FRAMEWORK FOR HIGHLY INTEGRATED, INTEROPERABLE CONSTRUCTION SIMULATION ENVIRONMENTS

Simaan M. AbouRizk

Professor, Hole School of Construction Engineering, Department of Civil and Environmental Engineering,
University of Alberta, Canada
Correspond to abourizk@ualberta.ca

ABSTRACT: This paper describes the use of a highly interactive and inter-operative application for complex simulation environments, or Synthetic Environments (SE), as deployed for construction as Construction Synthetic Environments (CSE). Based on the High Level Architecture (HLA), this research focuses on implementing simulation technology in a software environment, COSYE, that will be the foundation for building CSE applications. This framework is discussed in the context of tunneling and industrial construction applications, including steel fabrication and pipe-spool manufacture. The framework is demonstrated using the NEST sanitary tunnel project in Edmonton, Canada, in which COSYE was used for scenario-based analysis and planning.

Keywords: Simulation; Computer Models; Tunnelling; Industrial Construction; Planning

1. INTRODUCTION

Through simulation research, we seek to achieve a comprehensive representation of an entire construction project with all of its components, including: a model of the facility (product model), the production/construction operations (process models), the business models, the resources involved, and the environment under which the project takes place. Such a framework allows simulation models to be extended throughout the life of the project with real-time input and feedback to manage the project until it is handed over to operations. The goal is to provide a virtual world where a construction project is planned, executed, and controlled with minimum disruption to the actual project. The framework will provide means to establish: detailed and comprehensive modeling of the entire life cycle of facilities; collaboration amongst a variety of stakeholders in building the required virtual models that represent the project; seamless integration between various forms of simulation (discrete, continuous, heuristic, etc.) and simulation software and tools; reusable simulation components for many applications (e.g. weather generation, equipment breakdown processes etc); and man-machine interactions with the models.

This research aims at developing highly interactive and inter-operative applications for use in complex simulation environments, or Synthetic Environments (SE). Synthetic Environments can be defined as “computer-based representations of the real world, within which any combination of computer models, simulations, people or instrumented real equipments may interact” [21]. When these representations are brought into construction specifically through a framework that facilitates their implementation in construction, we refer to them as Construction Synthetic Environments (CSE).

Using this framework, our goals are to develop a suite of modeling, simulation, and analysis tools for: (1) the planning and management of construction projects throughout their life phases from conception to operation, (2) practical CSE developments, including documenting, mapping, and modeling specific types of construction projects specifically industrial and tunnelling projects and an educational bidding game, and (3) the exploration of construction management best practices. This entails understanding and documenting production processes and decision making in the three application areas as well as prototyping the CSE environment for each. These tools will be developed based on distributed simulation standards to allow parallel development and execution of the applications, which, in turn, facilitates better interoperability and reuse, multi-view representation of information to different users, a higher capacity for data capture and manipulation, and more effective model processing.

2. THE COSYE FRAMEWORK FOR CSE DEVELOPMENT

The CSE framework proposed in this research provides an opportunity for providing modeling and simulation techniques that enable: “a comprehensive representation of the natural environment; the ability to explore and visualize interactions and hence improve understanding of the ‘real world’; a flexible and relatively low cost (compared to live trials or prototypes) way to explore issues and proposed solutions ahead of major investment and commitment; enhanced training opportunities, free from environmental constraints including interaction with other co-operating or opposing participants; and a through-life approach to applications” [21].

The state-of-the-art in CSE research spans a wide range of theoretical developments in terms of the infrastructures and technologies that allow the building of SE, in addition to the applications in various technologies and knowledge domains. On the development front, research into the software and hardware platforms that support SE has been taking place since the early 1990’s. Areas of particular research include the algorithms and standards for distributed and parallel simulation, hardware and software for visualization and rendering, integration of intelligent agents, management of large size data sets, and network collaboration (see for example, [6] [10][13][18][2]. On the application front, defence applications are the most advanced users of the technology and the ones that are pushing research in the field [7][16] [19]. However, research into applying the technology in manufacturing and product assembly is also underway and tackles different challenges related to integration, data exchange, product modeling, model size, and intelligent behaviours [14][15]. In the construction discipline FIATECH [5] seeks (in the Capital Projects Technology Roadmap) a highly automated integrated environment for construction project planning and control that is similar in requirements to SE.

Our research is mostly focused on furthering our knowledge and technology base in simulation systems and in implementing the findings in a software environment that will be the foundation for building CSE applications. We refer to this environment as COSYE (Construction Synthetic Environment). The COSYE framework is developed based on the High Level Architecture [11]. The HLA approach is suited for complex applications such as the ones we are dealing with in construction. The HLA architecture supports building complex virtual environments (called federations) using distributed simulation technologies. In addition, it provides standards for building the individual components (federates) of such environments by different developers while maintaining interoperability between them. The HLA standards facilitate the reuse of the developed components as part of the new federations.

These standards consist of three main components [11]: the HLA rules, the interface specifications (, and the Object Model Template (OMT). HLA rules must be obeyed if a federate or federation is to be regarded as HLA-compliant. The interface specification defines the functional interfaces between federates and the run-time infrastructure (RTI). The RTI is software that conforms to the HLA specifications and provides software services such as synchronization, communication, and data exchange between federates to support an HLA-compliant simulation. Federates do not all need to be simulation models; instead, an HLA-compliant federate is any software that interfaces with the RTI as part of its standard services.

To promote collaborative modeling, reusability, and interoperability, all objects and interactions managed by a federate and visible outside the federate should be specified in detail with a common format. The Object Model Template (OMT) provides standards for documenting HLA object modeling information and consists of three main parts: Federation Object Model (FOM), the Simulation (or Federate) Object Model (SOM), and the Management Object Model (MOM). The design and proper documentation of these objects is central to developing large scale or complex simulation models. The object model template and the HLA services make it possible to scale the simulation down to manageable levels where multiple simulationists are involved in deploying specific components of the simulation. This also offers a medium for standardization whereby an entire community of simulationists can generate and reuse previously developed simulation components (e.g. libraries of common construction processes, weather generation models, breakdown of equipment events etc.). Inherited within the HLA are also features of distributed simulation and parallel computing, seamless integrations of different simulation algorithms within a federation (i.e. continuous, discrete event), human in the loop and other features.

We implemented the COSYE framework as a software application running on the Microsoft.NET platform which facilitates the development of CSEs in MS Visual Studio. During the design time, the framework provides tools to define and build the federation object model (FOM) and compile it into .NET assemblies. It also provides the abstract generic base federate that can be customized by the developer to produce particular simulation behaviours.

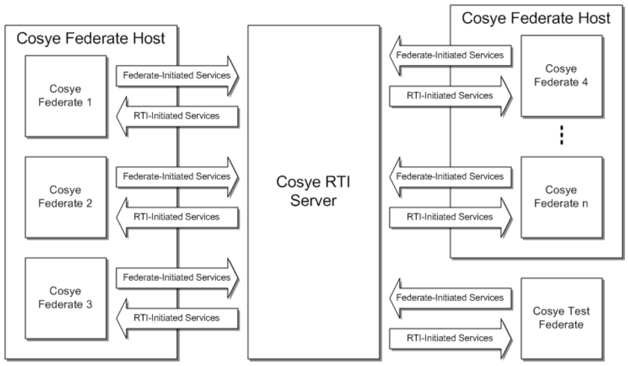


Figure 1. COSYE architecture with federates

During run time, the framework provides the necessary communication, information exchange, and data-sharing protocols through a run-time infrastructure (RTI) that assure simulation synchronization, coordination, and consistency between the different federates. The conceptual architecture of COSYE is shown in Figures 1 and 2. Figure 1 shows the structure of a federation, and Figure 2 shows the structure of a typical federate.

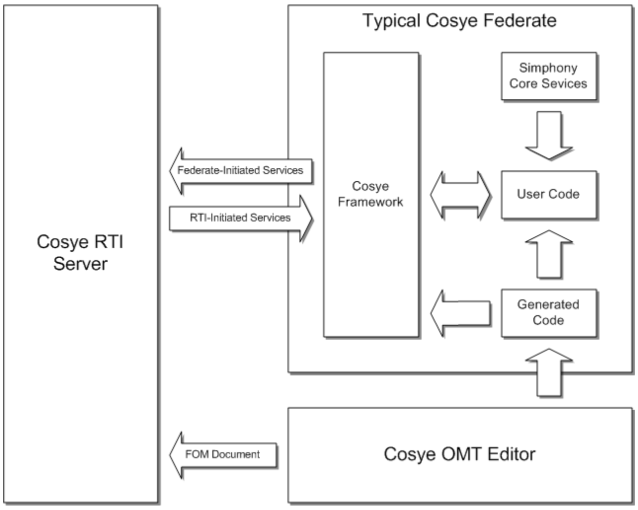


Figure 2. Conceptual framework and COSYE components

3. IMPLEMENTING CSE

We have defined specific projects that will enhance our chances of success with the first CSE implementations but maintain flexibility to change as research progresses to take advantage of other opportunities that can be defined by collaborating companies. The first set of projects deal with automated real-time input for enhanced decision support, 3D modeling, parallel processing and customized views for different users demonstrated through prototype CSE environments for 1) tunnelling, 2) industrial piping fabrication and construction and 3) a bidding game. The choice of tunnelling as the first CSE is primarily reflective of the fact that tunnelling projects lend themselves to CSE developments better than most other processes. We also have developed significant expertise in simulation modeling in this domain. The choice of the fabrication/construction CSE is reflective of the nature of the industrial collaborations that we have previously established.

Appendix 1 demonstrates a sample federate (Trucking), a discrete event simulation model written in Simphony.NET version 3.5. The visual interface is a Visual Studio form created with C#.

4. TUNNELLING CSE

Utility tunnel construction projects offer a reasonably good medium for exploring our new approach as they offers ample room for experimentation, the projects are well confined, the interactions and environments are fairly well defined, and the knowledge regarding construction is prevalent within our research team (e.g. [3][4]). To better understand the envisioned CSE, we will outline how it appears to/interacts with the user(s).

The basic components of the CSE (federates) are computer simulation models, real-time data acquisition components, visualization environment, real-time interacting resources (humans and machines), electronic databases, and the COSYE RTI that executes the simulations and regulates the operations of all interacting components. Those components will also be supported by decision support systems implanted as intelligent agents within the CSE. This representation is an integrated model of the real world tunnelling project on the computer. It is interacting with the decision makers in real-time and has the ability to forecast, propose decisions, evaluate scenarios and recommend actions. The components of this CSE can be summarized as follows.

a. Computer simulation models, which include:
   - Representations of the physical environment where construction takes place derived from 3D CAD (drawings and information) relevant for the simulation
   - Process interaction models of the construction processes involved
   - Analytical models of the random processes that affect tunnel construction (e.g. weather and equipment breakdown)

b. An implementation of a visualization environment provides a visual dynamic view of the progress and status of the project.

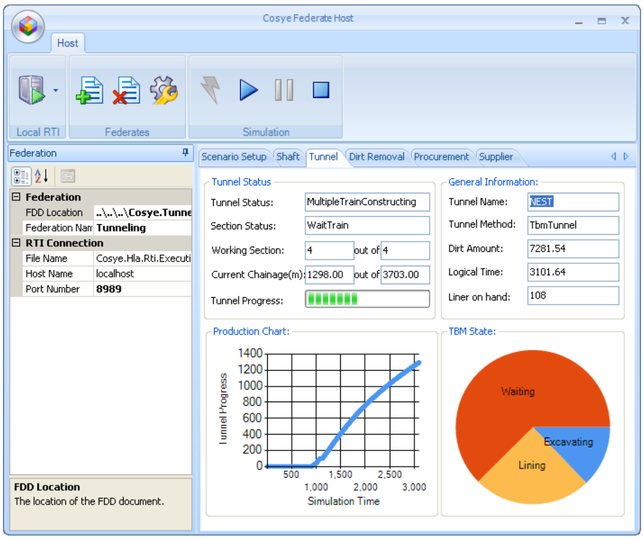
c. Data mining components providing intelligent gleaning of information from similar past projects and a comparison to the exiting project at hand.

d. Other decision support agents for monitor progress, comparing it to the planned one and updates the forecasts based on the new information available.

The components discussed above when integrated within the COSYE environment will provide different personnel involved in the tunnel construction process with a realistic, integrated, and flexible environment for evaluating different scenarios and management decisions throughout the lifetime of the tunnelling project from conception to completion. Figure 3 illustrates the model of CSE for tunnelling as deployed in COSYE.

The aim is to permit the CSE to evolve with time parallel to the evolution of the tunnelling process and into a more complex and realistic representation of the real project in order to allow more accurate decision making. Through this evolution, the CSE will have to be regularly updated based on the current status of the project and then extrapolated one or more steps beyond this status in order to examine future what-if scenarios. For this process to be practical, the components of the environment must be loosely coupled allowing them to engage/disengage from it without affecting any of the other components that need not be affected. More importantly, the flow of information regarding the real status of the project has to be expedited through automated data harvesting and data mining techniques.

Figure 3 shows six federates of the overall federation with the left window showing which federation these federates belong to, and other pertinent network information. A typical federation may contain federates residing on multiple computers or different hosts. The right side window is focused on the tunnel federate and only reflecting what the designer of this simulation wants the user to see in the output. Notice that the federate displays the tunnel progress during simulation and reflects various parameters from the simulation to engage the user. The various federates can receive input to the simulation (including during run time), and display output to reflect simulation status. The federates are supported with programs that encapsulate part of the simulation (see sample federate in Appendix 1).



Federation
FDD Location ..\..\..\Cosye.Tunne
Federation Nam **Tunneling**
RTI Connection
File Name Cosye.Hla.Rti.Executi
Host Name localhost
Port Number **8989**

FDD Location
The location of the FDD document.

Tunnel Status: MultipleTrainConstructing
Section Status: WaitTrain
Working Section: 4 out of 4
Current Chainage(m): 1298.00 out of 3703.00
Tunnel Progress:

Tunnel Name: NEST
Tunnel Method: TbmTunnel
Dirt Amount: 7281.54
Logical Time: 3101.64
Liner on hand: 108

Figure 3. Tunnel federation in COSYE

Please note that in this federation we had three developers each developing various components using the designed FOM. The scenario setup was developed by Hague [8] based on Al-Bataineh [1], the shaft and tunnel federates, the dirt removal, and procurement, by Xie [22], and the supplier federate by Labban [12]. The supplier federate demonstrates how two different contractors can be involved in the same simulation, for example. In this case, the procurement federate from the tunnelling contractor, observes how much of the liner segments are in inventory for the tunnel project. When a threshold is reached it places an order for segments to the Supplier. The supplier federate which may belong to a different party but can join this federation, receives the order and responds by scheduling a delivery of precast panels to this project. One should observe that this offers a medium to grow the simulation to encompass many components because of the ease by which we can compartmentalize the simulation exercise. All that is required is a common object model for the federation and few design issues agreed to in order to facilitate development of large scale simulations. In addition to the features described in here, we can take advantage of the integration between different simulation world views. For example, if we decide to use the federation as the basis from project control later on we can design and implement a to collect real time information from the tunnel site related to actual progress for example. The input information is continuous in nature as it mirrors the actual progress in the tunnel. While our other federates are discrete-event simulation, the HLA seamlessly allows us to integrate the two and synchronize the simulation in such a manner that the effort required to blend the real world with the simulated on is feasible and kept to a minimum level. The aim is to permit the CSE to evolve with time parallel to the evolution of the tunnelling process and into a more complex and realistic representation of the real project in order to allow more accurate decision making. Through this evolution, the CSE will have to be regularly updated based on the current status of the project and then extrapolated one or more steps beyond this status in order to examine future what-if scenarios. For this process to be practical, the components of the environment must be loosely coupled allowing them to engage/disengage from it without affecting any of the other components that need not be affected. More importantly, the flow of information regarding the real status of the project has to be expedited through automated data harvesting and data mining techniques.

A demonstration of using the tunnel federation for scenario-based planning is shown in Appendix 2 for the interested reader.

5. INDUSTRIAL CSE

In building a synthetic environment for structural steel projects or industrial piping, much of the framework found in the CSE for tunnelling can be retained. Although the processes of fabrication, erection and handling of structure steel/ pipe differ from tunnelling, the approaches for modeling will be founded on the same principles as depicted in the conceptual model of this CSE shown in Figures 2 and 3. A key difference in those applications based on our experience in the past, related to the fact that for those models to be effective, the product must be detailed and modeled explicitly [17]. While in the case of a tunnel, advanced excavation and precast liner installation is sufficient to track progress and thus model the operation, in this CSE each piece of steel as it is engineered, drafted and then fabricated, shipped and built must be tracked. In addition, the supply chain, which structural steel and piping projects are part of, is much more complex than tunnelling operations. Activities upstream of the chain like engineering drawings and material procurement significantly affect the fabrication and site installation processes [20]. The fabrication processes of structural steel and pipe spools also affect each other together with pipe module assembly. The complexity of the supply chain requires that the CSE account for the mutual effects between the different operations. This requires that some federates in the CSE interface with external data sources that models the actual/forecasted progress of other activities that affect structural steel or piping projects.

While the main objective of the development of this CSE is for production management, we also intend to experiment with utilizing the infrastructure provided by the CSE for training purposes. To achieve this, the same functionality provided by the different components of the CSE will be provided in parallel to a training CSE. For the training environment, real data acquisition components will be replaced by progress simulators and scenario generation simulators.

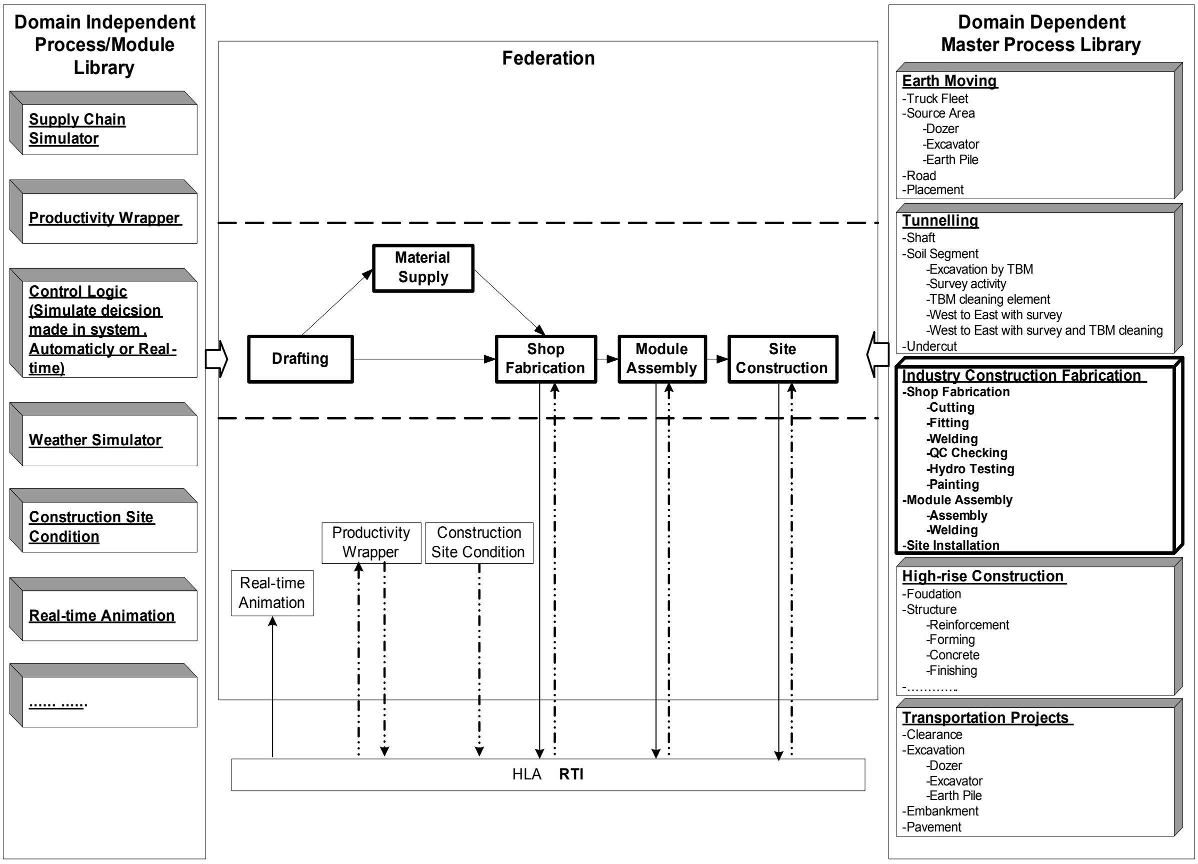


Figure 4. Industrial CSE

These simulators will provide the users with training scenarios and will utilize the same interfaces developed for the production management CSE. For example, the training CSE could provide simulated reports about the schedule and delivery dates of steel pieces in the shop for a particular project, man-hour availability of the different crews in the shop, and the material inventory. A trainee can then select between different courses of actions that influence the workload of the shop and delivery dates of the pieces. More than one trainee can interact at the same time through different interfaces to examine the effect of multiple decisions on the production system.

The basic components of this CSE are similar to those of the tunnelling CSE as demonstrated in Figure 4.

One of the key differences between the CSE for tunnelling and the CSE for fabrication/construction of steel or pipe-based facilitates is found in the number of inputs interacting with the real-time control federate. In the CSE for steel fabrication, for example, in addition to the fabrication shop itself, information is communicated via several integrated information management system (IMS) modules. These modules are sites, which collect data and subsequently disseminate that data to the appropriate federate. For example, the CSE for steel fabrication possesses a Field Control Module, a Fabrication and Shipping Control Module, a Project Control Module, and a Welding/Fitting/QA Inspection Module. The Field Control Module collects information regarding labour, equipment, and materials in the field. The Fabrication and Shipping Control Module gathers data on the pieces involved in steel fabrication and on the bills and invoice information related to the shipping of steel pieces. The Project Control Module keeps track of schedules, piece statuses, project cost data, project definition data, and electronic work order data. The Welding/Fitting/

QA Inspection module collects information on resource data, equipment data, and unit costs. These modules both communicate new data to the real-time control federate and are informed by the updates of complementary modules submitted to that federate. This interactivity ensures that the synthetic environment evolves in a manner parallel to the real steel fabrication process.

A major advantage for this CSE is the existence of information technologies (the modules we refer to) which we have implemented at our industrial collaborating companies over the past terms of the IRC. Our involvement in implementing these modules enables us to build portals and hooks to enable the envisioned CSE to extract the required information. In addition, the implementation of these modules provided our team with significance experience in exchanging information with CAD applications, which will allow us to develop practical solutions to the potential challenge of immature data exchange standards. This should enable seamless feeding of information into the required federates.

7. FURTHER RESEARCH

The HLA standard requires greater levels of resolution in preparing a model of a real system. The instructions required to drive a simulation exercise are generally more detailed than discrete-event instruction. This poses a challenge: while we require the HLA services to facilitate collaborative, modular, real-time input and integrated simulation developments, we cannot ask for more detail in describing models that are already large and complex. In other words, the HLA realities of detailed model instructions may offset the benefits of model decomposition and parallel computing, which were attractive in handling the large-scale, complex construction models.

Early feasibility studies direct us into developing modeling layers along the lines of the Simphony templates. Those modeling templates act as an intermediary between the model developer and the COSYE environment. A major issue to resolve in this context revolves around the nature, structure, and contents of those intermediate layers. We will first investigate generic modeling constructs that potentially possess smart components, which will facilitate model development without limiting the extendibility of the model. We will also develop a modeling template that mimics the same functionality of the current Simphony general purpose simulation template in order to facilitate easier transition of existing models. The challenge is to maintain the functionality of the HLA and extend it to the simulationist, while retaining the usability and ease of the existing simulation tools.
2005).

ACKNOWLEDGEMENTS

Special thanks to members of the research team at the Hole School of Construction Engineering at the University of Alberta: Stephen Hague, for implementing COSYE frameworks and providing sample code, Dr. Yasser Mohamed, for collaboration on HLA development, and PhD students Hua Xie, Ramzi Labban, and Farzaneh Saba, for contributing figures and data. Thanks also to Andrea Hasenbank for compiling and editing this work.

APPENDIX 1. SAMPLE CODE FOR DISCRETE EVENT MODEL (TRUCKING)

```
Imports Simphony.Distributions.Continuous
Imports Simphony.Resources
Imports Simphony.Simulation
Public Class Loading
        Private Const TotalLoaders As Integer = 2
        Private Const TotalTrucks As Integer = 12
        ‘Distribution to model truck loading time.
        Private ReadOnly LoadingTime As _
             New Triangular(5 * 60, 10 * 60, 7 * 60)
        ‘The simulation engine, loader resource, and
        ‘truck queue.
        Private MyEngine As New BasicDiscreteEventEngine
        Private MyResource As New Resource(TotalLoaders)
        Private MyFile As New WaitingFile()
        ‘Event delegates.
        Private TruckArrivedEvent As _
             New DiscreteEventHandler(Of Truck)_
        (AddressOf TruckArrived)
        Private LoaderCapturedEvent As _
             New DiscreteEventHandler(Of Truck, CapturedEventArgs)_
             (AddressOf _LoaderCaptured)
        Private TruckLoadedEvent As New _
```

```
    DiscreteEventHandler(Of Truck)(AddressOf TruckLoaded)
Public Sub New()
    ‘ This call is required by the Windows Form Designer.
    InitializeComponent()
    ‘ Add any initialization after the InitializeComponent() call.
    MyEngine.Resources.Add(MyResource)
    MyEngine.WaitingFiles.Add(MyFile)
    MyResource.WaitingFiles.Add(MyFile)
End Sub
Private Sub MyTruckFactory_RegisterInitialInstances( _
    ByVal sender As System.Object, _
    ByVal e As System.EventArgs) _
    Handles MyTruckFactory.RegisterInitialInstances
    ‘Register the appropriate number of trucks.
    While MyTruckFactory.Count < TotalTrucks
        MyTruckFactory.RegisterObjectInstance()
    End While
End Sub
Private Sub MyTruckFactory_InitializeInitialInstances( _
    ByVal sender As System.Object, _
    ByVal e As System.EventArgs) _
    Handles MyTruckFactory.InitializeInitialInstances
    For Each MyTruck As Truck In MyTruckFactory
    ‘Each truck begins in the loading state.
    MyTruck.TruckState = TruckStateType.Loading
    MyTruck.UpdateAttributeValues()
    MyTruck.UnconditionalAttributeOwnershipDivestiture(
        _”TruckState”)
    ‘Schedule a truck arrival event for each truck at time zero.
MyEngine.ScheduleEvent(MyTruck, TruckArrivedEvent,0)
    Next
End Sub
Private Sub MyTruckFactory_ReflectAttributeValues( _
    ByVal sender As System.Object, _
    ByVal e As Cosye.Hla.Rti.ReflectAttributeValuesEventArgs) _
    Handles MyTruckFactory.ReflectAttributeValues
    ‘Determine the truck in question.
    Dim MyTruck As Truck = MyTruckFactory(e.theObject)
    If MyTruck.TruckState = TruckStateType.Loading Then
    ‘If a truck has entered the loading state,
    ‘schedule a truck arrival event.
    MyEngine.ScheduleEvent(MyTruck, TruckArrivedEvent,
        _e.theTime - MyEngine.TimeNow)
    End If
End Sub
Private Sub TruckArrived( _
    ByVal MyTruck As Truck, _
    ByVal e As EventArgs)
    ‘Request a loader for the current truck.
    MyTruck.RequestResource( _
        MyResource, 1, LoaderCapturedEvent, MyFile)
End Sub
Private Sub LoaderCaptured( _
    ByVal MyTruck As Truck, _
    ByVal e As CapturedEventArgs)
    ‘Determine the amount of time it’s going to take to
    ‘load the truck.
    Dim Interval As Double = LoadingTime.Sample()
    ‘Tell the RTI when the truck will change to the hauling state.
    MyTruck.AttributeOwnershipAcquisition(“TruckState”)
    MyTruck.TruckState = TruckStateType.Hauling
    MyTruck.UpdateAttributeValues(MyEngine.TimeNow +
Interval)
MyTruck.UnconditionalAttributeOwnershipDivestiture_
(“TruckState”)
    ‘Schedule the truck loaded event.
    MyEngine.ScheduleEvent(MyTruck, TruckLoadedEvent,
Interval)
End Sub
Private Sub TruckLoaded( _
    ByVal MyTruck As Truck, _
    ByVal e As EventArgs)
    ‘Release the loader.
    MyTruck.ReleaseResource(MyResource, 1)
End Sub
Private Sub fedAmb_TimeAdvanceGrant( _
    ByVal sender As System.Object,  _
    ByVal e As Cosye.Hla.Rti.TimeAdvanceGrantEventArgs) _
    Handles fedAmb.TimeAdvanceGrant
    ‘Process any internal events that should occur at the
    ‘current time.
    MyEngine.Simulate(e.theTime)
    ‘Update the user interface.
    AverageQueueLengthTextBox.Text = _
        MyFile.FileLength.Mean.ToString()
    AverageWaitingTimeTextBox.Text = _
        MyFile.WaitingTime.Mean.ToString()
    LoaderUtilizationTextBox.Text = _
```

```
                MyResource.Utilization.Mean.ToString()
        SimulationTimeTextBox.Text = _
                e.theTime.ToString()
        'Advance time to the time of the next internal event.
        rtiAmb.NextMessageRequest(MyEngine.TimeNext)
End Sub
Private Sub fedAmb_FederationJoined( _
        ByVal sender As System.Object, _
        ByVal e As System.EventArgs) _
        Handles fedAmb.FederationJoined
        'Prepare the discrete event engine for simulation.
        MyEngine.InitializeScenario()
End Sub
End Class
```

APPENDIX 2. DEMONSTRATION OF TUNNELLING CSE

Our research team has had great success in using the Tunnel federate to undertake large-scale simulation and scenario-based planning for the North East Sanitary Trunk (NEST) tunnel project in Edmonton, Canada [1]. With a budget of $22 million CAD, the project involved constructing a new 2.3 m diameter tunnel with a total length of 3707 m from 76 Street to Manning Drive. Underneath the alignment of 153 Avenue, there should be an overflow weir structure between the existing NL1 pump station and NL2 (see Figure A1). The basic approach to construction is two-way tunnelling from a shaft at 59A Street, as two-way tunnelling has proven to be superior in most cases, especially when tunnel length exceeds a threshold of 1 km. However, due to several limitations on project area (necessary space for laydown, proximity to residential zones, site access), one-way tunneling from Manning Drive West was also proposed as an alternative. In order to test several different scenarios—incorporating the variables of two-way versus one-way tunneling, number of shifts, shift lengths, and productivity rate—the research team performed simulation analysis using the described framework and deployed in the Simphony platform [9]. The scenarios were evaluated based on overall project duration and projected budget variation.

The tested scenarios are presented as follows:

- Scenario 1: Two-way tunneling, single shifts of 8 hr or 10 hr (Figure A2)
- Scenario 2: One-way tunneling, two shifts of 8 hr or 10 hr (Figures A3-A5)
- Scenarios 2a, 2b, 2c, 2d give a range of planning options based on the placement of shafts and the location of an internal switch for the trains (an enlargement in the tunnel to facilitate using two trains and thus minimize waiting time of the TBM. Figure A3 shows the location of the switch at 50 Street as option 2a. Option 2b places the switch at 59 Street, for example, and so on. (Figures A3-A5)

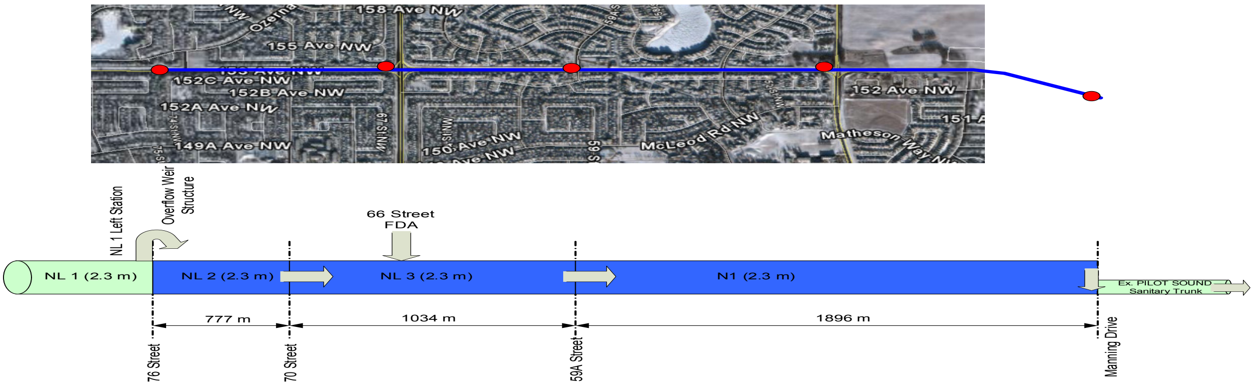


Figure A1. Planned construction of NEST project

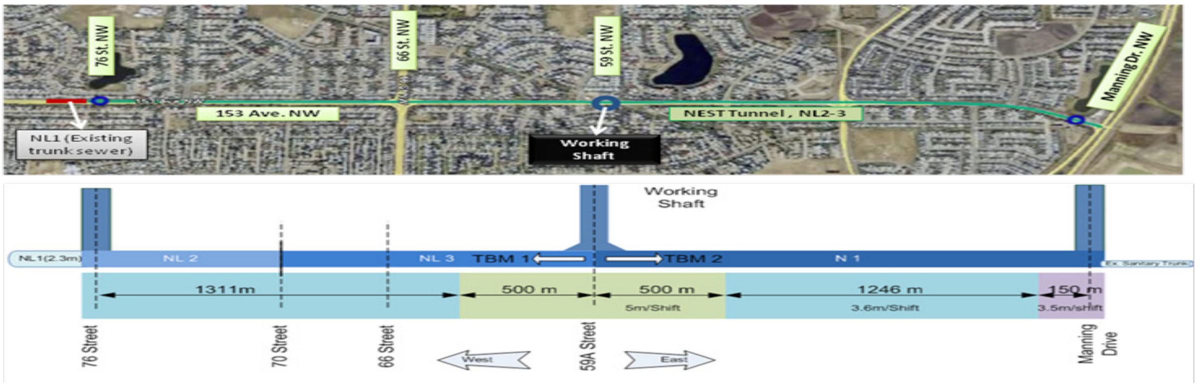


Figure A2. NEST project – Scenario 1

Figure A3. NEST project – Scenario 2a

The results of the simulations run for the above scenarios are given in Table 1. The overall best option was found to be Scenario 2c, one-way tunneling, two 8-hour shifts per day.

Table 1. NEST simulation results

Scenario	Alternative	Shift	No. of Shifts	Duration (days)	Productivity (m/day)	Switch
1	**Two way**	**8 hrs**	**1**	**540**	**6.86**	**NA**
	Two way	10 hrs	1	415	8.93	NA
2a	One Way	8 hrs	2	482	7.69	NA
	One Way	10 hrs	2	370	10	NA
2b	One Way	8 hrs	2	424	8.74	900
	One Way	10 hrs	2	328	11.3	900
2c	**One Way**	**8 hrs**	**2**	**401**	**9.24**	**1896**
	One Way	10 hrs	2	311	11.9	1896
2d	One Way	8 hrs	2	423	8.77	2632
	One Way	10 hrs	2	324	11.44	2632

REFERENCES

[1] Al-Bataineh, M. (2008) Scenario-Based Planning for Tunnel Construction. PhD thesis, University of Alberta.

[2] Bui, T., Sebastian, H., Dolk, D. R., and Gachet, A. (2005). “Virtual environments for advanced modeling: Conceptual foundations for research.” 38th Annual Hawaii International Conference on System Sciences, Jan 3-6 2005, Institute of Electrical and Electronics Engineers Computer Society, Piscataway, NJ 08855-1331, United States, Big Island, HI, United States, 91.

[3] Chung, T., Mohamed, Y. and AbouRizk, S. (2006) “Bayesian Updating Application into Simulation in the NEST Tunnel Project.” Journal of Construction Engineering and Management, ASCE, 132(8), 882-894.

[4] Fernando, S., Van Tol, A., AbouRizk, S. (2003) “Boring Deeper into Tunnelling Simulation.” Technical Note, Journal of Canadian Tunnelling, 181-187.

[5] FIATECH (2004). Capital Projects Technology Roadmap Element 1 Tactical Plan, Scenario-based Project Planning. Austin, TX. www.fiatech.org.

[6] Flanigan, W. F.,Jr. (1993). “Modeling and simulation standards products and processes.” Proceedings of 1993 Winter Simulation Conference - (WSC ‘93), 12-15 Dec. 1993, IEEE, Los Angeles, CA, USA, 968-70.

[7] Gaskell, R., Collier, J. B., Husman, L. E., and Chen, R. L. (2001). “Synthetic environments for simulated missions.” 2001 IEEE Aerospace Conference Proceedings, 10-17 March 2001, IEEE, Big Sky, MT, USA, 7-3556.

[8] Hague, S. (2009). “Tunnel federation for scenario planning.” Internal report. NSERC Industrial Research Chair in Construction Engineering and Management.

[9] Hajjar, D., and AbouRizk, S. (2002). “Unified Modeling Methodology for Construction Simulation” Journal of Construction Engineering and Management, ASCE, 128(2), 174-185.

[10] Hintze, A., Schumann, M., and Stuering, S. (1999). “Towards distributed synthetic environments for training in industry.” Proceedings of ICMCS99: IEEE Multimedia Systems ‘99: International Conference on Multimedia Computing and Systems, 7-11 June 1999, IEEE Comput. Soc, Florence, Italy, 72-6.

[11] IEEE (2000) Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Federate Interface Specification 1516.1 -2000.

[12] Labban, R. (2009) “Supplier federate for tunneling simulation in COSYE.” Internal report. NSERC Industrial Research Chair in Construction Engineering and Management.

[13] Ladner, R., Shaw, K., and Abdelguerfi, M. (1999). “3D synthetic environment representation using the “non-manifold 3D winged-edge” data structure.” Proceedings of INTEROPP’99: 2nd International Conference on Interoperating Geographic Information Systems, 10-12 March 1999, Springer-Verlag, Zurich, Switzerland, 305-14.

[14] Langley, P. A., and Morecroft, J. D. W. (2004). “Performance and learning in a simulation of oil industry dynamics.” Eur.J.Oper.Res., 155(3), 715-732.

[15] McGinnis, L. F., Xu, S., and Wang, K. (2005). “Evaluate simulation design alternatives for large scale manufacturing systems.” IEEE International Symposium on Semiconductor Manufacturing, Conference Proceedings, Sep 13-15 2005, Institute of Electrical and Electronics Engineers Inc., Piscataway, NJ 08855-1331, United States, San Jose, CA, United States, 344-347.

[16] Menendez, R. G., and Bernard, J. E. (2001). “Flight simulation in synthetic environments.” IEEE Aerosp. Electron.Syst.Mag., 16(9), 19-23.

[17] Song, L., Al-Battaineh, H., and AbouRizk, S. (2005). “Modeling uncertainty with an integrated simulation system.” Canadian Journal of Civil Engineering, CSCE, 32(3), 533-542.

[18] Sterling, G., and Dillon, T. (2002). “Cognitive assistants in synthetic environments - aspects of the concept and engineering development.” Knowledge-Based Intelligent Information Engineering Systems and Allied Technologies. KES 2002, 16-18 Sept. 2002, IOS Press, Polo di Crema, Italy, 594-9.

[19] Stytz, M.R., and Banks, S.B. (2003). “Progress and Prospect s for the Development of Computer-Generated Actors for Military Simulation”. Presence. 12(3). June 2003. pp. 311- 325.

[20] Tommelein, I. (1998) “Pull-Driven Scheduling for Pipe-Spool Installation: Simulation of Lean Construction Technique.” Journal of Construction Engineering and Management, ASCE, 124(4), 279-288.

[21] UK Ministry of Defence. (2005). “Synthetic Environment & Modelling & Simulation.” http://www.mod.

uk/issues/simulation/policy.htm. (July 6, 2005).
[22] Xie, H. (2009). “Component federates for tunnel simulation in COSYE.” Internal report. NSERC Industrial Research Chair in Construction Engineering and Management.