TRAFFIC CONTROL SPECIFICATIONS USING DISCRETE EVENTS CELLULAR MODELS

Gabriel A. Wainer Alejandra Davidson

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Planta Baja. Pabellón I. Ciudad Universitaria.
(1428) Buenos Aires. Argentina
{gabrielw, ad3n}@dc.uba.ar
http://www.dc.uba.ar/people/proyinv/celldevs

Keywords: Simulation methods: Discrete-event simulation Modeling methodology: DEVS models, Cell-DEVS models. Applications: traffic models.

Abstract

ATLAS is a specification language defined to outline city sections as cell spaces. The goal is that a modeler is able to define complex traffic models in a simple fashion. In this case, extensions allowing control mechanisms in the traffic model is presented. The models are formally specified, avoiding a high number of errors in the developed application, and the problem solving time is highly reduced.

INTRODUCTION

Traffic control is an important area in applications of computer simulation. These models can be used to improve traffic control, to measure the consequences of collisions or men at work, to avoid pollution, traffic jams, etc.

Cellular Automata have been widely used to describe these real systems (Davidson et al. 1999). The paradigm defines a grid of cells using discrete variables for time, space and system states (Wolfram 1986). The cells are updated according with a local rule function that uses a finite set of nearby cells (called the neighborhood of the cell). The update is done synchronously and in parallel. Cell-DEVS (Wainer and Giambiasi 1998) was proposed to describe cell spaces as DEVS models (Zeigler, 1976). This approach improves the definition of the cell spaces, and allows to use explicit timing functions.

Most existing cellular models used for traffic simulation are devoted exclusively to model simple aspects of the traffic flow (Davidson et al. 1999), and several real problems of the traffic cannot be analyzed. Many of the approaches are devoted to analyze the global traffic flow, but are not well suited for microsimulations.

The goal of this work is to present part of a specification language, called ATLAS (Advanced Traffic LAnguage Specifications). The language allows to define sections of cities as cell spaces. The language constructions have been mapped into DEVS and Cell-DEVS models, providing the benefits of a formal approach. We have focused in presenting the constructions used to define behavior of vehicles when railways or traffic controls are included in the models.

BACKGROUND

ATLAS is a specification language built on top of DEVS and Cell-DEVS formalisms. A DEVS model is seen as composed atomic submodels than can be combined into coupled models. A DEVS atomic model is described as:

$$M = < I, X, S, Y, \delta_{int}, \delta_{ext}, \lambda, D >$$

I is the model's interface, ***X*** is the input events set, **S** is the state set, and ***Y*** is the output events set. There are also several functions: $\boldsymbol{\delta_{int}}$ manages internal transitions, $\boldsymbol{\delta_{ext}}$ external transitions, λ the outputs, and **D** the elapsed time. A DEVS coupled model is defined as:

$$CM = < I, X, Y, D, \{M_i\}, \{I_i\}, \{Z_{ij}\} >$$

Here, ***I*** is the model's interface, ***X*** is the set of input events, and ***Y*** is the set of output events. **D** is an index of components, and for each $i \in D$, $\mathbf{M_i}$ is a basic DEVS model, where $M_i = < I_i, X_i, S_i, Y_i, \delta_{inti}, \delta_{exti}, ta_i >$. $\mathbf{I_i}$ is the set of influencees of model i. For each $j \in I_i$, $\mathbf{Z_{ij}}$ is the i to j translation function.

Cell-DEVS is an extension of DEVS models, especially devoted to define cellular models. Each cell is defined as an atomic DEVS model, and a procedure to couple cells is depicted. Timing delays allow to define different timing behavior. The atomic models (the cells) can be described as:

$$TDC = < I, X, Y, \theta, N, delay, d, \delta int, \delta ext, \tau, \lambda, D >$$

This work was partially supported by ANPCYT Project 11-04460 and UBACYT Project JW10.

X defines the external inputs, ***Y*** the external outputs, and ***I*** is the interface of the model. θ is the cell state definition, and **N** is the set of inputs. **Delay** defines the kind of delay for the cell, and **d** its duration. Finally, there are several functions: $\delta_{\mathbf{int}}$ for internal transitions, $\delta_{\mathbf{ext}}$ for external transitions, τ for local computations, λ for outputs and **D** for the state's duration. Each cell takes the set of inputs and computes the cell's future state using the τ function. The delay allows to defer the transmission of the results. This behavior is defined by the $\delta_{\mathbf{int}}$, $\delta_{\mathbf{ext}}$, λ and **D** functions. The modeler only must focus in defining the local computing function, the kind of delay and its length. The remaining parameters are defined by the formalism.

A Cell-DEVS coupled model is defined by:

$$GCC = < X_{list}, Y_{list}, I, X, Y, n, \{t_1,...,t_n\}, N, C, B, Z >$$

Here, $\mathbf{X}_{list}$ and $\mathbf{Y}_{list}$ are the input/output coupling lists, used by ***I*** to define the interface of the model. ***X*** and ***Y*** represent the input/output events. The **n** value defines the dimension of the cell space, $\{\mathbf{t_1,...,t_n}\}$ is the number of cells in each dimension and **N** is the neighborhood set. The cell space is defined by **C**, together with **B,** the set of border cells, and **Z** the translation function. For coupled models, the modeler only has to focus in the neighborhood shape, the size and dimension of the model, the definition of the border set, and the coupling lists.

In ATLAS, a city section is specified by a set of streets connected by crossings. The vehicles advance surpassing slower vehicles, up to their arrival to a crossingThe roads are defined by:

$$RoadSections = \{ (p1, p2, n, a, dir, max) \;/\; p1, p2 \in City \wedge n, max \in \mathbb{N} \wedge a, dir \in \{0, 1\} \}$$

Here, *p1* and *p2* represent the boundaries of each section, belonging to *City* = { (x,y) / x, y $\in \mathbb{Z}$}. Then, *n* defines the number of lanes in the segment. After, *dir* represents the vehicle direction. Variable *a* defines the shape of the section (straight or curve). Finally, *max* is used to define the maximum speed allowed in the segment.

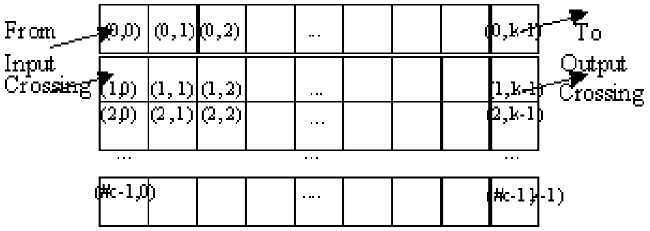


Figure 1. A road section with more than four lanes.

In (Davidson and Wainer 1999) a complete set of rules was defined for streets with one to five (or more) lanes. Different models were included because in each case a different behavior is defined for the borders. The speed of each vehicle is represented by a transport delay. This delay is generated using a random function related with the maximum speed allowed in the street.

The Crossings set represents places of the City where the road sections joins. Each crossing can connect any number of road sections, which will be its inputs or outputs. The set of crossings is defined as follows:

$$Crossings = \{ c \;/\; \exists\, t, t' \in RoadSections \wedge t = (p1, p2, n, a, dir, max) \wedge t' = (p1', p2', n', a', dir', max') \wedge t \neq t' \wedge (p1 = c \vee p2 = c) \wedge (p1' = c \vee p2' = c) \}$$

Every crossing in this set is represented as a ring of cells where the vehicles advance. In any given instant, they can get to another path, using the proposal defined in (Chopard et al. 1997). A car in the intersection has higher priority to obtain a position into the ring than the cars outside the crossing. The cars advance continuously in order to avoid deadlocks.

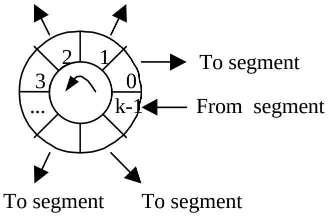


Figure 2. Crossing

Once the components of a city section are defined, complex behavior can be analyzed by using other constructions of ATLAS. Definitions for traffic lights, railways, men at work, street holes, transit signals and parked cars are included. Finally, special behavior has been defined for special vehicles: trucks, vans and high priority cars (ambulances, police, firefighters). The following sections will be devoted to present the behavior for road sections and crossings with different constraints in the traffic flow.

RAILWAYS

The existence of railways in a city section can influence the vehicle flow due to the level crossings. Railways models have been defined, and they are built as a sequence of level crossings overlapped with the road sections. Each train is defined as a Cell-DEVS following a predefined advance sequence through level crossings. The railway network is defined by:

RailNet = { Rail / Rail ∈ RailTrack }, where RailTrack =
{ (s, δ, seq) / s∈ RoadSections ∧ δ∈ N ∧ seq∈ N }

Every RailTrack ∈ RailNet represents a part of the rail network in a given city section. The shape of the network is defined by the places where level crossings are placed. Every tuple pn = (s, δ, seq) identifies the position of the level crossing. First, the road section crossed (s); then, the distance between the beginning of the section and the railway (δ), and finally, a sequence number defining the position of the crossing in the railway. This sequence starts in zero, and finishes in the number of tuples in the set.

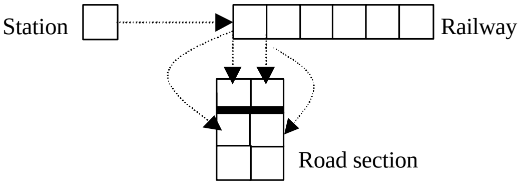


Figure 3. Level crossing definition.

For every element in RailNet, two models are defined. The first one is a DEVS, representing a station from where the trains depart. The second is a unidimensional Cell-DEVS whose cells model all the level crossings. The associated station is defined as a DEVS model. It represents the train departure according to given frequency. It is connected with a RailTrack model, defined as an unidimensional Cell-DEVS with transport delays. Every cell in the model is specified as:

$$C_{ij} = < I, X, S, Y, N, \delta int, \delta ext, delay, d, \tau, \lambda, D >$$

I = <η, Px, Py>, with η = 2; Px = { (X1, binary), (X2, binary) }; Py = { (Y1, binary), (Y2, binary) }.

X = Y = {0, 1};

S:

$$s = \begin{cases} 1 \text{ if a train is passing;} \\ 0 \text{ otherwise.} \end{cases}$$

N = { (0,-1), (0,0) } ;

delay = transport;

d = dtrain ();

λ, δint and δext are defined by Cell-DEVS;

τ is defined as:

τ(N)	N
0	(0,0) = 1
1	(0,-1) = 1
(0,0)	TRUE /* Otherwise it preserves the present state */

The cell's state represents the presence of a train that advances independently of the cells in front of it. Therefore, the neighborhood only includes the back cell. Transport delays define the train speed, using *dtrain()*. This function computes the delay of the train between crossings, depending on the distance between them. The train movement is defined by three rules. The first one represents a train leaving the crossing. The second one represents an empty crossing that is occupied by a train. The last one preserves the present state in every other case. A coupled model of the railway is built using the following definition:

$$TrainTrack(k, Out) = < X_{list}, Y_{list}, I, X, Y, n, \{t1,...,tn\}, \eta, N, C, B, Z, select >$$

Y_{list} = {(0,i) / i ∈ N ∧ i ∈ [0, k] }; X_{list} = { (0,0) }.

I=<P_x,P_y>, with P_x = {<$X_{\eta+1}$(0,0), binary>}, P_y = {<$Y_{\eta+j}$(0,i), binary>/ i ∈ N ∧ i ∈ [0, k] ∧ (i, #p) ∈ Out ∧ j ∈ [1, #p] } .

X = Y = { 0, 1 } ;

n = 1; t1 = k; η = 2; N = { (0,-1), (0,0) }

C = { C_{ij} / i = 0 ∧ j ∈ [0, k-1] }, with C_{ij} defined earlier;

B = { (0,0) };

Z is built using the definition given by Cell-DEVS.

select = { (0,0), (0,-1) }

This model represents a railway layout. Here, k represents the number of level crossings, and it is obtained counting the elements of the set (k = #RailTrack). The Out set contains the number of ports to be coupled to the road sections. It includes two output ports per road. One of them allows to see if a train has arrived to the crossing. The other one informs that a train is leaving. It is built as: Out = { (i, 2n) / i ∈ N ∧ i ∈ [0,k] ∧ ((c1, c2, n, a, dir, max), δ, i) ∈ RailNet}.

The input coupling of this model is defined through the cell (0,0), which is connected with a station. Then, the behavior of this border cell is different. The following parameters are changed:

η = 1; N = { (0,0) }

τ(N)	N
0	(0,0) = 1
1	portvalue(x-s-train) = 1
(0,0)	TRUE

The input port *x-s-train* is used to receive a new departure from the station.

A road section crossed by a railway should reflect the existence of the crossing. To define which cells in the road are affected, we compute col = ⌈δ / cell_size⌉, obtaining Cpn = { (i, col) / i ∈ [0,n-1] } ∪ { (i, col+1) / i ∈ [0,n-1] }, where n is the number of lanes in s. The Cell-DEVS model of the road section s (M_s) is modified by changing the behavior of the cells in the columns defined by Cpn. In these cells, a new output port will reflect the existence or absence of a train.

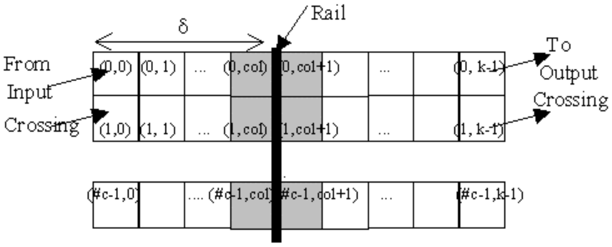


Figure 4. Road section with level crossing.

The cells in { (i, col) / i ∈ [0,n-1] } must reflect a stop in the straight movement when a train is crossing, as follows:

Delay: inertial; Delay length: dtrack().

τ(N)	N
0	(0,0) = 1 and (0,1) = 0 and portvalue(x-R-train) = 0
1	(0,-1) = 1 and (0,0) = 0
(0,0)	TRUE /*Otherwise: state unchanged */

The port *x-R-train* informs that a train is passing through the level crossing. If its state is 0, the rails are free. The delay function *dtrack* depends on the expected time for a train in the crossing. The inertial delay allows to define that the vehicles only advance into the crossing if it is empty for that period. Otherwise, the previous state is preempted, and the cars wait for the passing train.

The cells after the rails { (i, col+1) / i ∈ [0,n-1] }, are also affected because we should verify that no trains are crossing the railways before vehicles advance to them:

Delay: inertial; Delay length: dtrack

τ(N)	N
0	(0,0) = 1 and (0,1) = 0
1	(0,-1) = 1 and (0,0) =0 and portvalue(x-R-train) = 0
(0,0)	TRUE /*Otherwise: state unchanged */

Finally, we define the coupling between rails and road sections. In this case, for every R ∈ RailNet, a station and a TrainTrack models are built (M_E and M_R). Every (s, δ, seq) ∈ R generates the following influences: I_E = { M_R }, I_R = { M_s }. The coupling between the cellular models is defined using the parameters col (earlier computed), and n (number of lanes of the model M_s), resulting in:

$Z_{E,R}$: y-R-train$_E \rightarrow X_{\eta+1}(0,0)_R$;

$Z_{R,s}$: $Y_{\eta+j}(0,seq)_R \rightarrow X_{\eta+3}(j-1,col)_s$ $\forall j \in \mathbb{N}$, $j \in [1, n]$;

$Z_{R,s}$: $Y_{\eta+n+j}(0,seq)_R \rightarrow X_{\eta+3}(j-1,col+1)_s$ $\forall j \in \mathbb{N}$, $j \in [1, n]$.

CONTROL SIGNALS

This section includes several constructions that constrain the vehicle movements. They allow a modeler to analyze traffic behavior under real situations. They can be used to improve the programming of traffic lights, avoiding traffic jams, etc.

Traffic Lights

In ATLAS, the crossing with a traffic lights is defined using the following construction:

TLCrossings = { c / c ∈ Crossings }.

Each element in this set is mapped into one DEVS, transmitting the light color to the corresponding road section in the intersection. Another DEVS is in charge of synchronizing all the lights in the corner. An upper level Cell-DEVS can be built to coordinate all the controllers in a city section.

For every c ∈ TLCrossings, the DEVS model Sync(#tl) = < I, X, S, Y, δint, δext, λ, D > is created. This controller defines the color of each traffic light. The parameter #tl defines the number of traffic lights, depending on the number of input road sections:

T_{in} = { s / s ∈ RoadSections ∧ s = (c1, c2, n, a, dir, max) ∧
[(c1 = c ∧ dir = 0) ∨ (c2 = c ∧ dir = 1)] }

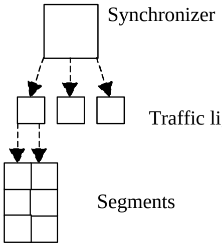


Figure 5. Crossing with traffic lights.

For each t ∈ T_{in}, a DEVS model representing the traffic light is built: TL(#c) = < I, X, S, Y, δint, δext, λ, D >. This model informs its color to an input road section. Specification details of these models can be found in (Davidson and Wainer 1999).

Every input road section of the crossing should reflect the existence of the traffic light. The basic rules for the road sections were modified to represent this behavior. The new model is defined by:

M_R = <X_{list}, Y_{list}, I, X,Y, n, {t_1,t_2}, η, N, C, B, Z, select>

Specialized ports inform the color of the traffic light to the cells in the road section. If its value is 1, it is red. Otherwise, it is green. These border cells are defined by $\{(i, t_2\text{-}1) / i \in [1,t_1]\}$.

Delay: inertial. Duration: speed.

τ(N)	N
1	(0,-1) = 1 and (0,0) = 0
0	(0,0) = 1 and portvalue(x-c-space) = 0 and portvalue(x-light) = 0 send(1, y-c-car)
(0,0)	TRUE /*Otherwise: state unchanged */

These rules represent that a vehicle can cross if there is space (*portvalue(x-c-space) = 0*) and the traffic light is green (*portvalue(x-light) = 0*). Then, a 1 value is sent to the crossing telling that the car will is pass. An inertial delay is used because a car can enter the crossing if there is a free space and the light is green during the time specified. Otherwise, the car must remain waiting.

Finally, $\forall$ c $\in$ Crossings we must define the cells' coupling. Crossing(c, k, In, Out) defines a crossing of k cells, where the set In defines are input positions, and Out their outputs. The number of cells in the crossing (k) is defined counting the number of input lanes. These ones depend on the road sections connected to it and their direction. We have defined a unique ordering for the input road sections, such that close roads are connected to neighboring cells. This ordering is built using the incidence angle between the road section and the line y = 0.

Once the In and Out sets are defined, every pair $(s,j) \in$ In tells that the road section s is connected to an input using the j-eth cell in the crossing. This numbering of the road sections is also used to establish the order to receive the green lights.

The synchronizer influences the TL models corresponding to the input road sections for the crossing c. That is, $I_{sync} = \{TL_i / i \in [0, \#(In)]\}$. In addition, every traffic light influences the behavior of the input road section, according to the established order. That is, $I_{TLi} = \{ M_s / (s,j) \in In \wedge i = \#(\{(s',j') \in In / j' < j \})\}$.

Therefore, the coupling function is defined as

$Z_{sync,TLi}$: $(y\text{-}ligth_i)_{sync} \rightarrow (x\text{-}light_i)_{TLi}$, $\forall\ i \in \mathbb{N}$, $i \in [0,\#(In)]$

$Z_{Tli,s}$: $(y\text{-}s\text{-}light_h)_{Tli} \rightarrow X\eta+2(h, k\text{-}1)_s$, $\forall\ (h \in \mathbb{N}, h \in [0, n\text{-}1])\wedge (s,j) \in In \wedge i = \#(\{(s',j') \in In / j' < j \})$. Here, n is the number of lanes in s (n = t1 of the model Mt), and k is the number of cells in every lane (k = t2 of the model Mt).

Men at work

The presence of men at work produces that the vehicles cannot advance in a road. These traffic obstructions are specified as:

$$\text{Jobsite} = \{ (s, ni, \delta, \#n) / s \in \text{RoadSections} \wedge s = (c1, c2, n, a, dir, max) \wedge ni \in [0, n\text{-}1] \wedge \delta \in \mathbb{N} \wedge \#n \in [1, n\text{+}1\text{-}ni] \wedge \#n \equiv 1 \bmod 2 \}$$

Every tuple o = (s, ni, δ, #n) defines the road section where the construction is being done. It includes the first lane affected, the distance between the center of the jobsite and the beginning of the section, and the number of lanes occupied. These values are used to define a rhombus over the road section where the vehicles cannot advance. The cars arriving to the jobsite must deviate.

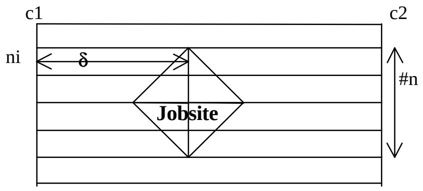
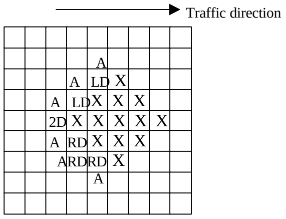


Figure 6. Road section with men at work.

A group of different models has been defined, according with the number of lanes, the size and position of the jobsite. Different border conditions were defined in every case. The basic behavior for these models was defined as a Cell-DEVS representing the working site. We use different behavior for the cell in the rhombus and those before it. The first ones will always have a 0 value. The other ones should avoid to advance into the jobsite.



				A				
			A	LD	X			
		A	LD	X	X	X		
		2D	X	X	X	X	X	
		A	RD	X	X	X		
		A	RD	RD	X			
				A				

Figure 7. Jobsite cells.

The cells before the jobsite have been grouped according to the movements that are allowed for the vehicles passing through them. Vehicles can only move to the left diagonal, the right diagonal, or advance in a straight line. Therefore, different rules have defined for each case. For instance, the cells whose movement to both diagonals is allowed are defined by:

$$C_{ij} = < I, X, S, Y, N, \delta_{int}, \delta_{ext}, \text{delay}, d, \tau, \lambda, D >$$

I = < η, P^X, P^Y>, with η = 9;

P^X = { (X_1, binary), (X_2, binary), (X_3, binary), (X_4, binary), (X_5, binary), (X_6, binary), (X_7, binary), (X_8, binary), (X_9, binary) };

P^Y = { (Y_1, binary), (Y_2, binary), (Y_3, binary), (Y_4, binary), (Y_5, binary) , (Y_6, binary), (Y_7, binary), (Y_8, binary), (Y_9, binary) }.

X = ***Y*** = {0, 1};

S:

$$s = \begin{cases} 1 \text{ if there is a vehicle in the cell;} \\ 0 \text{ otherwise.} \end{cases}$$

N = { (0,0), (0,1), (1,0), (1,1), (0,-1), (1,-1), (-1,1), (-1,-1), (-1,0) };
delay = transport;
d = speed(max);
λ, δint and δext are defined by Cell-DEVS;
τ is defined as:

τ(N)	N
1	(0,-1) = 1 and (0,0) = 0 or (0,0)=0 and (0,-1)=0 and (-1,-1)=1 and (-1,0)=1 or (0,0) = 0 and (0,-1) = 0 and (1,-1) = 1 and (1,0) = 1
0	(0,0) = 1 and (0,1) = 0) or (0,0) = 1 and (1,1) = 0 and (1,0) = 0 or (0,0) = 0 and (0,-1) = 0 and (-1,-1) = 1
(0,0)	TRUE /*Otherwise: state unchanged */

Traffic signs

A construction to specify speed control signals was defined. This construction is defined by:

Control = { (s, t, δ) / s ∈ RoadSections ∧ δ ∈ N ∧ t ∈ {bump, depression, school, pedestrian crossing, others} }

Each tuple identifies the road section, the kind of control element, and the distance up to it in the section.

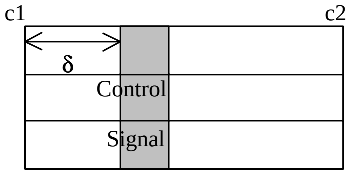


Figure 8. Definition of a control signal.

The cells where a control element has influence has changed by increasing the length of the cell's delay. In this way, a slower speed of the cars is represented.

CONCLUSION

We have presented an approach providing an application oriented specification language. It allows the definition of complex traffic behavior using simple rules. The constructions are mapped into DEVS and Cell-DEVS. These models are formally specified, avoiding errors in the application. Therefore, the problem solving time is highly reduced, allowing the analysis complex traffic behaviors.

At present, the specification language is being implemented. In addition, a graphical user interface is being defined, to allow easy definition of the models. Finally, efficient execution of the models is being considered by means of parallel execution of the Cell-DEVS models. We are also defining new extensions to the definition language, to include routing information, behavior of the drivers, and avoiding congestion mechanisms.

REFERENCES

Davidson, A.; Díaz, A.; Vázquez V. and Wainer, G. "A comparative study of cellular automata application for simulation of urban traffic models". Technical Report 99-005, Departamento de Computación, FCEN/UBA. Submitted. 1999

Davidson, A. and Wainer, G. "A specification language for traffic modelling and simulation". Technical Report 99-012, Departamento de Computación, FCEN/UBA. 1999

Chopard, B.; Dupuis, A. and Luthi, P. “A Cellular Automata Model for Urban Traffic and its applications to the city of Genoa”. Proceedings of Traffic and Granular Flow. 1997.

Wainer, G. and Giambiasi, N. "Specification, modelling and simulation of timed Cell-DEVS models". Technical Report 97-007, Departamento de Computación, FCEN/UBA. zSubmitted to publication. 1998.

Wolfram, S. Theory and applications of cellular automata. Vol. 1, Advances Series on Complex Systems. World Scientific, Singapore, 1986.

Zeigler, B. Theory of modeling and simulation. Wiley, 1976.