for the *Run* signal.

The Run states $q_1^r, q_2^r, ..., q_n^r$, each representing a phase of a driving maneuver, perform the maneuvering of the vehicle. Each of them includes checking of preconditions, such as the availability of required information and safety conditions. As long as the defined preconditions are met, the Run states execute closed-loop control algorithms. A driving maneuver finishes in one of the *final states* q_F (finished) or q_E (error). While q_F signals a successful completion of the driving maneuver, the error state q_E indicates its incompletion due to an error or some other reason.

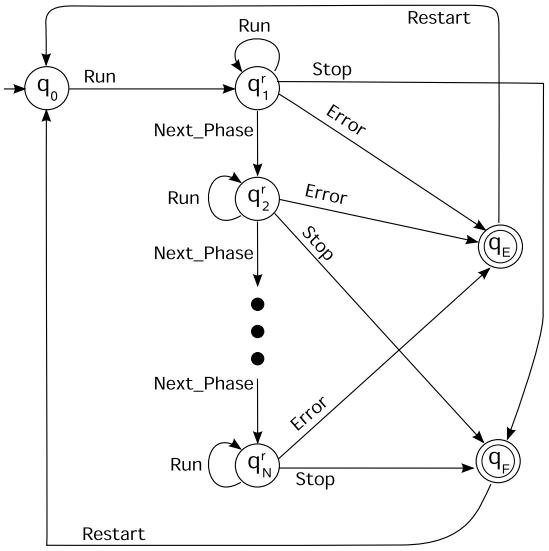

Fig. 2. General structure of a driving maneuver finite automaton. Each Run state $q_i^r, i = 1, 2, ...N$ represents a phase of a driving maneuver.

B. Decision Making Problem Definition

The decision making problem can be specified as follows. The following is given:

- a set $M_{all} = \{m_1, m_2, ...m_n\}$, $n \in \mathbb{N}$, of all available driving maneuvers which can be performed by the driverless vehicle (section II-A4),
- a k-tuple $(w_1, w_2, ..., w_k) \in W_{events}$ of World Model Events (section II-A2), $w_l \in \{0, 1\},\ l = 1, 2, ..., k$,
- a route planner direction indication $d_i \in D_{route} = \{forward\ straight, forward\ right, forward\ left, turn\ around\}$ (section II-A3).

The general problem of decision making in this context is to identify the most appropriate driving maneuver $m_{most\_appr.} \in M_{all}$, which leads to a driverless vehicle driving behavior conforming to the specification.

C. Problem Decomposition

The general decision making problem is decomposed into the following two consecutive stages:

1. First Stage: Decision regarding *feasible driving maneuvers* subject to World Model Events and route planner indication. A driving maneuver is defined as feasible if it can be safely performed in a specific traffic situation, and is conforming to the road traffic rules. In any traffic situation, there can be multiple feasible driving maneuvers (e.g. overtaking a stopped vehicle or waiting for it to continue driving).
2. Second Stage: Decision regarding *the* most appropriate driving maneuver. This stage selects and starts the execution of one single driving maneuver, which is the most appropriate for the specific traffic situation.

The decomposition into two stages leads to subproblems with manageable complexity, enabling the verification and testing of each stage in particular. Furthermore, different solution algorithms can be implemented for a specific stage, enabling the testing of a variety of algorithm combinations. However, this is beyond the scope of this work.

D. General Requirements

The following minimal list of general requirements is crucial for a solution of the real-time decision making problem for driverless city vehicles. These general requirements are fundamental for developing a solution which is usable for real-world applications.

- *Real-Time Capable*: Being part of a mission-critical real-time system, one of the main requirements of the decision making solution is to deliver correct decisions within specified time limits.
- *Explainable Results*: Safety-relevant systems, such as driverless vehicles, need to be analyzed, tested, and verified in order to ensure their correct operation. Consequently, one of the goals is to obtain results (i.e. driving decisions), which are explainable and reasonable.
- *Verifiable*: The verification, i.e. "the systematic approach to proving the correctness of programs" [5], is one of the most important aspects for safety-critical, hard real-time systems [1], such as driverless city vehicles. As solution approaches which are based on formal methods lead in general to an easier verification process [6], this work applies a formal method, namely Petri Nets.

- *Scalable*: The decision making algorithm should be scalable, having in mind the future integration of new information sources about the vehicle's environment, such as new sensors or information resulting from cooperation between driverless vehicles or communication with a vehicle management centre. Furthermore, the solution should allow the integration of additional driving maneuvers or traffic rules, without requiring significant changes or redesign of existing source code.
  However, a detailed discussion and analysis of these requirements would go beyond the scope of this work.

III. FIRST DECISION MAKING STAGE

A. General Approach

The goal of this stage is to select the feasible driving maneuvers, i.e. the subset of all driving maneuvers, which can be performed without putting any traffic participants at risk.

The following aspects are relevant for the selection of feasible driving maneuvers:

- Information about the vehicle's environment, which is provided in the form of World Model Events.
- Knowledge about traffic rules and compliance with them.
- Information about the planned travel direction, which is assumed to be provided by the route planner.

Figure 3 shows the processing steps for the selection of feasible driving maneuvers. Each driving maneuver requires certain world model information, which is provided by the World Model in the form of events. Therefore, occurring World Model Events define which driving maneuvers are operational (i.e. which can be performed). In order to comply to traffic rules, additional restrictions of driving maneuvers are required. In this work, the knowledge about traffic rules is embedded in the first step of this stage (DMU1A).

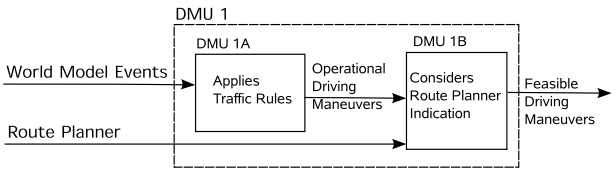

Fig. 3. General overview of the decision making unit for the selection of feasible driving maneuvers.

As a third aspect, the planned traveling route plays a further role in reducing the number of candidate driving maneuvers (DMU1B in Figure 3). Driving maneuvers which lead the vehicle into a wrong direction with respect to the planned route are omitted from the set of feasible driving maneuvers.

Consequently, the large number of factors to be considered in this decision stage require a model which enables the design and analysis of a highly complex operational behavior. As Petri Nets are a suitable modeling method for this purpose [7], this work uses Petri Nets to model this decision stage. This is elaborated in the following subsection.

B. Petri Net Model

In this work, the decision making unit shown in Figure 3 is modeled as a Petri Net (Figure 4). The Petri Net consists of two subnets, each modeling the decision making units DMU1A and DMU1B, respectively.

The structure of the Petri Net modeling DMU1 (Figure 4) is as follows. The input to the Petri Net consists of two sets of input places. The first set of input places represents World Model Events, the second set of input places represents the route planner indication. The output of the Petri Net modeling DMU1 represents driving maneuvers. There is one output place of the Petri Net for each available driving maneuver.

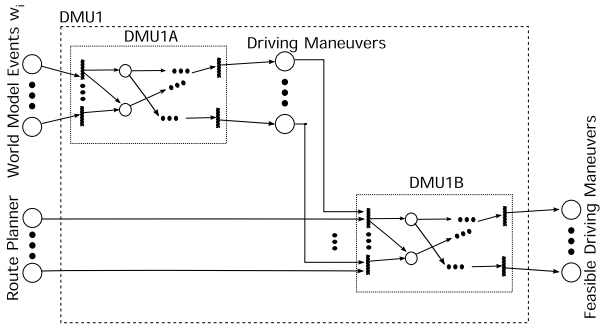

Fig. 4. Petri Net model of the decision making unit deciding about the feasible driving maneuvers. This stage consists of two steps (subnets): DMU1A and DMU1B.

In each execution cycle the World Model and the Route Planner mark the Petri Net's input places. A token is placed by the World Model into each place which represents a World Model Event which has the value *true*. Similarly, the Route Planner marks the input places which correspond to the planned travel direction.

After the execution of the complete Petri Net DMU1, only those Petri Net output places representing feasible driving maneuvers (i.e. operational and according to the route planner) are marked. After each decision making cycle, all marked places of the Petri Net are cleared, and a new decision making cycle begins.

IV. SECOND DECISION MAKING STAGE

This second stage of the decision making process selects and activates the most appropriate driving maneuver from the set of feasible ones. Because the set of feasible driving maneuvers only contains those maneuvers that can be safely performed in the specific road traffic situation, this stage is not safety-relevant, and its main objective is to maximize the efficiency and comfort.

In order to meet these non-safety-relevant objectives, a variety of approaches can be applied in this decision stage, including for instance priority-based or heuristic approaches. Our current implementation and test results of the decision making module is based on fixed priorities for each driving maneuver. In order to increase efficiency in the experimental tests, those driving maneuvers which move the vehicle faster

towards the destination have a higher priority (e.g. overtaking is preferred to platooning, if both maneuvers are feasible).

V. IMPLEMENTATION AND TEST RESULTS

The developed solution approach has been implemented and integrated into the control software for an experimental driverless vehicle. The identical control software is used for both, on-road testing and a 3D simulation environment. Figure 5 shows the graphical user interface of the decision making module.

A. Implementation Approach and Discussion

The selection of feasible driving maneuvers is the first stage of the decision making module. It consists of a Petri Net, which models the selection of driving maneuvers depending on World Model Events and Route Planner indication.

In order to decouple the decision making logic from the C++ implementation, the Petri Net structure is loaded from an XML (Extensible Markup Language) file into an object-oriented Petri Net structure, which is implemented in C++.

Having the Petri Net structure in an external XML file brings a variety of benefits, such as the possibility to make changes to the Petri Net structure without the need to make source code changes and to recompile the control software. Furthermore, this approach enables the design, simulation, analysis and verification of an eventually very complex Petri Net structure using already available Petri Net analysis tools.

As the number of Petri Net places and transitions is constant and does not include cycles, the execution of the Petri Net, and therefore the execution time of this decision making stage can be analyzed, in order to ensure that it meets the real-time requirements.

B. Simulation

A 3D simulation software has been used to test the presented decision making approach. The 3D simulation software includes the simulation of a driverless vehicle and its on-board sensors, such as GPS and LIDAR sensor. The vehicle interface for the simulated vehicle is identical to the vehicle interface for a real experimental vehicle. Furthermore, the 3D simulation environment is able to simulate static and dynamic obstacles, such as buildings, parked cars, pedestrians and other moving vehicles. Details about the simulation software have been published in [8]. The simulated traffic environment is identical to the real traffic environment at the test location for the real experimental vehicle at Griffith University, Nathan campus. Therefore, the simulation results reflect real road traffic situations.

C. Real-World Experiments

Experimental tests have been carried out using three vehicles, namely a driverless vehicle (Cycab, manufactured by Robosoft, France), a manually driven Cycab and a conventional car. All vehicles, sensors, and test facilities haven been provided by the French research institute INRIA (team IMARA).

All vehicles, including the conventional car, were equipped with Differential GPS (DGPS) and were able to exchange data

Fig. 5. Graphical user interface of the decision making module.

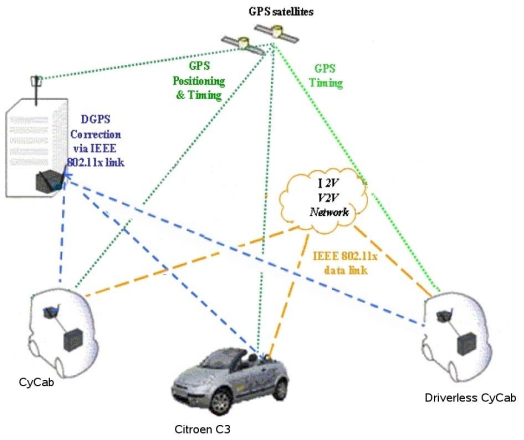

Fig. 6. The wireless communication setup (adapted from [9]).

over an ad-hoc wireless network. Figure 6 shows the wireless communication setup used in the experiments. Each vehicle receives DGPS correction signals from a base station over the wireless network. Furthermore, each vehicle sends its own position and speed to other vehicles, and each vehicle can receive this data from all others. The communication framework includes the possibility to integrate a traffic management centre, however this was not part of the experimental setup.

In our experiments with three vehicles, the driverless vehicle was able to receive the GPS positions of the other two vehicles. Furthermore, the driverless vehicle's world model included a priori information, such as the position of intersections and positions of imaginary stop signs.

In order to test the decision making approach, the three traffic scenarios have been set up, all showing a common decision situation: passing a stopped vehicle under different traffic conditions.

In the first traffic scenario, the driverless vehicle approached

a stopped vehicle. The distance to the next intersection was sufficient to enable safe passing, and the oncoming traffic lane was free of any obstacles.

In this first scenario, the driverless vehicle immediately started the passing maneuver when it approached the stopped vehicle.

The second traffic scenario was similar to the first, however another manually driven vehicle was oncoming, making safe passing impossible (Figure 7). In this second scenario, the driverless vehicle waited behind the stopped vehicle, and started passing the stopped vehicle when the oncoming traffic lane was free.

In the third traffic scenario, a manually driven vehicle was stopped at an intersection. In this third traffic scenario, the driverless vehicle waited behind the stopped vehicle until it crossed the intersection. Then the driverless vehicle continued driving, stopped at the imaginary stop sign and then continued crossing the intersection.

Fig. 7. Second traffic scenario: passing the stopped car is not possible due to an oncoming vehicle. After waiting until the oncoming lane is free, the driverless vehicle passes.

In the conducted experiments, the decision making module was repeatedly able to make the correct driving decision with respect to passing. Although it was not possible to continuously drive the vehicle for a longer distance due to a lack of adequate driving maneuvers, regarding only the correct decision making results made in real-time, it can be concluded that the developed decision making approach is capable of dealing with real-world traffic conditions in real-time.

D. Remaining Challenges

Although a prototype implementation of the developed decision making approach was successfully tested in both a 3D simulation and real-world tests with experimental vehicles, a number of issues remain to be addressed, before such vehicles can operate safely without human supervision. Only a few are addressed at this point:

- Adequate and reliable sensor systems need to be developed, which provide sufficient information in any light and weather condition. Without reliable World Model information, the decision making module cannot operate safely.
- The robustness, reliability, and correct operation of hardware systems and software components (i.e. sensors, computers, operating systems, etc.), which are integrated in such a driverless vehicle needs to be ensured.

VI. CONCLUSION

This work has addressed the topic of real-time decision making for driverless city vehicles, particularly focusing on the aspects on how to develop a reliable and safe approach which meets the requirements of city traffic. After specifying definitions and requirements, the problem of decision making was decomposed into two decision making stages.

The first decision making stage is based on discrete events in order to make the decision regarding the set of feasible driving maneuvers. In order to enable the formal algorithm verification of this safety-critical decision making stage, this work has presented a solution based on Petri Nets. Besides meeting the elaborated general algorithm requirements, the developed Petri Net based approach is suitable to enable the modeling of a large number of factors and the highly complex operational behavior.

The achieved simulation and real-world test results indicate that the presented decision making approach is able to meet the software requirements for safe on-road operation of driverless vehicles.

ACKNOWLEDGMENT

We would like to thank INRIA's team IMARA for the financial support provided towards conducting the experimental work at their test track in Rocquencourt, France. We are particularly grateful to Dr. Michel Parent, Laurent Bouraoui and Francois Charlot for their effort and assistance in performing the experiments.

REFERENCES

[1] P. A. Laplante, *Real-time systems design and analysis*. IEEE Press, IEEE Computer Society Press, 1997.
[2] R. O. Lewis, *Independent verification and validation: a life cycle engineering process for quality software*. John Wiley & Sons, Inc., 1992.
[3] C. G. Cassandras and S. Lafortune, *Introduction to discrete event systems*, 2nd ed. Springer, 2008.
[4] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to automata theory, languages, and computation*, 3rd ed. Pearson Education, Inc., 2007.
[5] K. R. Apt and E.-R. Olderog, *Verification of sequential and concurrent programs*. Springer-Verlag, 1997.
[6] C. Heitmeyer and D. Mandrioli, Eds., *Formal methods for real-time computing*. John Wiley & Sons, 1996.
[7] J. L. Peterson, *Petri net theory and the modeling of systems*. Prentice-Hall, Inc., 1981.
[8] S. Boisse, R. Benenson, L. Bouraoui, M. Parent, and L. Vlacic, "Cybernetic transportation systems design and development: Simulation software," *IEEE International Conference on Robotics and Automation - ICRA'2007*, 2007 2007.
[9] B. Molinete, L. Bouraui, E. Naranjo, H. Kostense, F. Hendriks, J. Alonso, R. Lobrino, and L. Isasi, *CyberCars-2: Close Communications for Cooperation between CyberCars*. INRIA Technical Report Project No IST-2004-0228062, 2009.