# A specification language for traffic modeling and simulation

**Alejandra Davidson    Gabriel Wainer**

{ad3n,gabrielw}@dc.uba.ar

*Departamento de Computación*
*Facultad de Ciencias Exactas y Naturales*
*Universidad de Buenos Aires*
*Pabellón I - Ciudad Universitaria*
*Buenos Aires (1428) – ARGENTINA*

## ABSTRACT

A specification language is defined to outline sections of cities as cell spaces. The characterization of the streets includes traffic direction, number of lanes, etc. Once the urban section is outlined, the traffic flow is automatically set up. The models are formally specified, avoiding a high number of errors in the developed application, and the problem solving time is highly reduced. Therefore, a modeler can concentrate in the problem to solve, instead of being in charge of defining a complex simulation system.

**Keywords:** DEVS models, Cellular Automata, Cell-DEVS models, traffic simulation.

## 1. INTRODUCTION

The traffic control is an important area in the field of computer simulation. These models allow the definition of new traffic rules, to test the effect of traffic control signals. It also permits to measure the consequences of collisions or men at work, to control pollution, avoiding bottlenecks, and so on.

A traffic simulation model should permit to define streets, avenues, crossings, semaphores, railways, traffic signs, and every aspect affecting the traffic. When a higher number of features of the traffic is included in the model, the simulated behavior is more accurate, and the results more precise.

Cellular Automata is formalism well suited to describe this problem. A model built using this formalism uses discrete variables for time, space and system states. The states in the lattice are updated according with a local rule in a simultaneous and synchronous way. The cell states change in discrete time steps according with a local computing function. The function considers the present cell's state and a finite set of nearby cells (called the cell's neighborhood). Several works have been proposed using cellular automata for traffic simulations. Besides, in most existing proposals the precision is reduced and the semantic is poor. Most of them are devoted exclusively to model simple aspects of the traffic flow, and real problems of the traffic cannot be analyzed.

Besides these problems, the cellular automata are synchronous, a fact that poses several constraints in the timing precision for the models. Also, compute time can be wasted. The use of a parallel approach can lead to serial execution due to the characteristics of the model. The Cell-DEVS [1] paradigm was proposed to solve this kind of problems. Cellular models can be described as discrete event models. Moreover, the inclusion of transport and inertial delays allow the accurate modeling of the vehicle movement in a two-dimensional space.

The goal of this work is to present the definition of a specification language used to define sections of cities as cell spaces. The streets can be defined, analyzing the traffic direction, number of lanes, traffic lights, etc. Once defined the urban section, the traffic flow is automatically set up. Therefore, a modeler can concentrate in the problem to solve, instead of being in charge of defining a complex simulation system.

## 2. SPECIFYING STREETS

This section will be devoted to present the definition of a specification language based on the Cell-DEVS formalism. The language will be defined as a set of components allowing the definition of traffic behavior.

The constructions of this language have been mapped into DEVS [2] and Cell-DEVS models. In this way, the benefits of a formal approach as a base can be obtained. Errors in the simulation can be detected by analyzing the specification, without considering the underlying software system.

A Timed Cell-DEVS atomic model can be formally described as:

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

**I** = $<\eta, \mu, P^X, P^Y>$ is the model's interface, where $\eta \in \boldsymbol{N}$ is the neighborhood size, $\mu \in \boldsymbol{N}$ is the number of other ports, and for $i =$ X | Y, $P^i$ is the definition of an Input/Output port,

$\boldsymbol{X} = \{0, 1, \phi\}$ is the set of external input events;

$\boldsymbol{Y} = \{0, 1, \phi\}$ is the set of external output events;

**S** is the state sets, with

$S = \{ (s, phase, \sigma queue, \sigma) \ / \ s \in \{0,1,\phi\},\ phase \in \{active, passive\},\ \sigma queue = \{ ((v_1,\sigma_1),...,(v_m,\sigma_m)) \ / \ m \in \boldsymbol{N} \wedge \forall\ i \in [1,m],\ i \in \boldsymbol{N},\ \sigma_i \in \boldsymbol{R_0}^+ \cup \infty,\ vi \in \{0, 1, \phi\} \},$ and $\sigma \in \boldsymbol{R_0}^+ \cup \infty \}$ ; for Transport Delays models, or

$S = \{ (s, phase, f, \sigma) \ / \ s \in \{0,1,\phi\},\ phase \in \{active, passive\},\ f \in \{0,1,\phi\},\ y\ \sigma \in \boldsymbol{R_0}^+ \cup \infty \}$; for inertial delays.

$\mathbf{N} \in \{0, 1, \phi\}^{\eta+\mu}$, is the set of input values for a cell;
$\mathbf{d} \in \boldsymbol{R_0}^+$ is the delay associated with the cell;
$\delta_{int}$: S→S is the internal transition function;
$\delta_{ext}$: QxX → S is the external transition function; $Q = \{ (s, e) / s \in S, \text{ and } e \in [0, D(s)]\}$;
$\tau$: N → S is the local computation function;
$\lambda$: S→$Y$ is the output function; and
**D**: $S \rightarrow \boldsymbol{R_0}^+ \cup \infty$, is the life duration for a state.

A Cell-DEVS coupled model can be defined as:

$$GCTD = < I, X, Y, Xlist, Ylist, \eta, N, \{m, n\}, C, B, Z, select >$$

**I** = $< P^X, P^Y >$ represents the interface definition for the modular model. In this case, for $i =$ X | Y, $P^i$ is a port definition;
$\boldsymbol{X} = \{0, 1, \phi\}$ is the set of external input events;
$\boldsymbol{Y} = \{0, 1, \phi\}$ is the set of external output events;
**Ylist** = $\{(k,l) \ / \ k \in [0,m],\ l \in [0,n]\}$ is the output coupling list;
**Xlist** = $\{ (k,l) \ / \ k \in [0,m],\ l \in [0,n]\}$ is the input coupling list;
$\eta \in \boldsymbol{N}$ is the neighborhood size;
**N** is the neighborhood's shape, defined as $N = \{ (i_p,j_p) \ / \ \forall\ p \in \boldsymbol{N},\ p \in [1,\eta] \Rightarrow i_p, j_p \in \boldsymbol{Z} \wedge i_p, j_p \in [-1, 1] \}$;
$\{\mathbf{m}, \mathbf{n}\} \in \boldsymbol{N}$ is the cell space size;
**C** is the cell space, defined as $C = \{C_{ij} \ / \ i \in [1,m],\ j \in [1,n]\}$, where

$$C_{ij} = < I_{ij}, X_{ij}, Y_{ij}, S_{ij}, N_{ij}, d_{ij}, \delta int_{ij}, \delta ext_{ij}, \tau_{ij}, \lambda_{ij}, D_{ij}>$$

is a Cell-DEVS component;

**B** is the set of border cells, with
- $B=\{\varnothing\}$ if the cell space is "wrapped", and
- $B=\{C_{ij} \ / \ \forall(i=1 \vee i=m \vee j=1 \vee j=n) \wedge C_{ij} \in C\}$ if the border cells have different behavior of the others, where

$$C_{ij} = < I_{ij}, X_{ij}, Y_{ij}, S_{ij}, N_{ij}, d_{ij}, \delta int_{ij}, \delta ext_{ij}, \tau_{ij}, \lambda_{ij}, ta_{ij}>$$

is a Cell-DEVS component.

**Z** is the translation function, defined as:
Z: $P_{ij}{}^{Y_q} \rightarrow P_{kl}{}^{X_q}$, where $P_{ij}{}^{Y_q} \in I_{ij}$, $P_{kl}{}^{X_q} \in I_{kl}$, $q \in [0,\eta]$ and $\forall(f,g) \in N$, $k = (i+f) \bmod m$; $l=(j+g) \bmod n$;
$P_{kl}{}^{Y_q} \rightarrow P_{ij}{}^{X_q}$, where $P_{kl}{}^{Y_q} \in I_{kl}$, $P_{ij}{}^{X_q} \in I_{ij}$, $q \in [0,\eta]$ and $\forall(f,g) \in N$, $k = (i-f) \bmod m$; $l = (j-g) \bmod n$; and

**select** = $\{(k,l) /(k,l)\in N\}$is the tie-breaking function.

A city section is specified by a set of streets connecting two crossings. The vehicles advance in a right line (surpassing slower vehicles), up to their arrival to a crossing. The speed of each vehicle is represented through a transport delay. A random function is used to define the delay.

Each street is represented as a sequence of segments. They represent a section of one block of length, where every lane has the same traffic direction (one way) and maximum velocity allowed. Consequently, to build a two-way street is necessary to define one segment for each direction.

**Segments**

Each segment is defined as a one-dimensional Cell-DEVS space. The behavior is different for two-track streets, and for streets with more than two tracks, due that it is necessary to define special behavior for each border.

Segments can be specified as:

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

Then, a segment t is a tuple with 6 elements:
t = (p1, p2, n, a, dir, max)
where,

- p1 and p2 represent the boundaries of each segment, and are included in the Crossing set, defined as City = $\{ (x,y) / x, y \in Z \}$;
- $n \in N$, is used to define the number of lanes in the segment;

- dir $\in$ {0,1}, represents the vehicle direction. If dir = 1 vehicles move towards p2; otherwise they go to p1;
- a $\in$ {0, 1}, defines the shape of the segment. Here,
  a = 0 $\Rightarrow$ the segment is a straight line from p1 to p2.
  a = 1 $\Rightarrow$ the segment is curve.
- max $\in$ N, is used to define the maximum speed allowed in the segment.

The following figure shows two segments, one straight and the other curve:

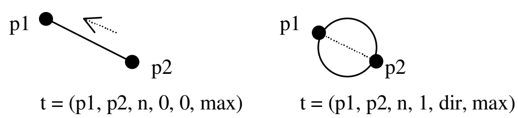


**Figure 1. Specification of a segment.**

Following, the definition of several types of segments is included. Each segment is supposed to be coupled with a crossing in each of the segment boundaries. In this case only one-lane models and general models (more than 4 lanes) will be presented. In [3], several other models are presented, including an experimental framework that allows the definition for traffic flow generation.

**One lane segments**

A one lane segment t = (p1, p2, 1, a, dir, max) is defined as a one-dimensional Cell-DEVS model with transport delays.

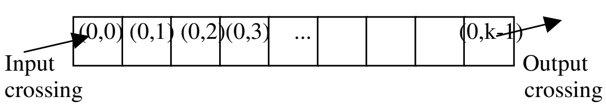


**Figure 2. One lane segment.**

Each cell can be defined as:

$C_{0j}$ = < I, $X$, S, $Y$, N, $\delta_{int}$, $\delta_{ext}$, delay, d, $\tau$, $\lambda$, D >

**I** = <$\eta$, $P^X$, $P^Y$>, with $\eta$ = 3;
$P^X$ = { ($X_1$, binary), ($X_2$, binary), ($X_3$, binary) };
$P^Y$ = { ($Y_1$, binary), ($Y_2$, binary), ($Y_3$, 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,-1), (0,0), (0,1) };

**delay** = transport;

**d** = random();

| | | |
|---|---|---|
| (0,-1) | (0,0) | (0,1) |

**Figure 3. Cell's Neighborhood.**

$\lambda$, $\delta_{int}$ y $\delta_{ext}$ behaves as defined in the Cell-DEVS formalism with transport delays [1].

$\tau$: S x N $\rightarrow$ S is defined as follows:

| New state | Neighborhood state |
|---|---|
| 1 | (0,-1) = 1 and (0,0) = 0 |
| 0 | (0,0) = 1 and (0,1) = 0 |
| (0,0) | TRUE /*Otherwise: state unchanged */ |

Here, the transport delays are used to model the time used by a vehicle to abandon one cell and get into the next one. It depends on the present speed of the vehicle, and it is generated using a random function.

The local computation defines the movement of a vehicle. The first two rules represent the linear movement of a vehicle. The third one represents a cell conserving the present value.

The coupled model corresponding with the segment is defined as:

TC1(k, max) = < Xlist, Ylist, I, $X$, $Y$, n, {$t_1$,...,$t_n$}, $\eta$, N, C, B, Z, select >

**Ylist** = **Xlist** = { (0,0), (0,k-1) }

**I** = <$P^x$, $P^y$>
$P^x$ = {<$X_{\eta+1}$(0,0), binary>, <$X_{\eta+1}$(0,k-1), binary>}
$P^y$ = {<$Y_{\eta+1}$(0,0), binary>, <$Y_{\eta+1}$(0,k-1), binary>}

These ports will be called as follows:

| Port | Name |
|---|---|
| $X_{\eta+1}$(0,0) | x-c-vehicle |
| $X_{\eta+1}$(0,k-1) | x-c-space |
| $Y_{\eta+1}$(0,0) | y-c-space |
| $Y_{\eta+1}$(0,k-1) | y-c-vehicle |

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

**n** = 1

$\mathbf{t_1}$ = k

$\eta$ = 3

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

**B** = { (0,0), (0,k-1) }

**Z** is built using the basic definitions of the Cell-DEVS formalism.

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

The parameters k and max are used to build the coupled model. The maximum speed is used as a parameter for the random number generation, and the k parameter is obtained computing the length of the segment (using the segment boundaries and the cell's size).

The cells (0,0) and (0, k-1) compose the external interface of the model. Each of them must interchange information with the crossings. The behavior of the cell (0,0) is defined as:

$\eta = 2$

**N** = { (0,0), (0,1) }

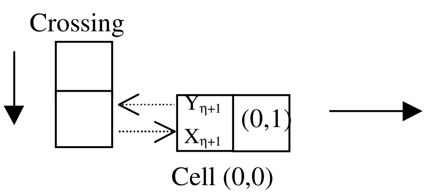


**Figure 4. Neighborhood/coupling for the cell (0,0)**

The function $\tau$ for this cell is defined as it was defined for the rest of the cell space, and the first rule is changed by:

| New state | Neighborhood state |
|---|---|
| 1 | ( portvalue(x-c-vehicle) = 1 and (0,0) = 0 ) |

The cell is coupled with another cell belonging to the crossing, as it can be seen in the Figure 4. Here, the function *portvalue* is used to return the present value of a given port. Whenever the port x-c-vehicle is 1, there is a vehicle wanting to get out from the crossing. Therefore, this rule is used to represent the arrival of a new car to the segment.

The cell (0,k-1) must be redefined using a neighborhood and behavior different from the rest. Here,

$\eta = 2$

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

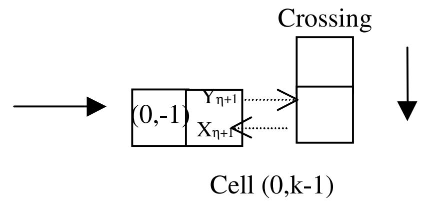


**Figure 5. Neighborhood/coupling for the cell (0,k-1)**

The function $\tau$ is defined like in the rest of the cell space, excepting for the second rule, specified by:

| New state | Neighborhood |
|---|---|
| 0 | ( (0,0) = 1 and portvalue(x-c-space) = 0 )<br>send(1, y-c-vehicle) |

Delay = inertial.

Here, the function *send* is used to define the new state of the selected external ports. When this function is not defined explicitly, it is assumed that all the output ports are updated using the new cell's value.

As it can be seen in the Figure 5., the cell (0,k-1) is coupled with the first cell of the crossing. The port x-c-space tells if there is space in the input cell of the crossing. Whenever this cell and the previous one in the crossing are empty, a car can get into the crossing. In this way, the cars in the crossing are modeled with higher priority than the cars trying to get into it. In this case an inertial delay is used, because one car can get into the crossing only if there is a free space during a specified time. Otherwise, the car must remain waiting before entering the crossing.

**Streets with two lanes**

A segment t = (p1, p2, 2, a, dir, max) is defined as a binary two-dimensional Cell-DEVS model with transport delays. The basic structure of these models is defined in the Figure 6.


**Figure 6. Two lane segment**

Each row of this space has a different behavior, defined by the neighborhood. Therefore, each row must be specified separately. That is, the vehicles in the cells of the first row only can move straight and to the right, and the second row can move straight and to the left.

Therefore, the cells in the first row will be defined as the one-lane cells. The main changes include the interface (I), neighborhood (N) and behavior ($\tau$), now defined as:

**I** = < $\eta$, $P^X$, $P^Y$>, where $\eta$ = 6;

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

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

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

| | | | |
|---|---|---|---|
| (0,-1) | (0,0) | (0,1) | Lane 0 |
| (-1,-1) | (-1,0) | (-1,1) | Lane 1 |

**Figure 7. Neighborhood for the Lane 0.**

The function $\tau$ for these cells is defined as in the one lane model. The following rules, modeling lane changes, are added to the local computation function:

| New state | Neighborhood state |
|---|---|
| 1 | (0,0) = 0 and (0,-1) = 0 and (-1,-1) = 1 and (-1,0) = 1 |
| 0 | (0,0) = 1 and (-1,1) = 0 and (-1,0) = 0 |

The diagonal movement considers that a vehicle tries first to move straight, and it has priority to access to the position in front of it. The first rule here represents a vehicle arriving in diagonal. To define the priority access, the diagonal movement checks if there is no car waiting to access the cell in diagonal. If that is not the case, it can advance.

The main difference in the definition for the lane 1 is in the neighborhood definition and the local computing function, as follows:

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

| | | | |
|---|---|---|---|
| (1,-1) | (1,0) | (1,1) | Lane 0 |
| (0,-1) | (0,0) | (0,1) | Lane 1 |

**Figure 8. Neighborhood for the Lane 1.**

The function $\tau$ for these cells is defined as in the one lane model. The following rules, modeling lane changes, are added to the local computation function:

| New state | Neighborhood state |
|---|---|
| 1 | ( (0,0) = 0 and (0,-1) = 0 and (1,-1) = 1 and (1,0) = 1 ) |
| 0 | ( (0,0) = 1 and (1,1) = 0 and (1,0) = 0 ) |

These rules also define the lane changes, considering priority movements for the cars advancing in a straight line.

The cells are combined into a coupled model corresponding to the segment, which is defined as:

TC2(k, max) = < Xlist, Ylist, I, *X*, *Y*, n, {$t_1$,...,$t_n$}, $\eta$, N, C, B, Z, select >

**Ylist** = **Xlist** = { (0,0), (1,0), (0,k-1), (1,k-1) }

**I** = <$P^x$, $P^y$>

$P^x$ = {<$X_{\eta+1}$(0,0), binary>, <$X_{\eta+1}$(1,0), binary>, <$X_{\eta+1}$(0,k-1), binary>, <$X_{\eta+1}$(1,k-1), binary>}

$P^y$ = {<$Y_{\eta+1}$(0,0), binary>, <$Y_{\eta+1}$(1,0), binary>, <$Y_{\eta+1}$(0,k-1), binary>, <$Y_{\eta+1}$(1,k-1), binary>}

These ports will be named as:

| Port | Name |
|---|---|
| $X_{\eta+1}(i,0)$, $0 \le i \le 1$ | x-c-vehicle |
| $X_{\eta+1}(i,k-1)$, $0 \le i \le 1$ | x-c-space |
| $Y_{\eta+1}(i,0)$, $0 \le i \le 1$ | y-c-space |
| $Y_{\eta+1}(i,k-1)$, $0 \le i \le 1$ | y-c-vehicle |

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

**n** = 2

$\mathbf{t_1}$ = 2

$\mathbf{t_2}$ = k

$\eta$ = 6

The neighborhood (**N**) for the cells in the space has been described when the cell model of each lane was presented.

**B** = { (0, k-1), (1, k-1), (0,0), (1,0) }

**Z** is built using the definition given by the definition given in the Cell-DEVS formalism, using the neighborhood definition.

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

The cells of the first and last columns, which are used to interchange information with each of the crossings, compose the interface of the coupled model. The external ports and the rules for the crossings are the same defined for the one lane model.

In [3] a complete set of rules was defined for streets with one to five lanes. In the previous section we have shown one lane segments, and here segments with two lanes will be presented. Further details for the complete specification can be found in [3].

## 3. CROSSINGS

The set of crossings defines cells in the two-dimensional space representing the places where the streets are connected together. Each crossing can connect any number of segments, which will be inputs or outputs for the crossing according with the direction defined for each track.

The crossings are represented as a ring of cells where the vehicles travel up to deciding to get through another path, using the proposal defined in [4], [5] and [6]. A car into the intersection has higher priority to obtain a position into the ring than the cars out of the crossing. In the crossing, the cars advance continuously in order to avoid deadlocks.

Each crossing is defined as a one-dimensional Cell-DEVS model with transport delays. The input and output cells should be specified, information that can be obtained through the segments specified.

The set of crossings is obtained using the previous definition for segments, as follows:

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

The set of crossings represents the places where the streets are connected together. Each crossing can connect any number of segments, which will be inputs or outputs for the crossing according with the direction defined for each lane. Each crossing should have at least one input and one output cell. This constraint can is defined by:

$\forall\, c \in$ Crossings:
$\exists\, t, t' \in$ Segments $\wedge\ t = (p1, p2, n, a, dir, max) \wedge t' = (p1', p2', n', a', dir', max') \wedge t \neq t' \wedge$
$[\ (p1 = p1' = c \wedge dir \neq dir') \vee (p1 = p2' = c \wedge dir = dir') \vee (p2 = p2' = c \wedge dir \neq dir')\ ]$

In this way, the vehicles getting into a crossing from a given segment will always have one available output.

The crossings are represented as a ring of cells where the vehicles travel up to deciding to get through another path, using the proposal defined in [4], [5] and [6]. A car into the intersection has higher priority to obtain a position into the ring than the cars out of the crossing. In the crossing, the cars advance continuously in order to avoid deadlocks.

Each crossing $c \in$ Crossings, will be defined as a one-dimensional Cell-DEVS model with transport delays, as shown in Figure 9. The input and output cells should be specified, information that can be obtained through the segments specified.

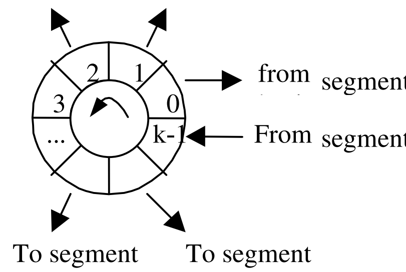


**Figure 9. Crossing**

The cells of this space are defined as the cells in the one lane model, except by the rules defining the cell's behavior. In this case, each cell will be connected with one segment influencing its behavior.

The coupled model corresponding to the crossing c is defined as:

Crossing(k, In, Out) = < Xlist, Ylist, I, $X$, $Y$, n, $\{t_1,...,t_n\}$, $\eta$, N, C, B, Z, select >

In this case, Crossing(k, In, Out) defines a crossing of k cells, where the positions in the set In are inputs for the crossing, and Out their outputs. The definition of the input and output cells depends on the specification of the segments connected to it, and by checking the sense of the traffic in them. A unique ordering for the segments acting as input is defined, in such a way that closer segment occupy neighboring cells. This ordering considers the angle between the segment and the line defined by $y = 0$. This ordering is also used to couple the models.

The number of lanes per segment, and the number of segments arriving to the crossing define the number of cells in the crossing (k).

This model represents the movement of vehicles into segment intersections. Each cell into the crossing is connected with any of the segments, therefore, the model interface include all the cells. The input/output ports are used to inform the state of each cell to the segments and viceversa.

The definition of each set in the model is the following:

**Ylist** = **Xlist** = { (0,i) / $0 \le i < k$}

**I** = <$P^x$, $P^y$>

$P^x$ = {<$X_{\eta+1}$(0,i), binary> / $0 \le i < k$ }
$P^y$ = {<$Y_{\eta+1}$(0,i), binary> / $0 \le i < k$ }

These ports will be called:

| Port | Name |
|---|---|
| $X_{\eta+1}(0,i)$, $i \in$ In | x-t-vehicle |
| $X_{\eta+1}(0,i)$, $i \in$ Out | x-t-space |
| $Y_{\eta+1}(0,i)$, $i \in$ In | y-t- space |
| $Y_{\eta+1}(0,i)$, $i \in$ Out | y-t- vehicle |

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

**n** = 1

$\mathbf{t_1}$ = k

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

**B** = {∅}

**Z** is built using the definition given in the Cell-DEVS formalism, considering the present neighborhood.

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

$\tau$: S x N → S,

In this case, the behavior will be different according if the cell is an input or output of the crossing.

In the case for *output cells*, that is, those cells in { (0,i) / $i \in$ Out } $\tau$ is defined by:

| New state | Neighborhood state |
|---|---|
| 1 | (0,0) = 0 and (0,-1) = 1 and (portvalue(x-t-space) = 1 or (portvalue(x-t-space) = 0 and random < $p_{out}$) )<br>Send(0, y-t-vehicle)<br>/* A car kept into the crossing arrives */ |
| 0 | (0,0) = 0 and (0,-1) = 1 and portvalue(x-t-space) = 0 and random ≥ $p_{out}$<br>Send(1, y-t-vehicle)<br>/* A car abandoning the crossing arrives */ |
| 0 | (0,0) = 1 and (0,1) = 0<br>Send(0, y-t-vehicle) |
| (0,0) | TRUE<br>Send(0, y-t-vehicle)<br>/*Otherwise, the cell keeps the present state */ |

Here, *random* is a probability function with Poisson distribution. If the value returned by this function is higher than the constant $p_{out}$, the vehicle abandons the origin cell towards to the corresponding segment.

The first rule of this definition represents the advance from the cell to the present one. It also considers the decision to stay into the crossing, due that the cell in the segment is occupied (portvalue(x-t-space) = 1) or because the value obtained by the random function. The output port y-t-vehicle, corresponding to the output segment must be set to 0 because the vehicle should not be detected by the rules in the segment.

The second rule represents the advance of a vehicle from the previous cell, when the vehicle must abandon the crossing. In this case, the cell in the segment must be empty, and the random function should return a value higher than the predefined constant.

The third rule represents that a vehicle in the origin cell advances to the next cell into the crossing, and the fourth represents the state conservation in other cases.

The *input cells* of the crossing are those defined by { (0,i) / $i \in$ In }, and in this case $\tau$ is defined by:

| New state | State of the neighboring cells |
|---|---|
| 1 | (0,0) = 0 and ( (0,-1) = 1 or portvalue(x-t-vehicle) = 1 )<br>send(1, y-t-space) |
| 0 | (0,0) = 1 and (0,1) = 0 and (0,-1) = 0<br>send(0, y-t-space)<br>/* No cars with priority into the crossing */ |
| 0 | (0,0) = 1 and (0,1) = 0 and (0,-1) = 1<br>send(1, y-t-space)<br>/* Cars with priority into the crossing */ |
| (0,0) | TRUE /* Otherwise, keep the previous state */ |

The input cells can receive vehicles from the neighbors or from a coupled segment (port x-t-vehicle). The first rule models that, if a cell has a vehicle, and the origin is empty, the vehicle advances. The port of the segment must be updated because the cell is busy.

The second and third rules represent that the origin cell is empty if the previous is empty too. If the previous cell is empty, the port is updated with a 0, because there is enough place to receive a vehicle from the segment. Otherwise, the port is updated with a 1, because there is a car into the crossing with priority to access to the cell, and the cars into the segment are not allowed to get into the crossing. That means that the port y-t-space is updated to 0 only when the origin and previous cell are empty, allowing the admission of a new car only if there is no other into thee crossing wanting to access to the same position.

## 4. CONCLUSION

This article was devoted to show part of a specification language used to define sections of cities as cell spaces.

Once defined the basic behavior for a city section, different components can be defined. Other components of this specification language include definitions for traffic lights, railways, men at work, street holes, transit signals, parked cars, and so on. Finally, special behavior can been defined for special vehicles: trucks, vans and high priority cars (ambulances, policemen, firemen).

This approach provides an application oriented specification language, which will allow the definition of complex traffic behavior using simple rules. The models are formally specified, avoiding a high number of errors in the developed application, and the problem solving time is highly reduced, allowing the analysis complex behavior in the traffic, and providing new solutions.

At present, the specification language is being implemented. Also, a graphical user interface is being defined, so as 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.

## 5. ACKNOWLEDGEMENTS

This work was partially supported by the UBA-SECYT research project TX-004, "Concurrency in Distributed Systems".

## 6. REFERENCES

[1] WAINER, G.; GIAMBIASI, N. “Specification, modeling and simulation of timed Cell-DEVS models”. Technical Report 97-007, Departamento de Computación, FCEN/UBA. Submitted to publication. 1998.

[2] ZEIGLER, B. "Theory of modeling and simulation". Wiley, 1976.

[3] DAVIDSON, A.; WAINER, G. “Definition of a specification language for urban traffic simulation using the Cell-DEVS formalism” (in Spanish). Technical Report 99-003, Departamento de Computación, FCEN/UBA. 1999.

[4] CHOPARD, B.; QUELOZ, P.; LUTHI, P. “Traffic Models of a 2D road network”. Proceedings of the 3rd CM users’ Meeting. October 1995. Parma. Italy.

[5] CHOPARD, B.; QUELOZ, P.; LUTHI, P. “Cellular Automata Model of Car Traffic in two-dimensional street networks”. J. Phys. A, vol. 29, PP. 2325-2336, 1996.

[6] CHOPARD, B.; DUPUIS, A..; LUTHI, P. “A Cellular Automata Model for Urban Traffic and its applications to the city of Genoa”. Proceedings of Traffic and Granular Flow. 1997.