

## **LOOSELY COUPLED VISUALIZATION OF INDUSTRIAL CONSTRUCTION SIMULATION USING A GAMING ENGINE**

Amr ElNimr  
Yasser Mohamed

Department of Civil & Environmental Engineering  
University of Alberta  
Edmonton, AB T6G 2W2, CANADA

### **ABSTRACT**

The use of simulation in construction project management is not widely adopted. Effective and intuitive tools and techniques to communicate simulation models with industry practitioners are needed. Visualization of simulation behaviors using three dimensional virtual worlds of the simulated construction operations is an effective medium of communication. However, developing visual behaviors to reflect hidden simulation behaviors is time consuming. The relatively small time window available for developing and using simulation models on real construction operations requires a time and cost effective approach for developing simulation driven visualization. This paper describes an approach that utilizes an open source gaming engine to develop parallel and loosely coupled simulation-driven visualizations of industrial construction operations in a distributed simulation environment. The paper focuses mainly on the development pipeline in a step-by-step approach to document and facilitate application of the same approach in similar simulations.

### **1 INTRODUCTION**

Simulation modeling is an effective approach for analyzing construction operations, yet it is not widely used by construction practitioners. Visualization in general, and particularly in construction projects, is a convenient and intuitive way of conveying project information among various project parties. Recently, construction management researchers have been investigating adding visualization components to construction simulation models in order to make these models more intuitive and appealing to decision-makers. These researchers argue that enhancing visualization and spatial representation of construction operations in a simulation environment can improve the adoption of simulation techniques by the industry.

Simulation is defined as the process of designing a model of a real system and conducting experiments with this model for the purpose of either understanding the behavior of the system or evaluating various strategies for the operation of the system (Shannon 1975). The strength of simulation approaches emanates from their ability to examine various scenarios of the simulated system, rather than giving a mathematically optimum result as is the case in mathematical optimizations. Most researches stress the fact that simulation is not used to its maximum potential in the construction industry (Ioannou and Martinez 1996; Kamat and Martinez 2001; Huang and Halpin 1994; Tucker et al. 1998). There are two major reasons for this lack of use:

- Most simulation models show results in statistical and text formats, which causes simulation to be perceived by the construction industry as a “black box.”

- The current simulation models do not consider the transformation of space that result from the evolution of the constructed product (Kamat and Martinez 2001).

In order to overcome the limitations of simulation and to encourage its application in the construction industry, researchers turned to SDV (Simulation Driven Visualization). It incorporates 2D and 3D depictions and animation to provide the end user with better representation and understanding of construction operations and space. It also addresses the limitations of using stand-alone animation or simulation systems where graphical representations of construction operations and simulation of activity timing and resource interactions are separate and independent of each other.

There are SDV commercial, off the shelf (COTS) packages that exist in the market. These include Delmia's Quest® and Brooks Software AutoMod®. However, these COTS are generally focused on manufacturing operations. They are usually unable to effectively handle the additional complications introduced by the changes in the geometry of the construction site as work progresses, and their use to model and animate construction operations requires a radical change in the model conceptualization and thought process followed by construction model developers (Oloufa 1993; Kamat and Martinez 2008). Another visualization and CAD package that is used in the construction industry is CATIA®. Although powerful on the CAD side, it is limited on the simulation side of processes as it lacks a discrete event simulation engine. CATIA can be a better solution for 4D modeling, final product development modeling, but not for simulation of construction processes.

The first construction simulation tools to adopt graphics did so in the early 80s (Rohrer 2000). This was done through attempts to link construction simulation packages to animation to achieve a post-process playback of the simulation (Halpin and Woodhead 1976; Kalk and Douglas 1980; Paulson et al. 1983). The "Utopian Framework" for earth moving operations was a good example (Oloufa and Crandall 1992). Another attempt to extract product information from a CAD design into a simulation was done through the PSE (Product-oriented Simulation Environment) (Xu and AbouRizk 1999). Recent research in construction operations SDV involves the use of simulation trace files to invoke post-simulation animation (replay) of the simulation. This includes the Dynamic Construction Visualizer (DCV) and Visualization of Simulated Construction Operations (VITASCOPE) (Kamat and Martinez 2001; Kamat and Martinez 2004).

The above simulation visualization packages (both commercial and research oriented) provide advanced visualization capabilities. However, they inherit one or more of the following characteristics that limit their use in visualizing simulation behaviors of construction operations.

- Post-processing visualization: After a simulation is complete, visualization is produced as a replayed record of what took place during the simulation run. It does not allow the decision-maker to interact with the simulation. This causes the visualization component also to rely on simulation trace statements produced by the simulation engine exclusively for visualization purposes. Recent efforts by Rekapalli and Martinez (2009) and Rekapalli and Martinez (2011) has focused on achieving two-way communication and user interaction with the visualization component of a DES visualization.
- Tight coupling between simulation and visualization engines: Existing SDV mechanisms and COTS packages are usually tightly coupled with whatever simulation engine they serve. A particular visualization component will usually work perfectly with a particular simulation engine but will be very time consuming or knowledgeably demanding to tailor it to work with other simulation engines. This leads to compromises on the strengths of both the simulation and visualization components. Also, this makes the two tightly coupled components strong in depicting certain applications and weak when it comes to depicting others. Finally this decreases the reusability these tightly coupled frameworks or COTS packages and limits their use in day to day construction operations.
- Compromising the power of the visualization components: Graphics usually demands specialized and high computer processing power. Running simulation models is also a demanding task on computer processors. Running the two in parallel significantly increases the demand on computer

hardware and may be deemed infeasible in case of highly detailed simulation and visualization models. In addition, there have been significant advancements and optimization of software specially developed for handling graphics and visualization tasks. Capitalizing on these advancements requires the development of a mechanism to allow different specialized simulation and visualization components to interoperate without limiting or compromising the strengths of each.

- The existing SDV mechanisms do not allow information to be sent back from the visualization to simulation. The simulation scenarios can't be evaluated or adjusted based on the visualization output (one way data flow only). This makes it complicated for a construction team to use the current SDV mechanisms to try various construction processes and site layout scenarios.

To overcome some of the above limitations, a loosely coupled SDV framework was proposed (ElNimr and Mohamed 2010; ElNimr and Mohamed 2011). In this framework, using distributed simulation strategies, the execution of the visualization components is parallel and independent from the simulation engine. Execution of these components can also be carried by completely separate and dedicated computer systems allowing the utilization of the strengths of each component independently and exclusively. In addition, the framework aims at capitalizing on the strengths of existing high-level 3D graphics or game engines to reduce the effort required to customize and develop visualization components.

This paper discusses the development process of visualization components for industrial construction simulation following the proposed framework. The paper summarizes the main components of this framework then it outlines the main development steps and illustrates the execution of each step using industrial construction visualization components as examples. The paper also highlight the main advantages and limitations experienced from applying the framework to the presented case.

## 2 BACKGROUND

### 2.1 Distributed Simulation Driven Visualization

The proposed visualization framework relies on distributed simulation concepts. The distributed simulation standards used in this work is the High Level Architecture (HLA) (IEEE 2000). As such, the generation of simulation behaviors is handled by dedicated simulation model(s) (federates), while the display of these behaviors in different visual forms is handled by the visualization components (visualization federates). An HLA compliant run time infrastructure (RTI) handles the communication and time synchronization between the different federates. Figure 1 shows an outline of the main components needed to enable development of the visualization components under the proposed framework. The main focus in this paper is on the development steps of the visualization federates.

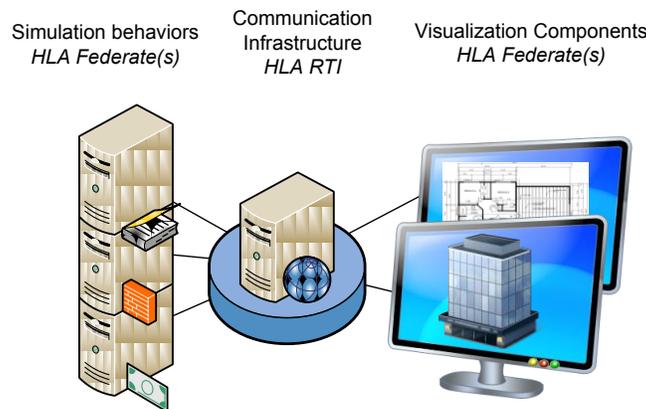


Figure 1: Distributed Simulation Driven Visualization

## 2.2 Industrial Construction Operations

Construction of industrial projects like oil refineries and petrochemical plants involves a large number of interdependent processes. It usually follows a modular construction approach where modules are assembled out of prefabricated structural steel, pipe spools, vessels, and equipment. Modules are later shipped and installed on site using heavy-lift cranes. Industrial construction operations include engineering, material procurement, fabrication, assembly, and site installation processes. A distributed simulation federation was developed to represent these processes as practiced in Alberta, Canada. The federation is described in AbouRizk et al. (2010) and it includes a number of federates that simulate the main processes in industrial construction. The visualization components described in this paper were developed to work as part of that federation and runs in parallel to other federates.

## 3 DEVELOPING VISUALIZATION COMPONENTS FOR INDUSTRIAL CONSTRUCTION

The development of visualization components for industrial construction simulation can be broken down into a number of steps that can be followed for structuring the development of similar visualizations in other construction operations domains. These steps are illustrated in Figure 2. They include 1) deciding on the simulation objects that need to be present in the visualization and the behaviors that need to be displayed for each of these objects, 2) preparation of visual assets that will represent each of these objects, 3) deciding on a visualization engine to process the display of the objects and their behaviors, 4) customizing the visual behaviors of the objects using the selected engine and developing the suitable triggers for these behaviors, and 5) connecting the visualization component to the simulation and hooking the triggers to the appropriate simulation events. The following sections explain each of these steps with examples from the industrial construction visualization components.

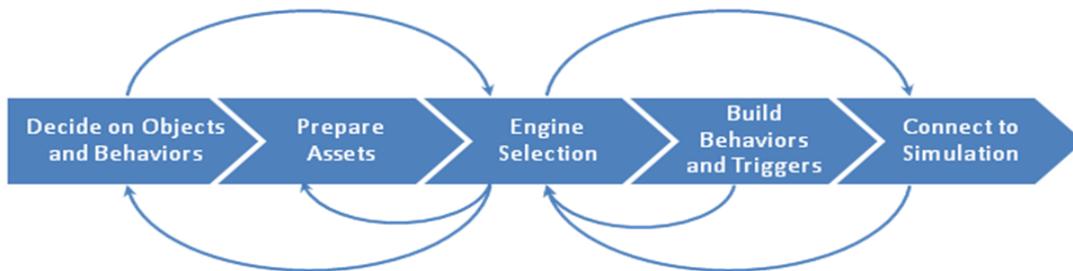


Figure 2: Main Steps in Developing Distributed Visualization Components

### 3.1 Deciding on Objects and Behaviors

A simulation model usually produces different behaviors for different entities involved in it. These “simulation behaviors” takes different forms and may include, for example, creation of new entities, entities waiting for service, entities or resources movements, or other kinds of time consuming tasks performed by resources on entities. Developing visualizations of these behaviors involves mapping all or some of them from abstract mathematical format to a different graphical representation to establish the link between them and their real-life counterparts. Therefore, a critical step in designing a visualization component for simulation is to decide on what objects need to be present in the visualization?, what simulation behaviors need to be mapped to visual behaviors?, and what is the degree of abstraction that is required in the visualization? For example, one may need to include the visualization of a major piece of equipment like a crane and its location at a given point in time but not the human drivers or workers interacting with it, nor the details of its movement from one point on site to another. In addition, the representation of the piece of equipment may be as simple as an icon or a point on a two dimensional (2D) map to a three di-

mensional (3D) photo-realistic rendering of the actual equipment. This step depends mainly on the objective(s) and intended use of the visualization components, and their intended audiences.

In the case of industrial construction operations, two points in the supply chain were selected for visualization. These are module assembly operations (Yard Viewer) and the installation of modules on site (Site Viewer). An HLA federate was built to represent each of these view points. On the module assembly side, the decision was to have representations of the actual site layout and the progress of modules at the different assembly bays at different points in time. On the site installation side, the decision was to have representations of the planned locations of the different modules, the progress of installation of modules at these locations, and the interaction of mobile cranes with these locations over time. It was also decided that the presentation of the objects will be in 3D low polygon geometry to be as close as possible to 3D CAD models that construction practitioners are familiar with. While these decisions were made for mainly experimental and research objectives, some are driven by actual needs from practitioners (e.g. crane interactions with site space at different times). Figure 3 shows sample snapshots of the different behaviors explained earlier.

It should be noted here that the design and decision making process at this stage are highly unstructured and could be iterative. Using the principles of display design (Wickens 2004) can be of value at this stage.

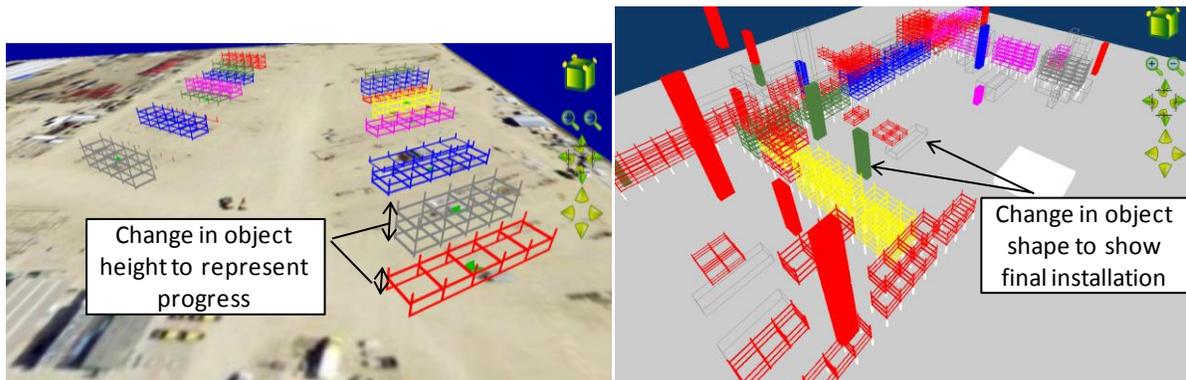


Figure 3: Sample Behaviors of Objects in Module Assembly and Site Installation Viewers

### 3.2 Preparing Assets

Depending on the required objects and behaviors sought from the previous step, a set of graphical assets needs to be prepared. These may include images, icons, 2D vector maps, or detailed 3D models. The formats of these assets depends highly on the graphics engine that will be utilized to display them, and the visual behaviors they need to comprise. Preparing these assets from scratch for a construction operation/project can be a time consuming task and therefore, developers should capitalize as much as possible on already available assets that may have been developed for other purposes.

In the industrial construction case, and most likely in other construction operations, the main sources for visual assets are the engineering drawings for the project to be simulated. However, these drawings are usually highly detailed, and include far more information than is required for simulation visualization. This required some clean-up and detail reduction steps where drawing layers that includes unneeded information were removed. Some objects (e.g. modules) were also preprocessed inside the CAD system to only keep their outer dimensions and main structural steel elements and remove their detailed inner design. For the overall site, bounding boxes that represent each module and its final location on site were also automatically generated from the CAD system. For mobile cranes, 3D models that represent the critical dimensions of the cranes but with low level of details were used. Some crane models with higher level of details were also imported from Google's online 3D warehouse mainly for experimental purposes. These

models were also processed to reduce their polygon counts. For transferring these assets from one system to another, 3D file formats represented a challenge due to the loss of some attributes when translating from one format to another. 3DS format was used as the medium format as it maintained most of the needed attributes between translations. The choice of a particular format depends on the 3D applications in use and their capabilities. Figure 4 shows samples of the assets used in the industrial construction visualization federate.

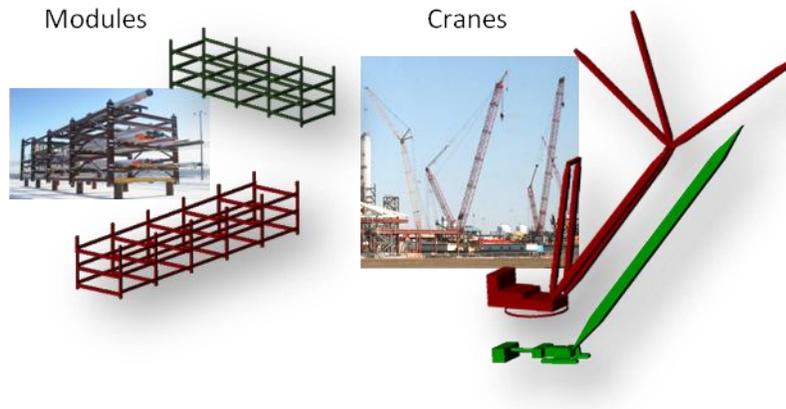


Figure 4: Sample 3D Objects Used by the Visualization Component

### 3.3 Selection of a Graphics Engine

Processing and displaying graphical objects require specialized software libraries and computations that are different from simulation computations. Dedicated graphics engines and applications are usually optimized for handling such computations but they vary in the level of flexibility and control provided to the developer or user. Selecting a suitable graphics engine to handle the visualization of construction simulations in a distributed environment depends on the objectives of the visualization, the type of graphics that needs to be processed, and the programming language the engine needs to interface.

In the case of industrial construction simulation, the authors experimented with a number of graphics engines. The goal was to select an engine that can handle 3D graphics with medium to low degree of realism, enable wide range of visual behaviors for the displayed 3D models, requires minimum programming experience and effort, provide high level services for rapid prototyping and fast development, provide an application programming interface (API) in Visual Basic (VB) .NET, has high degree of extensibility to facilitate connection to other programs at run time, and has support for a wide range of graphical formats especially CAD oriented formats. Among these requirements high extensibility, fast development, and minimum programming effort were the highest on the list. After experimenting with a few engines and applications, the choice was made to use the open source software Blender (Blender Foundation 2011). The program has a high level game customization environment that depends mainly on graphical constructs for producing visual behaviors and interactions with users. However, it also provide a very extension capabilities for producing complex behaviors through a Python scripting interface. Although it does not have an API for VB .NET, its Python API is highly efficient and provide access to a wide range of services including external Python modules. The following section discusses the use of the game engine in Blender to customize the behaviors of industrial construction objects. Another application that could provide similar capabilities is VirTools® (Dassault Systems 2011). However, there were issues with openness of the development environment and programming language that made connectivity to a running HLA federation more challenging.

### 3.4 Customization of Visual Behaviors

Generating the required visual behaviors from different graphical objects is greatly affected by the capabilities of the graphics engine used and the amount of effort allocated to this activity. As discussed earlier, construction projects and hence their simulations usually have short time windows for decisions. Therefore developing visualizations for construction simulations needs to be time efficient.

For the industrial construction visualization, a number of behaviors were customized for the 3D objects using Blender’s game engine logic bricks. These logic bricks provide an intuitive approach for customizing the behavior of any object in the 3D scene. The bricks are graphical elements as shown in Figure 5 and are broken down into three main classes: sensors, controllers, and actuators. An object in the 3D scene equipped with these bricks uses sensors to receive messages from other objects, user, or system interactions; runs these messages through controllers to decide on appropriate responses, and generate the appropriate visual behaviors through the actuators. Samples of these behaviors are explained in the following sections to illustrate the graphical approach used for building these behaviors.

#### 3.4.1 Display of a Finished Module on Site

As federates responsible for simulation behaviors run, they generate information on the state of different modules on site at different points in time. One part of the information generated is an attribute (*ModuleFieldLocation*) that describes whether a module is at *storage*, *pick point*, or *set point* (i.e. final location). The visual behavior described in this section is triggered by changes in this attribute. If a module is not shipped to site yet (i.e. attribute value is not available), the module is represented visually by a an object that shows only the outer boundaries (bounding box) of the module. The object has a sensor that listens to messages related to the *ModuleFieldLocation* attribute. When a message is received and indicates a change in location to *set point* the object controller checks the module ID to make sure the message is handled by its rightful owner. Then, two actuator are triggered. The first actuator adds a prototype model of the module to the scene at its final location. The second one deletes the bounding box object from the scene as it is no longer required. The sensors, controllers, and actuators bricks are shown in Figure 5.



Figure 5: Logic Bricks for Displaying Modules at Final Location in Site Viewer

#### 3.4.2 Crane Changing Location on Site

Another visual behavior that is produced by the Site Viewer represents the changes in mobile crane locations over time. Similar to the previous behavior, message sensors in the Site Viewer Federate receive the changes in crane attribute (*Crane ID* and *Location ID*) values. Messages are time-stamped and fired when the simulation federates schedule a lift activity involving a certain mobile crane. Once the *Location ID* value for a certain crane changes, actuators delete the crane object from its current location and add it to the new location. The change in location is discrete in both simulation and visualization federates. The incremental movement of a crane from one location to another is not simulated or displayed. In addition, crane locations are preset and finite. This matches the actual practice on site where crane locations and

module pick points are decided and prepared prior to construction start. In an extension of the site viewer federate, a path finding algorithm implemented in a Python script is used to make sure a feasible path exists between the start and end locations of a moving crane.

### 3.4.3 Progress of Module Assembly on Assembly Yard

In module assembly operations, simulation federates assign modules to assembly bays and simulate their progress with time until they are shipped out to site. The yard viewer displays visual behaviors to some of the simulated operations. A bay object in the 3D scene handles the production of some of these behaviors. Figure 6 shows the logic bricks for a bay object. The object is equipped with three sensors to receive messages with subjects “ASSGN”, “PRGRS”, and “SHIP”. These messages instruct the object to produce three different visual behaviors that represent 1) assignment of a bay to a module, 2) a certain progress of the module, or 3) shipping of a module out of the yard.

An “ASSGN” message will trigger four actuators: the first one adds the module to the bay that is receiving this message, the second and third resets progress properties to 0, and the fourth resets an interpolated animation curve (IPO) to its original state, which represents 0 progress of the module. The IPO curve animates the visibility of the module object in the Z dimension and can be controlled to show the progress at any animation frame to represent different states of the module from 0% complete to 100% complete. A “PRGRS” message will first signal its controller to read the percentage of progress sent from the federate. Then, the controller will trigger two actuators. The first actuator will calculate an animation frame to represent the percentage of progress sent by simulation federates. The second actuator uses the calculated frame number to change the look of the 3D object that represent the module based on the predefined IPO. The last message (“SHIP”) triggers three actuators that reset progress properties and remove the 3D module prototype from the scene.

Although this behavior demonstrates the flexibility of the utilized tools, visualizing the progress of a module’s assembly through controlled visibility in the Z dimension in the 3D scene is not a strongly noticeable behavior unless the simulation speed is reduced and the view angle is adjusted, which emphasizes the importance of proper selection and design of visualization objects and their behaviors. A color or transparency change of the whole module object may be a more effective signal in such case.

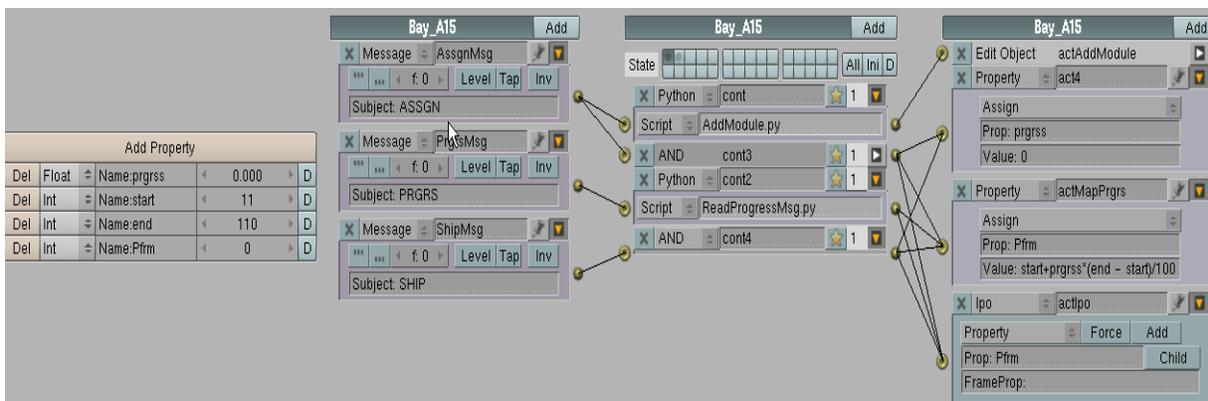


Figure 6: Bay Object logic bricks in the Yard BGE

### 3.5 Connecting to Simulation

The connection between simulation and visualization components in the industrial construction case is handled using HLA standards. A group of federates are mainly responsible for generating simulation behaviors while viewer federates handles visualization only. An HLA federate usually require an RTI am-

bassador components to be embedded in it. The language support of the HLA SDK and RTI in use can be restrictive. In this case the used SDK is .NET based, which is not support by the visualization tool (Blender). Therefore a bridge federate was developed to marshal messages at run time between the .NET-based RTI and the visualization interface in Blender. The bridge federate communicates with Blender through user data gram protocol (UDP) and the messages are received and distributed in Blender through a Python server script that transforms these messages to internal game engine messages. Figure 7 shows an outline of the message flow within this architecture.

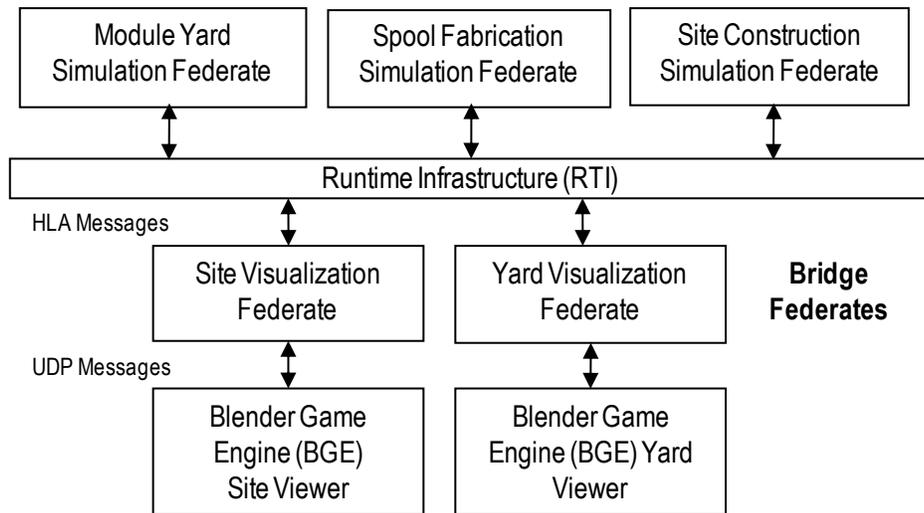


Figure 7: Message Flow between Simulation and Visualization Federates

#### 4 CONCLUSIONS AND FUTURE WORK

This paper presented an approach for visualization of distributed construction simulations. The approach relies on separation and loose coupling between visualization components and simulation components. A sample case is presented for visualization of industrial construction operations by utilizing a game engine with graphical constructs for customization of visual behaviors. The proposed approach facilitates rapid development and capitalizes on the strengths of dedicated graphics engines. Future work should focus on the design of the visual behaviors that objects generate in response to simulated behaviors, which is usually an overlooked step but can affect the effectiveness of the visualization and hence user’s perception and confidence in the simulation study. Focus should also be on utilizing the capabilities of graphics engines in processing geometry and space to handle spatial changes and constraints of construction sites and feed them back to simulation components.

#### ACKNOWLEDGMENTS

The authors would like to acknowledge the contribution of the Natural Science and Engineering Research Council of Canada (NSERC) and collaborating construction companies for their funding of this research through the Collaborative Research and Development Grant number CRDPJ 335345-05.

## REFERENCES

- AbouRizk, S., Mohamed, Y., Taghaddos, H., Saba, F., and Hague, S. 2010. "Developing complex distributed simulation for industrial plant construction using High Level Architecture" In *Proceedings of the 2010 Winter Simulation Conference*, edited by B. Johansson, S. Jain, J. Montoya-Torres, J. Huggan, and E. Yücesan, 3177-3188. Piscataway, New Jersey: Institute of Electrical and Electronic Engineers, Inc.
- Hajjar D. and Abourizk S. 1999. "Symphony: An Environment For Building Special Purpose Construction Simulation Tools". In *Proceedings of the 1999 Winter Simulation Conference*, edited by P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, 998-1006. Piscataway, New Jersey: Institute of Electrical and Electronic Engineers, Inc.
- Halpin D. and Woodhead R. 1976. *Design of construction and process operations*. John Wiley & Sons, New York, N.Y., USA.
- Dassault Systems. 3DVIA Virtools. Accessed June 2011. <http://a2.media.3ds.com/products/3dvia/3dvia-virttools/>
- Blender Foundation. 2010. Help & Education . Accessed June 2011. <http://www.blender.org/education-help/>.
- Huang R. and Halpin D. 1994. "Simulation Of Cable Stayed Bridges Using DISCO". In *Proceedings of the 1994 Winter Simulation conference*, edited by S. Manivannan, D. A. Sadowski, and A. F. Seila, 1130-1136. Piscataway, New Jersey: Institute of Electrical and Electronic Engineers, Inc.
- Ioannou P. and Martinez J. 1996. "Animation Of Complex Construction Simulation Models. In *Proceedings of the 1996 Winter Simulation conference*, edited by J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain, 1321-1328. Piscataway, New Jersey: Institute of Electrical and Electronic Engineers, Inc.
- Kalk A. and Douglas S. 1980. "Insight: Interactive Simulation Of Construction Operations Using Graphical Techniques". Technical Report No. 283, Construction Institution, Department of Civil Engineering, Stanford Univ., Stanford, California, USA.
- Kamat V. and Matrinez J. 2001. "Visualizing Simulated Construction Operations In 3D". *Journal of Computing in Civil Engineering* 15(4):329-327.
- Kamat V. and Matrinez J. 2008. "Software Mechanisms For Extensible And Scalable 3D Visualization Of Construction Operations". *Journal of Advances in Engineering Software* 39(8): 659-675.
- Oloufa A. 1993. "Modelling And Simulation Of Construction Operations". *Journal of Automation in Construction* 1(4):351-359.
- Oloufa A.A. and Crandall K.C. 1992. "Feedback Mechanism For Operational Simulation". *Journal of Computing in Civil Engineering* 6(2):161:177.
- Paulson B.C., Douglas S.A., Kalk A., Touran A. and Victor G. 1983. "Simulation And Analysis Of Construction Operations". *Journal of Technical Topics in Civil Engineering* 109(2):89-104.
- Rekapalli P. and Martinez J. 2009. "Runtime User Interaction In Concurrent Simulation- Animations Of Construction Operations". *Journal of Computing in Civil Engineering* 23(6):372-383.
- Rekapalli P. and Martinez J. 2011. "Runtime user interaction in concurrent simulation- animations of construction operations". *Journal of Construction Engineering and Management* 137(3):214-224.
- Rohrer M. and McGregor I. 2002. "Simulating Reality Using Auto-Mod". In *Proceedings of the 2002 Winter Simulation Conference*, edited by E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes,, 173-181 Piscataway, New Jersey: Institute of Electrical and Electronic Engineers, Inc.
- Shannon R.E. 1975. *System Simulation: The Art And Science*, Prentice-Hall, Englewood Cliffs, N.J., USA.
- The Institute of Electrical and Electronics Engineers, Inc. 2000. *IEEE 1516 Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) -Framework and Rules*.

- Wickens C., Lee J., Liu Y. and Becker S. 2004. *An introduction to human factors engineering, 2nd edition*, Prentice-Hall, Upper Saddle River, NJ, USA.
- Xu J. and Abourizk S. 1999. "Product-Based Model Representation For Integrating 3D CAD With Computer Simulation", In *Proceedings of the 1999 Winter Simulation Conference*, edited by P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, 971-977. Piscataway, New Jersey: Institute of Electrical and Electronic Engineers, Inc.

#### **AUTHOR BIOGRAPHIES**

**YASSER MOHAMED** is an Assistant Professor in the Department of Civil and Environmental Engineering at the University of Alberta. His research interest includes modeling and simulation of construction processes and the integrating of simulation models and knowledge engineering tools into construction management and decision making processes. His e-mail address is [yaly@ualberta.ca](mailto:yaly@ualberta.ca).

**AMR ELNIMR** is a PhD candidate at the Department of Civil and Environmental Engineering at the University of Alberta. His research focuses on simulation driven visualization of construction processes, and modeling dynamic spatial site changes. His e-mail is [elnimr@ualberta.ca](mailto:elnimr@ualberta.ca).