CELL-BASED REPRESENTATION AND ANALYSIS OF SPATIAL RESOURCES IN CONSTRUCTION SIMULATION

Cheng Zhang [a], Amin Hammad [b, *], Tarek M. Zayed [a], and Gabriel Wainer [c]

[a] *Department of Building, Civil & Environmental Engineering, Concordia University,*
1515 Sainte-Catherine St W., EV-6.139, Montréal, Québec, Canada, H3G 2W1
zha_che@encs.concordia.ca, zayed@bcee.concordia.ca

[b, *] *Corresponding author, Concordia Institute for Information Systems Engineering, Concordia University,*
1425 René Lévesque Blvd., CB-410-14, Montréal, Québec, Canada, H3G 1T7
hammad@ciise. concordia.ca

[c] *Systems & Computer Engineering Department, Carleton University,*
1125 Colonel By Drive, Minto C.A.S.E.- 3048, Ottawa, Ontario, Canada, K1S 5B6
gwainer@sce.carleton.ca

Abstract

Space is one of the resources that may cause crucial problems during construction. Discrete event simulation has been widely used in construction to allocate resources and improve productivity or mitigate conflicts. However, simulation research that provides an explicit method to investigate possible space conflicts is still limited. This paper suggests a cell-based modeling approach to represent space resources in construction simulation, which enables conflict analysis and visualization of the worksite and the occupation of spaces. A detailed procedure for the cell-based modeling method is discussed. The new modeling approach is compared with MicroCYCLONE as a representative of conventional construction simulation tools to identify the advantages and limitations of each method in spatial resource representation. Furthermore, a case study is used to demonstrate the feasibility of the new approach.

Keywords: Cell-based modeling, Construction simulation, Spatial resources, Conflicts, DEVS.

1. Introduction

Workspace conflicts are serious problems that can delay construction activities, reduce productivity, or cause accidents that threaten the safety of workers [1]. Mallasi and Dawood [2] discussed that workspace interference could result in decreasing work productivity by about 40%. Several studies have been done to detect space conflicts during construction [1,2,3,4,5]. Workspace conflicts have three characteristics that differentiate them from other conflicts: (1) They have temporal aspects, i.e., they occur only during certain periods of time; (2) They exist in different forms that could change with the requirements of construction activities; and (3) They create different types of problems on site, e.g., compromising the safety of workers or reducing the productivity of equipment [4]. Simulation has been used in construction for process planning and resource allocation. However, due to the specific characteristics of workspaces, it is not easy to detect spatial conflicts without an explicit representation of space in the construction simulation. Early in the research of Halpin [6], it was mentioned that location or space-type flow units usually constrain the access to certain work processed and thus constrain the movement of other type units. However, in MicroCYCLONE [6], space is not represented based on any spatial modeling. Instead, space is represented implicitly as any other resources using abstract symbols and it is limited to the operation spaces of equipment, excluding other workspaces, such as moving paths. Kamat [7] proposed detecting conflicts between any pair of mobile or static objects on a construction site based on collision detection methods implemented within visualization tools of discrete event simulators. However, this approach is based on visualizing the results of the simulation rather than considering spatial issues in the simulation itself.

Some research has been done to investigate space visualization in construction simulation. Zhang et al. [8] used 2D icons to represent the resources, which can move along the path between activities. However, this research did not clearly represent the spatial relationships between different activities although there are icons moving from one activity to another. Zhong et al. [9] developed the GIS-based Visual Simulation System (GVSS) to offer planning, visualizing, and querying capabilities of complex construction processes. However, workspace conflicts are not discussed in their research.

As can be seen from the above review, space conflicts are not clearly represented in the previous simulation research. This could result in spatial constraints being ignored in the simulation, and the output may not reflect the real situation of the construction site. Spatial problems need to be studied in a general way that is easy to understand, and a model should be built in a way that the space can be represented explicitly. Furthermore, the space representation can improve the accuracy of calculating the duration of some activities, such as the transportation between two locations, with the real representation of the geometrical relationships between different points on the work site instead of using approximate average values. The objectives of the current research are: (1) To investigate the limitations of conventional construction simulation tools in analyzing spatial conflicts; (2) To propose a new approach for developing cell-based construction simulation models to represent the worksite layout and to analyze spatial conflicts; and (3) To demonstrate the feasibility of the new approach using a case study.

2. Cell-based modeling

In 1948, John Von Neumann and Stephan Ulam defined a modeling formalism, called Cellular Automata (CA), suited to define spatial systems, and allowing the description of cell-based models by using simple rules [10]. In CA, space is represented by a uniform grid, with each cell containing a few bits of data. At each step, each cell computes its new state from that of its close neighbors. CA are well suited to describe spatial distribution of resources. The cells in the lattice are updated according to a local rule in a simultaneous and synchronous way, using a local computing function. This function considers the state of the present cell and a finite set of nearby cells (called the neighborhood). Unfortunately, these CA have the constraint of being synchronous. This fact reduces the timing precision for the models, and, consequently, the computations related to time resources cannot be fully used. Moreover it should be considered that in most cell spaces there are a large number of quiescent cells.

In the '70s, Bernard Zeigler [11] defined a theory for Discrete-EVent systems Specification (DEVS). It is a formal method to build models using a hierarchical and modular approach. This approach allows the developer to build a Model Base permitting easy reuse of models that have been validated. A real system modeled with this paradigm can be described as several sub-models coupled into a hierarchy. Each model can be behavioral (atomic) or structural (coupled), consisting of a time base, inputs, states, outputs and functions to compute the next states and outputs. The basic idea is that each model uses input/output ports in the interface to communicate with other models.

Also, Zeigler [11] defined a cell space model, which consists of an infinite set of geometrically defined cells, each cell containing the same computational apparatus as all other cells and connected to other cells in a uniform way. As shown in Fig. 1 (a), for a cell located at point (0,0), the nearest neighbors would be those located at: (0,1) (1,0) (0,-1) (-1,0), which are at a distance of one cell away orthogonally and (1,1) (-1,1) (-1,-1) (1,-1) which are at a distance of one cell away diagonally [11]. The cell model could be two dimensional or three dimensional. Figs. 1 (b) and (c) show the generalization of the cell neighborhood representation in three dimensions. For 2D models, one layer is enough; while for 3D models, we may have as many layers as necessary to cover the physical space of the problem. In simple models, one value per cell could be sufficient. However, cell modeling in engineering applications often needs several attributes for each cell. For simplicity, these attributes could be attached to several interrelated layers so that the corresponding cells in different layers will contain one attribute each. In this case, we may have as many layers as the number of the attributes. This approach will be used in the rest of the paper.

Based on the concept of cell spaces, Wainer [12] developed an approach called Cell-DEVS [13]. Cell-DEVS describes cell spaces as discrete event models using the DEVS formalism, including delay functions to have a simple definition of the timing of the cell. The proposal of the Cell-DEVS paradigm considers each cell of a CA as a hierarchical and modular discrete event model [14]. In this way, complex models can be defined using a continuous time base. It also allows associating several kinds of delays for each cell, allowing the definition of complex models easily. The cell state changes according to a local function that uses the present cell state and a finite set of nearby cells. Many applications of Cell-DEVS have been developed including surface tension analysis, studies of ecological systems, and a specification language used to define traffic simulations [15].

Based on the cell representation of the spatial model, this paper investigates the possibility of a cell-based approach for the representation and analysis of spatial resources in construction simulation. In the next section, the limitations of space representation in conventional simulation models are discussed, and the usefulness of the cell-based representation is investigated using a case study of a bridge re-decking project. In addition, the benefits of the proposed approach, such as improving the accuracy of the simulation and detecting conflicts are discussed.

3. Limitations of available simulation models

There are several software packages available to develop construction simulation models. MicroCYCLONE [6] and Stroboscope [16] are two popular programs used in construction simulation to assist the manager in making decisions about resource allocation and conflict prediction. These two programs use similar representation of activities and resources. In this paper, MicroCYCLONE is used to demonstrate how space resources are represented in a case study described in Section 3.1 and to identify the limitations of available construction simulation tools.

3.1 Case Study

The case study of Jacques Cartier Bridge in Montreal is used throughout the rest of the paper to discuss the limitations of available simulation models and the feasibility of the proposed approach. The deck of this bridge has been replaced in 2001-2002. The new deck is constructed of precast, prestressed and post-tensioned panels made of high performance concrete which were prefabricated in a temporary plant installed near the south end of the bridge. The case study will focus on the two activities of removing existing deck sections and installing new panels in the main span of the bridge. These two activities were critical for the success of the project from the point of view of spatial and temporal constraints. The existing deck was removed by saw-cutting the deck into 410 sections similar in dimensions to the new panels being installed. Each existing deck section was removed by two telescopic cranes and a new panel was lifted from a truck and lowered onto the new bearing assemblies. Old sections were transported to a dumping area near the bridge. New panels were transported from the plant located at the south end of the bridge. Table 1 shows the tasks involved in these two activities and their durations. Fig. 2 shows a schematic representation of the worksite layout during the deck replacement. Most of the time, two teams worked in parallel in different parts of the bridge. Each team used two telescopic cranes located at both sides of the section to be replaced. Two types of semi-trailer trucks were used to transport the old sections and new panels.

3.2 Simulation Model of the Re-decking Project Using MicroCYCLONE

As shown in Fig. 3, active-state rectangular nodes represent tasks and include the triangular time distribution for each task. Idle-state circles represent delays or waiting positions for resource entities; and directional flow arrows represent the path of resource entities as they move between idle and active states. Tasks can be executed only when all the queues (resources) needed are available. For example, task "Dumping" (Task 13) is the task of dumping an old section of the deck in the dump area. It takes an average time of 5 minutes and needs two resources: "Truck waiting for dump" (Task 11) and "Forklift waiting" (Task 12). After the dumping of one section is finished, these two resources are released: the truck will return to the bridge (Task 14) and the forklift will go back to idle state (Task 12). The procedure for modeling these processes involves four basic steps: (1) Flow unit identification; (2) Development of flow unit cycles; (3) Integration of flow unit cycles; and (4) Flow unit initialization. Because of the space limitation we will not discuss the details of MicroCYCLONE modeling techniques, which are available in [6].

To investigate the space representation in this model, a conceptual mapping of the space is applied. In Fig. 3, spaces are represented as abstract symbols-queues. The site layout can be approximately divided into several areas according to the geographic locations, including *Bridge*, *Plant*, and *Dump Area*. There are two spaces implicitly represented as queues in this model (see Fig. 2): (1) Empty deck space of a removed section (ED); and (2) Truck working space (TWS). Other spaces, such as the moving paths of trucks, are considered to be available all the time and are not represented in this model. In addition, workspaces can change from one task to another and it is the responsibility of the modeler to identify these spaces in MicroCYCLONE.

Moreover, in MicroCYCLONE and other conventional simulation tools, spatial conflicts are not discovered except by visualizing the results using post processing applications [17]. In fact, our observation is that most of the time space resources are attached to other resources (equipment, materials, etc.) and heavily depend on the site layout. Therefore, our proposed approach, which will be discussed in detail in Section 4, is based on linking spatial resources with other resources.

4. Proposed Approach: Cell-based Modeling

As discussed in Section 2, space can be divided into cells and every cell is a discrete event model. A cell can change its state according to its own time delay and external events. A dynamic information exchange can be achieved during the simulation period. Conflict detection can be simplified by checking the state of each cell and avoiding an occupied cell being used by other objects. Based on this idea, a cell-based modeling approach is used to investigate the spatial issues in construction simulation. It should be noted that in spite of the availability of cell-based simulation tools developed for research purposes, to the best of our knowledge, this is the first trial to investigate the applicability of these tools in construction simulation. In the rest of this section, we will introduce the process used in building cell-based models using several examples related to the case study of Jacques Cartier Bridge re-decking project. Cell-DEVS modeling techniques for representing spatial constraints in construction simulation will be discussed in detail. A full description of the implementation of the case study will be given in Section 5. Fig. 4 shows the general procedure of cell-based modeling. This procedure has ten steps:

(1) **Identify Cell-DEVS and DEVS models.** Cell-DEVS models are used where the spatial representation is important; while the DEVS models are used to simulate the parts that do not need spatial representation or could not be represented using cells. For example, in our case study, a Cell-DEVS model could be extended to cover the whole working area, including the bridge, the dump area and the plant. However, representing all the roads between the bridge and the dump area or the plant as cells is not necessary because there are less spatial constraints on these parts. Therefore, only the main three working areas need to be represented as Cell-DEVS models: *Bridge, Plant* and *Dump Area.* DEVS models are used to build the models of queues and transports between the above Cell-DEVS models.

(2) **Define the relationships and information exchange between models and connect them using input and output ports.** For example, whenever there is an old section to be replaced, a request for a truck to go to the *Bridge* model will be sent to the queue of waiting trucks. More details about the implementation of this step are given in Section 5.

(3) **Decide the suitable size of cells and the dimensions of each Cell-DEVS model.** For example, Fig. 5 shows the cell representation of the bridge model. The cell size is assumed to be 3*3 meters. The main span of the bridge can be approximately represented by 200*6 cells.

(4) **Define the layers of each model.** As explained in Section 2, several layers are needed to represent the attributes related to a Cell-DEVS model. Based on our experience in cell-based modeling [18], three layers should be created for the *Bridge* model to represent the occupancy, mobility conditions and IDs of the objects occupying the cells (Fig. 5 (a)). The first and main layer is the *occupancy layer*, which has the occupancy states of the cells (e.g., the type of equipment occupying a cell). The second layer is the *control layer*, which

decides the mobility state and moving direction, detects conflicts between objects, and sets the priority of moving depending on the types of objects as defined in the *occupancy layer*. The cells in the *control layer* will detect the corresponding cells in the *occupancy layer* and set the priority based on certain rules (as will be explained in step 6). The *ID layer* contains the ID numbers of each piece of equipment. Combining the information about a certain location (cell) collected from the corresponding cells in the three layers gives a triplet of attributes for that location: <occupancy, mobility, ID>. For example, a cell can be occupied by a truck moving west and having an ID of '2'.

(5) Identify the resources needed and define the codes that will be used in Cell-DEVS models. Different encodings can be used to represent equipment states, the occupation of cells, and the IDs of equipment, such as those given in Table 2. For example, Fig. 5 (b) shows an old section (2) and two cranes (3) beside it, which occupies 3 cells each. For simplicity, cells with zero value are shown as blank cells. The same code can be used in different layers with different meanings. In the *control layer* of the *Bridge* model, different mobility conditions are represented as (1): move north; (2): move south; (3): move east; (4): move west; and (5): static object. Any number can be used in the *ID layer* to represent the IDs of equipment.

(6) Analyze the activities and develop rules for each Cell-DEVS model. Cells can communicate with each other through rules that detect the state changes of a cell's neighborhood and change the cell's own state accordingly. Rules governing the interactions between layers will guarantee the coupling of the attribute triplets introduced in step 4 and the consistency across layers. There are several types of rules that could be applied for simulating construction activities, such as rules for moving trucks, conflict detection, truck generation, direction changes, etc. Each rule has a condition part, an action part, and a time delay. The conditions and actions may involve the attributes defined in one or more layers. Examples of rules that could be used in the *Bridge* model are shown in Fig. 6. The first rule represents an empty truck for loading an old section. From the *occupancy layer*, the truck should move into the next cell if it is not facing an old section. From the *control layer*, both the condition part and the action part show that the direction of the truck movement is towards west (4). This means that if a truck is moving in certain direction at a certain time step, it will continue moving in the same direction in the next time step. The time delay for this action is the time needed to move one cell that can be calculated from the average speed of the truck. The second rule represents a truck arriving at the location of an old section. In the *occupancy layer*, the truck should stop to load the old section. After the time delay needed for this task, it will change its occupancy state to (4), which means the truck is now carrying the old section. In the *control layer*, the mobility state is changed to static object (5) when the truck is loading the old section. After the loading is done, the mobility state changes to (4) and the truck will move west again (assuming that this is the direction towards the dump area). The third rule is about conflict detection. The cells representing static objects, such as old sections and cranes are set to the highest value (5 = static object), which means that a moving object (e.g., truck) with a lower value should change its direction to avoid collision with these static objects. When the truck is moving west (the cell value is (4)), the rule in the *control layer* checks whether the cell to the west is occupied. In this case, conflict analysis will be applied to decide the new direction (i.e., move north or south) until there is no obstacle on the way of the truck. Then the direction will be changed back to (4) and the truck will continue moving west.

(7) Define the zones of each Cell-DEVS model. In general, rules in a Cell-DEVS model are applied on all the cells of that model. However, in many cases, it is necessary to specify a part of the model as a zone in order to define local rules that apply only within that zone. In the *Bridge* model, both the *occupancy* and *control layers* should be divided into two zones to define different moving directions when a conflict is detected (Fig. 7). The main difference of the rules of these two zones is that a truck has to take a different direction when it meets an obstacle. For example, when the *control layer* detects a conflict between a truck and a crane, the direction of the truck will be changed to south or north to avoid the obstacle depending on whether the crane is in *zone 1* or *zone 2*, respectively. This example shows the limited intelligence that can be represented by rules for controlling direction changes.

(8) Develop the DEVS models. DEVS models can be developed based the functions that they perform, such as queues or transport functions. Examples of these DEVS models are given in Section 5.

(9) Initialize the resources. The number of the resources should be initialized before running the simulation. For example, the number of trucks can be initialized in the *Dump Area* and the *Plant* models. Multiple teams can be initialized in the *Bridge* model by defining the location of each team (Fig. 7).

(10) Run the simulation and visualize the simulation results. The simulation tool will generate the discrete changes of states in each model. These results can be visualized as an animation that gives a quick method for checking the results.

As an example of the overall mechanism of the simulation process, Fig. 8 shows the state changes in the three layers at different time steps that will be generated based on rules similar to those explained above. At time t_1 an empty truck (1) is moving west with an ID number “2”. At time t_2, it meets an obstacle, which is a crane (2). The *control layer* detects that the mobility state of the crane is static (5), which should be given the highest priority. Therefore, the direction of the truck is changed to (1) to move north because it is in *zone 2*. After checking the neighboring cells again at t_3, the moving direction is changed back to (4) and the truck is moving west again (t_4). At time t_5, the truck arrives to the location of an old section that will be removed. It stops and the *control layer* changes the corresponding cell state to (5). After some time delay, the truck occupancy state will change to (4), which means it carries an old section and is ready to move. At the same time, the cells that represent the old section change their occupancy state to (5), which means the space is empty and ready for the installation of a new panel.

5. Implementation of the case study using the cell-based modeling

Fig. 9 shows the structure of the cell-based simulation model of the re-decking project of Jacques Cartier Bridge using CD++, a tool for cell-based discrete-event modeling and simulation based on the DEVS formalism [19]. This model is a combination of Cell-DEVS and DEVS models. Arrows show the input and output information flow between different models through ports. *Bridge, Plant*, and *Dump Area* are Cell-DEVS models. The following DEVS models are built to facilitate communications between these Cell-DEVS models: *Control Unit* model, *Reposition* model, and *Transport (T)* and *Queue (Q)* models. For example, the *T-Plant-Bridge* is a *Transport* model representing the transportation of a panel from the plant to the bridge, and the *Q-Old* is a *Queue* model representing the queue of trucks that will carry the old sections. The *Control Unit* model is built to provide overall control on the system, such as permitting a queue to send a truck to the bridge when a truck is needed.

The computing functions of the cells are defined in CD++ using a set of rules with the form: VALUE DELAY {CONDITION}. This format indicates that when the CONDITION is satisfied, the state of the cell changes to the designated VALUE, and it is DELAYed for a specific time. To calculate the new value for a cell's state, the simulator takes each rule and evaluates the condition clause. If the condition evaluates to true, then the action and delay clauses are evaluated. The result will be the new cell state and will be sent as an output after the delay. The time delays used in rules follow the duration of activities as given in Table 1. The above models are briefly discussed in the following paragraphs:

Cell-DEVS Models

The discussion about the *Bridge* model given through examples in Section 4 covered most of the aspects of this model. The following simplifications are made: (1) A truck is represented by one cell (3×3m) while a telescopic crane is represented by three cells (Fig. 5 (b)); and (2) The moving direction of all trucks on the bridge is always from east to west.

There are two input ports of the *Bridge*: *in-old* and *in-new*. The queues will send the ID number of each truck to the *Bridge* through the respective input port. The output port of the *Q-Old* is linked with the *in-old* port, while the *Q-New* is linked with the *in-new* port. Different layers act differently when receiving an ID number of a truck from a queue. The *occupancy layer* generates trucks; the *control layer* generates the directions; and the *ID layer* keeps the received value as it is and moves it with the truck.

There is one output port of the bridge, which is linked with *T-Bridge-Dump*, *T-Bridge-Plant*, and *Control-Unit* models to send the ID number of each truck. The ID numbers are initialized in the *Dump Area* or in the *Plant* models.

The techniques used in implementing the *Dump Area* and the *Plant* models are similar to those used in the *Bridge* model. About 100 rules were applied to implement all the functions of the Cell-DEVS models. It takes about 20 minutes to simulate a 9-hour construction period using this model on a computer with Pentium 4 CPU.

DEVS models

The DEVS models are defined using C++ language supported by the CD++ simulation software. The atomic models that were used are:

- Queues: *Q-Old*, *Q-New*
- Transport models: *T-Dump-Bridge*, *T-Plant-Bridge*, *T-Bridge-Dump*, *T-Bridge-Plant*
- *Control-Unit*
- *Reposition*

As an example of transport models, the *T-Dump-Bridge* model receives the trucks for carrying an old section from the *Dump Area* and keeps it with some time delay to represent the transportation time from the dump area to the bridge.

The *Control-Unit* model is the central part of the whole model. It communicates with other models and checks the need for trucks, then decides when to send a truck to the bridge. When the *Control-Unit* receives a truck from the *Q-Old*, it will check if a truck for carrying an old section is needed. In that case, it will send a signal to the *Q-Old* to send a truck to the bridge. The *Q-Old* will send a truck to the bridge when the queue is not empty. Similarly, when the *Control-Unit* model receives a truck from the *Q-New*, it will check if a truck for carrying a new panel is needed. In that case, it will send a signal to the *Q-New* to send a truck to the bridge.

When the *Control-Unit* model receives a signal from the *Bridge*, it will check the type of the truck. If the truck is carrying an old section, which means there is an empty space available on the bridge, it will send a signal to the *Q-New* to send a truck for carrying a new panel to the bridge provided that the queue is not empty. If the signal from the *Bridge* is an empty truck that finished unloading a new panel, this means that the work cycle of a team replacing an old section with a new panel has been finished. In this case, the *Control-Unit* will send a signal to the *Reposition* model to simulate the delay of the time needed to move the equipment to the next old section to be replaced. After the time delay, the *Reposition* model will send a signal to *Q-Old* asking for a truck. At the same time, a signal will go to the *Control-Unit* to increase the number of the old sections available on the bridge. In addition, the *Control-Unit* is used to accumulate the number of the removed old sections and the installed new panels and to calculate the productivity of the work.

6. Simulation results and discussion

We did the simulation of the Jacques Cartier Bridge re-decking case study using the following resources based on the data from the real project: (1) two teams, (2) two trucks for carrying old sections, and (3) two trucks for carrying new panels. The locations of the teams are initialized in the *Bridge* model, and the numbers of trucks are initialized in the *Dump Area* and the *Plant* model.

The construction time that has been simulated is 9 hours, which is the real construction period from 8:30 p.m. to 5:30 a.m. the next day. The result of the simulation shows that 18 panels have been installed during this period, which is the same as the result obtained from MicroCYCLONE. This similarity could be explained by the simplicity of the spatial aspects of the case study where the explicit representation of spatial constraints in the cell-based simulation did not affect the result.

The work site layout can be displayed at every time step to show the space occupation in the *Bridge* and other Cell-DEVS models. Fig. 10 shows part of the *occupancy layer* of the *Bridge* model, where two teams are working on the bridge in parallel. Figs. 10 (a) and (b) show *Truck-1* loading an old section while *Truck-2* is coming to unload a new panel at another location. Fig. 10 (c) shows *Truck-1* carrying the old section to the dump area. Fig. 10 (d) shows *Truck-3* coming with a new panel to be installed. Fig. 10 (e) shows *Truck-3* after it changed its direction to avoid *Truck-2*.

Comparison is made between the cell-based model and the MicroCYCLONE model of the case study to investigate their spatial representation. The three Cell-DEVS models explicitly represent the three major areas that are implicitly identified in the MicroCYCLONE model. The physical space and the occupation situations on the bridge, the plant, and the dump area can be clearly shown using the Cell-DEVS models. Transport models are similar to the transportation tasks in the MicroCYCLONE model, such as “Truck with new panel travels to bridge” activity (Task 23). Queue models are similar to the queues in the MicroCYCLONE model, such as “Truck for old section” queue (Task 15). *Reposition* model is similar to the task of “Team repositioning” (Task 30).

The advantages and limitations of each model are shown in Table 3. The advantages of using cell-based simulation are: (1) The space can be represented explicitly and the simulation models can be visualized so that the occupation of the workspace and other spatial information of the construction environment can be understood more easily than using conventional models. The physical resources can be initialized to simulate several site layouts, and thus to find the optimal layout based on the availability of workspaces; (2) The conflicts between spaces can be detected based on the site layout; (3) The accuracy of the duration of activities is expected to improve, especially for situations where spatial conflicts are present; and (4) The models created for a project can be reused in other similar projects because the rules are flexible and easy to modify. However, there are also some limitations of the cell-based modeling, such as the difficulty to represent complex layouts and mobility on curved roads. The simplified version of the model may not exactly match the worksite. Furthermore, there are some limitations of the cell-based simulation tools themselves, such as the complexity of defining rules. In addition, there are many functions readily available in the conventional tools, such as productivity calculation and sensitivity analysis, which have to be implemented in the cell-based modeling tools.

7. Conclusions and future work

This paper proposed analyzing spatial issues in construction sites using cell-based simulation. This is the first step in investigating the feasibility of applying cell-based simulation in construction. Simulation models were built to investigate space representations and conflict analysis during construction. The following conclusions can be stated: (1) The implicit representation of space using conventional simulation tools makes it difficult to check availability of workspaces, thus the spatial conflicts could not be detected during the simulation; (2) In contrast, cell-based simulation can be used as a general method to represent the worksite layout and to analyze and visualize the activities that may be affected by the space constraints, and thus the spatial conflicts that may happen during construction can be detected; (3) The proposed general procedure of developing cell-based construction simulation models showed the specific steps for defining layers and conflict detection rules and how these steps can be used in simulating construction activities; (4) The case study demonstrated the feasibility of cell-based construction simulation; and (5) In spite of the present limitations of cell-based simulation for practical usages, it has several advantages over conventional tools in representing spatial constraints and detecting spatial conflicts.

Tools for creating more realistic models should be developed, and further study is needed to improve the usability of cell-based modeling. In addition, the feasibility of using the cell-based approach for site layout planning should be investigated.

References

[1] S. Guo, Identification and Resolution of Workspace Conflicts in Building Construction, Journal of Construction Engineering and Management, 128(4), 2002, pp. 287-295.

[2] Z. Mallasi and N. Dawood, Workspace Competition: assignment, and quantification utilizing 4D Visualization Tools, In Proceeding of Conference on Construction Application of Virtual Reality, ADETTI/ISCTE, Lisbon, 2004, pp. 13-22.

[3] D.R. Riley, and V.E. Sanvido, Patterns of construction-space use in multistory buildings. Journal of Construction Engineering and Management, 121(4), 1995, pp. 464-473.

[4] B. Akinci, M. Fischer, R. Levitt, and R. Carlson, Formalization and Automation of Time-Space Conflict Analysis, Journal of Computing in Civil Engineering, 16(2), 2002, pp. 124-134.

[5] D. Heesom, L. Mahdjoubi, and D. Proverds, A Dynamic VR System for Visualizing Construction Space Usage, Proc., Construction Research Congress, 2003, Honolulu, Hawaii.

[6] D.W. Halpin, and L. S. Riggs, Planning and analysis of construction operations. Wiley Interscience, New York, 1992.

[7] V.R. Kamat, VITASCOPE – Extensible and Scalable 3D Visualization of Simulated Construction Operations, Doctoral dissertation, Department of Civil Engineering, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 2003.

[8] H. Zhang, J.J. Shi, and C. M. Tam, Iconic Animation for Activity-based Construction Simulation, Journal of Computing in Civil Engineering, 16(3), 2002, pp. 157-164.

[9] D. Zhong, J. Li, H. Zhu, and L. Song, Geographic Information System-Based Visual Simulation Methodology and Its Application in Concrete Dam Construction Processes, Journal of Construction Engineering and Management, 130(5), 2004, pp. 742-750.

[10] S. Wolfram, Theory and Application of Cellular Automata, Vol. 1, Advances Series on Complex Systems, World Scientific, Singapore, 1986.

[11] B. P. Zeigler, Theory of Modelling and Simulation, Wiley-Interscience Publication, 1976.

[12] G. Wainer, Discrete-events cellular models with explicit delays, Doctoral dissertation, Université d'Aix-Marseille III., 1998.

[13] G. Wainer and N. Giambiasi, N-Dimensional Cell-DEVS. In *Discrete Events Systems: Theory and Applications*, Kluwer. Vol. 12, No. 1. January 2002. pp. 135-157.

[14] G. Wainer and N. Giambiasi, Application of the Cell-DEVS Paradigm for Cell Spaces Modeling and Simulation, SIMULATION 76:1, 2001, 22-39.

[15] A. Davidson, and G.,Wainer, "ATLAS: a specification language for traffic modelling and simulation". Accepted for publication in *Simulation, Practice and Experience*. Elsevier. Accepted for publication: July 2005.

[16] J.C. Martinez, EZStrobe - General-Purpose Simulation System Based on Activity Cycle Diagrams, Proceedings of the 1998 Winter Simulation Conference, Washington, D.C., 1998, pp. 341-348.

[17] V.R. Kamat, Visualizing Simulated Construction Operations in 3D, Journal of Computing in Civil Engineering, 15(4), 2001, pp. 329-337.

[18] C. Zhang, A. Hammad, T.M. Zayed, and G. Wainer, Representation and Analysis of Spatial Resources in Construction Simulation, Winter Simulation Conference, December, 2005, Florida, U.S.A. (Accepted, reference number: hammada54350)

[19] G. Wainer, "CD++: a toolkit to define discrete-event models". *Software, Practice and Experience*. Wiley. Vol. 32, No.3. pp. 1261-1306. November 2002.

Authors' Biographies

Cheng Zhang is currently pursing her Master of Science degree at the Department of Building, Civil and Environmental Engineering at Concordia University. Her research interests are focused on spatio-temporal issues in infrastructure management systems. Her email address is <zha_che@encs.concordia.ca>.

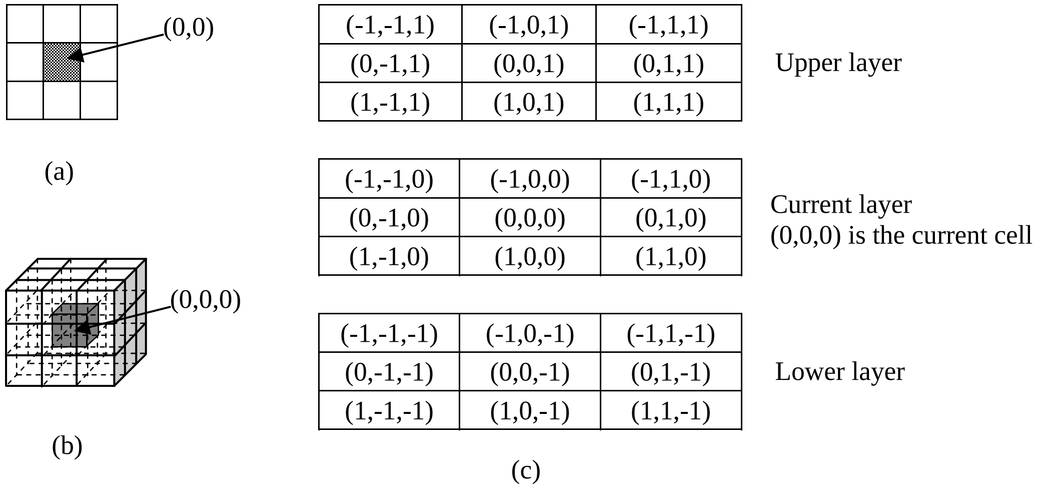
Amin Hammad is Associate Professor at Concordia Institute for Information Systems Engineering. His research interests are mainly focused on telegeoinformatics, infrastructure and urban management systems. His email address is <hammad@ciise. concordia.ca> and his Web address is <www.ciise.concordia.ca/~hammad>.

Tarek M. Zayed is Assistant Professor at the Department of Building, Civil and Environmental Engineering at Concordia University. His research interests are focused on construction engineering and management. His email address is <zayed@bcee.concordia.ca>.

Gabriel Wainer is Associate Professor in the Department of Systems and Computer Engineering at Carleton University. His research interests are focused on Discrete Event Modeling and Simulation, Parallel and distributed simulation, and Real-Time systems. His e-mail and web addresses are
<gwainer@sce.carleton.ca> and <www.sce.carleton.ca/faculty/wainer>.

List of figures:

List of tables:



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

Upper layer

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

Current layer
(0,0,0) is the current cell

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

Lower layer

Fig. 1. Cell representations
(a) 2D representation of cells (b) 3D representation of cells (c) Neighborhood cells in 3D

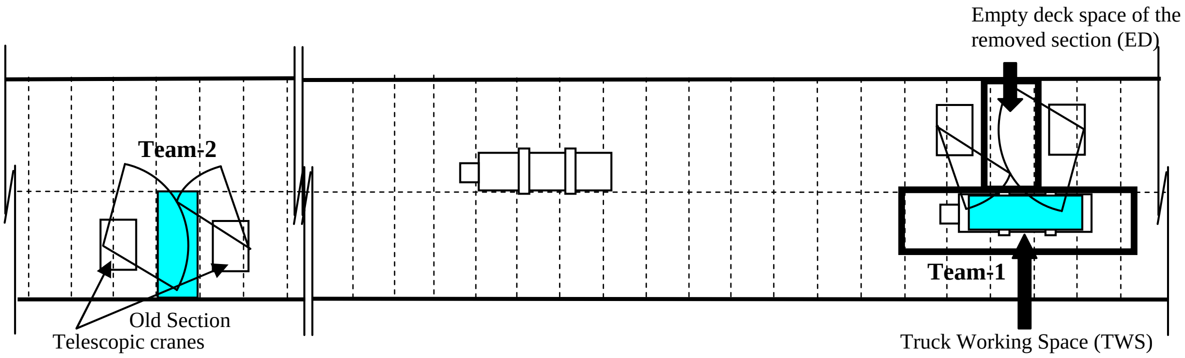

Empty deck space of the removed section (ED)
Team-2
Team-1
Old Section
Telescopic cranes
Truck Working Space (TWS)

Fig. 2. Worksite layout of the bridge re-decking

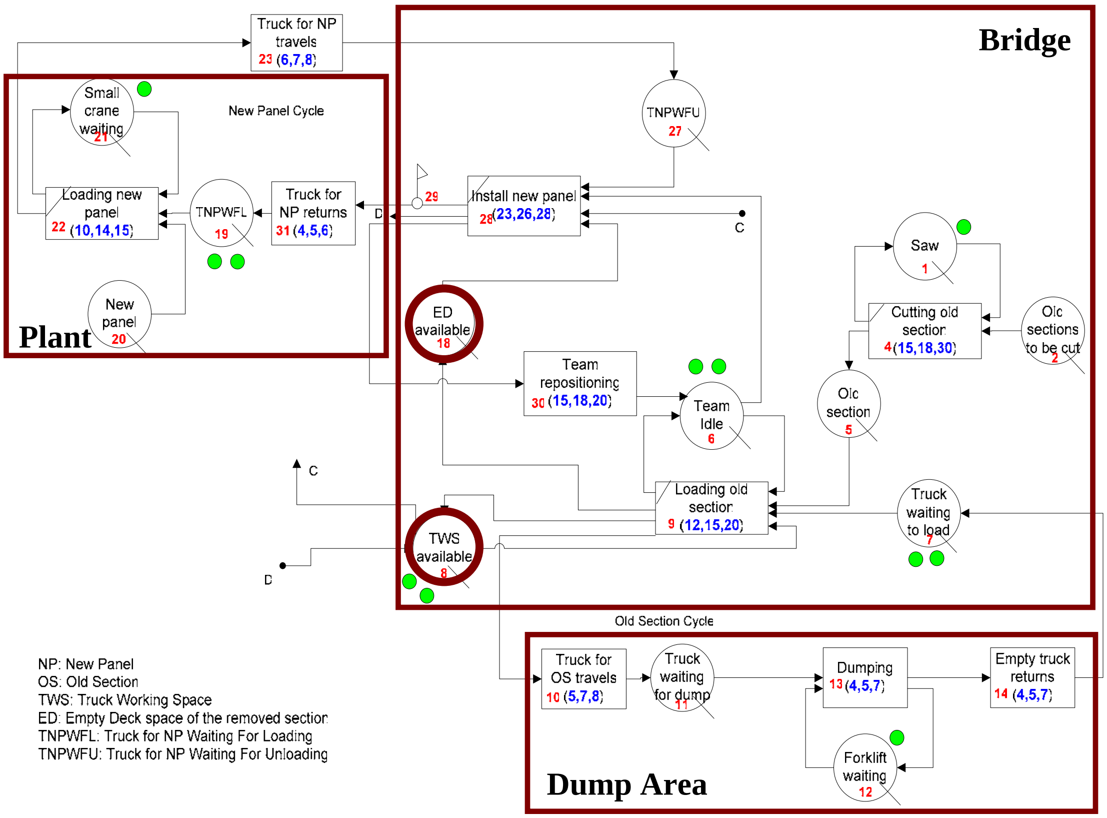


Fig. 3. MicroCYCLONE Model of Jacques Cartier Bridge Re-decking Project

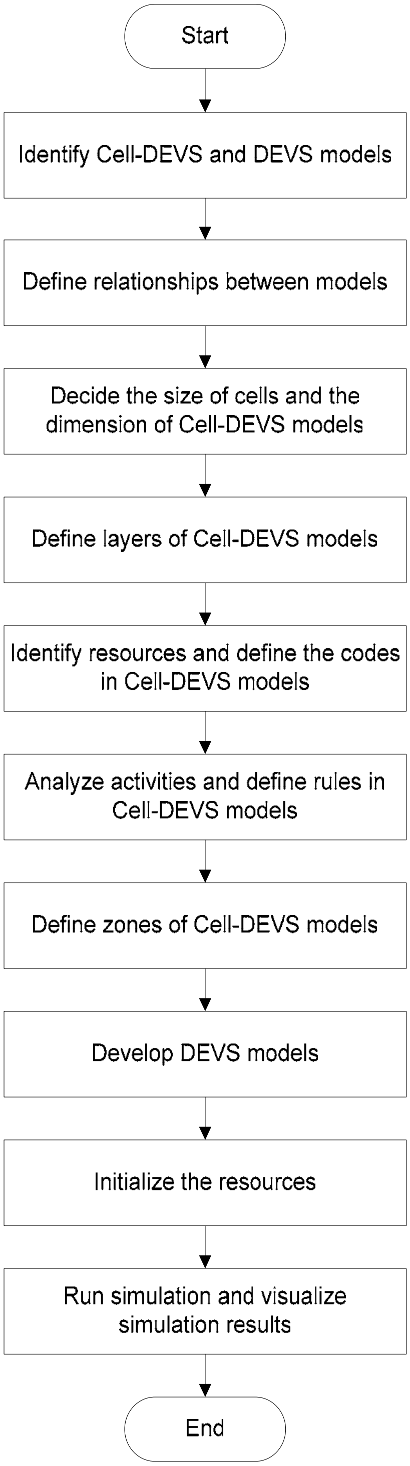
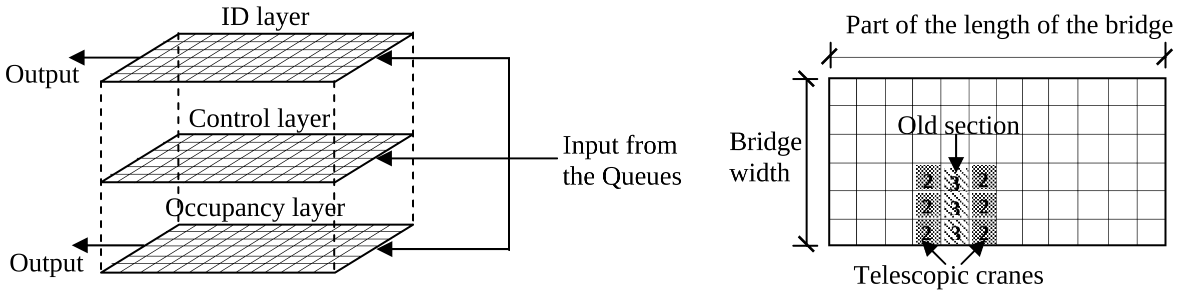

Start
Identify Cell-DEVS and DEVS models
Define relationships between models
Decide the size of cells and the dimension of Cell-DEVS models
Define layers of Cell-DEVS models
Identify resources and define the codes in Cell-DEVS models
Analyze activities and define rules in Cell-DEVS models
Define zones of Cell-DEVS models
Develop DEVS models
Initialize the resources
Run simulation and visualize simulation results
End

Fig. 4. Flowchart of general procedure of cell-based modeling



(a)

(b)

Fig. 5. Cell representation of the *Bridge* model

(a) The three layers used to model occupancy, direction, and ID information

(b) Cell representation of the *occupancy layer*

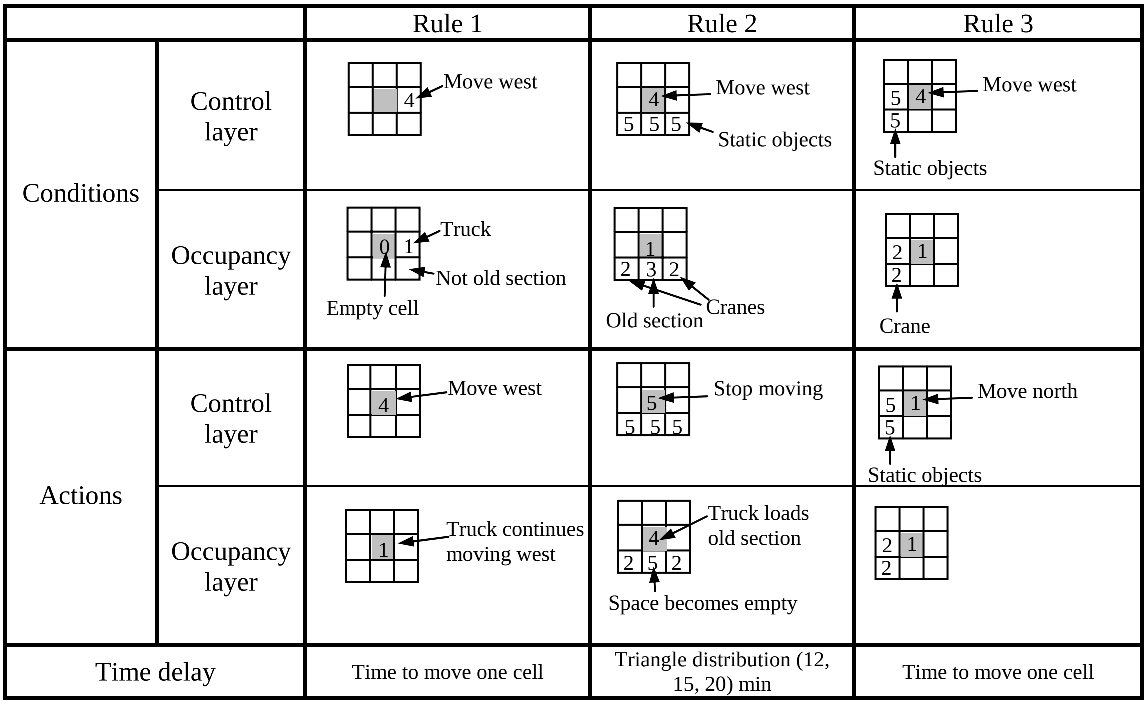


Fig. 6. Examples of rules about truck movements

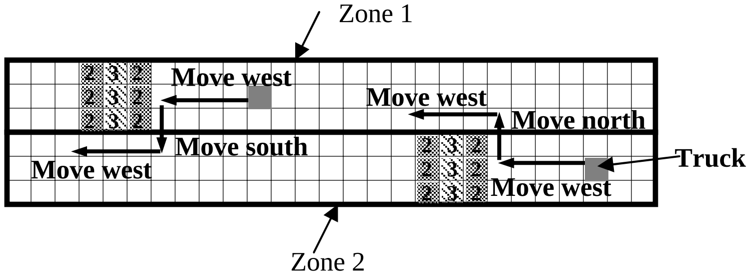


Fig. 7. Zone division and the rules for changing directions in each zone

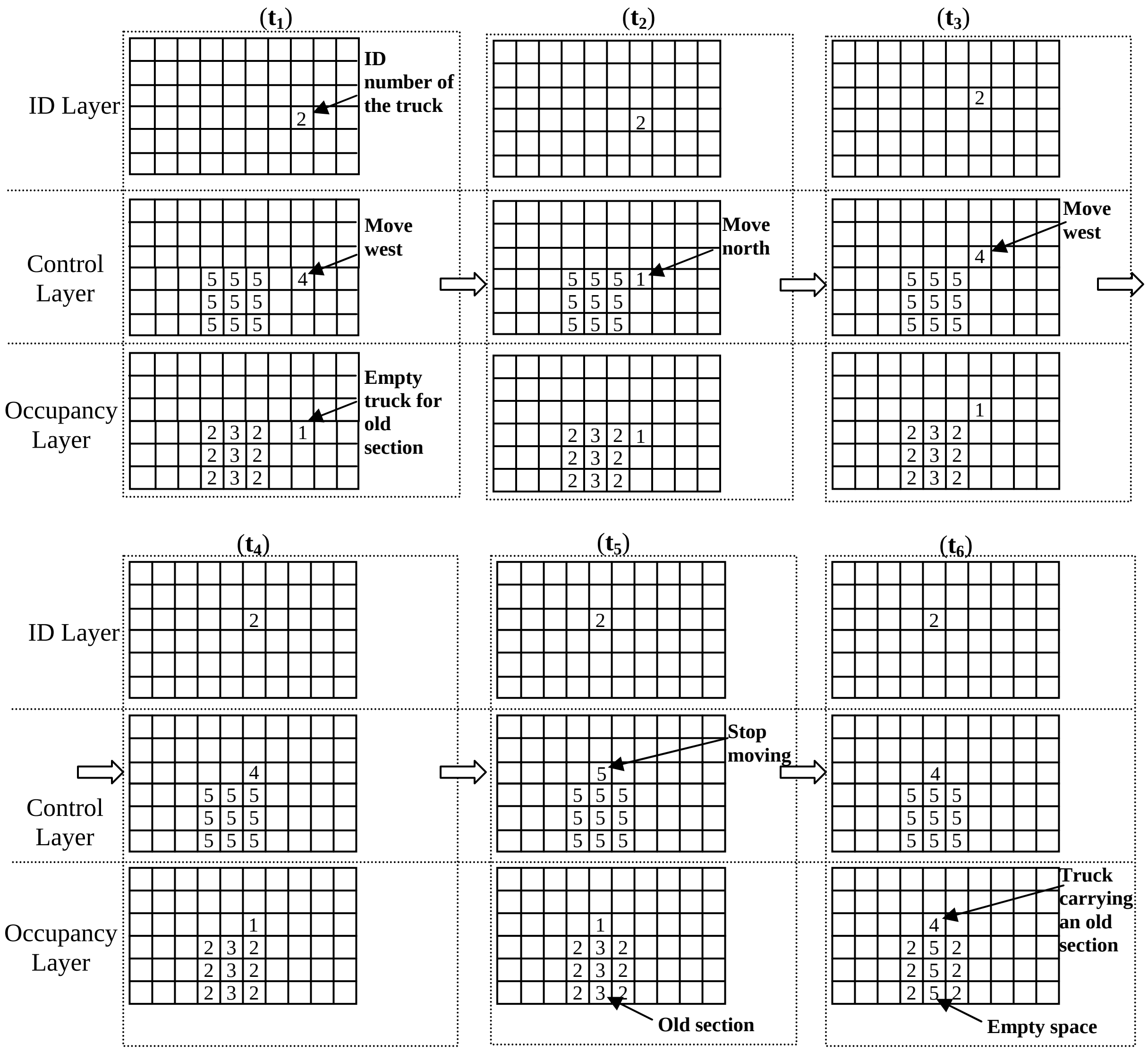
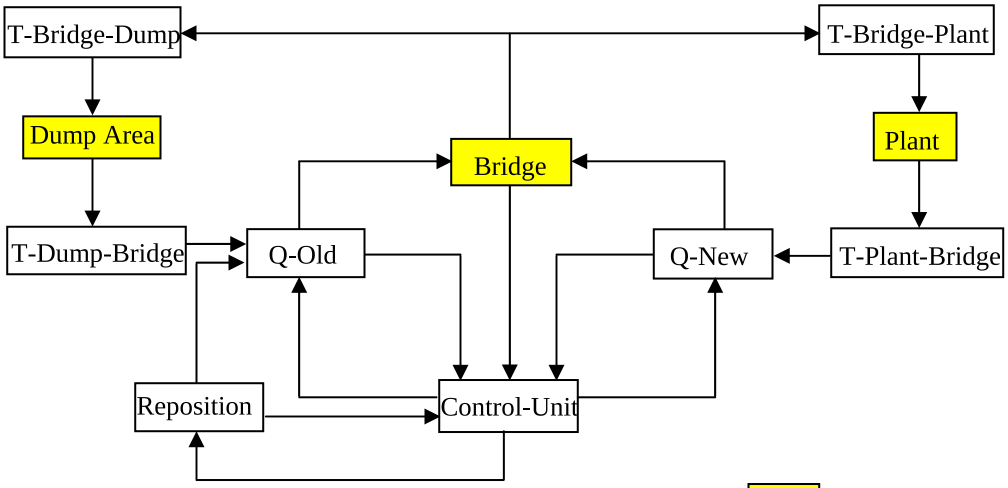


Fig. 8. States changes in the three layers at different time steps

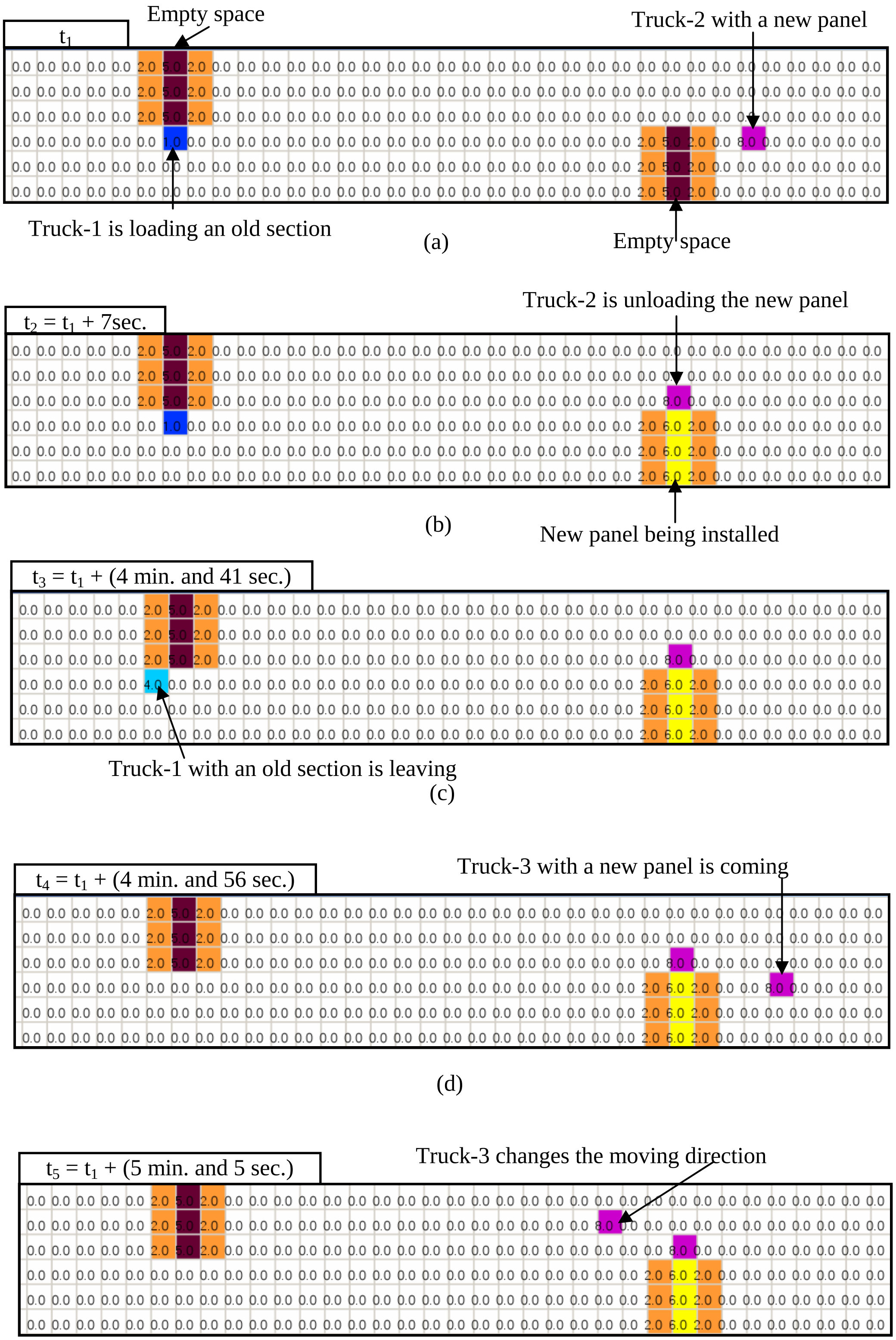


🟨 Cell-DEVS model

□ DEVS model

Q-Old: Queue of trucks for old sections
Q-New: Queue of trucks for new panels
T-*A*-*B*: Transport model from *A* to *B*

Fig. 9. Structure of the cell-based model of Jacques Cartier Bridge re-decking project



t_1

Empty space

Truck-2 with a new panel

0.0	0.0	0.0	0.0	0.0	2.0	5.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	2.0	5.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	2.0	5.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	5.0	2.0	0.0	8.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	5.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	5.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Truck-1 is loading an old section

(a)

Empty space

$t_2 = t_1$ + 7sec.

Truck-2 is unloading the new panel

0.0	0.0	0.0	0.0	0.0	2.0	5.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	2.0	5.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	2.0	5.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	8.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	6.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	6.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	6.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

(b)

New panel being installed

$t_3 = t_1$ + (4 min. and 41 sec.)

0.0	0.0	0.0	0.0	0.0	2.0	5.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	2.0	5.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	2.0	5.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	8.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	6.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	6.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	6.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Truck-1 with an old section is leaving

(c)

$t_4 = t_1$ + (4 min. and 56 sec.)

Truck-3 with a new panel is coming

0.0	0.0	0.0	0.0	0.0	2.0	5.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	2.0	5.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	2.0	5.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	8.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	6.0	2.0	0.0	0.0	8.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	6.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	6.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

(d)

$t_5 = t_1$ + (5 min. and 5 sec.)

Truck-3 changes the moving direction

0.0	0.0	0.0	0.0	0.0	2.0	5.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	2.0	5.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	8.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	2.0	5.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	8.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	6.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	6.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	6.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Fig. 10. Snap shot showing part of the *occupancy layer* of the *Bridge* model

Table 1: Task durations and IDs used in MicroCYCLONE model

Activity	**Task ID**	**Task Description**	**Triangular Distribution of Durations (min)**
Remove old sections	4	Cut old section	15, 18, 30
	9	Load old section	12, 15, 20
	10	Truck with old section travels to dumping area	5, 7, 8
	13	Dump old section	4, 5, 7
	14	Empty old-section truck returns to bridge	4, 5, 7
Install new panels	22	Load new panel	10, 14, 15
	23	Truck with new panel travels to bridge	6, 7, 8
	28	Install new panel	23, 26, 28
	31	New-panel truck returns to plant	4, 5, 6
	30	Team repositioning	15, 18, 20

Table 2: Codes used in the *occupancy layer* of different Cell-DEVS models

Model	**Code**	**Descriptions**
Bridge	0	Empty cell
	1	An empty truck for old sections
	2	Crane (occupies 3 cells)
	3	Old section on the bridge (occupies 3 cells)
	4	A truck carrying an old section
	5	Empty space after removing the old section
	6	New installed panel
	8	A truck carrying a new panel
	9	An empty truck for new panels
Dump Area	1	An empty truck for old sections
	4	A truck carrying an old section
	7	Forklift
Plant	6	Small crane
	8	A truck carrying a new panel
	9	An empty truck for new panels

Table 3: Comparison between cell-based simulation and conventional simulation tools

Simulation method	**Cell-based simulation**	**Conventional simulation tools**
Spatial representation	Explicit representation	Implicit
Spatial conflict detection	Based on site layout	Not available
Duration of transportation activities	Based on predefined delay times and the site layout	Based on predefined durations only
Practical usability	Possibility of reusing models in similar cases	Special purpose functionalities for construction simulation