

Research Article

An Innovative Approach for Modelling Urban Road Traffic Using Timed Automata and Formal Methods

Eduardo Valente,^{1,2} Camelia Avram,¹ José Machado ,² and Adina Astilean¹

¹Technical University of Cluj-Napoca, Faculty of Automation and Computer Science, Str. G. Baritiu, nr. 26-28, 400027 Cluj-Napoca, Romania

²University of Minho, MEtRICs Research Center, Campus of Azurém, 4800-058 Guimarães, Portugal

Correspondence should be addressed to José Machado; jmachado@dem.uminho.pt

Received 1 May 2018; Revised 8 August 2018; Accepted 30 August 2018; Published 18 September 2018

Academic Editor: Abdelaziz Bensrhair

Copyright © 2018 Eduardo Valente et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Modelling and implementing adequate controllers for urban road traffic control constitute a huge challenge nowadays because of the complexity of systems, as well as possible scenarios and configurations, in each road in a city. A series of issues related to modelling these behaviours are common to arise when using formalisms, tools, and computation machines to perform complex calculations and limitations. This paper presents a formal, flexible, and adaptable approach, with no limitations, from the scientific point of view. For this purpose, modelling formalisms (cellular automata and timed automata) and analysis techniques (simulation and formal verification) are proposed to reach the main goals of modelling complex and adaptable behaviours in urban road traffic with multiple over time changeable configurations. A case study is presented, in order to illustrate the approach and demonstrate in detail the unlimited application of the presented approach.

1. Introduction

The continuous increasing of the number of vehicles in parallel with a slight possibility of building new roads and the corresponding infrastructure are only two of the main reasons that are permanently leading to the search for new solutions, capable of preventing congestion and improving road safety.

Numerous authors highlighted the necessity of a thorough analysis of various traffic conditions, having the role to reveal the most important specific characteristics. They found their expression in the three main categories of models, developed as a result of macro, meso, and micro approaches, which were modified, extended, and improved in a multitude of studies [1–3].

Based on different categories of information, these models are used in a lot of scenarios and for various purposes. While macroscopic model simulations determine the traffic flow or average velocities for different vehicle densities, where vehicles are considered moving entities in the traffic infrastructures, micromodels need a large amount of data to reproduce the dynamic behaviour of individual vehicles in

different traffic conditions [4, 5]. A lot of research has been dedicated to developing microscopic car-following models applicable in situations in which collisions can occur, with the offered information being useful for operational analysis, too [6]. Considering the importance and the complexity of lane-changing operations as components of various traffic simulation tools [7], different approaches and algorithms were proposed for macro and especially micro modelling applications [8].

Practical reasons generated by inherent disadvantages of each category of the basic models, such as computational cost and some unsatisfactory evaluation results regarding the capabilities of different groups to adapt to real situations, lead to the necessity to consider appropriate hybrid approaches, respectively, the possibility of combining models to work together in a common framework [9]. It must be mentioned that different model combinations may lead to network representation consistency problems, and new specific aspects related to this approach must be taken into account [6].

Traffic simulation systems, as dedicated tools used for the analysis of traffic congestions, played a crucial role in the development of traffic models. In existing research directions,

the event driven, or time driven, evolution of the traffic was considered along with the deterministic or stochastic nature of processes. Under particular conditions for many traffic models, two possible approaches were focused upon in an attempt to realize efficient models within a unified framework; the theory of traffic modelling focused on two major directions, attempting to realize efficient models and a unified framework.

A considerable number of road traffic simulators were proposed, and a particular importance has been given to microsimulation models [10]. VISSIM [11–14] TRANSMIS [15], TransModeler [16], AIMSUN [17], MITSIM [18, 19], CORSIM [20], and PARAMICS [21] are only representative examples, some of the most used simulation tools for road traffic monitoring, management, control, prediction, and other related applications.

Comparative aspects regarding the characteristics and capabilities of traffic simulators constituted standalone subjects of many studies, taking into account different criteria by the authors. While comparisons referring to traditional traffic [22–30] only consider basic criteria, others aim to evaluate the actual simulators as useful tools for the analysis of intelligent transportation systems. The most common characteristics that have been taken into account in the first case are the flow model category, modelled objects/entities and phenomena, realized functionalities, the facilities that the model can analyse, implemented restrictions, the system complexity, the simulation speed, and the dimension of the simulated network. In the second case, adaptability, flexibility, extensibility, real-time data integration and analysis, computational cost, visualization facilities, and possibilities to use wireless sensors and to integrate Geographic Information Systems (GIS) as components used in the decision process form some elements of the comparison basis.

Recent researches show that the potential of traffic simulators must be increased, in terms of pedestrian safety traffic [31–35], new algorithms [36], and calibration and validation [37–39], and the growing complexity of the next generation of transportation systems requires automated verification techniques.

Focusing especially on the necessity of validation and verification of microsimulation traffic models and based on the theory of timed automata [38], UPPAAL model-checker [40–42], and the corresponding associated language, this paper presents a unitary framework to modelling, simulation, and verification of urban traffic. A flexible traffic micromodel, easily changed and adaptable to various local traffic conditions, was designed. Having an increased integration capacity, its use can be extended to the scale of an entire city. An adaptive and systematic approach to modelling traffic behaviour considers different configurations of the several actors (vehicles, lanes, road, crossovers, and pedestrians) that could intervene in this process. It must be specified that, in adopting these solutions, a unitary approach is possible because UPPAAL is a tool designed to check systems that can be modelled as networks of timed automata, extended with integer variables, structured data types, user defined functions, and channel synchronisation [40, 41].

The rest of the paper is organised as follows: Section 2 is devoted to a discussion on a few work hypotheses related to formalisms and tools used in the developed study; Section 3 presents the problem and a systematic methodology; Section 4 details a case study for illustration and the obtained results are analysed. Section 5 presents some conclusions and future work.

2. Considered Hypotheses

The key elements of the urban road traffic modelling approach to be considered are presented in this section. A careful choice of the theoretical and implementation support was needed to ensure a systematic, flexible, and adaptive solution (Section 4). Relevant characteristics of the main components, their corresponding role in the system, and their interactions are highlighted.

2.1. Cellular Automata. Cellular automata were used for the first time in the 40s by Ulam and von Neumann in order to analyse possible behaviours of complex systems. Later, due to a considerable number of studies in the field, the theory of cellular automata was continuously developed, and many derived structures have been used to simulate a large variety of behaviours.

The basic components of cellular automata are cells that are organised in different configured n -dimensional grids. The system evolution can be described by a variety of sets of rules that are defined, taking into account the states of neighbouring cells, and are applied in a preestablished number of phases. A complete formal definition of this theoretical concept is presented in [29, 35]. In a simplified manner, it can be defined as follows:

Cellular automaton CA is a quintuple:

$CA = \langle S, s_0, G, d, F \rangle$;

S is a finite set of states;

s_0 is the initial state, $s_0 \in S$;

G is the cellular neighbourhood;

$G = \{i, i + r_1, i + r_2, \dots, i + r_n\}$, where n specifies the neighbourhood size;

d is the dimension of C ;

$f: S^n \rightarrow S$ is the local transition rule, or the local cellular interaction rule;

$CA(t)$ is the configuration at time t ;

$CA(t) = (s_0(t), s_1(t), \dots, s_N(t))$, where N is finite size of CA and $s_i(t)$ is state of cell i at time t ;

F is the global mapping, $F: C(t) \rightarrow C(t+1)$.

The utility of cellular automata was proved in many categories of applications, with the urban road traffic being one of the most important [45–47]. Taking into account the above considerations, in the first stage of this work, cellular automata based formalism was chosen for modelling purposes. It was considered a good option not only because it was suitable to model specific characteristics of the application, but also because it could constitute a possible basis to further implement discrete events and real-time modelling features [48]. Their subsequent integration played a crucial role in the process of obtaining a realistic simulation and a correct

formal verification, with both being two important final goals of the presented application.

2.2. Timed Automata. Timed Automata were developed as a consequence of the need to model the timing behaviour of certain categories of systems that was not possible, using discrete automata. In order to include time variables [40], the theory of the finite automata was extended. In this context, the time was related to a set of real-valued variables modelling clocks [41]. There is a direct relation between the restrictions on the behaviour of timed automata and the restrictions imposed by clock variables.

Consequently, this formalism is well adapted for modelling processes in which the time must be taken into account such as in urban road traffic modelling. In this case, time is important in order to calculate the speed or the acceleration, or to model other various aspects.

A complete formal description of timed automata is presented in [40], while a brief, formal definition of timed automata is given below [49]:

A timed automaton is a six-tuple $A = (L, L^0, C, A, E, I)$, where

L is a set of locations;

$L^0 \in L$ is the set of initial locations;

C is the finite set of clocks;

A is a set of actions and coactions and the internal τ -action;

$E \subseteq L \times A \times B(C) \times 2^C \times L$ represents the set of transitions. An arc (l, l', a, λ, b) represents a transition from the location l to the location l' when entering the symbol a . The set $\lambda \subseteq C$ represents the clocks to be reset in this transition, and b is a time restriction over C .

$I : L \rightarrow B(C)$ assigns invariants to locations.

To define the semantics of a timed automaton, a clock valuation is a function $u : C \rightarrow R \geq 0$ from the set of clocks to the nonnegative real values. Let R^C be the set of all clock valuations. Let $u_0(x) = 0$ for all $x \in C$. It is considered that the notation means that guards and invariants are sets of clock valuations; thus writing $u \in I(l)$ means that u satisfies $I(l)$.

During a run of a timed automaton, all clock values increase with the same speed [50].

The clocks can be reset to zero (independently of each other) in the transition of the automaton, keeping the time elapsed since the last reset [51].

Based on the above considerations, the application presented in the paper was developed starting with a model based on cellular automata, which corresponds to the requirements to be a dynamic, discrete space and discrete time formalism and then translating the resulting cellular automata to a discrete automata [45]. Furthermore, temporal aspects were taken into account.

It must be pointed out that even though a large number of formalisms could be used to model timed systems and in particular the proposed traffic system, timed automata were adopted here especially because the behavioural analysis of the considered models was realized using UPPAAL model-checker. This choice has the advantage that the necessity to convert models when the simulation and verification processes are performed is eliminated.

2.3. UPPAAL. There are several tools used to analyse timed automata and extensions, including UPPAAL, Kronos, and TIMES. These are becoming more and more mature, but they are all exclusively academic research tools.

UPPAAL is a model-checker [38], adapted to verify real-time system behaviours. The considered models are built in a network of processes, with each process being considered an automaton. The entire model is composed of three parts: its global and local declarations, the automata templates, and the system definition. Because the use of templates is possible in UPPAAL, it is suitable for modelling complex systems to use modules corresponding to the component parts of the system that is being modelled. This characteristic allows the instantiation of models of the same kind and the reusing of models for systems of the same structure, having similar component parts. Also, the utilization of modelling modules allows more simplicity in terms of the modelling complex systems behaviour.

Besides the mentioned advantages, UPPAAL allows obtaining results for simulation and formal verification techniques, using the same environment [40, 48]. This tool is also well adapted to simulate and verify networked complex mechatronic systems [41].

Due to the above specified reasons, UPPAAL was chosen for simulation and formal verification purposes contributing to a more complete understanding of the presented application behaviour.

2.4. Timed Computation Tree Logic. In order to formalize the desired behaviour of the system modelled with timed automata, it is important to choose a formal logic compatible with the verification tool. Consequently, this will be a timed version of Computational Tree Logic (CTL), a simplified version of Timed Computation Tree Logic (TCTL) [52], used by UPPAAL.

CTL [43] is a propositional, branching-time temporal logic that enables expressing queries over the possible (desired and/or undesired) behaviours of a model. In CTL, the behaviour of a system is seen as a tree, representing behaviour possibilities (Figure 1). CTL formulae consists of path formulae and state formulae. State formulae are evaluated in individual states. Path formulae quantify over paths in the behaviour of the model and over the states in those paths.

3. Modelling of Urban Road Traffic

This section is divided into two main parts: firstly, a series of issues are dealt with in relation to modelling urban road traffic; secondly, a flexible, adaptive, and systematic approach for solving these issues is presented.

3.1. Aspects of Modelling Urban Road Traffic. In order to correctly represent urban road traffic, aspects to be considered include modelling the following:

- (i) modelling different vehicles (cars, buses, trams, trucks, etc.)
- (ii) modelling the driver's behaviour;

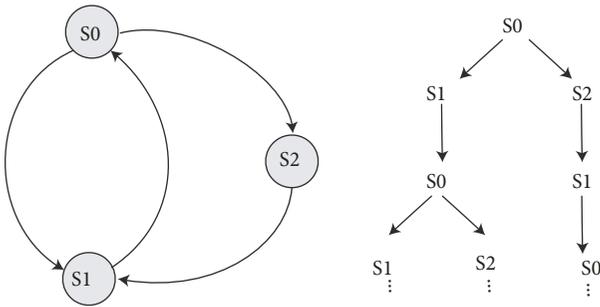


FIGURE 1: A state machine and its computation tree. adapted from [43].

- (iii) modelling road and respective traffic-lanes, considering different sizes for road and lanes;
- (iv) modelling crossroads and upcoming road;
- (v) modelling car parks, stops, and changing traffic-lanes for different kinds of vehicles (except trams);
- (vi) modelling the velocity and acceleration of vehicles;
- (vii) modelling the space occupied by each vehicle taking into account the size of the vehicle and the speed corresponding to each road (dynamic cell length).

When considering modelling tasks, there is also an interaction between the actors of the system.

The automobiles can come across the following implemented traffic-elements:

- (i) pedestrian crossings for pedestrians;
- (ii) car parks placed transversely/alongside on the right side of the traffic-lane.

The buses can come across the following implemented traffic-elements:

- (i) pedestrian crossings;
- (ii) bus-stops inside or outside the traffic-lane.

The trams can come across the following implemented traffic-elements:

- (i) pedestrian crossings;
- (ii) tram-stops inside the traffic-lane.

For the purposes mentioned above, a modular modelling approach is adopted, using timed automata. This approach considers a modular model for each vehicle, road, lane, crossroad, upcoming road, and physical component.

In fact, this approach seems to be efficient, but it is neither flexible nor adaptive, despite being systematic. The reuse and readaptation of modules are not simple and when, for instance, the size of one road or one lane changes, the modules must be built again, with new specific conditions for each model, for each actor of the system. Another important issue is the fact that a huge number of models (composed of different modules) pose issues when it comes to extending the reachable state space of the overall system model.

In order to deal with this complexity, it was decided to create a model to representing accurately by a matrix the movement of the vehicles and the traffic-elements and being easily extendable allowing addition of more rows or columns. Matrices were used for modelling all parts of these complex systems.

This model is the first model with relevant traffic road rules included along with a significant level of detail. This methodology features the following main characteristics:

- (i) The physical environment is a one-dimensional grid of rectangular cells, all equal in size (7.5 meters of length).
- (ii) It is a single cell model because each automobile will occupy only one road's cell for each time iteration, and the cells can have only two possible states: occupied by an automobile or empty.
- (iii) The size of the neighbourhood is the same for each cell; the model is anisotropic because the automobiles only respond to stimuli in front of them.
- (iv) This model is a dynamic system with a limited number of automobiles, buses, trams, and roads that evolve and change in time and space depending on the same rules; only if the cell in front is free, can the different vehicles move along the lane.
- (v) The time is a stochastic feature and its choice is completely nondeterministic. In an instant t the automobile can circulate in a cell at a velocity of 50 km/h, and in the instant $t+1$ it can move at 2 km/h. This model considers, for vehicles, the possibility of severe deceleration and acceleration even if the road is completely free in order to model the individual driver's behaviour.
- (vi) This model can be easily extended. The number of roads and automobiles can be easily changed; the only requirements are to define in global declaration those variables and the matrix for extending the map, to define the inputs in terms of the traffic flow used in the simulation, to define the upcoming road, and to set up the variable for each automobile.

By using this approach, vehicles can circulate irrespectively of the different traffic-elements and interact with them. For example, a pedestrian crossing will influence the behaviour of the automobiles, buses, and trams when pedestrians cross the traffic-lane. If there is also a bus or a tram stop on the next lane, this will not affect an automobile on the road because it does not need to change its behaviour; rather buses and the trams will have to stop. These different behaviours are given by the functions and channels implemented in the road declarations explained in the next subsection.

3.2. Modelling Approach. Different structures are defined in order to represent the complexity of the urban road traffic. Two types of urban road traffic structures are taken into consideration: infrastructure (road, crossroads, traffic signs, parking lots, and pedestrian crossings) and moving vehicles (automobiles, trams, and buses). The formalism adopted

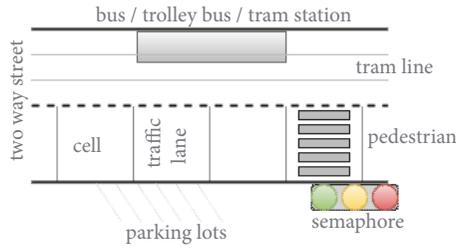


FIGURE 2: First layer of the structure.

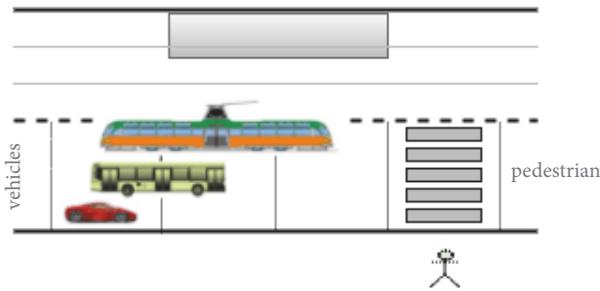


FIGURE 3: Second layer of the structure.

to model the considered system is timed automata (TA). UPPAAL environment will be used for performing the simulation and formal verification analysis techniques.

To model and analyse the two types of structures, a two-layered model based on timed automata is considered; one represents the components of the infrastructure and the relations between them (Figure 2) and the second one represents the mobile components of the urban road traffic (Figure 3). The interactions between layers are defined into a set of rules which are implemented in “C” programming language.

Several equations are considered to represent the evolution of each component. Arrays and matrices are built to memorize the components (as arrays) and their (current and future) behaviour and their interactions (as a matrix).

The circulation of the individual vehicles in a traffic flow is described by a set of rules that reflect the movements of cars and the lane-changing behaviour, evolving in time and space.

The characteristics of the defined structures include:

- (i) *traffic-lanes*: name, length, number of cells (determined for each road separately taking into consideration the length and the safety distance), bus/tram stop, and parking lot;
- (ii) *road*: number of lanes;
- (iii) *crossroad*: number of roads which form the crossroad, name of the road;
- (iv) *traffic signs*: (stop, give way, traffic-lights, etc.) defining the rule that needs to be observed;
- (v) *car parks/parking lots*: located inside a road or aside from the road and can be parallel, perpendicular, or placed at an angle of 45° from the direction of travel;
- (vi) *pedestrian crossings*: located inside a road;

- (vii) *automobile*: number of automobiles; they occupy a single cell at a given time and can travel anywhere or have a predefined route;
- (viii) *tram*: number of trams; they occupy three cells and have a predefined route and predefined stops;
- (ix) *bus*: number of buses; they occupy two cells and have a predefined route and predefined stops.

The automobiles can come across the following implemented road-elements:

- (i) pedestrian crossings;
- (ii) car parks placed transversely on the right side of the traffic-lane;
- (iii) car parks placed on the right side of the traffic-lane;
- (iv) (fixed or mobile) obstacles that can be overtaken if the traffic allows it.

The buses can come across the following implemented road-elements:

- (i) pedestrian crossings;
- (ii) bus-stops inside the traffic-lane;
- (iii) bus-stops outside the traffic-lane, on the right side of the traffic-lane;
- (iv) obstacles that can be overtaken if the traffic allows it.

The trams can come across the following implemented road-elements:

- (i) pedestrian crossings;
- (ii) tram-stops inside the traffic-lane;
- (iii) overtaking obstacles is not possible.

Using this formalism to represent urban road traffic and the interactions within it allows us both a general representation and the ability to model complex structures. This way, the model allows any modification of the simulated map without changing the UPPAAL model, only by changing the initialization variables. Since the length of road is different and the matrix cannot be defined with variable number of columns, the value -1 is introduced.

Global declarations include the number of automobiles, buses, trams, and roads, using the function “typedef”. The maximum number of cells that can be defined for a road (the longest road) has been declared by using the variable “maxnoCells”. With this variable and the variable number of road (noS) the matrix “idexSC” was created. This matrix is a map of road cell coordinates. Each line corresponds to a road ID—the ID of the first line is 0 and the ID of the last line is the last road’s ID. The column number one corresponds to the number of cells that each road covers and the following columns are filled with -1 which means that a road’s cell is empty. The first column will be important as it allows the limitation of the road size.

Due to UPPAAL’s limitations it is not possible to create arrays with different sizes, and some of the values of “-1” present in the matrix have no significance. Therefore, it is

easy to extend the map and to implement other features in subsequent models.

The choice of the next road in an intersection was implemented with a similar application. First, a new variable “maxNextRoad” (maximum next road) was declared. This variable is equal to three because a crossroad features a choice of no more than three roads usually. In case of a crossroad with more than three streets the problem is managed in a similar manner. Using the variable “maxNextRoad” and with the variable number of the road (“noS”), another matrix “indexMAP” was created. Each line corresponds to a road ID whose first line ID is 0 and whose last line ID is the last road’s ID. The first column contains the number of possible next road choices at the end of the current road, and the other columns have the road’s IDs of the next road.

To implement these road-elements it was necessary to instantiate the matrix “indexSC” for new negative numbers which will have a different meaning from an empty cell, affecting the behaviour of the vehicles. The elements present inside the road (pedestrian crossings, bus-stops inside the road, and tram-stops inside the road) were coded in the respective road ID with a negative number.

However, to create the road-elements outside the road, it was necessary to define adjacent roads to implement a turning movement that a vehicle will execute to go to an adjacent road where “i” the road-element “s” is located. The implementation rule is the following: if a traffic-lane has to its right an exterior road-element (Parking lots across the road, Parking lots alongside, and bus-stops outside the road), the road ID-1 will have the corresponding negative digit of the road-element, and in the road ID-2 will be necessary to create the border of the road that will give information that the vehicle cannot move to another road ID on the right.

In summary, the different negative digits that a matrix “indexSC” can have will give the instructions for an interaction between a vehicle and a road-element. The meaning (the road-element) of the negative digits is as in Table 1.

With the “indexSC” variable and the variable number of the roads (noS), three matrices “indexMapAUTOMOBILE”, “indexMapBUS”, and “indexMapTRAM” are created. Each line of those matrices corresponds to a road ID, whose first line ID is 0 and whose last line ID is the last road’s ID.

To define the physical environment several variables were defined: “maxnoCells” (represents the maximum length of the road considered, maximum number of cells); “indexSC” matrix (size of noS x maxnoCell+1: this matrix is the description of the physical infrastructure; each line corresponds to a road ID, whose first line ID is 0 and whose last line ID is the last road’s ID and the column number one corresponds to the number of cells that each road has and the following columns are filled with -1 which means that a road’s cell is empty); “maxNextRoad” (maximum number of next roads); “indexMAP” matrix (each line corresponds to a road ID, whose first line ID is 0 and whose last line ID is the last road’s ID; the first column contains the number of possible next roads at the end of the road, and the other columns have the road’s IDs of the next road.); and “currentRoadA” (the current road for each automobile was defined; an array is created and presented as the initial road’s ID where an automobile

TABLE 1: Description of the matrix number of cells per different road (with different road-elements).

-1	Empty cell
-2	Bus station inside the traffic-lane
-3	Tram station inside the traffic-lane
-4	Pedestrian crossings
-5	Possible parking lot alongside for automobiles on the right side of the road
-55	Border of a parking lot alongside for automobiles on the right side of the road
-6	Possible transversal parking lot for automobiles on the right side of the road (right place)
-66	Border of a transversal parking lot for automobiles on the right side of the road (right place)
-7	Possible transversal parking lot for automobiles on the right side of the road (left place)
-77	Border of a transversal parking lot for automobiles on the right side of the road (left place)
-22	Bus station outside the right side of the road
-222	Border of a bus station outside the right side of the road
-8	Road’s end and cells with no meaning for the simulation

will start the simulation). This array contains the inputs of the traffic flow used in the simulation. The value of “current Road” will change every time an automobile leaves a road and continues to move along another road of the map. The current road for each automobile is defined, and the value defined in the array “current Road” represents the initial road’s ID where an automobile will start the simulation. Figure 4 represents the interactions between the modelling layers.

The average speed of the vehicles along a traffic-lane can be expressed by the following equation.

$$\bar{v} = \frac{\sum_{i=1}^n v_i}{n}, [m/s] \quad (1)$$

The traffic density is calculated as follows.

$$\rho = \frac{n}{L}, [vehicles/m] \quad (2)$$

Based on these values, the traffic flow is:

$$J = \frac{\sum_{i=1}^n v_i}{L}, [vehicles/s]. \quad (3)$$

The average speed of the vehicles that cross the lane section is expressed by:

$$\bar{v} = \frac{\sum_{j=1}^m v_j}{m}, [m/s]. \quad (4)$$

Two types of traffic behaviours were implemented: the “car-by-car” model and the free traffic model (this involves that when a car is driving on the road with no constraints and an obstacle emerges, the traffic allows the driver to overtake it).

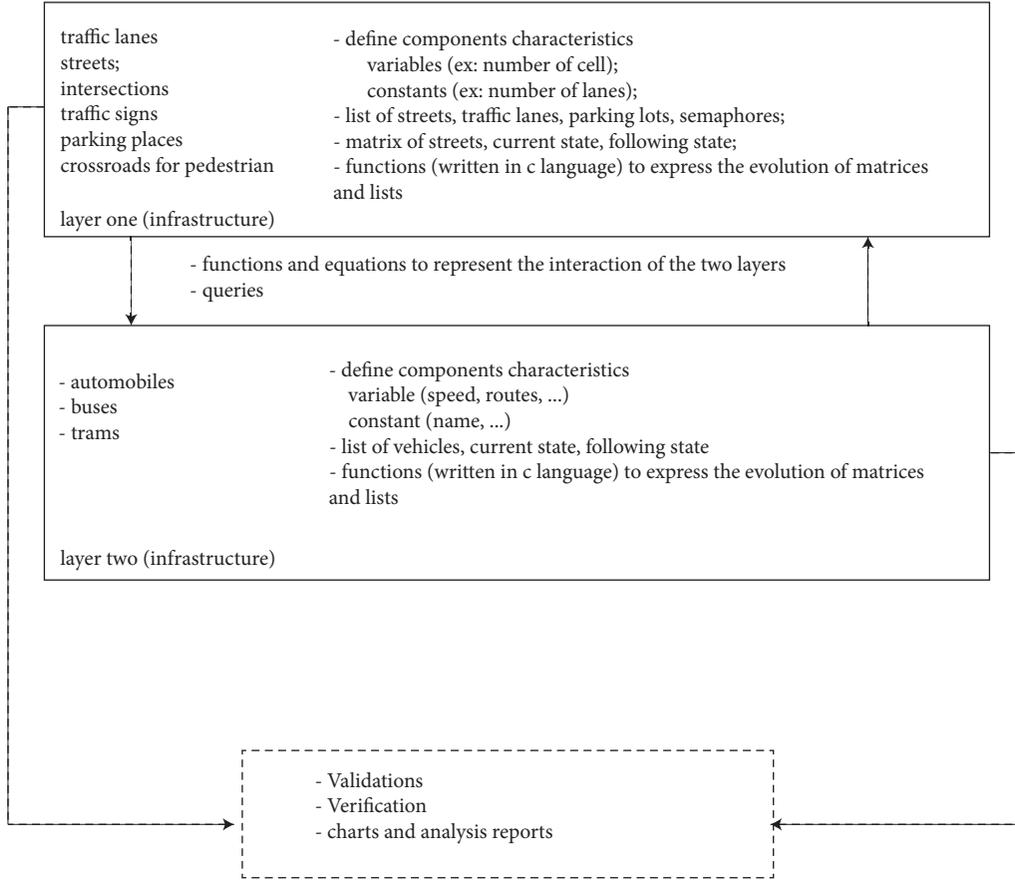


FIGURE 4: Interactions between layers.

The “car-by-car” model used in this model is presented in Figure 5.

In general, the “car-by-car” models are centred on the following relation.

$$[Response]_n \propto [Stimulus]_{n+1} \quad (5)$$

The vehicle $n = 1, 2, \dots$ can only accelerate or decelerate as a response to the different flow conditions. The equations of “car-by-car” models consider the vehicle’s analysed speed (v_n), the speed difference between it and the leader vehicle (Δv_n), and the distance between these vehicles (d_n) among others. In general, it has the following form [53].

$$\bar{x}_n(t) = f(v_n(t), d_n(t), \Delta v_n(t)) \quad (6)$$

The stimulus function, f , has different interpretations for each “car-by-car” model. The simplest model is a follow-the-leader one which only uses the speed difference between the vehicle’s leader and its tracker, represented by the following equation.

$$\bar{x}_n(t) = \frac{1}{\tau(\dot{x}_{n+1}(t) - \dot{x}_n(t))} \quad (7)$$

where τ is a setting parameter related to the time scale; $1/\tau$ can be interpreted as the sensitivity of the tracker vehicle’s speed to the leader vehicle’s speed variations.

Changing the traffic-lanes in order to overtake an obstacle or not is presented in Figure 6 and is done by using the following steps: First, three cells (in front) on both traffic-lanes need to be free (current road lane and neighbour road lane); the checking is done in cell X [t or t+1, i]; if yes, secondly, the vehicle changes the traffic-lane. This works only to prove the concept because in real life the traffic is congested or at least there is more than one car on the streets. To be more accurate a third step (or check) is also introduced: simultaneously, in front of the car one cell should be free (cell i+1) and two cells on the next traffic-lane should be free (cells i and i+1), too.

4. Application of the Methodology: Case Study

After presenting the developed and tested approach, we include a case study in order to allow illustrating the application of the approach and to allow the extrapolation of the obtained results for other similar systems.

4.1. Presentation of Different Situations. One of the key aspects of creating good traffic road models is to define in a precise and unambiguous way the road that will be studied with all relevant information for a correct vehicular circulation.

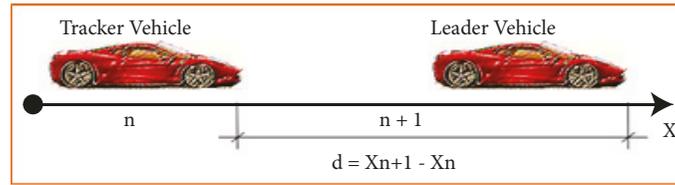


FIGURE 5: Scheme of the car-by-car model [44].

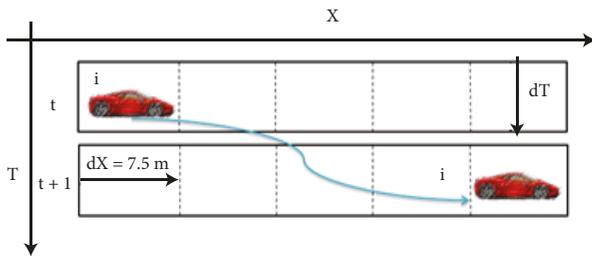


FIGURE 6: Changing traffic-lanes model.

In order to test and validate the proposed model a case study is considered, and the simulation involves a small group of roads in the city centre of Cluj-Napoca, containing the maximum road-elements that a road can have in order to test and verify the proposed model, as presented in Figure 7.

This section details the physical environment of the simulation, describing the road (number, direction, and length of the traffic-lanes), the location of the pedestrian crossings, bus stations and tram stations, tram railways, the intersections between roads, and the way traffic-lights for road, rail, and pedestrian traffic operate.

The total network length (Figure 7) is 3735 meters and this simulation contains 12 pedestrian crossings, 32 traffic-lanes for automobiles, 22 lines of buses, 2 tram railways, 8 traffic-lights for drivers and other 8 traffic-lights for pedestrians, 4 bus-stops, and 2 tram-stops, as well as 2 parking spaces.

The simulated map contains the maximum number of elements that can be simultaneously simulated due to software constraints. Observing a large number of elements, tracking their status over time, is also costly.

In order to analyse some specific road traffic problems from the simulated network, some small parts are simulated and tested separately.

The image in Figure 7 is a 2D model of the entire studied traffic environment. For each road various pieces of information were collected and structured in Table 2.

The time measurements were taken at midday and at 6 pm, when roads are the busiest. The mode of operation of the traffic-lights taken into consideration in the simulation is the same at both times.

4.2. Illustration of Methodology Application. The model is based on templates for roads, traffic-lights, and vehicles (automobiles, buses, and trams, because three different types of vehicles are considered). In the global declaration, the

numbers of automobiles, buses, trams, roads, and traffic-lights are first declared, using the function “typedef” according to the model described in Section 3.2.

The automaton road was simplified because there are only three channels, which means

- (i) ROAD_STARTS: when an automobile, a bus, or a tram was detected by a sensor at the beginning of a road and the road is free of any automobile;
- (ii) UPDATING_CELLS: when an automobile, bus, or tram or more than one vehicle on the road are moving; several functions were created in order to model the driving behaviour on the road;
- (iii) ROAD_ENDS: when an automobile, a bus, or a tram was detected by a sensor at the end of the road and it leaves the road.

The automaton road is the “brain” of the simulation, thus storing all the information regarding traffic conditions, the set of rules, and the subsequent transition rules.

The automaton automobile has three places:

- (i) OUT_of_the_ROAD_START_MOVING: when an automobile is moving towards the beginning of the road;
- (ii) MOVING_INSIDE_the_ROAD: when an automobile was detected by a sensor and is moving on the road;
- (iii) OUT_of_the_MAP: when an automobile was detected by a sensor and there is no possible next road and it exits the considered map.

The bus and tram automatons have structures similar to that of an automobile, but they also feature a predefined path on the map.

The traffic light automaton has three states defining the three colours (red, green, and yellow). Yellow can be skipped, consequently reducing the loop to only two colours (red and green).

In the road considered in the case study, traffic behaviour is not exclusively influenced by the state of the road cells, as it was considered in the previous model. The vehicles interact with other road-elements present in the road such as pedestrian crossings, bus stations, tram stations, and traffic-lights. In the studied model, the interaction between vehicles and the road-elements contained in the group of the roads was taken into consideration.

The automobiles can come across the following implemented road-elements:

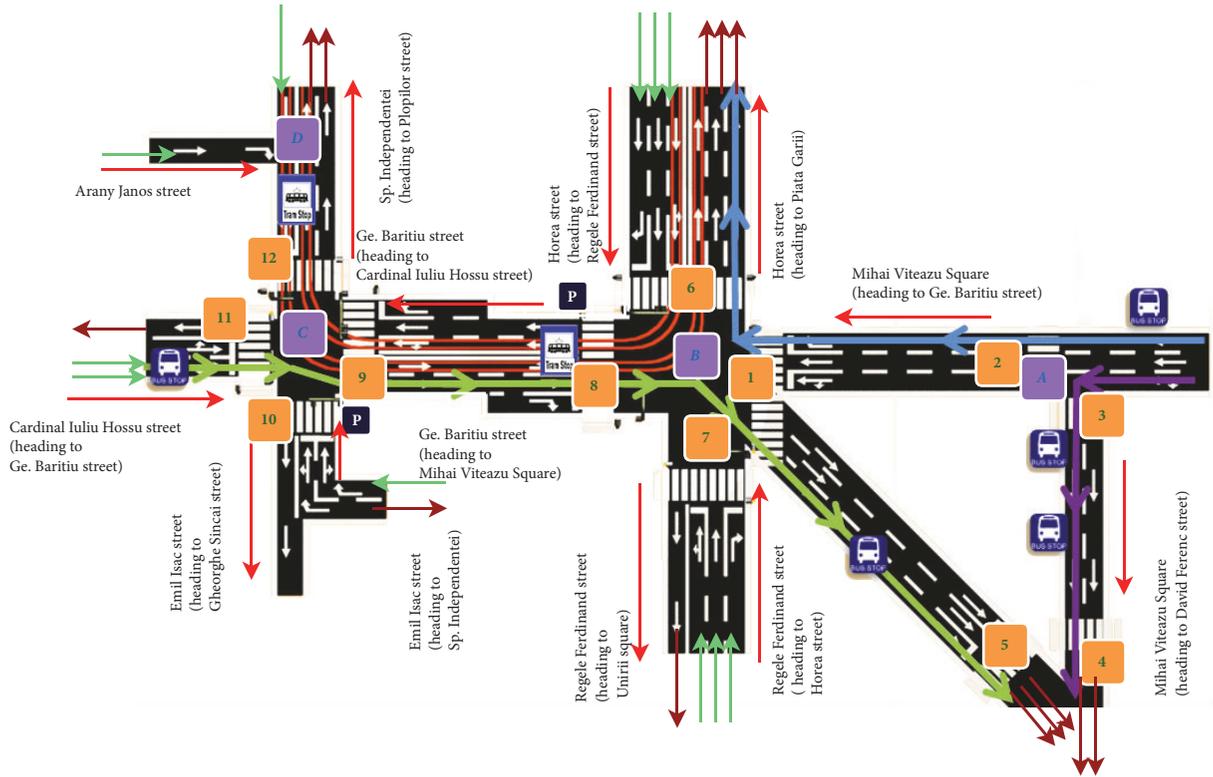


FIGURE 7: Representation of the entire road network considered in this simulation.

TABLE 2: Structured information for each modelled and simulated road.

Length considered	Length of the traffic lane (number of cells)				
Number and direction of traffic-lanes	Number of traffic lanes - direction				
Possible next road in the intersection A	Next road to follow				
Possible next road in the intersection B	Possible next road to follow (directions)				
	Traffic-lane	Right	Left and central	Time	
Traffic flow control at the intersection B for each traffic-lane	Red:	Time	Time	Time	seconds
	Yellow:	Time	Time	Time	seconds
	Green:	Time	Time	Time	seconds
Pedestrians flow control in the pedestrian crossing number one		Traffic-light:			
		Green: time [s]			
		Red: time [s]			
Features of the Bus stop	Location	coordinates			
	Length	Length of the bus station [m]			
	Buses	Buses no. lanes			
Pedestrian crossing number two	Pedestrians' desire				
Maximum velocity	Max. speed allowed on this sector km/h				
Types of vehicles allowed	Automobiles, Motorbikes and Buses				

- (i) pedestrian crossings;
 - (ii) parking lots placed transversely on the right side of the traffic-lane;
 - (iii) parking lots placed alongside the right side of the traffic-lane;
 - (iv) traffic-lights.
- The buses can come across the following implemented road-elements:
- (i) pedestrian crossings;
 - (ii) bus-stops inside the traffic-lane;

- (iii) bus-stops outside the traffic-lane, on the right side of the traffic-lane;
- (iv) traffic-lights.

The trams can come across the following implemented road-elements:

- (i) pedestrian crossings;
- (ii) tram-stops inside the traffic-lane;
- (iii) traffic-lights.

4.3. Simulation Results. To implement the considered map, 74 traffic-lanes, 480 automobiles, 30 buses, and 18 trams were declared. Afterwards, the traffic-elements and the predefined routes for buses and trams were implemented. Given the large volume of interactions between vehicles, road-elements, and roads, the entire map was divided into several small networks in order to simulate and analyse each modelled feature.

During the simulation and validation process, the authors faced several problems related to computational power that UPPAAL needed. The allocated memory that UPPAAL could use during the simulation was restricted and could not be extended. To overcome this problem, the authors defined several scenarios to validate all the considered traffic-elements. First the roads from the original case study were considered (74 traffic-lanes, 90 automobiles, 10 buses, and 1 tram) in order to verify and test all the already presented interactions. In that map, the road-elements were not considered because of the limitations of UPPAAL (e.g., pedestrians, car parking, drivers' behaviour, and dynamic cell length). Another simulation was performed in order to study the interaction between traffic-elements. This simulation included a road which contained all the different road-elements (pedestrian crossing, bus stop, and tram stop cells within the road and dynamic cell length).

Figure 8 presents possible time interactions for an automobile travelling inside the map (route: intersection $D \rightarrow C \rightarrow B \rightarrow A \rightarrow$ road 4; see Figure 6) with this sequence of road-elements being presented. Basically, an automobile will only react to the stimuli given by road cells and pedestrian crossing cells, and when it is passing through a bus stop or a tram stop it will react with the same behaviour as for a road cell. These stimuli will influence the time needed to move through each different cell, with each red line being the minimum time needed to travel each cell and each green line being the maximum time needed to travel each cell. The sums of the minimum and maximum times needed to travel each cell of the road, by an automobile, are 4 and 239 time units, respectively. In this figure, two concrete ways of interaction between road-elements and an automobile are presented. The values presented in these two possible interactions were extracted from the option "concrete simulation" in UPPAAL.

Figure 9 shows the possible time interactions at the moment that a bus is travelling on the road with the same sequence of road-elements. When a bus is travelling on this road, it will react to the stimuli given by road cells, pedestrian crossing cells, and bus stop cells, and when it is passing through a tram stop cell, it will react as for a road cell. These

stimuli will influence the time needed to travel through each different cell, with each red line being the minimum time needed to travel each cell and each green line being the maximum time needed to travel each cell. The sums of the minimum and maximum times needed to travel each cell of the road, by an automobile, are 33.5 and 272 time units, respectively. In this figure two concrete ways of interaction between road-elements and a bus are also presented. The values for the two possible interactions were extracted from the separator concrete simulation in UPPAAL.

Figure 10 shows the possible time interactions in the moment that a tram is travelling on the road with the same sequence of road-elements. When a tram is travelling on this road, it will react to the stimuli given by road cells, pedestrian crossings cells, and tram stop cells, and when it is passing through a bus stop cell, it will react like a road cell. These stimuli will influence the time needed to travel through each different cell, with each red line being the minimum time needed to travel each cell and each green line being the maximum time needed to travel each cell. The sum of the minimum time needed to travel through each cell of the road by an automobile is 272 time units. In this figure two concrete ways of interaction between road-elements and a bus are also presented. The values for the two possible interactions were extracted from the separator concrete simulation in UPPAAL.

4.4. Formal Verification. Formal verification is used in this work as a complementary technique to simulation. In fact, simulation considers some possible scenarios of evolution of the developed models, but formal verification checks all possible behaviours of those models. Due to this reason, the validation of the presented approach seems important by using this exhaustive technique.

For formal verification of the model it is enough to know that, in the UPPAAL version, A is the universal quantifier on paths (for any path...), E is the specific quantifier on paths (there is a path...), $[]$ is the universal quantifier over states in a path (for any state...), and $\langle \rangle$ is the specific quantifier over states in a path (there is a state...).

In UPPAAL, in the option Verifier, several queries to check the correct behaviour of this model were implemented. The initial values and variables taken into consideration for the formal verification of the map in Figure 6 are the following:

- (i) number of automobiles = 90;
- (ii) number of buses = 10;
- (iii) number of trams = 1;
- (iv) number of roads = 9;
- (v) number of road cells per road = [30, 30, 8, 22, 14, 14, 14, 14];
- (vi) $\text{indexSC}[\text{noS}][\text{maxnoCells} + 1]$: status (free or occupied) of each cell in each road;
- (vii) $\text{novis}[\text{noS}]$: number of vehicles on the road.

Cellular automata:

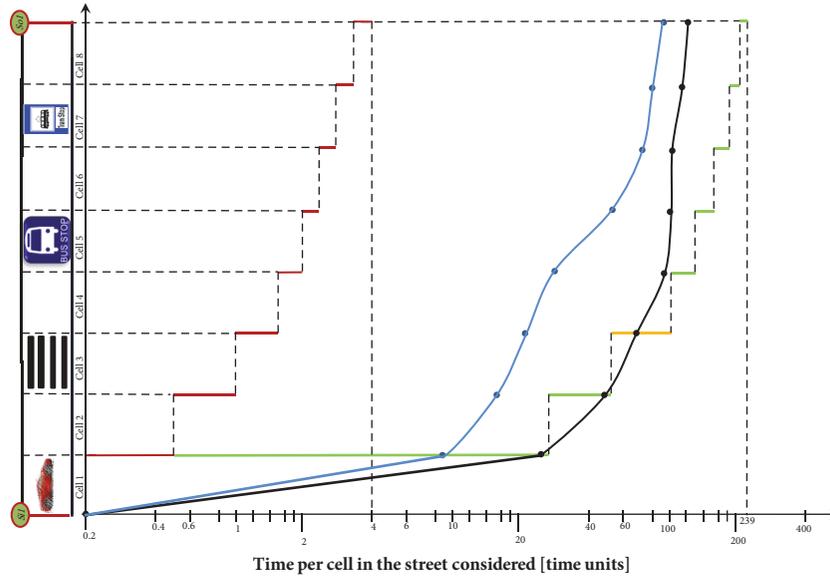


FIGURE 8: Time intervals needed for an automobile to go through a sequence of cells.

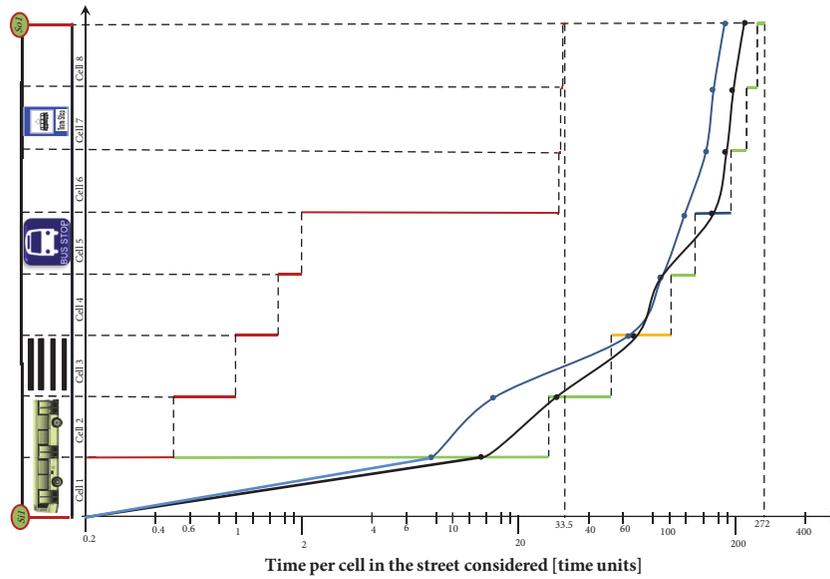


FIGURE 9: Time intervals needed for a bus to go through this sequence of cells.

- (i) automobile (idA, time_per_road, time_per_cellA);
- (ii) bus(idB, time_per_road, time_per_cellB);
- (iii) tram(idT, time_per_road, time_per_cellT);
- (iv) road (idS, GLOBAL, novis);
- (v) traffic-lights (idSem, tR, tY, tG).

A list of queries was generated in order to verify various properties. Some of the queries were built in order to double check some important traffic behaviours. The response of the query could be either “yes” or “no”. Based on this response, a report of a traffic simulation scenario can be generated.

The following properties can be verified and validated:

- (i) Properties that need to be verified: greater than a minimum value
 - (a) $E[] \text{ forall } (i : idA) \text{ automobile}(i).time_per_road(idS) > 0.5 * indexSC[p][q]$
 - (b) $E<> \text{ forall } (i : idA) \text{ automobile}(i).time_per_road(idS) > 0.5 * indexSC[p][q]$
- (ii) Properties that need to be verified: smaller than a maximum value
 - (a) $E[] \text{ forall } (i : idA) \text{ automobile}(i).time_per_road(idS) < 27 * indexSC[p][q]$

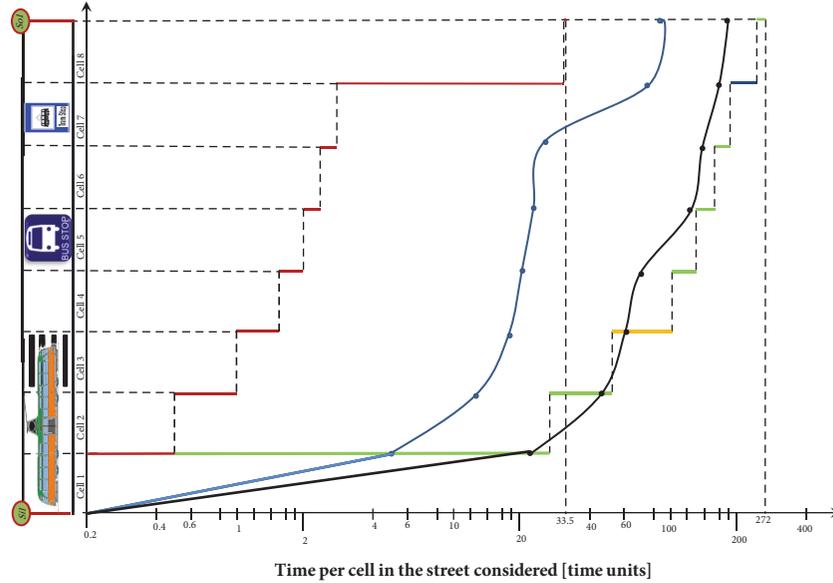


FIGURE 10: Time intervals needed for a tram to go through this sequence of cells.

- (b) $E \langle \rangle \text{forall } (i : \text{idA}) \text{ automobile}(i).\text{time_per_road}(\text{idS}) < 27 * \text{indexSC}[p][q]$
- (iii) Properties that need to be verified: greater than a minimum value and smaller than a maximum value
- (a) $E[] \text{ (forall } (i : \text{idA}) \text{ automobile}(i).\text{time_per_road}(\text{idS}) > 0.5 * \text{indexSC}[p][q]) \ \&\& \text{ (forall } (i : \text{idA}) \text{ automobile}(i).\text{time_per_road}(\text{idS}) < 27 * \text{indexSC}[p][q])$
- (b) $E \langle \rangle \text{ (forall } (i : \text{idA}) \text{ automobile}(i).\text{time_per_road}(\text{idS}) > 0.5 * \text{indexSC}[p][q]) \ \&\& \text{ (forall } (i : \text{idA}) \text{ automobile}(i).\text{time_per_road}(\text{idS}) < 27 * \text{indexSC}[p][q])$
- (iv) Properties that need to be verified: smaller than a maximum value, greater than a minimum value, greater than a minimum value, and smaller than a maximum value (automobile)
- (a) $E[] \text{ automobile}(\text{idA}).\text{time_per_road}(\text{idS}) > 0.5 * \text{indexSC}[p][q]$
- (b) $E \langle \rangle \text{ automobile}(\text{idA}).\text{time_per_road}(\text{idS}) > 0.5 * \text{indexSC}[p][q]$
- (c) $E[] \text{ automobile}(\text{idA}).\text{time_per_road}(\text{idS}) < 27 * \text{indexSC}[p][q]$
- (d) $E \langle \rangle \text{ automobile}(\text{idA}).\text{time_per_road}(\text{idS}) < 27 * \text{indexSC}[p][q]$
- (e) $E[] \text{ (automobile}(\text{idA}).\text{time_per_road}(\text{idS}) > 0.5 * \text{indexSC}[p][q]) \ \&\& \text{ (automobile}(i).\text{time_per_road}(\text{idS}) < 27 * \text{indexSC}[p][q])$
- (f) $E \langle \rangle \text{ (automobile}(\text{idA}).\text{time_per_road}(\text{idS}) > 0.5 * \text{indexSC}[p][q]) \ \&\& \text{ (automobile}(\text{idA}).\text{time_per_road}(\text{idS}) < 27 * \text{indexSC}[p][q])$
- (v) Properties that need to be verified: smaller than a maximum value, greater than a minimum value, greater than a minimum value, and smaller than a maximum value (bus)
- (a) $E[] \text{ bus}(\text{idB}).\text{time_per_road}(\text{idS}) > 0.5 * \text{indexSC}[p][q]$
- (b) $E \langle \rangle \text{ bus}(\text{idB}).\text{time_per_road}(\text{idS}) > 0.5 * \text{indexSC}[p][q]$
- (c) $E[] \text{ bus}(\text{idB}).\text{time_per_road}(\text{idS}) < 27 * \text{indexSC}[p][q]$
- (d) $E \langle \rangle \text{ bus}(\text{idB}).\text{time_per_road}(\text{idS}) < 27 * \text{indexSC}[p][q]$
- (e) $E[] \text{ (bus}(\text{idB}).\text{time_per_road}(\text{idS}) > 0.5 * \text{indexSC}[p][q]) \ \&\& \text{ (bus}(i).\text{time_per_road}(\text{idS}) < 27 * \text{indexSC}[p][q])$
- (f) $E \langle \rangle \text{ (bus}(\text{idB}).\text{time_per_road}(\text{idS}) > 0.5 * \text{indexSC}[p][q]) \ \&\& \text{ (bus}(\text{idB}).\text{time_per_road}(\text{idS}) < 27 * \text{indexSC}[p][q])$
- (vi) Properties that need to be verified: smaller than a maximum value, greater than a minimum value, greater than a minimum value, and smaller than a maximum value (tram)
- (a) $E[] \text{ tram}(\text{idT}).\text{time_per_road}(\text{idS}) > 0.5 * \text{indexSC}[p][q]$
- (b) $E \langle \rangle \text{ tram}(\text{idT}).\text{time_per_road}(\text{idS}) > 0.5 * \text{indexSC}[p][q]$
- (c) $E[] \text{ tram}(\text{idT}).\text{time_per_road}(\text{idS}) < 27 * \text{indexSC}[p][q]$
- (d) $E \langle \rangle \text{ tram}(\text{idT}).\text{time_per_road}(\text{idS}) < 27 * \text{indexSC}[p][q]$
- (e) $E[] \text{ (tram}(\text{idT}).\text{time_per_road}(\text{idS}) > 0.5 * \text{indexSC}[p][q]) \ \&\& \text{ (tram}(i).\text{time_per_road}(\text{idS}) < 27 * \text{indexSC}[p][q])$

- (f) $E \langle \rangle (\text{tram}(\text{idT}).\text{time_per_road}(\text{idS}) > 0.5 * \text{indexSC}[\text{p}][\text{q}] \ \&\& \ (\text{tram}(\text{idT}).\text{time_per_road}(\text{idS}) < 27 * \text{indexSC}[\text{p}][\text{q}])$
- (vii) Properties that need to be verified: if the vehicles leave the road
- (a) $E \langle \rangle \text{forall } (i : \text{idA}) \text{ automobile}(i).\text{response} = -1$
 (b) $E \langle \rangle \text{forall } (i : \text{idB}) \text{ bus}(i).\text{response} = -1$
 (c) $E \langle \rangle \text{forall } (i : \text{idT}) \text{ tram}(i).\text{response} = -1$
- (viii) Properties that need to be verified: if the time needed to travel through a cell is small or greater than the predefined values, or if it is in a fixed interval
- (a) $E[] \text{ automobile}(\text{idA}).\text{time_per_cellA}[\text{m}] < 27$
 (b) $E[] \text{ automobile}(\text{idA}).\text{time_per_cellA}[\text{m}] > 1/2$
 (c) $E[] ((\text{automobile}(\text{idA}).\text{time_per_cellA}[\text{m}] > 1/2) \ \&\& \ (\text{automobile}(\text{idA}).\text{time_per_cellA}[\text{m}] < 27))$
- (ix) Properties that need to be verified: if the number of vehicles does not exceed the size of each road
- (a) $E[] \text{ novis}[\text{idS}] \geq 0$
 (b) $E[] \text{ novis}[\text{idS}] \leq \text{indexSC}[\text{p}][\text{q}]$
 (c) $E[] ((\text{novis}[\text{idS}] \geq 0) \ \&\& \ (\text{novis}[\text{idS}] \leq \text{indexSC}[\text{p}][\text{q}]))$
 (d) $E[] ((\text{novis}[\text{idS}] \geq 0) \ \&\& \ (\text{novis}[\text{idS}] \leq \text{indexSC}[\text{p}][\text{q}]/3))$
 (e) $E[] ((\text{novis}[\text{idS}] > \text{indexSC}[\text{p}][\text{q}]/3) \ \&\& \ (\text{novis}[\text{idS}] \leq \text{indexSC}[\text{p}][\text{q}] * 2/3))$
 (f) $E[] ((\text{novis}[\text{idS}] > \text{indexSC}[\text{p}][\text{q}] * 2/3) \ \&\& \ (\text{novis}[\text{idS}] \leq \text{indexSC}[\text{p}][\text{q}]))$
 (g) ...
- (x) Properties that need to be verified: determine the load of each road during the simulation process
- (a) $E \langle \rangle ((\text{GLOBAL} > 0 \ \&\& \ \text{GLOBAL} \leq 10) \ \&\& \ ((\text{novis}[\text{idS}] \geq 0) \ \&\& \ (\text{novis}[\text{idS}] \leq \text{indexSC}[\text{p}][\text{q}]/3))$
 (b) $E[] ((\text{GLOBAL} > 0 \ \&\& \ \text{GLOBAL} \leq 10) \ \&\& \ ((\text{novis}[\text{idS}] \geq 0) \ \&\& \ (\text{novis}[\text{idS}] \leq \text{indexSC}[\text{p}][\text{q}]/3))$
 (c) $E \langle \rangle ((\text{GLOBAL} > 0 \ \&\& \ \text{GLOBAL} \leq 10) \ \&\& \ ((\text{novis}[\text{idS}] \geq 0) \ \&\& \ (\text{novis}[\text{idS}] > \text{indexSC}[\text{p}][\text{q}]/3) \ \&\& \ (\text{novis}[\text{idS}] \leq \text{indexSC}[\text{p}][\text{q}] * 2/3))$
 (d) $E[] ((\text{GLOBAL} > 0 \ \&\& \ \text{GLOBAL} \leq 10) \ \&\& \ ((\text{novis}[\text{idS}] \geq 0) \ \&\& \ (\text{novis}[\text{idS}] > \text{indexSC}[\text{p}][\text{q}]/3) \ \&\& \ (\text{novis}[\text{idS}] \leq \text{indexSC}[\text{p}][\text{q}] * 2/3))$
 (e) $E \langle \rangle ((\text{GLOBAL} > 0 \ \&\& \ \text{GLOBAL} \leq 10) \ \&\& \ ((\text{novis}[\text{idS}] \geq 0) \ \&\& \ (\text{novis}[\text{idS}] > \text{indexSC}[\text{p}][\text{q}] * 2/3) \ \&\& \ (\text{novis}[\text{idS}] \leq \text{indexSC}[\text{p}][\text{q}]))$

- (f) $E[] ((\text{GLOBAL} > 0 \ \&\& \ \text{GLOBAL} \leq 10) \ \&\& \ ((\text{novis}[\text{idS}] \geq 0) \ \&\& \ (\text{novis}[\text{idS}] > \text{indexSC}[\text{p}][\text{q}] * 2/3) \ \&\& \ (\text{novis}[\text{idS}] \leq \text{indexSC}[\text{p}][\text{q}]))$
 (g) ...

In these queries inserted in UPPAAL, maximum values for validation variables taken into consideration are the values presented at the beginning of the simulation scenario. The minimum time is the multiplication of 0.5 time units by the number of cells contained in the first column of the $\text{indexSC}[\text{noS}][\text{maxnoCells} + 1]$ Matrix. The maximum time is the multiplication of 27 time units by the number of cells, which is contained in the first column in the $\text{indexSC}[\text{noS}][\text{maxnoCells} + 1]$ matrix.

This structuring modality of the presented simulation scenarios was chosen to prove the capacity of the system to simulate in detail the behaviour of each traffic component in a realistic context. During different verification processes, depending on the reachable states which are taken into account for investigation purposes, the element types of the simulation scenario have to be changed in a corresponding manner. At the same time, because not all the existing traffic-elements influence the considered evolution of the system on the interest state, the number of implied traffic components can decrease significantly, extending the urban area in which the verification process is applicable from the point of view of UPPAAL limitations.

4.5. Extrapolation of Results for Similar Situations. Cellular automata allow the observation of different phenomena, managing to break down components into individual variables. They allow the understanding of how local changes affect the whole grid of cells. The formalism of timed automata, due to their elementary structure, is appropriate for a modular approach, and the resolution (level of detail) and system size obtained (the network size that needs to be covered) are appropriate to the model proposed. In the context of urban traffic, cellular automata based on microscopic models have the capacity to simulate in detail all the elements presented in this environment. The quantity of traffic-elements implemented generates a model containing a large number of evolutionary rules and interactions.

This model has a high flexibility in accordance with the environment where it will be applied, due to the fact that it is implemented on a large group of road-elements and possible interactions. The stimuli created for each vehicle will depend on the road-elements contained in the road and the traffic conditions, due to the modular approach used.

In the context of formal verification, this model presents all the possible scenarios that can occur, even the ones that are physically impossible, and the presented results allow the maximum level flexibility.

It was demonstrated that UPPAAL software is capable of dealing with low/medium complexity models, but for the implementation of high complexity models it is limited, due to the computational power that UPPAAL can access and cannot be extended. Taking its limits into consideration, the application proved to be a success in this context.

5. Conclusions and Future Work

The new road traffic simulation approach presented in this paper is based on the theory of timed automata and UPAAL model-checker. It was able to overcome the disadvantages regarding the lack of adaptability and flexibility of some previous urban road traffic models and to respond to the increasing necessity to verify the behaviour of traffic simulation models.

The behavioural characteristics of a representative and complex urban road traffic system are rendered in a realistic manner, considering the interactions between the infrastructure component and the moving vehicles. A two-layered timed automata based model was developed for this purpose. Unlike existing systems, this approach offers real extension possibilities, easily applied in practice due to the modular components based on matrix structures. They can be extended by adding a corresponding number of rows or columns. All the modifications of the simulated map could be performed simply by changing the initialization variables, without the necessity of modifying the UPAAL model. A case study highlighted some limitations of the UPAAL model-checker, more suitable for low/medium complexity models.

Features of this work can be developed and explored more deeply, providing room for new perspectives in this domain. Taking into consideration the scenario in which this approach will be implemented, some improvements could be made in terms of the specific problem and of the application itself.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This research was carried out under European funding, Structural Projects: POSDRU/89/1.5/S/52603-4D-POSTDOC.

References

- [1] R. Mardiyati, N. Ismail, and A. Faruqi, "Review of microscopic model for traffic flow," *ARPN Journal of Engineering and Applied Sciences*, vol. 9, no. 10, pp. 1794–1800, 2014.
- [2] K.-C. Chu, L. Yang, R. Saigal, and K. Saitou, "Validation of stochastic traffic flow model with microscopic traffic simulation," in *Proceedings of the 2011 7th IEEE International Conference on Automation Science and Engineering, CASE 2011*, pp. 672–677, Italy, August 2011.
- [3] P. J. Gundaliya, T. V. Mathew, and S. L. Dhingra, "Heterogeneous traffic flow modelling for an arterial using grid based approach," *Journal of Advanced Transportation*, vol. 42, no. 4, pp. 467–491, 2008.
- [4] M. Treiber and A. Kesting, "Microscopic calibration and validation of car-following models – a systematic approach," *Procedia - Social and Behavioral Sciences*, vol. 80, pp. 922–939, 2013.
- [5] E. Amini, M. Tabibi, E. R. Khansari, and M. Abhari, "A vehicle type-based approach to model car following behaviors in simulation programs (case study: Car-motorcycle following behavior)," *IATSS Research*, 2018.
- [6] J. Casas, J. Perarnau, and A. Torday, "The need to combine different traffic modelling levels for effectively tackling large-scale projects adding a hybrid meso/micro approach," in *Proceedings of the 14th Meeting of the Euro Working Group on Transp. - In Quest for Adv. Models, Tools, and Methods for Transp. and Logist., EWGT, 26th Mini-EURO Conf. - Intelligent Decis. Making in Transp. and Logist., MEC, 1st Eur. Sci. Conf. on Air Transp. - RH*, pp. 251–262, Poland, September 2011.
- [7] M. Rahman, M. Chowdhury, Y. Xie, and Y. He, "Review of microscopic lane-changing models and future research opportunities," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1942–1956, 2013.
- [8] V. L. Knoop, S. P. Hoogendoorn, Y. Shiomi, and C. Buisson, "Quantifying the number of lane changes in traffic: An empirical analysis," in *Proceedings of the 91st Annual Meeting Transportation Research Board*, Washington, DC, USA, 2012.
- [9] J. Barcelo, D. Garcia, and J. Perarnau, "Methodological notes on combining macro, meso, and micro models for transportation analysis," in *Proceedings of the Workshop on Modeling and Simulation*, 2005.
- [10] E. Bernauer, L. Breheret, S. Algers et al., "Review of micro-simulation models, smartest/D3," in *The "SMARTEST" Project, Simulation Modelling Applied to Road Transport European Scheme Tests*, Institute for Transport Studies, University of Leeds, 1998.
- [11] *VISSIM 5.2 User Manual*, PTV Planung Transport Verkehr AG, Germany, 2010.
- [12] D. Lin, X. Yang, and C. Gao, "VISSIM-based simulation analysis on road network of CBD in Beijing, China," *Procedia - Social and Behavioral Sciences*, vol. 96, pp. 461–472, 2013.
- [13] M. Fellendorf and P. Vortisch, "Microscopic traffic flow simulator VISSIM," in *Fundamentals of Traffic Simulation*, J. Barcelo, Ed., pp. 61–93, Springer, New York, NY, USA, 2010.
- [14] "Surrogate "safety measures from traffic simulation models"," FHWA-RD-03-050, Office of Research, Development, and Technology, Office of Safety, RDT, 2012.
- [15] B. W. Bush, *An Algorithmic Overview of TRANSIMS*, Los Alamos National Laboratories, 2010.
- [16] *TransModeler Users Guide, Traffic Simulation Software*. Caliper, MA, 2008.
- [17] J. Barceló and J. Casas, "Dynamic network simulation with AIMSUN, simulation approaches in transportation analysis recent advances and challenges," in *MasoKuwahara Operations Research/Computer Science Interfaces Series*, R. Kitamura, Ed., pp. 57–98, 31 edition, 2005.
- [18] MITSIMLab, *User's Guide for MITSIMLab and Road Network Editor (RNE)*, Massachusetts Institute of Technology, Intelligent Transportation Systems Program, 2002.
- [19] T. Toledo, C. F. Choudhury, and M. E. Ben-Akiva, "Lane-changing model with explicit target lane choice," *Transportation Research Record*, no. 1934, pp. 157–165, 2005.
- [20] CORSIM, *Advanced CORSIM Training Manual*, Minnesota Department of Transportation, 2004.
- [21] P. Quadstone, "QuadstoneParamics V4.2 analyser reference manual," in *Quadstone Limited*, 2003.
- [22] C. R. Long, T. Yanzhi, and L. Der-Hornig, "Comparison of PARAMICS and GETRAM/AIMSUN microscopic traffic simulation," in *Proceedings of the 83rd Annual Meeting of the Transportation Research Board*, 2003.

- [23] J. Olstam and A. Tapini, "Comparison of Car-Following Models," in *VTI Publication M960A.2004*, Swedish National Road and Transport Research Institute, Linköping, Sweden, 2004.
- [24] S. Panwai and H. Dia, "Comparative evaluation of microscopic car-following behavior," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 3, pp. 314–325, 2005.
- [25] C. Krogscheepers and K. Kacir, "Latest trends in micro simulation: an application of the PARAMICS model," in *Proceedings of the The 20th Annual South African Transport Conference*, South Africa, 2001.
- [26] H. Sbayti and D. Roden, *Best Practices in the Use of Micro Simulation Models Prepared for: American Association of State Highway and Transportation Officials (AASHTO)*, repaired as part of NCHRP Project -36, Task 90, National Cooperative Highway Research Program, Transportation Research Board, 2010.
- [27] E. Ugnenko, E. Uzhvieva, and Y. Voronova, "Simulation of traffic flows on the road network of urban area," *Procedia Engineering* 134, pp. 153–156, 2016.
- [28] J. Shen and X. Jin, "Detailed traffic animation for urban road networks," *Graphical Models*, vol. 74, no. 5, pp. 265–282, 2012.
- [29] L. Li, "Research on traffic flow mathematical model in urban traffic," *International Journal of Smart Home*, vol. 9, no. 11, pp. 123–132, 2015.
- [30] Z. Tian, L. Jia, H. Dong, F. Su, and Z. Zhang, "Analysis of urban road traffic network based on complex network," *Procedia Engineering*, vol. 137, pp. 537–546, 2016.
- [31] M. M. Ishaque and R. B. Noland, "Behavioural issues in pedestrian speed choice and street crossing behaviour: A review," *Transport Reviews*, vol. 28, no. 1, pp. 61–85, 2008.
- [32] G. Waizman, S. Shoval, and I. Benenson, "Micro-simulation model for assessing the risk of vehicle–pedestrian road accidents," *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, vol. 19, no. 1, pp. 63–77, 2013.
- [33] V. Astarita, G. Guido, A. Vitale, and V. P. Giofrè, "A new micro-simulation model for the evaluation of traffic safety performances," *Transport*, no. 51, pp. 1–16, 2012.
- [34] F. Cunto and F. F. Saccomanno, "Microlevel traffic simulation method for assessing crash potential at intersections," in *Proceedings of the 86th Annual Meeting of the Transportation Research Board*, Washington, DC, USA, 2007.
- [35] L. P. Hurd, "Formal language characterizations of cellular automaton limit sets," *Complex Systems*, vol. 1, no. 1, pp. 69–80, 1987.
- [36] H. Zhang, X. Liang, and K. Xu, "VISSIM network simulation under actuated signal control," *Journal of Computational Information Systems*, vol. 10, no. 4, pp. 1573–1580, 2014.
- [37] R. Dowling, A. Skabardonis, J. Halkias, G. McHale, and G. Zammit, "Guidelines for calibration of microsimulation models: framework and applications," in *Proceedings of the 83rd Annual Meeting of the Transportation Research Board*, Washington, DC, USA, 2004.
- [38] B. P. Byungkyu and W. Jongsun, *Microscopic Simulation Model Calibration and Validation Handbook*, Final Contract Report, Traffic Operations Laboratory, Center for Transportation Studies, University of Virginia, 2006.
- [39] Q. Chao, Z. Deng, and X. Jin, "Vehicle-pedestrian interaction for mixed traffic simulation," *Computer Animation and Virtual Worlds*, vol. 26, no. 3-4, pp. 405–412, 2015.
- [40] R. Alur and D. L. Dill, "A theory of timed automata," *Theoretical Computer Science*, vol. 126, no. 2, pp. 183–235, 1994.
- [41] G. Behrmann, A. David, and K. G. Larsen, "A tutorial on uppaal," in *Formal Methods for the Design of Real-Time Systems*, vol. 3185 of *Lecture Notes in Computer Science*, pp. 200–236, Springer, Berlin, Germany, 2004.
- [42] C. Avram, J. Machado, and A. Aştilean, "Modeling and formal analysis of urban road traffic," in *Proceedings of the 11th International Conference of Numerical Analysis and Applied Mathematics 2013, ICNAAM 2013*, pp. 1408–1411, Greece, September 2013.
- [43] E. M. Clarke, E. A. Emerson, and A. P. Sistla, "Automatic verification of finite-state concurrent systems using temporal logic specifications," *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol. 8, no. 2, pp. 244–263.
- [44] E. B. Lima, *Modelos microscópicos para simulação do tráfego baseados em autômatos celulares*, Dissertação de Mestrado em Computação, Universidade Federal Fluminense, Niterói, Brazil, 2014, <http://www2.ic.uff.br/PosGraduacao/Dissertacoes/361.pdf>.
- [45] J. Vasic and H. J. Ruskin, "A discrete flow simulation model for urban road networks, with application to combined car and single-file bicycle traffic," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 6782, no. 1, pp. 602–614, 2011.
- [46] M. Tiriolo, L. Adacher, and E. Cipriani, "An urban traffic flow model to capture complex flow interactions among lane groups for signalized intersections," *Procedia - Social and Behavioral Sciences*, vol. 111, pp. 839–848, 2014.
- [47] J. Shi, L. Cheng, J. Long, and Y. Liu, "A new cellular automaton model for urban two-way road networks," *Computational Intelligence and Neuroscience*, vol. 2014, Article ID 685047, 6 pages, 2014.
- [48] G. A. Wainer, "Cellular modeling with cell-DEVS: a discrete-event cellular automata formalism," in *ACRI 2014, LNCS 8751*, J. Was, G. C. Sirakoulis, and S. Bandini, Eds., pp. 6–15, 2014.
- [49] J. Bengtsson and Y. Wang, "Timed automata: semantics, algorithms and tools," in *Lectures on Concurrency and Petri Nets*, vol. 3098 of *Lecture Notes in Computer Science*, pp. 87–124, Springer, Berlin, Germany, 2004.
- [50] M. U. Iftikhar, J. Lundberg, and D. Weyns, "A model interpreter for timed automata," in *ISO/LA*, pp. 243–258, 2016.
- [51] J. Wiedermann and P. Lukas, "Communicating mobile nanomachines and their computational power," in *Proceedings of NanoBet Conference*, vol. 125, 2008.
- [52] G. Kunz, E. Perondi, and J. MacHado, "Modeling and simulating the controller behavior of an Automated People Mover using IEC 61850 communication requirements," in *Proceedings of the 2011 9th IEEE International Conference on Industrial Informatics, INDIN 2011*, pp. 603–608, Portugal, July 2011.
- [53] L. Evans and R. Rothery, *Experimental Measurements of Perceptual Thresholds in Car following*, Highway Research Record 454, Transportation Research Board, Washington, DC, USA, 1973.

