# Integrating Building Information Modeling & Cell-DEVS Simulation

**Ahmed Sayed Ahmed, Gabriel Wainer, and Samy Mahmoud**
**Department of Systems and Computer Engineering**
**Carleton University**
**1125 Colonel By Drive**
**Ottawa, Ontario, Canada K1S 5B6**
{**asahmed, gwainer, and mahmoud**}**@sce.carleton.ca**

**Keywords:** Cell-DEVS, CD++, BIM, Revit, API

## Abstract

We present the development of an Interactive Environment System (IES). The IES is used for simulating Cell-DEVS models built in CD++ that interact with a Building Information Modeling(BIM) system using Autodesk Revit architecture and Autodesk 3ds Max. CD++ is a modeling and simulation tool that was created to study complex systems by using a discrete event cell-based approach. It was successfully employed to define a variety of models for complex applications using a cell-based approach. The system developed has a highly modular collection of software packages designed to facilitate the creation of device independent simulation for BIM. The integration of the proposed system is investigated via the simulation of Diffusion Limited Aggregation (DLA) which represents the growth of mold in a building wall. The results affirmed the potential of the (IES) system for interactive simulation application.

## 1. INTRODUCTION

The Discrete Event System Specification (DEVS) formalism [1] has gained popularity to model a variety of problems in the recent years. DEVS is a framework for the construction of discrete-event hierarchical modular models, in which behavioral models (atomic) can be integrated forming a hierarchical structural model (coupled). The Cell-DEVS formalism [2] extended DEVS allowing the simulation of discrete-event cellular models. The formalism extends the traditional Cellular Automata [3] by defining each cell as a DEVS atomic model and the space as a DEVS coupled model. DEVS and Cell-DEVS were implemented in the CD++ toolkit [3].

DEVS and Cell-DEVS were used to solve different problems in Construction and Architecture projects [4]. For successful constructions projects, an enormous amount of data should be collected and analyzed in the pre-design phase of the project. Until recently, the success of this phase was dependant on the experience of experts. However, the information required to complete the project and the amount of data that must be analyzed to complete the project is now greater and more complex. Therefore, there is a necessity for tools that can directly support this pre-design phase. Building information Modeling (BIM) has been considered as a tool that can support this part of the construction project [5]. BIM has resulted in improvements to the way architects-contractors and fabricators have been working [6].

In order to solve different design and simulation problems in the field of Building Information Modeling, it is important to be able to produce prototype solutions easily and quickly. The Interactive Environment System (IES) that we present here provides a reconfigurable system to support the simulation of BIM that can run in a Cell-DEVS environment and then feedback the output result of Cell-DEVS Simulation to the BIM system. We decided use a modeling and simulation toolkit CD++ for Cell-DEVS [3], and Autodesk Revit architecture and Autodesk 3ds Max toolkits for BIM [7, 8]. We show a Cell-DEVS/BIM integration and describe a prototype implementation in the form of a BIM add-in tab for Cell-DEVS simulation, and then visualize the output simulation of Cell-DEVS on the BIM model.

We have surveyed the available systems to see if it supports our proposal, such as Bentley Architecture, Graphisoft ArchiCAD, VectorWorks Architect, Autodesk Revit Architecture, and Autodesk 3ds Max [7–11]. These systems allow us to use Building Information Modeling applications and present a 3D visualization that improves the productivity in building design and construction.

Section 2 of this paper describes the background of Discrete Event Systems Specification (DEVS) ,Cell-DEVS modeling simulation approach, and Building Information Modeling (BIM). Section 3 and Section 4 discuss the simulation of Building Information Modeling (BIM)and the 3D visualization of this simulation on a modeling example. Finally a conclusion of this research is presented in Section 5.

## 2. BACKGROUND

Discrete Event systems Specification (DEVS) is a system that allows us to define hierarchical modular models [2]. As shown in Figure 1, a real system

modeled using DEVS can be represented as a set of atomic or coupled submodels. The atomic DEVS model is defined as:

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

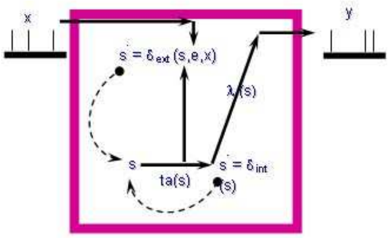



**Figure 1.** DEVS Atomic Model semantics

where **X** is the input events set, $S$ is the state set, $Y$ is the output events set, $\delta_{int}$ is the internal transition function, $\delta_{ext}$ is the external transition function, $\lambda$ is the output function, and $ta$ is the time advance function.

The atomic model can be considered as the base element in which we define dynamics of any system, while the coupled structural model consists of one or more atomic and/or coupled models. Coupled models are defined as a set of basic components (atomic or coupled).The coupled model can be defined as:

$$CM = < X, Y, D, \{M_d \mid d \in D\}, EIC, EOC, IC, select >$$

where X is the input events set, Y is the output events set, D is the set of component names, $M_d$ is a DEVS basic model(i.e., atomic or coupled), EIC is the set of external input couplings, EOC is the set of external output couplings, IC is the set of internal coupling, and select is the tie-breaker function. The coupled model explains how to convert the outputs of a model into inputs for the other models, and how to handle inputs/outputs to and from external models.

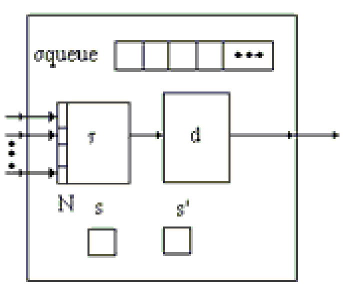



**Figure 2.** A Cell-DEVS atomic model with transport delay

Cell-DEVS [12] has extended the DEVS formalism, allowing us to implement cellular models with timing delays. Once the behavior of a cell is defined, a coupled Cell-DEVS can be created by interconnecting a number of cells with their neighbors. Each cell is defined as a DEVS atomic model, as shown in Figure 2.

Each cell uses N inputs to compute its next state. These inputs, which are received through the model's interface, activate a local computing function (τ). A delay (d) can be associated with each cell. A coupled Cell-DEVS model is the resulting array of cells (atomic models) with given dimensions, borders, and zones (if applicable). Cell-DEVS were implemented using CD++. CD++ has solved a variety of complex problems [13, 14].

The basic features of the CD++ toolkit can be shown by providing an example of an application. Figure 3 shows the definition of a maze solving algorithm example as an application of such models.

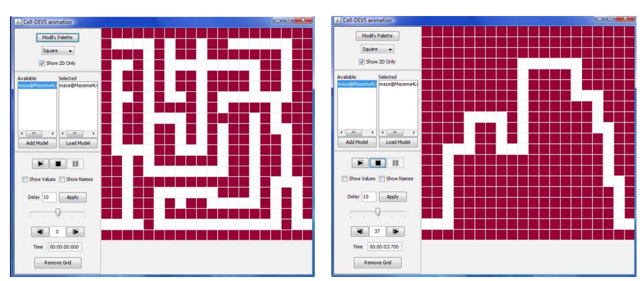
```
[top]
components : maze
[maze]
type : cell
dim : (15, 15)
delay : transport
defaultDelayTime  : 100
border : nowrapped

neighbors :              maze(-1,0)
neighbors : maze(0,-1)  maze(0,0)  maze(0,1)
neighbors :              maze(1,0)

initialvalue : 0
initialCellsValue : maze.val
localtransition : maze-rule

[maze-rule]
rule : 1 100 { (0,0) = 0 and (truecount = 3
        or truecount = 4) }
rule : 0 100 { (0,0) = 0 and truecount < 3 }
rule : 1 100 { t }
```

**Figure 3.** Definition of the maze game

In the example of the maze, the rules are as follows:

- If the cell is a wall cell, the cell remains a wall cell.
- If the number of neighborhoods of a cell is three or more, the cell becomes a wall cell.

When the maze model is executed using these rules, all non-solution paths in the maze are closed successfully. One example of the initial cell state to the maze and the final steady state of the given initial maze cells is shown in Figure 4.



(a) (b)

**Figure 4.** (a)Initial State of Maze (b)Final State of Maze

A Building Information Modeling (BIM) is the process of generating and managing building data during the life cycle of the project [15]. BIM uses three-dimensional, real-time, dynamic building modeling software to increase productivity in building design and construction. BIM allows us to get better and accurate constructional projects with minimized financial costs [16].

BIM software creates parametric 3D models instead of 2D perspective drawings and operates on a digital database where any change made to this database will be reflected in the whole drawing produced. BIM is often associated with IFCs (Industry Foundation Classes) which are data structures used to represent information used in BIM. IFCs were developed by the International Alliance for Interoperability [17].

BIM is considered as an important improvement to the way architects-contractors and fabricators have been working [6] in that it allows us to minimize conflicts between them, and present a 3D visualization of the building during design and fabrication. Therefore, errors made by design team can be minimized, resulting in costs reduction.

There are different simulation applications for BIM such as IDA Indoor Climate and Energy (IDA ICE) and Design-Builder Software [18, 19]. IDA Indoor Climate and Energy (IDA ICE) is a dynamic simulation application that allows us to calculate the thermal indoor climate of individual zones and the energy consumption of the entire building. Design-Builder Software is a simulation software used to check building energy consumption, CO2 emissions, and more building environmental.

# 3. SIMULATION OF BUILDING INFORMATION MODELING (BIM)

To use Cell-DEVS simulation for Building Information Modeling (BIM) models, we built an Interactive Environment System (IES). IES is composed of IES_Revit and IES_Max. IES_Revit is a piece of software, used to integrate BIM and Cell-DEVS, whereas IES_Max is used to visualize the output simulation results on BIM models. IES_Revit will be discussed in this section and IES_Max will be discussed in Section 5. To build IES_Revit, there are two main phases:

- Receive the required data to be simulated from the BIM model.
- Simulate the received data of the BIM model by using Cell-DEVS.

For first phase we used Autodesk Revit Architecture [7] as the implementation software for BIM. This part of the Interactive Environment System for Revit (IES_Revit) was written in Visual C# that provides a graphical user interface invoked in Autodesk Revit Architecture (Add-in tab). IES_Revit sends data to be simulated and visualized in Cell-DEVS.

In the second phase we use CD++ as an implementation software for Cell-DEVS. CD++ is a toolkit that models complex systems using a discrete event cell-based approach. CD++ provides 2D and 3D visualization using VRML and Java [20]. 2D and 3D visualization enables visualization of Cell-DEVS models so that the output of our simulation model will be visualized. Figure 5 shows how IES_Revit interfaces with CD++, Revit Architecture, and other software libraries.

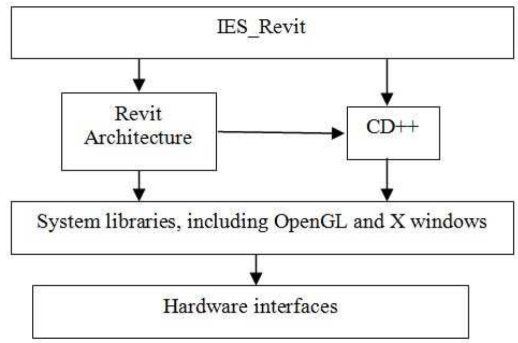



**Figure 5.** IES_Revit Software hierarchy

## 3.1. Receiving BIM data

AutoDesk Revit Architecture is a Parametric Building Information Modeling (BIM) tool, in which we can build 3D models and 2D drawings. We can develope different tasks using Revit API. The Revit API allows us to create and delete different model elements like floors, walls, and more. We also use Revit API to get different model parameter data and model model graphical data. Revit Platform API applications can be develop using Visual C# or VB.NET. We decide to use Visual C# for our system.

We create a 'SendToDEVS' class library to be invoked in the add-in tab of AutoDesk Revit Architecture. On the Add-Ins tab, 'SendToDEVS' appears in the External Tools menu-button, as shown in Figure 6. The SendToDEVS is responsible to execute the program to get the required parameters and send it to Cell-DEVS to be simulated.

SendToDEVS includes two functions:

- The GetParameter function, and
- The WriteMacro function

SendToDEVS receives the required parameters of the chosen item (e.g. a wall) in the active Revit document by calling the GetParameter function. The GetParameter function receives the parameters of the selected element in the active document of Autodesk Revit architecture. The GetParameter function sends the parameters to the WriteMacro function. The WriteMacro function is responsible to write the value of the received parameter of the chosen element of the active

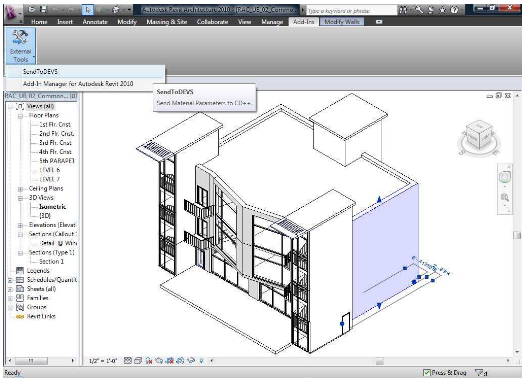


**Figure 6.** IES_Revit Interface

Revit document into RevitMacro.inc file. The RevitMacro.inc file now contains the new value, which will be simulated in Cell-DEVS. The RevitMacro is explained briefly in Subsection 3.2. If we choose another element in the same active document and click the SendToDEVS add-in tab in the External Tools menu-button, the GetParameter function will receive the new parameters of the new selected element. Then the WriteMacro function will write the new value of the received parameter into the RevitMacro.inc file to be modified.

Finally to invoke the application into AutoDesk Revit Architecture, we should modify the Revit.ini file to register it into Revit by adding the following code to the end of the existing code of Revit.ini file:

```
ECCount=1
ECName1=SendToDEVS
ECClassName1=Integration.CS. SendToDEVS
ECAssembly1=C:\Revit\bin\Debug\SendToDEVS.dll
ECDescription1=Send Material Parameters to CD++
```

## 3.2. Simulation of the received BIM data

Diffusion Limited Aggregation occurs when diffusing particles stick to and progressively enlarge an initial seed represented by a fixed object. The seed typically grows in an irregular shape resembling frost on a window [21]. We use the Cell-DEVS Diffusion Limited Aggregation (DLA) model as an example of a Cell-DEVS simulation. The DLA model is defined using CD++ toolkit. CD++ includes an interpreter for a specification language that describes Cell-DEVS models. A set of rules is used to define the specification language. Each rule indicates the output value for the cell's state after satisfying the precondition in this rule. These rules are performed sequentially until one rule produces the solution. Figure 7 illustrates the architecture of the IES_Revit. We have modified the Cell-DEVS model of DLA [22] by using a macro definition. We use the macro definition to read the parameters value received from BIM model. We have defined a revit-macro in the main .ma file (model file) of DLA. The model file (.MA) allows us to define the behavior of Cell-DEVS models. This model file reads the initialization data from a .VAL file. The val file has the initial values of each cell of the model in the simulation. The log file is generated when we ran the simulation. The output simulation could be shown in 2D visualization using CD++ modeler tool of CD++ toolkit.

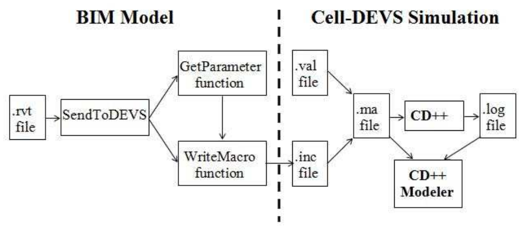



**Figure 7.** BIM/Cell-DEVS architecture

The DLA model is represented by two types of particles; fixed particles (seeds) and mobile particles. There are one or more seeds in each DLA Cell-DEVS model. A cell that has a seed is fixed and has a value equal to 5. There are a mobile particle percentage of the cells in each DLA Cell-DEVS model. A mobile particle can move according to its value in four directions: up (1), right (2), down (3) and left (4). We set an initial value from 1 to 4 randomly to occupy the cells in a certain concentration. This concentration is calculated and obtained in the Rivet Macro from the BIM model.

```
% initialize the cells with mobile particles
% in the range with value of concentration
rule : {round(uniform(1,4))} 100 {(0,0) = -1
       and random < #macro(Revit)}
```

The following rule presents that fixed particles remains fixed:

```
% fixed particles remains to be fixed
rule : 5 100 { (0,0)=5 }
```

The following rules present the moving of mobile particles:
- A cell has a mobile particle with value equal one can move to the above empty cell if there is no other mobile particle trying to move in to this empty cell.

```
% direction = 1 (up)
% stay and change direction when nowhere to move
rule : {round(uniform(1,4))} 100 {(0,0)=1
    and (-1,0)!=0}
rule : {round(uniform(1,4))} 100 {(0,0)=1
    and (-1,0)=0 and (((-2,0)=3 and(-2,-1)!=5
    and(-3,0) !=5 and(-2,1)!=5) or ((-1,-1)=2
    and(-1,-2)!=5 and(-2,-1)!=5 and(0,-1)!=5)
    or ((-1,1)=4 and(-2,1) !=5 and(-1,2) !=5
    and(0,1)!=5))}
% move otherwise
rule : 0 100 {(0,0)=1 and (-1,0)=0 and t}
```

- A cell has a mobile particle with value equal two can move to the right empty cell if there is no other mobile particle trying to move in to this empty cell.

```
% direction = 2 (right)
% stay and change direction when nowhere to move
rule : {round(uniform(1,4))} 100 {(0,0)=2
    and (0,1)!=0}
rule : {round(uniform(1,4))} 100 {(0,0)=2
    and (0,1)=0 and (((0,2)=4 and (−1,2)!=5
    and (0,3)!=5 and (1,2)!=5) or((−1,1)=3
    and (−1,0)!=5 and(−2,1)!=5 and (−1,2)!=5))}
% move otherwise
rule : 0 100 {(0,0)=2 and (0,1)=0 and t}
```

- A cell has a mobile particle with value equal three can move to the down empty cell if there is no other mobile particle trying to move in to this empty cell.

```
% direction = 3 (down)
% stay and change direction when nowhere to move
rule : {round(uniform(1,4))} 100 {(0,0)=3
    and (1,0)!=0}
rule : {round(uniform(1,4))} 100 {(0,0)=3
    and (1,0)=0 and ((1,1)=4 and (0,1)!=5
    and (1,2) !=5 and (2,1) !=5 )}
% move otherwise
rule : 0 100 { (0,0)=3 and (1,0)=0 and t}
```

- A cell has a mobile particle with value equal four can move to the left empty cell if there is no other mobile particle trying to move in to this empty cell.

```
% direction = 4 (left)
% stay and change direction when nowhere to move
rule : {round(uniform(1,4))} 100 {(0,0)=4
    and (0,−1)!=0}
% move otherwise
rule : 0 100 { (0,0)=4 and (0,−1)=0 and t}
```

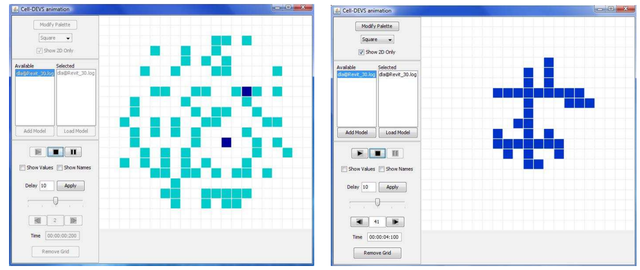
- A cell has a mobile particle becomes fixed if there is an adjacent fixed particle cell.

```
% the particle becomes fixed if an adjacent cell
% contains fixed particle
rule : 5 100 { (0,0) >0 and (0,0)<5 and
    ((−1,0)=5 or (0,−1)=5 or (0,1)=5 or (1,0)=5)}
```

We assume a DLA Cell-DEVS model with two initial seeds. The concentration percentage of mobile particles will vary due to the material parameter type value received from BIM. We ran the simulation for two different materials for the specified two seeds DLA Cell-DEVS model; one for concrete and the other for brick. We assume that the percentage of concentration will be 30% for the concrete material and 40% for the brick material. The simulation output of each run for the concrete and the brick is shown in Figure 8 and Figure 9 respectively.

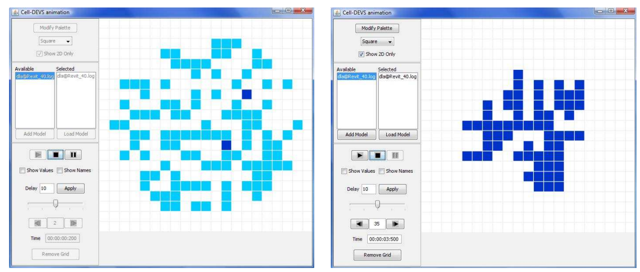
## 4. 3D VISUALIZATION DESCRIPTION

IES_Max is used to visualize the output simulation results of BIM models. IES_Max is implemented by using Autodesk 3ds Max [8]. Autodesk 3ds Max is used because it supports BIM and has a great 3D environment scene. Figure 10 illustrates how IES_Max interfaces with CD++, 3ds Max, and other software libraries.



(a) (b)

**Figure 8.** (a)Initial state for 30% concentration (b)Final state for 30% concentration



(a) (b)

**Figure 9.** (a)Initial state for 40% concentration (b)Final state for 40% concentration

### 4.1. IES_Max Configuration

We build a graphical display output using 3D visualization tools. We decided to expand our visual environment using Autodesk 3ds Max. Autodesk 3ds Max is a powerful application for 3D modeling and animation, using special effects and rendering. 3ds Max allows users to create 3D animation and visual effects. More functions can be added to Autodesk 3ds Max by using MAXScript. MAXScript is a built-in script language that facilitate the creation of functions and tools to enhance 3ds Max efficiently. We used 3ds Max modeling and animation toolkit to create 3D visual environments for Cell-DEVS Simulation of Diffusion Limited Aggregation as example. IES_Max is an application written in MAXScript that provides a graphical user interface that allows CD++ files (*.ma and *.log files) to interact with 3ds Max, and to visualize the corresponding Cell-DEVS simulation in a 3D visual environment scene of the BIM model. This BIM model, which is exported as FBX file(type of Autodesk file formats) from Autodesk Revit Architecture, is imported in the 3ds Max. Then IES_Max animates the 3D visual scene file in accordance with the CD++ files.

Figure 11 illustrates the architecture of the IES_Max. We use the ReadMa function to read the model file (.MA) of the Cell-DEVS model, which contains the definition of behavior of Cell-DEVS models and the initial data or the path of the

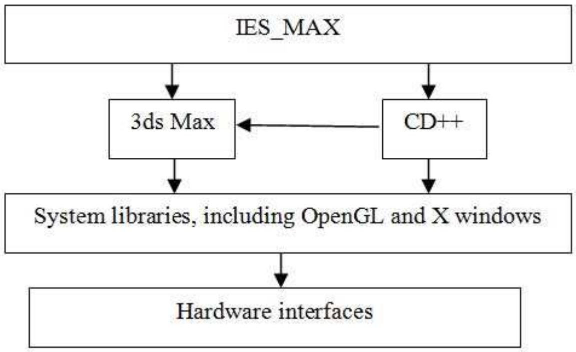



**Figure 10.** Software hierarchy

val file (.val). The val file has the initial values of each cell of the model in the simulation. So the ReadMa function has the initial status of the Cell-DEVS. The ReadLog function is used to read the log file (.log) of the Cell-DEVS model, which contains all the steps of output simulation results with time. So the ReadMa function has all the status during the simulation lifetime and the final simulation results. Finally, the Draw function draws the collected simulation data in the 3d Max visualization scene.

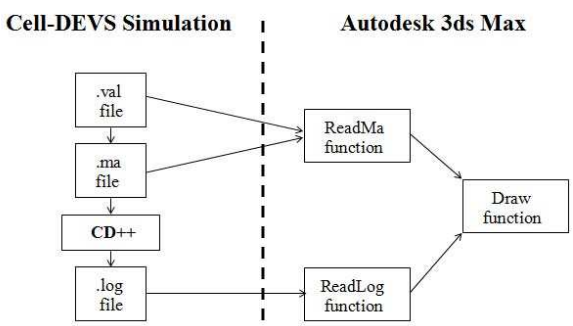



**Figure 11.** IES_Max architecture

IES_Max consists of Graphical User Interface (GUI). This GUI, as shown in Figure 12, is the graphical interface that requests the user to select a particular file. The GUI consists of 4 buttons. The"Select Ma File" button is used to open the *.ma file corresponding to the file name provided for reading the dimension of the Cell-DEVS and the val file which contains the initial values for the simulation, and "Select Log File" button is used to open the *.log file corresponding to the file name which is provided for reading the output of the Cell-DEVS simulation. The"ExecuteLog" button is used to execute and display the 3D visualization scene in which 3D models are created to be animated due to the data loaded from the ma and val files. Finally, the"Clear" button is used to remove all objects in the Display Window to run another simulation model if needed.

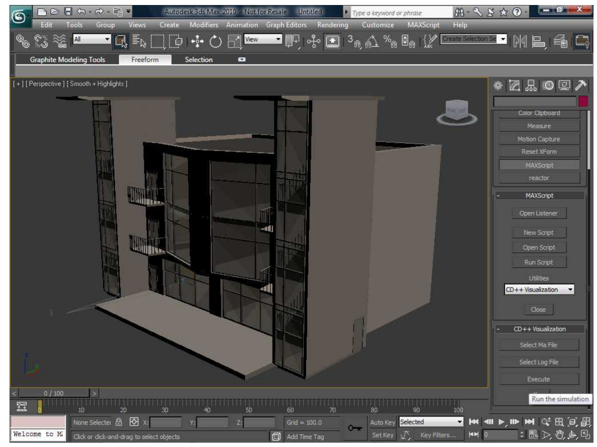


**Figure 12.** IES_Max 3ds Max Interface

IES_Max allows us to create a 3D visualization from the CD++ files Created by CD++ toolkit. 3ds Max has implicit support for hardware accelerated rendering. The 3ds Max visualization tool provides basic services that enable simple visualizations, including design and implementation of a graphical user interface based on the MAXScript within 3ds Max toolkit.

## 4.2. IES_Max Implementation

We use the MAXScript language which is the 3ds Max Toolkit script for writing the program to initialize the GUI interface window for the 3D visualization scene. The GUI consists of four buttons:

- The"Select Ma File" button is designed for selecting and opening the *.ma file.

- The"Select Log File" button is designed for selecting and opening the *.log file.

- The "ExecuteLog" button is designed for executing the 3D visualization in the Display Window.

- The "Clear" button is designed for clearing the 3D visualization in the Display Window from any object if you need to run another simulation model.

We use the ReadMa function and the ReadLog function for reading output simulation data of CD++ file, and Draw function for displaying the 3D visualization. The ReadMa function is used to read the *.ma file which contains the dimensions of the simulation model and the val file name, then reformat it to be used in the required argument that passed to the ReadVal function. The ReadVal function is used to read the *.val file which contains the initial values of each object from the simulation model and reformat it to be used in the required argument that passed to the Draw function. The ReadLog function is used to read the *.log file which contains time and position of each object from the simulation model and reformat it to be used in the required argument that passed to the Draw function. The Draw function is used to create objects and display the 3D visualization of the CD++ simulation model in the Display window of 3ds Max. The Draw function receives the dimension of the simulation model which controls the size of the drawing area and the position of each cell to be drawn in the specified location.

## 5. CONCLUSIONS

This paper describes our implementation of the Interactive Environment System (IES) as an integration of Cell-DEVS formalism into Building Information Modeling (BIM). Cell-DEVS approach can be applied to improve the development of Building Information Modeling. CD++ is used as a toolkit for Cell-DEVS models. We use Autodesk Revit Architecture and 3ds Max as toolkits for BIM. We implemented IES_Revit as an integration between BIM into Cell-DEVS simulation. The output simulation result of the Cell-DEVS model has been visualized in 3ds Max using IES_Max.

The Diffusion Limited Aggregation (DLA) model example was used to verify the feasibility of combining these two technologies by using IES.

## REFERENCES


[1] Bernard P. Zeigler, Tag Gon Kim, and Herbert Praehofer. *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*. Academic Press, 2000.

[2] Gabriel A. Wainer and Norbert Giambiasi. *Timed Cell-DEVS: modelling and simulation of cell spaces*. Springer-Verlag, 2001.

[3] Gabriel A. Wainer. CD++: a toolkit to define discrete-event models. *Software, Practice and Experience*, 32(3):1261–1306, November 2002.

[4] Hong Pang, Cheng Zhang, and Amin Hammad. Sensitivity analysis of construction simulation using cell-devs and microcyclone. In *WSC '06: Proceedings of the 38th conference on Winter simulation*, pages 2021–2028, 2006.

[5] Nam-Hyuk Ham, Kyung-Min Min, Ju-Hyung Kim, Yoon-Sun Lee, and Jae-Jun Kim. A study on application of bim (building information modeling) to pre-design in construction project. In *Third International Conference on Convergence and Hybrid Information Technology, 2008. ICCIT '08.*, volume 1, pages 42–49, November 2008.

[6] Brad Hardin. *BIM and Construction Management: Proven Tools, Methods, and Workflows*. Wiley, 2009.

[7] AutoDesk. Autodesk Revit Architecture, November 2009. `http://en.autodesk.ca/adsk/servlet/index?id=12436708&siteID=9719649`.

[8] AutoDesk. Autodesk 3ds Max, November 2009. `http://en.autodesk.ca/adsk/servlet/index?id=12341323&siteID=9719649`.

[9] Bentley. Bentley Architecture, November 2009. `http://www.bentley.com/en-US/Products/Bentley+Architecture/`.

[10] Graphisoft. Archicad 13, November 2009. `http://www.graphisoft.com/products/archicad/`.

[11] VectorWorks. Vectorworks Architect, November 2009. `http://www.nemetschek.net/architect/index.php`.

[12] Gabriel A. Wainer. *Discrete-Event Modeling and Simulation: a Practitioner approach*. Taylor and Francis, 2009.

[13] Javier Ameghino, Alejandro Troccoli, and Gabriel A. Wainer. Modeling and simulation of complex physical systems using cell-devs, 2001.

[14] Gabriel A. Wainer. Modeling and simulation of complex systems with cell-devs, 2004.

[15] Ghang Lee, Rafael Sacks, and Charles M. Eastman. Specifying parametric building object behavior (bob) for a building information modeling system. *Automation in Construction*, 15(6):758–776, 2006.

[16] S. Mihindu and Y. Arayici. Digital construction through bim systems will drive the re-engineering of construction business practices. In *International Conference Visualisation*, pages 29–34, July 2008.

[17] IAI/IFC. International Alliance for Interoperability, Industrial Foundation Classes, November 2009. `http://www.iai-tech.org/`.

[18] EQUA. Equa simulation, simulation software, simulation consultancy, November 2009. `http://www.equa.se/`.

[19] DesignBuilder. Building design, simulation and visualization, November 2009. `http://www.designbuilder.co.uk/`.

[20] Gabriel A. Wainer and Wenhong Chen. A framework for remote execution and visualization of Cell-DEVS models. *Simulation: Transactions of the Society for Modeling and Simulation International*, 79(11):626–647, 2003.

[21] Tommaso Toffoli and Norman Margolus. *Cellular Automata Machines: A New Environment for Modeling*. MIT Press, 1987.

[22] Wei Ding, Changshuang Lin, Laurentiu Chechiu, X. Wu, and Gabriel A. Wainer. Definition of cell-devs models for complex diffusion systems, 2005.