University of Alberta

A Simulation-Based Framework for as-built documentation of Construction Process and Product Information

by

Elmira Moghani

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Construction Engineering and Management

Department of Civil and Environmental Engineering

©Elmira Moghani

Spring 2013
Edmonton, Alberta

Permission is hereby granted to the University of Alberta Libraries to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only. Where the thesis is converted to, or otherwise made available in digital form, the University of Alberta will advise potential users of the thesis of these terms.

The author reserves all other publication and other rights in association with the copyright in the thesis and, except as herein before provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatsoever without the author's prior written permission.

To my lovely parents, for their endless encouragement and support

To my dear husband for his love, patience and care

And to my beloved brother and sister

ABSTRACT

In the construction industry, project documentation that covers all aspects of a project from planning through construction to the completion phase not only plays a significant role in managing an ongoing project, but is also considered a main resource in claim and dispute resolution, and in the planning of future projects. Therefore, project managers are always looking for efficient methods to document project information. Various software applications have been developed to assist project managers in data collection, storage, and retrieval including all aspects from drawings to cost estimates, schedules, resource information, change orders, and work logs. Studying existing commercial applications and research developments reveals that these systems, which mainly capture and integrate project information through database systems, CAD models, or scheduling software, focus on the static information of a project, thus losing the dynamic aspects of project progress including project work flow, resource interaction, and effects of external factors in an integrated format.

Therefore, the main objective of this research is the development of a generic framework for information representation that incorporates process model, product model, and external factors of a project into one system to create a complete project chronology, including all changes to the original plan. The system integrates the information and presents it to the user in a dynamic format, providing an overview of the project at every stage for comparison with the project as planned. This framework will help project managers to control the project during the execution phase while compiling a history of the project for future use.

The proposed framework utilizes computer simulation's functionality in modeling the dynamic nature of a project. This application employs a distributed simulation concept based on the High Level Architecture (HLA) rules in a Construction Synthetic Environment (COSYE) to facilitate integration of simulation components, to promote interoperability between the components, and to aid reusability of models in the future. This research uses a case study that focuses on repetitive construction such as tunnelling, which lends itself better to simulation planning. The proposed framework is created based on existing tunnel projects and is validated with an ongoing tunnelling case study.

ACKNOWLEDGEMENT

I would like to take this opportunity to express my sincere gratitude to many people who have helped and inspired me during my doctoral study at the University of Alberta.

First and foremost, I would like to express my appreciation and respect to my advisor Dr. Simaan M. AbouRizk for his continuous support, patience and encouragement. His immense knowledge, valuable advice and insightful guidance carried me throughout my research work and it was a great privilege to work under his supervision. I am also very thankful to Dr. Yasser Mohammed, whose knowledge and direction were crucial to my research.

Particular gratitude is extended to my doctoral committee members, Dr. Gunnar Lucko, Dr. Micheal Lipsett, Dr. Alireza Bayat, Dr. Aminah Robinson Fayek, and Dr. Mohamed Al-Hussein, for their constructive comments and valuable suggestions in my thesis defense.

I extend my sincere thanks to all my colleagues and the faculty and staff in the Construction Engineering and Management group for their advice and support. In particular, I would like to thank Brenda Penner, Amy Carter, Maria Al-Hussein and Holy Parkis for assisting me during this research project. Sincere gratitude goes to Stephen Hague, who provided valuable insight and support during my research development. I also wish to thank Han Zhang and James Wang for their great programming skills and quick response to my requests.

Last, but certainly not least, I would like to express my sincere gratitude to my beloved husband, Mohsen, for his love, patience, understanding and sacrifice which gave me the incentive to finish my research work. Thank you for your constant support and for always being there for me during this journey. I also would like to express my deepest appreciation to my parents, for their endless love, guidance and encouragement throughout my life. I could never thank them enough and I always appreciate everything they have done for me. My deepest love and thanks goes to my kind brother and lovely sister for their constant motivation.

TABLE OF CONTENTS

List of Tables

List of Figures

Chapter 1 - Introduction

1.1 Problem Statement

In the construction industry, information plays an essential role in managing projects; during the construction phase, documentation of on-site information can be effectively used for various management purposes such as project planning and control, forecasting, resource management, quality control, operation and maintenance, preparing a closure report, and lesson-learned studies for the ongoing project. In addition, construction records are used extensively in the resolution of construction disputes through arbitration. Since the records are the only resource that can be used to reconstruct the situation under which the disputes occurred (Kangari 1995), complete and accurate site documentation is extremely important in settling claims. Furthermore, project documentation is helpful in the planning of future projects, in the form of historical data stored within the involved organizations.

Therefore, one of the challenges in the construction industry is to capture, store, and manage project information and make it easily available to construction managers for management and decision-making purposes. Since the majority of documents are being produced during the construction phase, the main concern is usually to deploy the information management system to capture and store all the information generated on site including the progress report, equipment use, crew size, weather condition, working hours, quality control, change orders, site condition, interruptions, inspections, and material inventory. In addition to the site information, keeping track of design and construction changes adds yet another layer of documentation, which assists in project management and future possible claims.

It should therefore be unsurprising that project managers and contractors are always looking for efficient ways to collect and store construction histories and retrieve them effectively (Hendrickson and Au 1989). Many information management systems are being used in the industry to represent project information. These systems mainly utilize a database management system to facilitate information integration and data sharing in the project and between various project stakeholders.

In the construction management field, different studies have focused on developing integrated and computer-based information management systems to integrate the collection, processing, and storage of information. A number of these studies focused on integrating a database with a commercial application such as CAD systems, scheduling software or cost estimating systems. There are other studies that focus on automating data collection to provide access to the site data in a timely manner and provide a control system for early error detection on the construction site and update project progress and as-built schedule.

Literature review shows that current methods do not fully capture the construction process and do not reflect the story of a project in a detailed format and in a dynamic fashion; existing methods usually focus on capturing and storing physical information rather than capturing the actual building process. They are also limited in scope, focusing on smaller parts rather than the entire project. For example, project managers who are interested in equipment control try to capture information such as working hours, breakdown, or fuel consumption of onsite equipment, but estimators are interested in collecting the workers' daily log or material costs (Navon 2005; Fayek et al. 1998). Even integrating the project information database with a CPM (Critical Path Method) network does not support capturing the complete operation as it does not provide resource interactions and the dynamic nature of projects (Sawhney 1994). CPM network only describes tasks to a certain level of information; it mainly records the latest status of each activity and does not provide the daily progress of activities and possible issues such as work slowdown in a specific day, or interruptions and delays to the activity (Hegazy et al. 2005).

To solve these limitations, the solution is a well-structured system that can capture, store, process, and access all static and dynamic information, from the planning stage, through the building process, to the completed project, including all changes to the original plan. Ideally, the system should integrate this information and present it to the user, providing an overview of the project at every stage for comparison with the project as planned. It should be able to incorporate process models, product models, resource models, and static information into one system. A user should be able to see the resources that completed a given scope of work in a given period of time under the influence of external factors (e.g. weather). The solution must be easy-to-use from a manager's perspective, and it must also be reusable for different types and sizes of construction projects. One promising method for implementing the solution is computer simulation as it provides an environment to model a construction process while demonstrating the dynamic aspect of a project such as the resource interaction, work flow, and effect of external factors on a project progress in an integrated format.

Computer simulation has been successfully implemented in the construction industry for process modeling for purposes such as productivity measurement, risk analysis, resource allocation, site planning, and claim and dispute resolution (Sawhney 1994). Current simulation applications in the construction industry are usually based on Discrete Event Simulation method to model project activities, work sequence, resources and related factors. But to be able to utilize simulation technique for the documentation purposes especially for recording the complete story of a project including product and process model, a more advanced method needs to be developed in the construction simulation domain. This study proposes the use of distributed simulation technique to document a complete history of a project from planning to completion including site information, process and product model, and the dynamic nature of a project in an integrated environment.

1.2 Research Objectives

The main objective of this research is to provide an efficient approach for the industry to document project information that not only captures product as-built information, but also documents the actual construction process including all the changes, along with detailed information of resource use, external environment, and other relevant information considering static and dynamic aspects of construction projects. The strategy for achieving this goal is to develop a framework for information representation that covers both static and dynamic aspects of a project, from planning and construction phases.

This research employed the simulation modeling approach to facilitate the construction process modeling, as well as recording process changes to demonstrate what actually happened on the job site including all the changes to the original plan. In addition, utilization of computer simulation in this research enhances project documentation by adding a new level of information by recording work flow, resource interaction, and other dynamic features of construction projects. In particular, implementation of the framework in the distributed simulation environment enables the presentation of plan and as-built process models, along with product model information, in a single simulation environment that enables comparison of the results for various decision-making purposes.

Having complete plan and as-built documentation, the project manager will be able to control the project during the execution phase while compiling a complete history of the project for future use (e.g. for claims analysis, lessons learned, planning future projects, etc.).

1.3 Research Methodology

This research requires a thorough knowledge of construction projects as well as extensive programming skills. To develop a process model that includes all aspects of a construction project including activities, resources, and external factors in such detail that it provides valuable information as a decision-making tool, while still remaining easy to implement and use, an in-depth technical knowledge of a construction process and a good understanding of construction management techniques are required. Also, to develop the proposed framework in a distributed simulation environment, a broad range of computer programming skills are required. To gain the required knowledge and skills and ultimately develop the framework, a research plan consisting of specific steps was defined for this research. Research activities included a literature review, building a conceptual framework, choosing a case study, collecting, processing and storing site information for the selected case study, building components of the framework, implementing the documentation process, analyzing results, and testing and validating the framework with the ongoing construction project. The detailed explanation of each step is presented as follows:

1. Reviewing literature to recognize the issues in the current as-built documentation approaches.
2. Developing a conceptual model for the proposed approach to capture project as-built information through utilization of computer simulation.

3. Studying distributed simulation and High Level Architecture (HLA) and its implementation in Construction Synthetic Environment (COSYE) and evaluating various means of representing information within the COSYE environment.
4. Implementing the proposed framework, utilizing distributed simulation that allows integration of a process model developed in stand-alone Discrete Event Simulation software, product model, and different simulation systems into an integrated simulation environment.
5. Developing a Special Purpose Simulation (SPS) template for a case application of a tunnel construction that captures construction product and process information and details of project resources, calendar, shifts, material, unit costs, material delivery, and external factors such as ground condition.
6. Developing a data collection and storing system during construction phase that includes process information, resources, interruptions, etc. The application includes the data collection system that gathers information on a daily basis through a web-based form and stores it in a relational Microsoft Access database.
7. Implementing the framework for a case application of a tunnel construction project.
8. Documentation of the actual process model using a hybrid simulation technique which combines Discrete Event Simulation with a time-step simulation.
9. Experimenting with the framework using a case application, validating the framework through an actual ongoing project, and providing suggestions for future improvements.

1.4 Research Scope

The as-built documentation studies mainly focus on data collection from the construction site, including site daily reports and product information, as discussed in Chapter 2. Investigation of existing methods in the data collection area illustrated that these methods lack in documentation of dynamic aspects of projects, so a framework which can lead to an integrated application for project documentation that can record all aspects of a project is a unique and ideal solution for project history documentation, especially for claim analysis. Therefore, the scope of this research focuses primarily on as-built documentation of the construction process and product information that contains every detail of a construction project.

In this project, after investigation of various methods for as-built documentation of projects, the research focuses on application of simulation systems in documenting project information. This research will utilize simulation technique, not only for planning purposes, but also for recording the actual process information, along with product data and external factors during the construction stage of a project. For the implementation phase, a tunnel construction project was considered as a case study in this research, and the ultimate goal is to provide a generic tool for as-built documentation of different types of construction projects.

This research does not propose an information management system that works as a data warehouse for the entire project, but it utilizes simulation technique to record the actual construction product and process and provide additional information about project progress, resource utilization, and many other simulation related results in the presence of different external factors.

1.5 Implementation Environment for the Proposed Framework

The software applications that are employed to develop this framework are:

- COSYE
- Simphony 4.0
- Visual Studio 2010
- Microsoft Access 2010

In 2006, AbouRizk introduced an HLA-based distributed simulation environment called Construction Synthetic Environment (COSYE) to the construction simulation domain to simulate more complex projects and facilitate interactions of different parties and various simulation components in a single system. Based on HLA rules, each component of a system is called federate, and the simulation model is known as a federation. The COSYE environment is a Microsoft.NET software application that supports the development of distributed simulation models in Microsoft Visual Studio (AbouRizk and Hague 2009). COSYE supports:

1. Modeling of different aspects of the project including process model, product model, resource interactions (equipment, material, crews) and external factors' influences (weather).
2. Integration of different software applications to work together on multiple computers and from different locations (e.g. consultant, project management or contractor offices) as one simulation system to facilitate collaboration in the project.

Simphony 4.0 is the latest version of Simphony which was originally created in 1999 as a Microsoft Windows-based Special Purpose Simulation tool to develop a more flexible and easy to use application for the construction industry to model construction processes using Discrete Event Simulation technique. The provided features in the software, specifically the graphical, hierarchical, modular and integrated modeling environment, simplify the development of a simulation model. This version of Simphony was developed utilizing the Microsoft.NET 4.0 framework under the object-oriented paradigm to provide a flexible and extensible environment for modeling and integration purposes. Unlike the previous Simphony, the development of a template is not tied to the user interface, which allows the use of the provided services in other simulation environments.

Both Simphony template and COSYE federation are developed using Visual Studio 2010. And finally, Microsoft Access is utilized to capture and store project information including product and process data.

The proposed framework in this research is a simulation-based system that is created in COSYE. As it is shown in Figure 1-1, the framework consists of different components that communicate together through the COSYE environment's features.

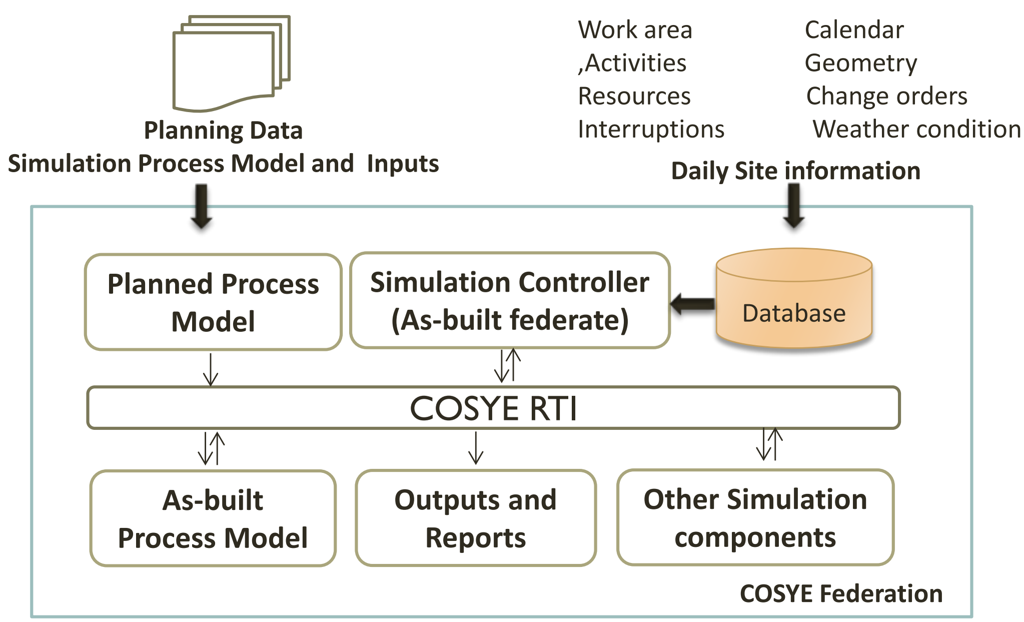


Figure 1-1: Envisioned Simulation-Based As-Built Dcumentation Framework

To develop process model federates for the both plan and as-built processes, Simphony software is integrated into COSYE federation to facilitate process model development. In the Simphony model, project activities and resources such as crew, equipment, and material are modeled. The interruptions in the project including equipment breakdowns, the external environment such as geotechnical information, and material delivery and supply are features considered in the development of the process model. In this framework, these inputs are all updated along with the process model during the construction phase.

A simulation controller is another module which connects to the database, reads data on a daily basis, sends updates to the simulation components, and controls the process updates in the Simphony model. The as-built data was documented in a Microsoft Access database which is a common application in construction projects documentation.

To update the product model, in the provided case study, surveying information and coordinates from 2D drawings are used to create a 3D model for the project. All the updates happen within the framework to keep everything in one single place and to make it accessible and easy to use for various purposes.

1.6 Thesis Organization

This thesis consists of seven chapters and five appendices. Chapter 1 introduces the research motivation, identifies research objectives and scope, and summarizes the methodology of the research. Chapter 2 contains a brief overview of literature and previous studies in the field of as-built documentation in construction projects, use of computer simulation in construction, Simphony applications, and introduction to distributed simulation in the COSYE environment. Chapter 3 discusses a case study for compiling data for the Post Implementation Review (PIR), and the challenges and limitation of existing methods that served as a motivation for this research. Chapter 4 presents the proposed simulation-based framework for documenting as-built records in construction projects. Chapter 5 provides the design and development of a Special Purpose Simulation template in Simphony 4.0, and integration of the process model in the COSYE environment for a case study project. Chapter 6 describes the implementation of the framework in a distributed simulation environment, and contains validation of the proposed framework with an ongoing tunnel construction project. Finally, Chapter 7 summarizes the research conclusion, the limitations of the research, research contributions, and suggestions for future development.

Chapter 2 - Literature Review

2.1 Introduction

In Chapter 1, the research objective, to develop a simulation-based framework to record construction product and process information and site data in an integrated and dynamic fashion was outlined. This framework utilizes a computer simulation technique to dynamically document as-built product and process, and all the related data during the construction phase of the project.

The first step to develop the application was to study previous research in the area of construction project documentation, especially those related to the construction process as-built documentation. Also, the applications of site information in the management process at construction projects were studied to show the value of the research in the industry. In the next step, the available simulation techniques and applications needed to be reviewed to choose the best approach for this research.

2.2 Construction As-Built Information

Construction as-built information is valuable and has multiple uses throughout the project lifecycle; the data collected during the construction phase is essential for controlling construction progress as well as monitoring material deliveries, product quality, project cost, and future forecasting of the project performance (Akinci et al. 2006). In addition, as-built information can assist in planning future projects, and in increasing knowledge and improving organization performance. Also, documentation of project as-built information presents what actually happened during the construction phase, which is essential in the case of claims and dispute resolution. In addition to the contract specifications and supporting documents such as project correspondence, meeting minutes, payroll, and cost reports, in case of construction claims, usually daily records can provide essential information of the construction site, especially when any changes happen in the construction process. It is entirely possible for as-built data to be used in claims and litigation by either a contractor or owner, or even by a third party in the project. Scott and Assadi (1999) conducted a survey to assess the importance of site records in construction. The analysis showed that the most important use of daily records is dealing with claims. Among all the reasons that create claims in projects, change orders are a major cause of disputes between owners and contractors (Kangari 1995).

There is typical as-built data in any construction project that needs to be collected and analyzed; this data can be categorized in three groups; the product data that contains data related to work locations and physical components of a project, which is usually documented in a form of as-built 2D/3D drawings; the process data that includes work sequence, activity progress, and delays with information of allocated resources, which are documented in the scheduling and estimating software; and the last category is site daily data that contains the daily state of activities, site condition, and effects of external factors, which are recorded in daily logs (Chiu and Russell 2011).

The following list includes common data related to product, process, and site condition that is collected frequently at construction projects (Russell 1993; Cox et al. 2002; Chin et al. 2005; Chassiakos and Sakellaropoulos 2008; Ailland et al. 2010):

- Project name, location and date,
- Weather conditions such as temperature, wind speed, precipitation, etc.,
- Crew information such as number of workers, working hours,
- Number and type of equipment on the project,

- Equipment usage and breakdowns,
- Site condition,
- Site measuring quantities,
- Material delivery,
- Activities information including daily progress,
- Construction methods (discrepancies in plans and/or conflicts),
- Scope changes, change work orders, and extra works,
- Delays and issues, and
- Dated photographs and videos.

Product, process and site daily records are usually stored by different parties and in separate files and formats. Several studies have been done in the area of project as-built documentation; many focused on data capturing techniques and others tried new methods for integration of project information. In the first area, researchers are working on automating data capturing processes at a jobsite to replace the traditional method of manual data collection which is time-consuming, inaccurate and expensive (Davidson and Skibniewski 1995). These studies mainly focused on using automation technologies to facilitate as-built data collection (either product or process data) during construction to provide accurate and complete information in real time for monitoring project performance and updated project progress. They studied various applications in construction projects such as mobile computing devices, bar-code systems, the Global Positioning System (GPS), photogrammetry, video capturing, laser scanning, and Radio Frequency Identification (RFID) tags (Hajian and Becerik 2009; Sardroud and Limbachiya 2010; Cheng and Teizer 2010). In the other research approach, researchers deal with information management in construction projects and how to document project information. They attempt to develop an integrated system that is able to store and integrate process and product information from different sources and with different formats during project lifecycle.

2.3 Documentation of As-Built Information in Construction Projects

There are several commercial software programs in the field of construction management that assist in project documentation including Computer-Aided Design (CAD) systems that document product information; Microsoft Project and Primavera Project Planner that are used for CPM scheduling and resource allocation; Expedition that tracks bid packages, submittals, changes, daily logs, and purchase orders; and Timberline that is used for cost estimation (Stumpf et al. 1995). These tools can capture and present as-built information during the construction phase of a project. But they do not provide integrated project information, and their interoperability is not easy due to the different data formats that they generate.

Therefore, researchers try to develop integrated systems to document as-built information in a more detailed and more accurate format. Hegazy et al. (2005) developed an intelligent bar chart system that links spreadsheets with scheduling software to assist project managers in cases of claims by providing detailed as-built processes compared to regular scheduling software. In this method, the daily percent complete for each activity is captured and compared to the plan progress using a spreadsheet. In the system, activities are presented in a group of adjacent cells (of daily progress values) that make up a total duration of the activity. This system can automatically recognize delays and ask for user inputs for documentation of the cause of delay, captured from the daily record. The bar chart is directly linked to the project schedule and automatically updates the state of activities. Dawood et al (2002) developed a system composed of a central database integrated with AutoCAD 2000 and Microsoft Project to document product and project schedule. Chin et al. (2008) proposes an integrated system for progress control of structural steel works that documents as-built and plan processes using 4D CAD. In this work, they integrate RFID data collection method and 4D CAD application to capture and document as-built progress during construction operation. However, none of the developed systems provide a complete history of a project.

In a more integrated approach, Goedert and Meadati (2008) tested the application of Building Information Modeling (BIM) in documenting construction as-built processes. They indicated that BIM software is not specifically prepared to accomplish these objectives and some modification to procedures as well as to software were necessary for the BIM to capture the construction process documentation.

In many studies, an integrated project database is proposed to store information from different sources in the project. This approach is known as an integrated project management system and facilitates effective communication between project parties and software applications during the lifecycle of a project. For modeling and integration approach, various data exchange standards have been developed for interoperability in the system.

Literature review reveals that integration may happen in three different categories, including integration between different computer software applications, integration through geometry and CAD packages, and knowledge-based integration that focuses on integrating multiple databases. These studies usually focus on one aspect of a project rather the entire project, so they cannot capture all the aspects of a project, and are usually interested in either schedule update, productivity study, or the as-built product through integration of CAD with other project management tools. The record is made on the contract documents, usually, but not necessarily limited to, the design drawings. Or the interest is sometimes in capturing resource information or recording site data including equipment, working hours, crew names, and any interruption in the site.

2.4 Simulation Application in Construction

In this research, a use of computer simulation is proposed to document construction information, especially the as-built process model, which includes actual work sequence in more detail than scheduling software, and records resource interaction, and all the factors that affect project progress. To implement the proposed framework, the application of computer simulation in the construction industry, and the capabilities and limitations of current systems to develop the framework were reviewed. In this section, an overview of construction simulation is presented in more detail to demonstrate its capabilities and limitations as part of this research.

In the construction industry, computer simulation has been utilized to create a construction process for different construction management and decision-making purposes. As an example, computer simulation enables construction managers to analyze complex construction processes, evaluate different scenarios, and optimize the time and required resources in a project.

Over the years, the main challenge in the use of computer simulation in the construction industry was to make it acceptable to the users by presenting it in a simple and more graphical context. Therefore, in 1973, Halpin introduced CYCLONE to simplify the simulation modeling for construction practitioners by allowing the graphical representation of a model. CYCLONE was developed to model processes based on discrete event simulation technique, which is an effective method in simulation of construction projects. A number of simulation systems have been developed based on

CYCLONE; the early applications such as RESQUE (Chang and Carr 1987) and Stroboscope (Martinez and Ioannou 1994) are known as General Purpose Simulation (GPS) tools which can represent almost any process. To create a model using a GPS tool, a user needs to have a deep understanding of simulation techniques, which makes it difficult for industry personnel to use. So, researchers tried to customize modeling tools for specific construction domains. They introduced Special Purpose Simulation (SPS) to facilitate modeling of specific type of projects. Simphony (Abourizk and Hajjar 1998) is one of the Special Purpose Simulation tools developed specifically for modeling construction processes. This research utilizes Simphony to model construction process in the proposed framework. The next section describes the specification of Simphony and the reason that Simphony is chosen in this study to demonstrate construction plan and as-built process models.

2.5 Construction Simulation Modeling in Simphony

Hajjar and AbouRizk (1999) introduced Simphony, a Microsoft Windows-based SPS tool to develop a more flexible and easy to use application for industry. Simphony facilitates modeling the construction process by providing a graphical, hierarchical, modular, and integrated modeling environment. In Simphony, a developer can use Discrete Event Simulation programming to create a template, a collection of elements linked together, and represent construction activities in one domain. Each element can encapsulate more than one activity in the process and its graphics represents the actual process in the site. This approach helps users to select elements based on their graphics, drag and drop them on the modeling surface, and link them together to create a complete model, without a deep knowledge of computer programming. However, the template development requires broad knowledge of the construction domain to be able to include all the process details while keeping the flexibility of the model for industry use. Several Simphony templates were created to model real projects and help construction managers in decision making and planning processes including productivity measurement, risk analysis, resource allocation, site planning, claim and dispute resolution, scenario planning, and cost estimation (Sawhney 1998). A number of successful construction simulation models were created in Simphony including earth moving, pavement construction, concrete placement on high-rise buildings, tunnelling, underground pipe-jacking, tower crane utilization, equipment management, site layout optimization, etc. Figure 2-1 shows a simulation model created in Simphony using a tunnel construction SPS template.

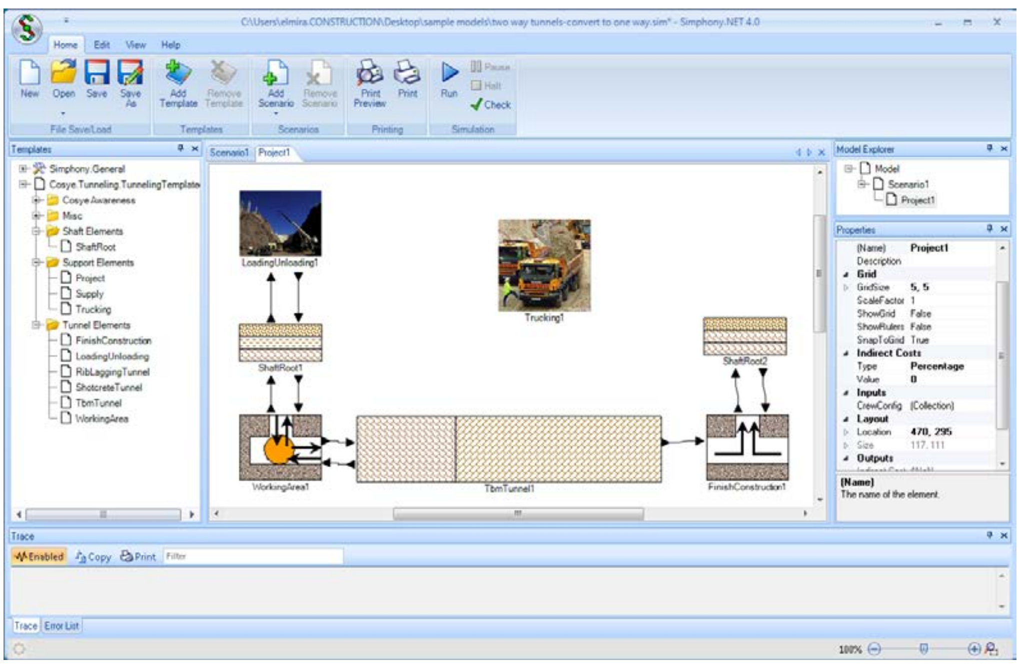


Figure 2-1: An SPS Template for Tunnel Construction Projects

A new version of Simphony called Simphony.NET 4.0 was developed to provide a more flexible and extensible environment for modeling and integration purposes. Simphony 4.0 was developed utilizing the Microsoft.NET 4.0 framework under the object-oriented paradigm. It consists of the following modules:

1. Services that facilitate Discrete Event Simulation programming such as Simphony Core Services (contains components such as the discrete event engine as well as math libraries, statistics, and distribution) and Simphony Modelling Services (contains abstract versions of models, scenarios, and various types of modeling for template development), and
2. Simphony templates and Graphical User Interface.

Each modeling element in Simphony.NET 4.0 is a Visual Basic or C# class developed in one Microsoft Visual Studio project (which represents a template in Simphony). Unlike the previous Simphony, the development of a template is not tied to the user interface, which allows the use of the provided services in other simulation environments. Figure 2-2 shows a sample model and the coding environment in Simphony.NET 4.0.

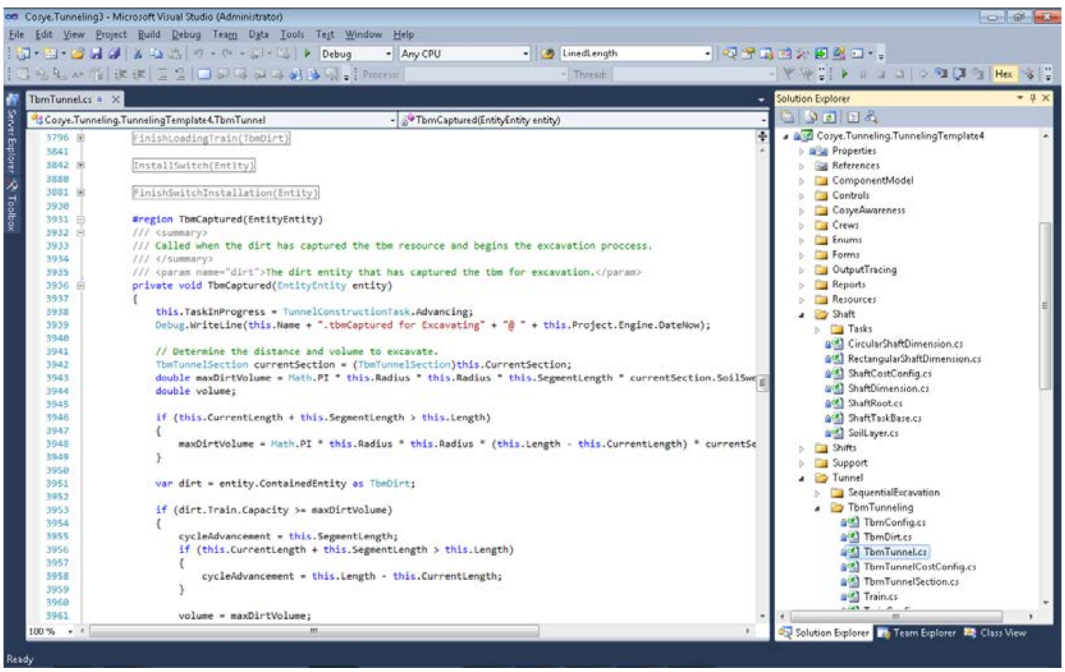


Figure 2-2: Simphony 4.0 Coding Environment in Visual Studio 2010

Due to the capabilities of Simphony 4.0, specifically its ability to integrate into other simulation environments, it has been selected in this research to model construction processes, but to develop a reusable and interoperable framework for information documentation in construction projects, there is a need for an integrated environment to facilitate product and process model documentation in one system.

Since the focus of current simulation applications is process modeling, more modern simulation tools are needed to cover project life cycle from the product model, the process model, the environment model, and all the related resources. They should support integration of visualization tools (CAD systems, animation), databases, different simulation techniques (continuous, discrete event, real time, agent-based, etc.), and software applications (such as schedule and estimation software and control systems) (AbouRizk, 2010). Therefore, a new simulation technique needs to be utilized to develop the framework. In 2006, AbouRizk introduced a distributed simulation technique to support integration of different computer applications, and various simulation techniques in a multi-user environment (AbouRizk 2006). Here, an overview of distributed simulation and its implementation for the construction domain is presented. Review of distributed simulation and its capabilities and current applications in the construction industry define that it is a suitable choice to fulfill the requirements of the proposed framework.

2.6 Distributed Simulation and High Level Architecture (HLA)

Distributed simulation was developed by the United States military and is known as a technology “that enables a simulation program to execute on a computing system containing multiple processors, such as personal computers, interconnected by a communication network” (Fujimoto 1999). In a distributed system, modellers divide a complex simulation system into multiple subsystems that are interrelated to represent the entire system functionality, and reduce the execution time of a system, and to decrease the time and cost of model development by reusing sub models in similar future projects. The other advantage of a distributed model is that it enables the interoperability of different simulation techniques in one system. It can also provide a collaborative simulation environment by enabling different parties to interact through a set of geographically distributed computers and reducing the time of communication and data transfer (Boer 2005). To fulfill all these requirements, a common framework and certain rules and standards were initiated to facilitate the development process. Based on the early experiences in distributed simulation, two well-known and widely accepted standards have been established: Distributed Interactive Simulation (DIS, IEEE standard 1978) and High Level Architecture (HLA, IEEE standard 1516). HLA, published in 2000, is a successor of DIS, and overcomes some limitations of DIS including data integrity and incapability of supporting event-driven and faster (than real-time) applications (Pederson et al. 2006).

High Level Architecture (HLA), the most advanced standard available, was developed in 1996 by the United States Department of Defense (DoD). HLA is a set of general rules created to manage the development of distributed simulation environments. The main intent of HLA is to promote interoperability between simulations and to aid the reuse of models in different contexts, ultimately reducing the time and cost required to create a new distributed simulation (Taghaddos et al. 2008). In HLA terms, each sub-model is called a federate and the entire simulation system is known as a federation. Figure 2-3 illustrates the functional view of HLA.

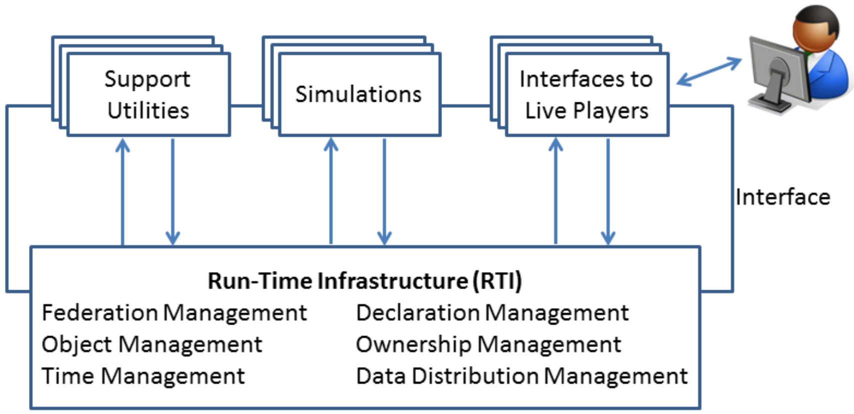


Figure 2-3: Structure of High Level Architecture (Wilcox et al. 2000)

HLA consists of three main components: HLA rules (IEEE 1516), the HLA interface specification (IEEE 1516.1), and the Object Model Template (OMT) (IEEE 1516.2). The interface specification defines the functional interface between federate and run-time infrastructure (RTI). The RTI provides software services such as synchronization, communication, and data exchange in the federation. 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 under a common format. The Object Model Template (OMT) provides standards for documenting HLA object modeling information and consists of three parts: Federation Object Model (FOM), Simulation Object Model (SOM), and Management Object Model (MOM). The FOM defines the data to be handled by the RTI in a federation execution; all the simulation entities and interactions represented in a federation are defined in FOM. The entities and interactions associated with managing the federation execution are defined in the MOM. The simulation functionality and all the data that the federate might expose in the federation are documented in federate SOM.

The HLA architecture provides a flexible simulation framework and enables connection of autonomous and heterogeneous simulators through RTI software (Fujimoto 1999). It also supports integration of discrete event, continuous, or agent-based simulation in a single federation (Kratkiewicz et al. 2004; Lees et al. 2007; Minson and Theodoropoulos 2008). In a federation, a federate can act as a collector / passive viewer that receives data from other federates, or may serve as a simulator which can receive human input and convey control inputs to the rest of the federation (Kuhl et al. 1999).

Although HLA and distributed simulation originated in the computer simulation domain for military application, they have been widely adopted in various industries (Wilcox et al. 2000).

2.7 Construction Synthetic Environment (COSYE)

The COSYE framework developed for the construction simulation domain (AbouRizk et al. 2009) is an implementation of a distributed simulation under HLA standards. The COSYE federation is capable of modeling the lifecycle of a construction project including process model, product model, and external factors, and integration of various software applications. In 2010, AbouRizk envisioned a fully-integrated and highly-automated construction simulation application in the COSYE environment that represents the entire construction project from the design phase, to the construction and completion phases of a project (Figure 2-4).

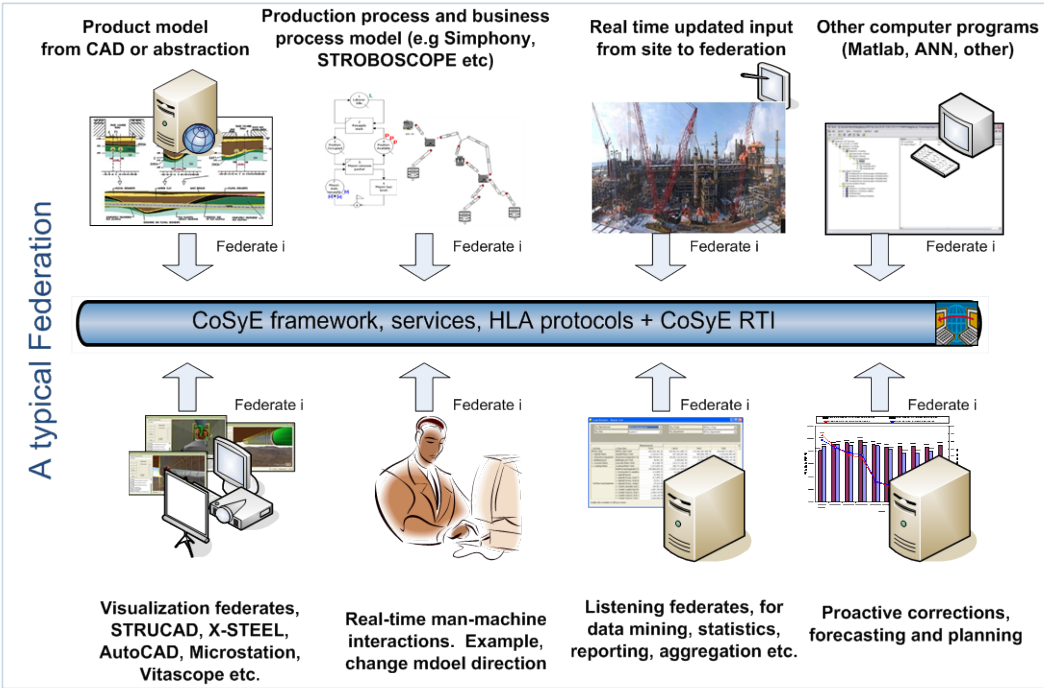


Figure 2-4: Comprehensive Model of a Construction Project in COSYE

COSYE federation consists of three major components (AbouRizk and Hague 2009)

1) The COSYE RTI server, a .NET implementation of IEEE standards 1516-2000. Currently, the COSYE RTI server can provide services such as federations, declaration, objects, ownership, and time management.

2) The Object Modeling Template (OMT) editor, which is used to define objects and their attributes. It defines the structure of the Federation Object Model (FOM) for the federation. In turn, the FOM for a single federation defines the communication language through which federates included in the federation speak with each other, including the names of items and occurrences. The OMT editor is integrated into Microsoft Visual Studio.NET.

3) The COSYE framework, an application programming framework that allows developers to create federates and handles details of communication with the RTI. It

integrates with the code generated by the OMT editor and uses many of the visual programming features supported by Visual Studio. In addition, the framework supports both discrete-event and time-stepped federates.

A COSYE federation is a Visual Studio solution that contains various C# classes that represent different federates. In the COSYE federation, federates are running in an application called Federate Host. It can host one or more federates at once and facilitate starting a federation execution.

A number of studies have been done in construction management using simulation technique in the COSYE environment. The first implementation of distributed simulation in the COSYE environment at University of Alberta was the development of a bidding game as a construction management training tool to teach students the bidding process in a real construction environment (AbouRizk et al. 2009, AbouRizk et al. 2010). Alvanchi et al. (2010) developed an HLA-based hybrid simulation framework that integrates System Dynamic into Discrete Event Simulation to consider the impact of dynamic behavior of a construction system on the productivity. They argued that due to their complexity, the development of Hybrid models involves different experts and requires extensive time and effort. Also, simulation execution in Hybrid models is usually lengthy and requires a huge amount of calculation. So distributed simulation and HLA can be an efficient solution to reduce the model development duration, as well as simulation execution time, by dividing it into smaller components and running them on different computers; this method also allows the hybrid–system-related tools and applications to join the federation as individual federates. Azimi et al. (2010) utilized HLA architecture to create an automated real-time monitoring and control framework for a construction project. The proposed framework assists project managers in detecting all deficiencies and deviations from the original plan, and allows them to take appropriate actions immediately. Various components for monitoring and control of projects such as data acquisition system for capturing actual site data, performance evaluation, forecasting, and visual reports are created in the form of separate federates in the COSYE environment. Taghaddos et al. (2008) developed a distributed simulation model based on HLA Architecture to manage resource allocation and leveling in the construction industry. In his framework, multiple agent-based federates are responsible for managing resource allocation in the project.

Chapter 3 - A Case Study of As-Built Documentation for a Post Implementation Review in a Tunnelling Project

3.1 Introduction

During the construction phase of a project, multiple activities are taking place. To effectively control the project, the management group tries to monitor the construction site and track all the activities happening in the site by collecting different site information; data such as site condition, activity progress, working hours, number of crew on site, material use, and number of equipment and their working hours and utilization can be monitored and recorded at the construction site. Documentation of construction as-built information on a daily basis is a challenging task which takes a lot of time and effort, but having access to the site information not only helps to effectively control the construction process, but also provides a clear story of what is happening in the site. Accurate as-built documentation assists project managers in various decision-making purposes such as tracking progress and delays, identify risks, forecasting and controlling future performance, claims analysis, lesson-learned studies, and planning of future projects.

A common practice in many projects is for the involved parties to be responsible to document the information being generated in their domain and to share it with the project management group so they can update the overall cost and schedule of a project. Since the data is usually collected by many parties and is being kept in multiple places, producing a complete history of a project is difficult, especially when there is a need to access to the project information at the completion phase when each party is involved in a new project.

One field that requires construction information is the Post Implementation Review (PIR) of a project, which aims to review the entire project to provide lessons learned and assure a satisfactory delivery of future projects through a comparison of planned and as-built information. The focus of this chapter is to implement a post implementation review for the 137 Avenue Grade Separation Tunnel Project constructed in Edmonton, Alberta. In this study, the process of compiling data and reconstructing project history after the completion phase, and the challenges that we faced during collecting and recording data from multiple places and different parties are discussed. Not capturing the actual construction process and missing important factors because of poor documentation was a motivation to propose a simulation-based method to store and re-use as-built information in a structured format to facilitate the accessibility of information in the project. The study shows that the discrete event simulation model created in the planning stage of a project can be effectively used to record and replay the detailed history of a project. The proposed system is applied on the case study and the plan and as-built simulation models were created as part of PIR report to demonstrate the scope and process changes in the construction site.

3.2 Project Post Implementation Review

In the closure phase of a project, two reports usually need to be generated; closure report and PIR report. The closure report is prepared by the project manager as an initial step in the project closure phase. This report mainly includes the list of activities required to complete the project such as terminating all the material supply agreements, planning for the release of project resources and equipment, informing stakeholders of the project completion and requesting project closure approval, and finally submitting it to the sponsors. In the next phase, a project is completely reviewed and lessons learned are documented in the PIR report. In this report, the actual outcome of a project is compared to the original plan and then the achievements and failures in the project are evaluated. The evaluation of the project performance requires studying project scope, schedule, cost, resources, risks, changes, quality, issues and communications both during the planning and construction phase to specify achievements, failures, deviations and any effects they had on the project, and then providing lessons learned and recommendations for future projects to improve project planning and execution. As it is shown in Figure 3-1, in general, the PIR is a simple and distinct process (Method 123 Ltd. 2006).

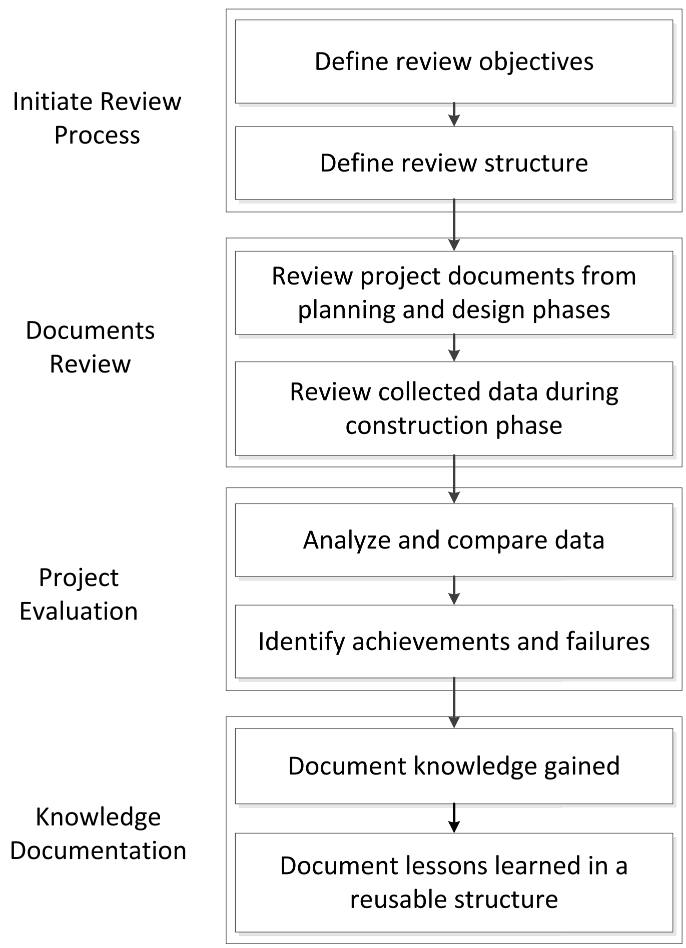


Figure 3-1: Post Implementation Review Process

Since the PIR deals with the original plan and actual outcomes, this phase would benefit greatly from the availability of the documents in an organized and structured format. In the planning phase, different documents need to be produced by the project management team so it is easier to access during the project and after the completion phase. The challenging part of PIR is to access all the information generated and all the people involved during the construction phase of a project. To have a successful review, a documentation management system should be available to provide easy access to project data. In addition, a close collaboration of the reviewers and project personnel would provide full access to the project history and guarantee the professional assessment of a project.

To study the performance of a project, different documents need to be reviewed including schedule performance which requires the comparison between the project plan and actual schedule and the productivity analysis for major activities during construction. Also, a detailed analysis of cost items should be conducted specifically for different equipment, materials, and workers. The resource plan needs to be compared to the actual resource allocation in the project. The same comparison should be done for other factors such as risks, changes, and communication plan. If the data management system is not implemented in the project, or it gathers specific information, other information can be captured through interviews, site visits, workshops, and collection and analysis of documents from different parties. Finally, after providing feedback, lessons learned and recommendation, the organization can utilize the PIR report for planning of future projects.

3.3 Post Implementation Review for the 137 Ave. Grade Separation Tunnel Project

In construction projects, lessons learned documentation for a project can be a valuable source of information for future projects to help contractor and owner organizations to increase the knowledge of their personnel, improve the decision-making process, enhance the quality of the work, and offer better performance in the future. Therefore, during the completion phase of a project, lessons learned study is performed and documented in the project PIR report to evaluate the overall performance and review the success and failure of the project and record knowledge gained during construction phase of the project. Since in the Post Implementation Review the actual outcome of a project is being compared to the plan, all the related information from schedule, cost, quality, and resource utilization to site data need to be analyzed. In the case study discussed in this section, the reviewing process and all the challenges in the data accessibility and collection are discussed.

3.3.1 Project Overview

The case study is a tunnel construction project for the City of Edmonton. The 137 Avenue tunnel and related works (Figure 3-2) are one of the major components of the grade separation at CN rail & 137 Avenue project. The project components include 1070 meters (m) of tunnel construction, a working pit, two removal shafts, 58 m of hand tunnel to the existing manhole, one additional hand tunnel close to the east removal shaft, and a number of access shafts and manholes. The Tunnel Boring Machine (TBM) was utilized to construct the tunnel. The working pit was constructed in the middle of the tunnel and divided the tunnel into east and west sections. It was used to provide access to the tunnels and to lower down the TBM, material and equipment, and to remove the excavated dirt from two tunnels. The tunnel construction process starts with excavation and lining of the west tunnel with the TBM machine. After removing the TBM from the west removal shaft, it was transferred to the working pit to start the construction of the east tunnel.

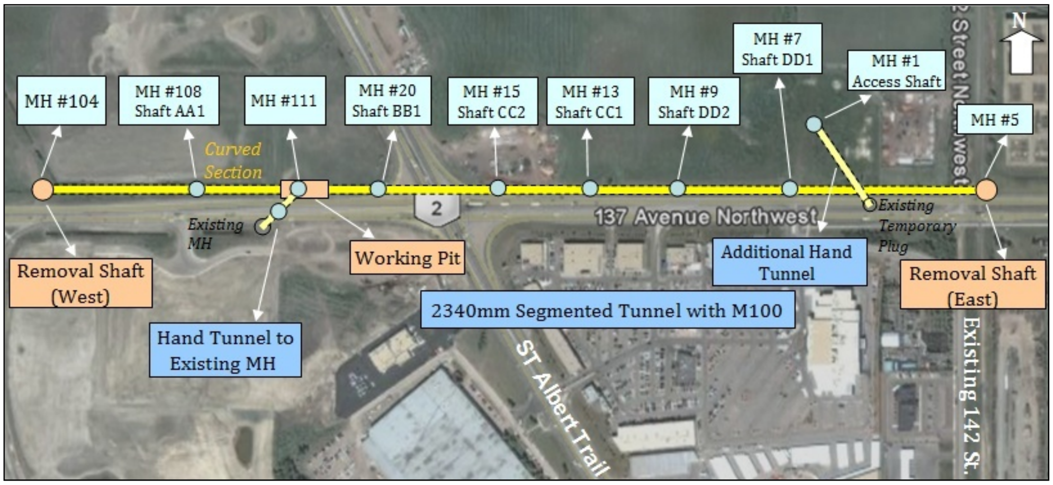


Figure 3-2: 137th Ave. Grade Separation Storm Tunnel Project (S.M.A. Consulting Ltd., 2008)

In 2009, a trenchless construction component was added to the project which consists of seven connections between different manholes (MH). Figure 3-3 illustrates a trenchless component of the project. A new construction method, which has never been implemented in City of Edmonton projects before, was used to construct the connection tunnels.

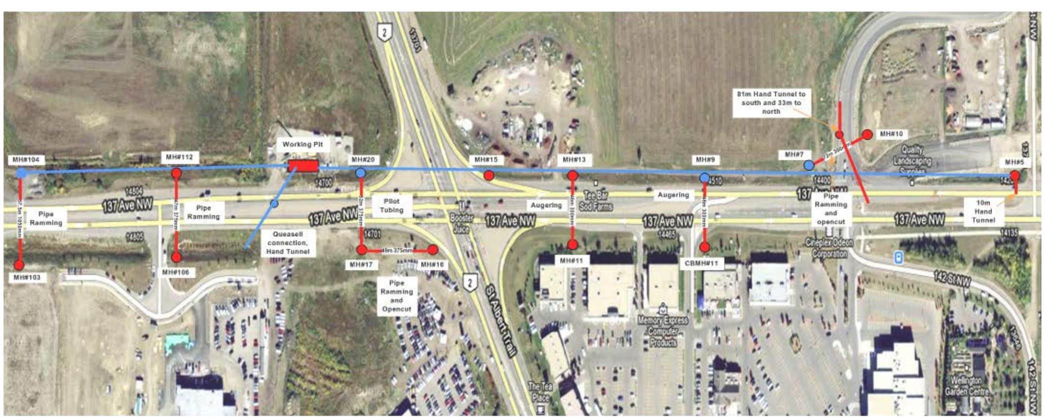


Figure 3-3: Connection Tunnels in the 137 Ave. Tunnel Project (S.M.A. Consulting Ltd., 2008)

The original estimate for the duration of the project was 762 days, with the start date of Aug. 1st, 2007, and the completion date of Sept. 13th, 2010. The project schedule was revised after a request in Sept. 2008, which shortened the total duration of the project to 685 days.

In the planning phase of the project, three different crews were assigned to the project; the tunnel crew was responsible for the working pit, west and east tunnel, and manhole construction. The drilling crew was working on access shafts, and the shaft crew was assigned to the removal shafts, undercuts, and west and east hand tunneling work (Figure 3-4).

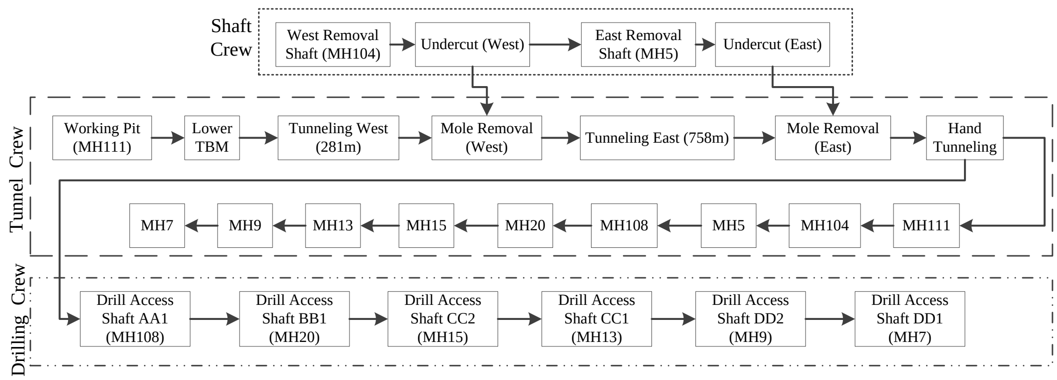


Figure 3-4: 137 Ave. Tunnel Project Crew Allocation (S.M.A. Consulting Ltd., 2008)

For both tunnels, the productivity was assumed to be 3.5 m/shift for the first 100 m and 5m/shift for the remaining parts, considering two shifts per day for the west tunnel and one shift per day for the east tunnel.

3.3.2 As-Built Data Collection and Post Project Review

The 137 Avenue tunnel construction project included many activities and new work orders, and also utilized new construction technology that made the project challenging; therefore, the City of Edmonton was interested in preparing a PIR report to analyze the achievements and failures, and document lessons learned for future projects. The Post Implementation Review for the project was conducted in August 2010 at the completion phase of the project. The focus of the review was mainly information collection from various sources to control the accuracy of the actual schedule, cost, and resource allocation of the project, and documenting the resource usage and detailed cost analysis of crew, equipment, and material, and comparing the final results with the original plan.

In this case study, the project schedule, cost, and resource performance were reviewed against the project original plan. In addition, the project plan for risk management, change management, quality management, communication, and record management were analyzed. For the risk management plan review, the process of risk identification, analysis, and mitigation were studied to find out whether all the risk factors were discovered in the early stages of the project, and whether they were managed through a structured procedure. For the change management review, the impacts of changes on schedule, cost, and resource requirements were analyzed. The change management process can help to reduce any conflicts and probable disputes, and help to manage all the changes properly. In the PIR report, all the changes in the project were listed, and the reasons for them were documented. For the quality management review, the quality control system for control of the actual construction product should be reviewed. Since there was no access to documented records for quality tests, only surveying information was controlled and documented for further use. Finally, the communication and records management system for the organization was investigated to find out who was responsible to prepare, update, and store documents. Based on the findings in this project, the lack of communication and information management system in the owner and contractor organizations produces a number of issues for accessing data in different phases of a project and for various uses within the company.

The main challenge in this study was the process of data collection and as-built documentation for the project schedule, cost, resource allocation, and product model at the project completion phase. The first step in the PIR process is to gather all the available information that exists in the project. All the information required for the PIR process can be obtained from multiple sources, including:

1. CPM schedule and cost estimate and their regular updates during the project, which are usually created and stored in the scheduling and cost estimating software. In this case study, project schedule was created in Microsoft Project and various project costs were documented in the in-house application.
2. Resource allocation. It is common that at the planning phase, the required resources are defined for different activities or work areas in the project. For the as-built, it can usually be captured from daily logs in the job site. A number of resources assigned to each activity were captured from the initial resource allocation assessment report.
3. Drawings. In the construction project, one major task is to prepare detailed drawings based on the design specifications. Usually 2D CAD systems are utilized to create project drawings. These drawings are usually stored in a soft and hard copy format. Figure 3-5 illustrates a sample drawing for the west removal shaft which is captured from a PDF file format sent by the client for the PIR study.

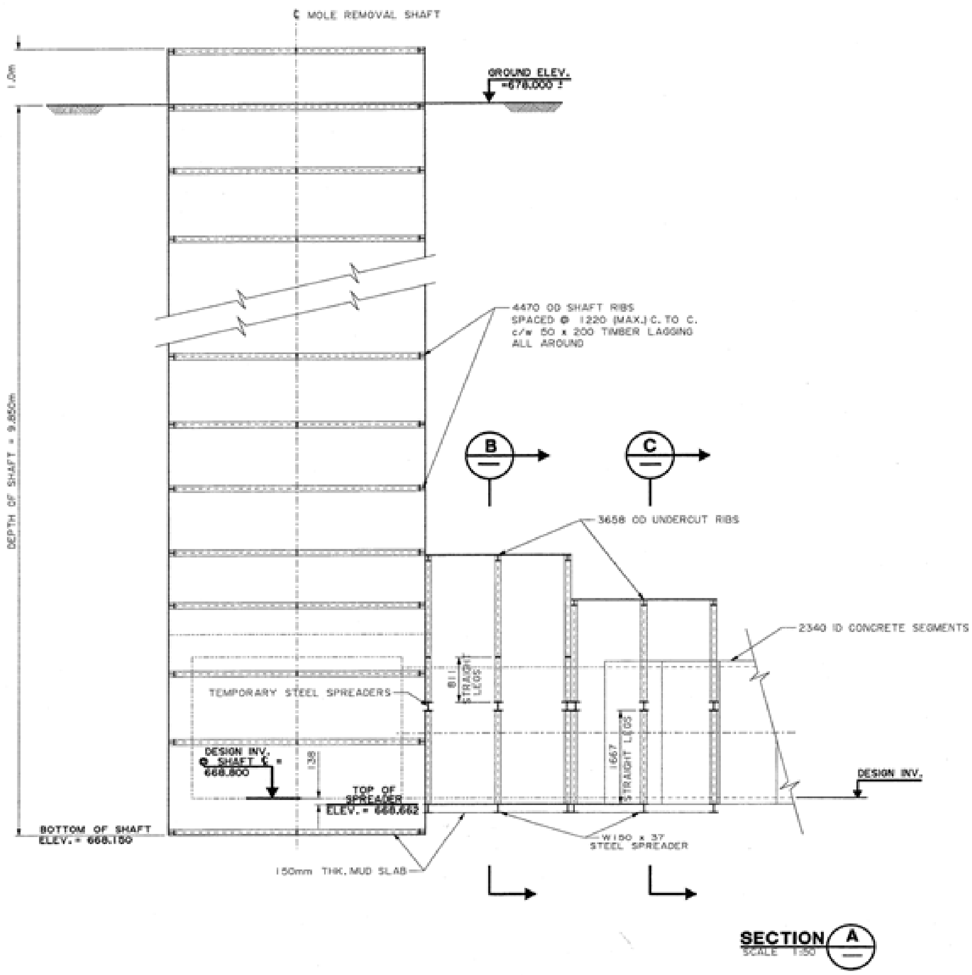


Figure 3-5: 137 Ave. Tunnel Project West Removal Shaft Section (City Of Edmonton, 2008)

4. Project correspondence that is used to confirm discussions in writing (sometimes includes emails).
5. Meeting minutes/notes. As it is shown in Figure 3-6, meeting minutes usually include a list of participants, subjects covered, all the decisions made, future works, and a proposed agenda for the next meeting.

DESIGN AND CONSTRUCTION
137 AVENUE GRADE SEPARATION PROJECT MEETING #6
MINUTES OF MEETING

DATE: July 16, 2009 TIME: 3:00 pm to 3:30 pm

PRESENT:

REGRET:

COPY TO:

ITEM	SUBJECT
1	Meeting Plan/Introduction:
2	Review of Last Meeting Minutes:
3	Safety:
4	Environmental:
5	Permits & Approvals:
6	Tunneling:
7	Hand Tunneling:
8	142 St Hand Tunneling:
9	Schedule:
10	Budget:
11	Design
12	Others:
13	Next Meeting:

These minutes were taken by:

Figure 3-6: 137 Ave. Tunnel Project Meeting Minutes Form (City Of Edmonton, 2008)

6. Construction field records (site daily diaries), which are prepared by superintendents, foremen, or site managers and include site condition, labor and equipment usage, major works, activity progress, any delays and problems, material delivery, and weather summary. Field records are usually captured and stored on a daily basis and they are kept on site until the end of the project.
7. Monthly management reports (includes latest cost estimate, project updated schedule and progress, earned value analysis, project risk management, and project control analysis).
8. Timesheets that document the number of workers and working hours, and the amount of work performed for different work areas in the project.
9. Site images that are captured on a weekly or monthly basis to show the progress and problems at work.
10. Submittal logs including change orders, drawing revisions, transmittals, etc.
11. Interviews of experts such as the project manager and project site manager to document construction method, problems, and challenges for different activities within the project.

For the PIR report preparation, all the management reports and all supporting documents such as meetings minute, correspondence, cost and schedule updates, and resource information were collected from different individuals and in several file formats including Microsoft Office Project, Excel, PDF, and Word files. Site images, site daily diaries and timesheets were obtained from the project manager, site coordinators and superintendents. After the data collection, the next step is the analysis of data to create as-built information. Then the final results are compared to the original plan to evaluate project performance and document it in a PIR report and provide recommendations for future organization use.

3.3.3 Challenges in the Data Collection and As-Built Documentation

In this section, the main challenges in the data accumulation phase are discussed. Table 3-1 lists all these challenges based on the subject of the study.

Table 3-1: Challenges in the Data Documentation and Analysis for 137 Ave. Tunnel Project

Subject	Challenges
Project Schedule	• There was more than one project scheduler in charge of updating the project schedule. As a result, the updated versions were not stored in one location, and in some points they did not match. • There were separate Microsoft Office Project files for different work areas in the project which made the interpretation of the activity sequence quite difficult. • The main issue in the project schedule analysis was that the actual construction methods for tunnels and removal shafts were changed from the original plan but were not updated in the scheduling files. • Since the review process was performed after the project completion date, the involved personnel had moved to new projects so it was hard to arrange interviews. • The retrieval of site records was time consuming if they were not documented properly.

Cost Performance	• There was no integration between schedule and cost (the cost order numbers did not match to the task items in the scheduling files), so it was time consuming to calculate costs for different tasks in the project. • In some cases, equipment costs were considered as part of the activity cost, so cost analysis for resources was really challenging.
Resource Allocation	• The resource analysis could be done by accessing site records and cost items. But the utilization of crew and equipment did not capture in the project records.

The PIR result shows that the main problem in this project is the lack of a systematic communication and record management plan to effectively access project information.

Even though the site information such as crew size, equipment working hours, interruptions and work progress were recorded on a daily basis, there was missing data for particular dates and in some cases, the recorded data was not accurate.

The next issue was the work flow documentation; the sequence of activities was not captured thoroughly and the daily progress report did not contain activities detail. Consequently, the new construction method utilized in this project was not properly documented and it was really challenging to find responsible experts to confirm the information and the actual process. And in some situations, they could not recall the particular details of the process. This is an issue especially when the project history is required for planning of future projects, and claims and dispute resolution.

3.3.4 Post Implementation Review Findings

The challenges encountered for the data collection in this case study clearly demonstrate that project documentation, specifically during the construction phase, is an essential task that can facilitate the project performance control, while recording a project history for the claims, lessons learned studies, and planning of similar projects in the future.

Although static data collected from a site would be useful for different decision-making purposes, there is still a need to capture detailed aspects of a construction process such as work flow, resource interactions, and effect of external factors on the project progress. This information, which demonstrates the dynamic nature of a project, assists the project team in discovering the effect of resources, weather, site condition, and other factors on the progress of activities on a daily basis, and helps them to enhance project performance by adjusting different project elements such as working hours, crew size, and equipment type. Therefore, development of an integrated and comprehensive documentation system for a project would provide more valuable information for managing ongoing projects and also for future use.

Usually among project documents, CPM schedule, created by the scheduling software, provides the project plan, and tracks project progress and resources assigned to the tasks. But the limitation of project schedule in detailing tasks and providing resource usage does not fulfil the need for detailed documentation of the construction process. So there is a need to have an integrated system that can record both static and dynamic aspects of a project in one system.

In this project, a simulation model was used during the planning stage to predict tunnel excavation advancement, resource utilization, and total cost and duration of the project. Since the Simphony model previously created during planning stage was not available during PIR review, a new model was developed based on available data from the planning phase. And because of the changes in the construction method, another model was created to show the actual process and resource interaction in the project. The development of a simulation model based on documented information demonstrates that simulation can also be used during the construction stage by creating an as-built process model with all the actual information related to resources, site condition, and the external factors. It shows that if a simulation model had been created during the planning phase and updated during the construction stage based on what was happening in the site, then a documented process model could be available at the end of the project to show the changes in the process, in the resource utilization, and in the performance of the project regarding the cost and duration of different activities, in a dynamic and integrated format. A user can run a simulation model during the construction phase for monitoring construction progress or replay it after the project completion date to obtain specific reports. An accurate model can provide various results from project total cost and duration, detailed project progress and product information, to project work flow, interruptions, equipment breakdowns, number of resources, utilization of resources, and site condition. In the next section, the process of creating a simulation model and its available features for project documentation are discussed.

3.4 Use of Simulation to Document Construction As-Built Information

In this section, a simulation model for the 137 Avenue tunnel project (developed as part of the PIR report) is presented. As previously discussed, it has been determined that a simulation model is a solution for construction project documentation that is also capable of recording the process model and its details in a dynamic fashion. For simulating construction activities, a simulation model was developed in Simphony.Net 4.0, and the General Purpose Simulation (GPS) template was utilized to create a model. Two separate simulation models were created to demonstrate the baseline and the actual process of the project. Since these models were developed at the latest stage of the project, the provided details were limited to the existing documents, and due to lack of data, they do not cover all aspects of the process; however, they are still effective for process recording.

3.4.1 General Purpose Simulation Model for a Process Documentation

A GPS template in Simphony 4.0 provides general elements to model various activities and different types of resources in a simulation model. All the elements in the GPS template, and available features in the software, such as calendar, facilitate developing a process model in such a way that it is easy and understandable for the user. For the development of the simulation model for the original plan, the only available documents were project baseline schedule and resource allocation chart (no information about delays or number of equipment for the simulation use were specified in the CPM schedule), but for the actual process model, the site data, along with the project schedule, were used to observe number of resources (mainly crews), activity sequence, actual duration, and any interruptions that happened in the project. It is important to mention that the actual project schedule (as-built schedule) was revised during the PIR process based on existing documents, so as it is shown in Figure 3-7, it provides more details than baseline schedule and it has the actual construction methods used in the project. The as-built schedule also includes the trenchless construction including Pilot Tubing and Pipe Ramming methods, which were used by the City of Edmonton for the first time.

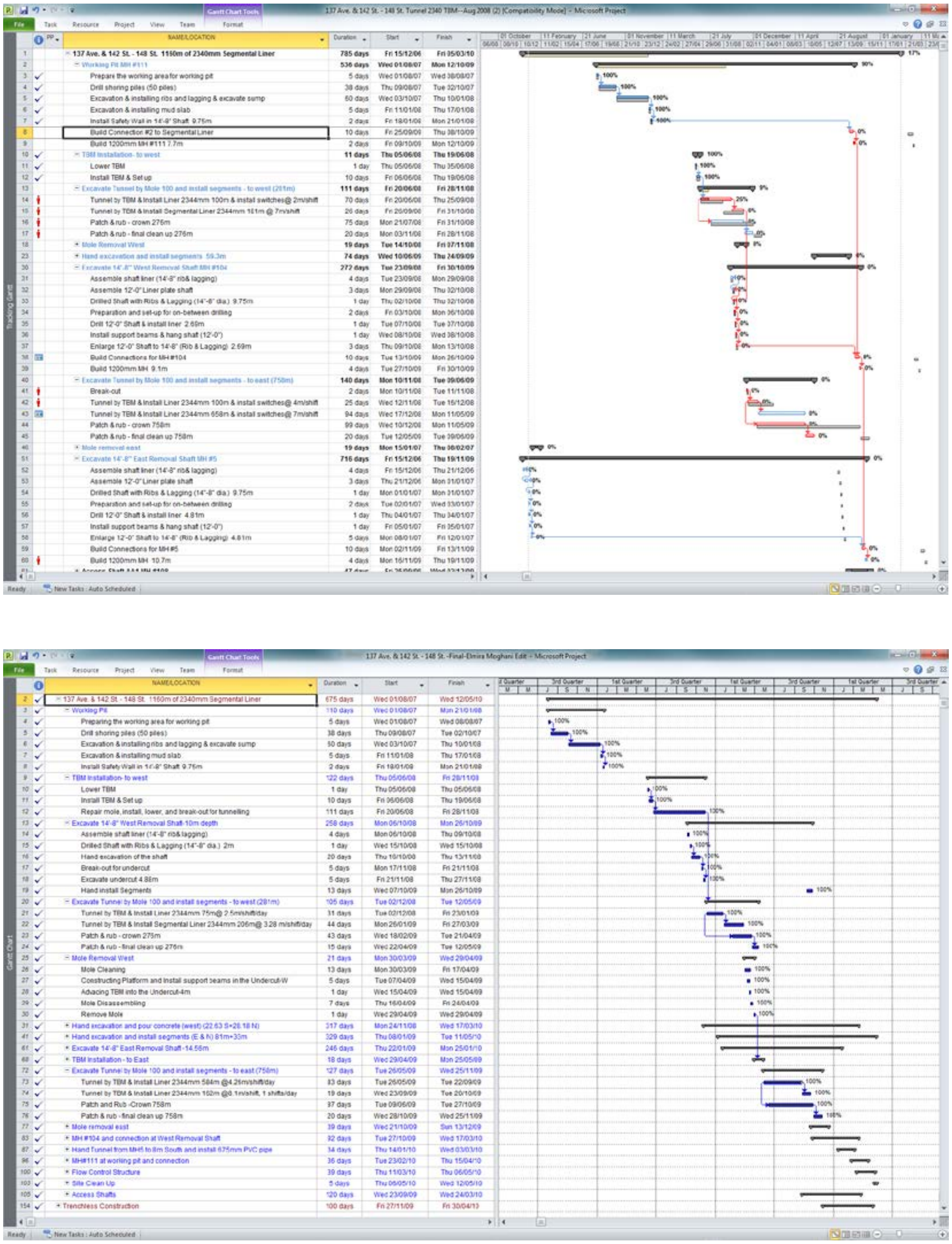


Figure 3-7: (Top) Baseline vs. (Bottom) As-Built Project Schedule

3.4.2 Simulation Model for the Plan and As-Built Process Model

The plan and as-built process models were created based on the CPM schedule developed in MS Project. A list of activities, and their sequence and duration were extracted from the CPM schedule. It is mentioned earlier that due to the lack of a systematic communication and record management plan, data was not documented accurately and there is missing information in the project. Therefore, simulation models developed in this stage of a project do not exactly present the actual story of the project; however, they are still useful for future project planning. As it is depicted in Figure 3-8, the construction process for the original plan does not include details of activities.

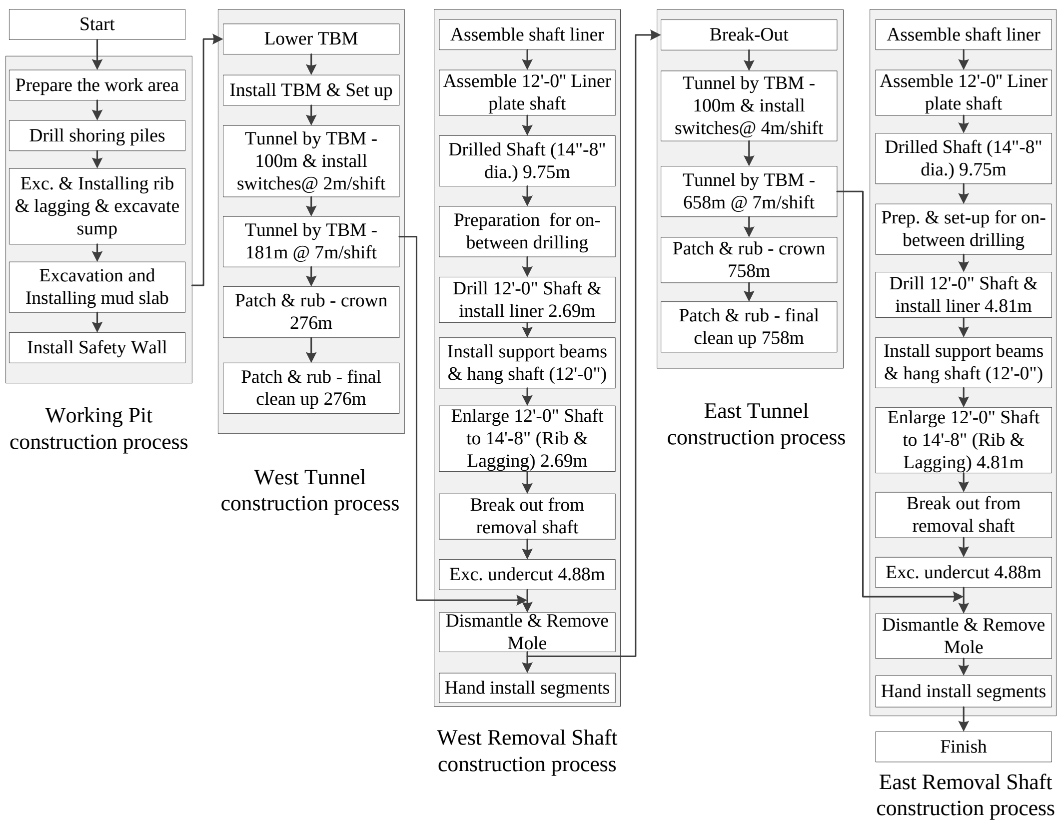


Figure 3-8: 137 Ave. Tunnel Project Activities Sequence for the Original Plan

A Simphony model developed for this tunnel construction project includes details such as product information (e.g. tunnel and shaft dimensions), shaft and tunnel excavation procedures based on construction method and soil type, TBM advancement cycle (excavation, lining, and resetting), dirt removal process, resource allocation, machine breakdowns, and effect of weather conditions on advancement of the work.

Both simulation models utilized main elements of General Purpose Template such as task and resource elements. To have a similar structure as the project CPM schedule, simulation models were divided into separate processes based on the work packages in the MS Project file. Also, the relationship between processes and the start date of each process were derived from the same document. Figure 3-9 illustrates the tunnel construction simulation model for the baseline (plan) schedule.

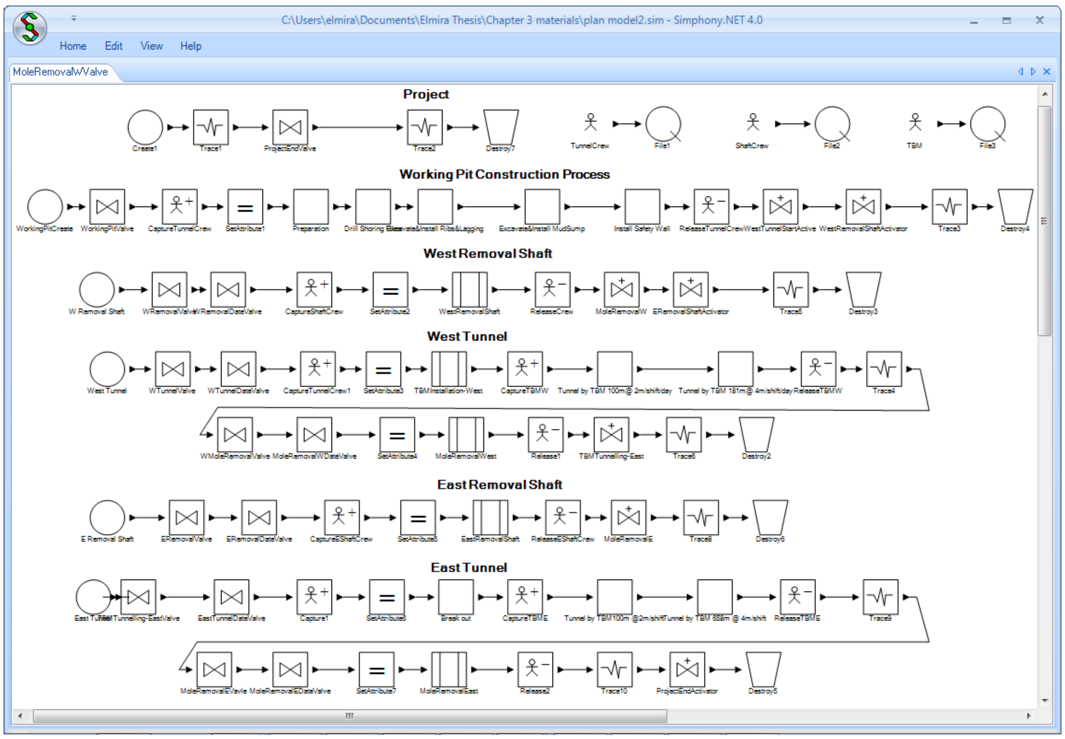


Figure 3-9: 137 Ave. Baseline Process Model Using Simphony GPS Template

In this model, six processes were created to simulate the project, working pit (working shaft), west removal shaft, west tunnel, east removal shaft, and east tunnel construction. Three resources were defined in this model. The TBM resource was utilized for the excavation and lining cycle of the east and west tunnel. The TunnelCrew resource was assigned to construction of the working pit, both tunnels and TBM removal activities and the ShaftCrew resource were allocated to construction of removal shafts and undercuts. To create a manageable and organized model, all the activities for TBM installation, removal shaft construction, and TBM removal are encapsulated in elements called composite elements, which are used for grouping a set of elements.

As an example, for west tunnel construction, as it is illustrated in Figure 3-10, a tunnel crew is captured at the beginning of the process. After TBM installation, the TBM resource is captured for the tunnel excavation and lining process. Since there is no information for the excavation and lining duration, a total process of tunnel is divided in two elements that represent construction of two tunnel sections. After tunnel construction, the TBM resource is released, and after TBM removal element, the crew resource is also released. At this point, construction of the east tunnel can begin.

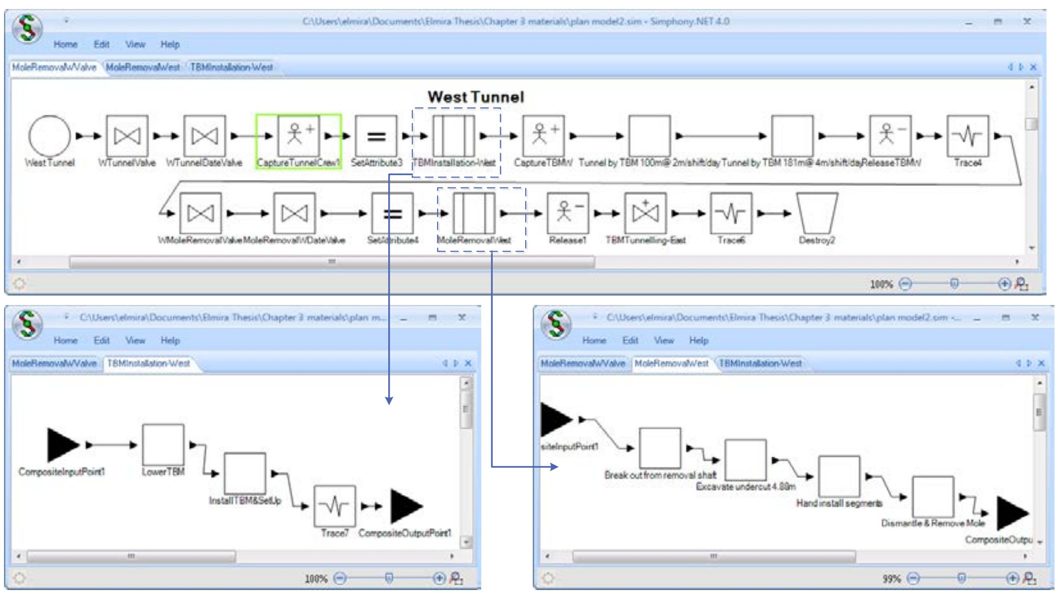


Figure 3-10: Simulation of West Tunnel for 137 Ave. Tunnel Project Baseline Process

The as-built process model (previously shown in Figure 3-7), has more details about activities, process interruptions, resource allocation, and equipment breakdown information, and also includes the simulation model for the new work added to the scope of the project. In addition, all the changes that occurred in the shaft and tunnel construction processes were documented in the model. In 2010, trenchless construction was added to the project, which consists of constructing all the connections between working shafts in a period of three months. Accordingly, the number of working crews increased in the project. Figure 3-11 shows the distribution of crews during the construction phase. This histogram is created based on the site daily information and time sheets, but since it was created at the completion phase, it may not represent the actual number of crews during the project lifecycle.

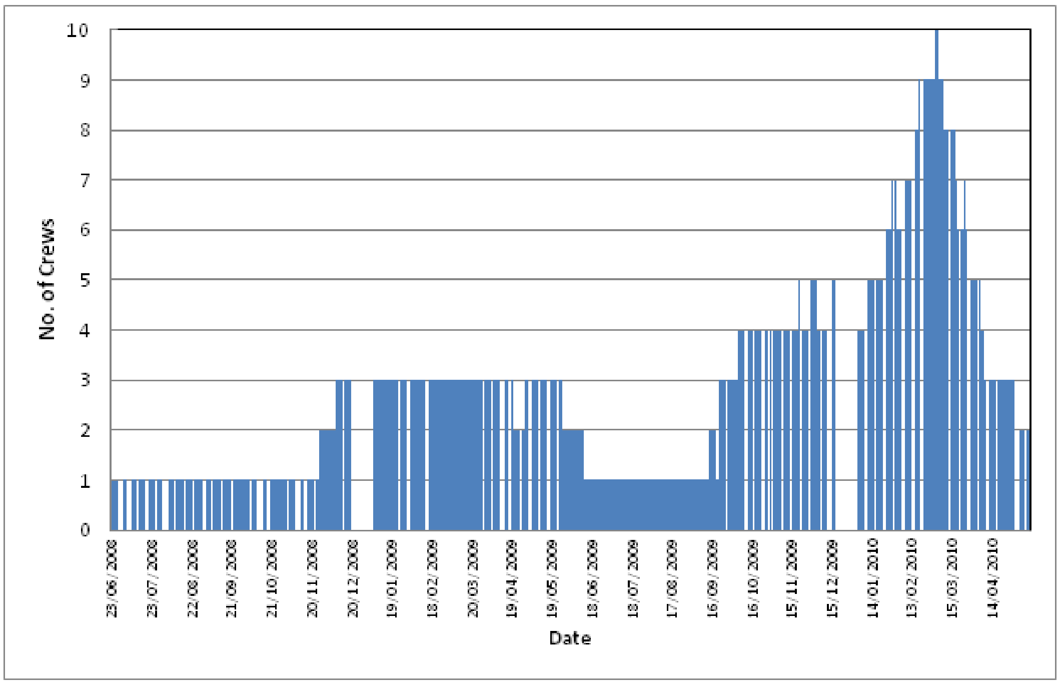


Figure 3-11: Actual Number of Crew in the 137 Ave. Tunnel Project

Based on the actual project schedule, until the end of November 2008, the major work was related to the west removal shaft and then west tunnelling. From November 2008 until April 2009, three crews were working on west tunnel excavation, hand tunnelling in the working pit location, and hand tunnelling at 142 Street, which is a separate project. In 2010, two crews were working on east and west removal shafts and the others were involved in trenchless construction. Increasing the number of crews helped to finish the work ahead of schedule (The plan schedule shows that the construction finish date is projected in June, but they finished all the construction activities by mid-April). The actual crew allocation was considered in the simulation model as well. Figure 3-12 illustrates the actual process simulation in Simphony.

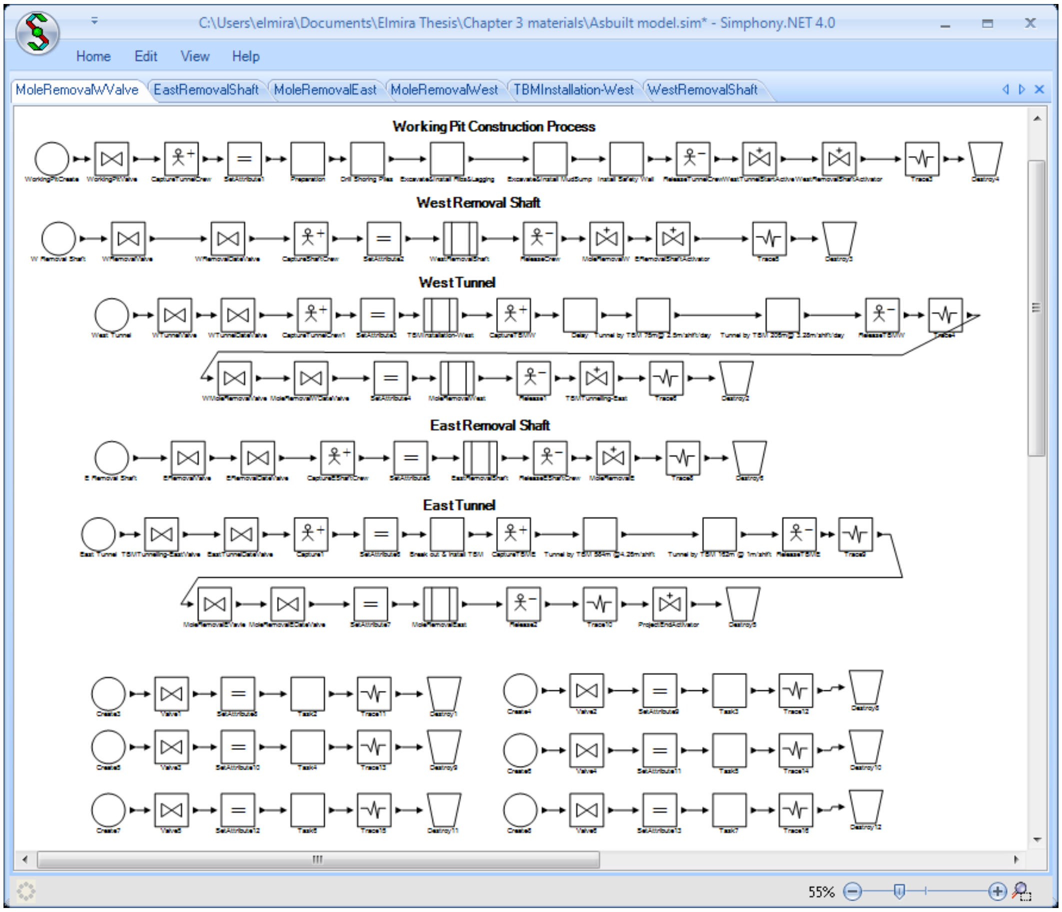


Figure 3-12: 137 Ave. As-Built (Actual) Process Model Using Simphony GPS Template

In this project, despite the details that the as-built model provides, it still simulates process model at a higher level and in a simplified format; similar to the baseline process model, the as-built model does not include the details of the excavation process for shafts and tunnels. In addition to the TBM, TunnelCrew, and ShaftCrew, the crane resource is also considered in the actual process model based on the information captured from site daily entries.

As a simulation results, a user can obtain the duration for each work package and the total duration of the project from the simulation (Figure 3-13).

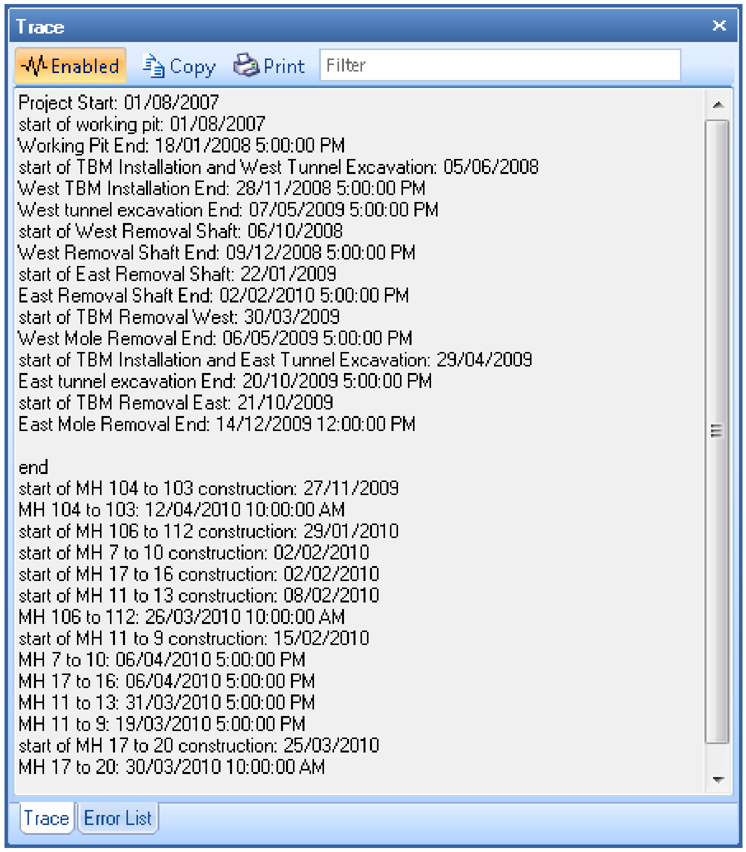


Figure 3-13: Start and Finish Date for Work Packages in the As-Built Simulation Model

Also, the resource utilization can be captured from the statistical outputs of the resource element in the Simphony model, as shown in Figure 3-14. The simulation model can also provide additional information for the user such as total project cost including crew cost, material cost, and equipment cost, as well as the cost of each activity, and the project cost for a specific period of the construction project.

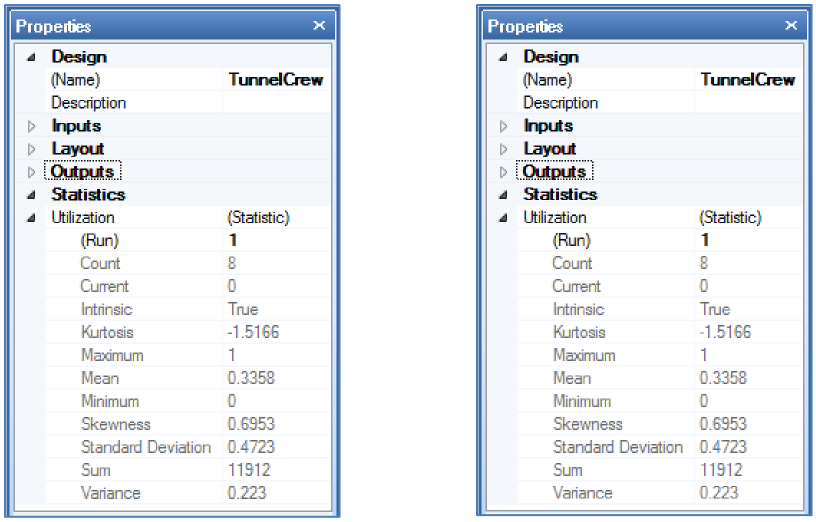


Figure 3-14: Tunnel Crew Utilization in the (Left) Plan and (Right) As-Built Process

3.5 Conclusion and Proposed Advancement

This case study demonstrated that a simulation technique can effectively document the process and product information of a project. By reproducing the actual process and replacing the input values with actual data and running the simulation, it will replay the actual story of the project and regenerate the outputs for the duration, cost, and any other information that a user intends to extract from the model. Furthermore, the ability of a simulation system to integrate schedule and cost, and to consider resource interactions and the effect of external factors on the advancement of a project makes it a powerful tool that combines different aspects of the project into one system, and facilitates accessing the required data for claims or future project evaluation, or even planning of future projects.

In this study, the main goal was to demonstrate the application of a simulation system in process documentation, and to present various outputs that can be generated from a simulation model. However, except for resource utilization which does not accurately represent the real situation in the construction site, simulation results for the 137 Avenue tunnel project did not provide more information than the existing scheduling documents, due to the limitations listed below.

- The waiting times for different resources were not considered in the model. This affects the final simulation results and leads to inaccurate calculation of resource utilization in the system.
- Interruptions due to external factors such as weather conditions were not completely recorded; therefore, this was not included in the simulation model.
- No breakdown was simulated for the TBM in the baseline process (there was no access to the historical data at the completion phase of the project).
- The concept of shifts was not implemented in the simulation model, and a fixed 8-hour shift was applied to calculate the duration of different tasks in the simulation, specifically for the baseline process model (GPS is not capable of modeling shifts).
- The information for the Crane breakdown was not documented completely during the construction stage, so the input values for the crane breakdown were not accurately entered in the as-built simulation model.

This case study illustrated that project documentation through the simulation approach can assist project personnel in recording dynamic aspects of a project, in addition to the typical information that is usually captured through the available tools and methods. However, to utilize a simulation technique for project documentation in the industry, a practical and generic tool is required. Also, due to the limitation of the GPS template to model all aspects of a project, this research proposes a simulation-based framework to assist project managers in developing a complete history of a project for different project management purposes. In the next chapter, a conceptual model, the proposed structure, and the component of the framework are discussed in more detail.

Chapter 4 - A Simulation-Based Framework for As-Built Documentation in Construction Projects[1]

4.1 Introduction

According to the literature review and a case study presented in Chapter 3, there is need for a well-structured system to record, process, and access all project information from planning stage to the completion phase, including both static and dynamic aspects of a project. This should include as-built product and process information and all the changes that occurred during the construction phase. The proposed system must provide easy access to the project information in an integrated fashion to give a complete overview of the project at any stage and enable comparison between planned and actual project performance. To apply the proposed framework in industry, not only should it be easily reusable for various construction projects, it also needs to offer a user-friendly interface to be acceptable.

Through early studies and experimentation presented in previous chapters, a simulation modeling approach was identified as the most suitable method for our requirements as it presents unique features that facilitate representation of construction processes and dynamic aspects of a project. But current simulation systems are not suited for integration purposes or to model the entire project lifecycle, which was a motivation to look for new techniques that can facilitate integration into the simulation environment. Studies show that distributed simulation, and in particular, the High Level Architecture, presents a promising simulation tool to facilitate the deployment of the envisioned framework by

[1] Part of this chapter has been previously published (Moghani and AbouRizk 2010; Zhang et al. 2012) - See Appendix A.

integrating product models, process models, and other computer software in a single, interoperable system. The proposed system is designed to collect actual data and processes, as in the physical as-built documentation that is traditionally required at project completion, during the construction phase; more importantly, it provides factual information about the project's actual activities, logic, durations, resources, and costs to create an accurate history of the project. This framework will assist project managers in project close-out, lessons learned studies, automation of the production of as-built information, claim analysis, dispute resolution, and general project control.

This chapter focuses on the development of the proposed framework; in the first step, a conceptual model is created and its components are determined. Then, the implementation of the system is detailed, and a case study is presented to demonstrate the structure of the framework.

4.2 An Integrated Framework for Documenting Construction Projects

The study of simulation application for the documentation of construction processes for the 137 Avenue tunnel construction project, described in Chapter 3, demonstrated that implementation of the proposed system in industry requires the development of a flexible and generic application. This research utilizes a software application framework approach for the development of a reusable, extensible, and modular system with flexibility in representing different modeling approaches and data forms by deploying a distributed simulation technique in the Construction Synthetic Environment (COSYE).

Software application frameworks help developers capture and reuse both the design and code of an application to reduce the programming time and increase the reliability of the system for future developments. Usually, an application framework encapsulates common design patterns and code for a specific application domain and also provides necessary features to extend its behaviour. The framework also applies specific rules which can facilitate the development process and enhance the functionality of the application. Construction simulation has utilized this concept to facilitate application use in the industry (Hajjar and AbouRizk 2000) by identifying the common design and components for simulation applications. In order to design an application framework, the first step is to define the scope of the framework and all the services that it provides for the intended domain. Based on background discussions, a conceptual framework to record, process, and access project as-built information through a simulation approach is illustrated in Figure 4-1.

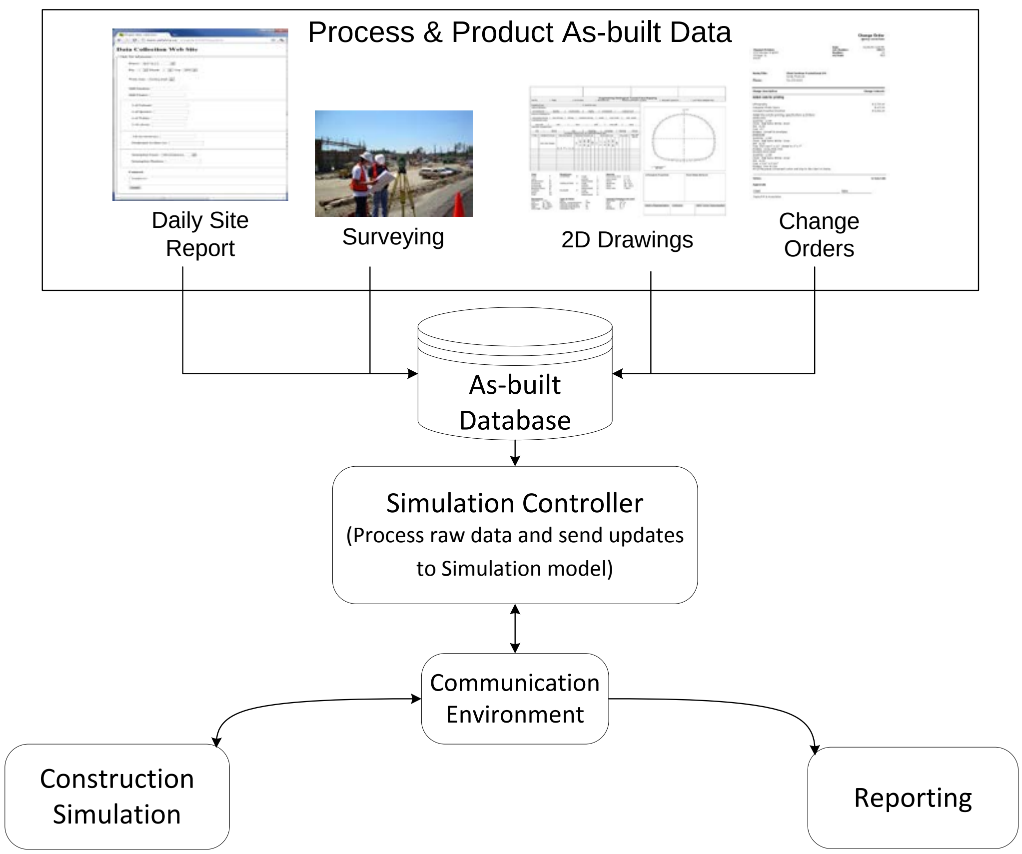


Figure 4-1: A Conceptual Framework for Project as-built Documentation

The proposed framework is divided into three main modules; the first module is responsible for the documentation of design, planning, and construction information. In this module, the main goal is to create a data repository, mainly for the construction phase of a project. A data collection system is designed to collect and send information to a central database on a daily basis. In addition to that, all the information required for the plan and as-built comparison of the product model is also stored in the database. For the development of this module, Microsoft Access is utilized to store and process information using various tables and queries. The second module consists of process model simulators to model plan and as-built process models. Finally, the third module provides output and comparison analysis for the project performance in the form of management reports. This module should be able to provide valuable information on different stages of a project. In the implementation process, each module can be composed of more than one component, as listed below. In this section, each component is described briefly, and later on in this chapter, the structure and the implementation of each one will be detailed using a case study.

4.2.1 Data Collection System

The data required for the proposed framework should be identified, collected, and stored in a manageable format for easy access and use in the simulation-based system. As part of the framework, a data collection system is formalized in order to collect and transfer information to a central database. Reviewing the studies conducted in the area of data collection in construction projects shows that the available automated data collection systems are not capable of capturing all the information required for updating a process model in this research. El-Omari (2008) demonstrates the capabilities of the automated data acquisition technologies as shown in Table 4-1. Based on the table, available methods are limited in scope, and even utilization of various automated systems in the construction site may not provide enough data for our research purposes, yet would add additional costs to the project.

Table 4-1: Automated Data Acquisition Technologies Capabilities (El-Omari 2008)

	Bar Code/ Radio Frequency Identification (RFID)	Laser Distance and Ranging (LADAR)	Pen-Based Computer	Multimedia		
				Photo Images	Video Images	Voice Records
Labor	✓		✓			
Material	✓		✓			
Equipment	✓		✓			
Quantity		✓	✓			
Task Progress		✓	✓	✓	✓	✓
Weather Conditions			✓			✓
Productivity		✓	✓	✓	✓	✓
Problem Areas		✓	✓	✓	✓	✓

Therefore, as a simple solution to implement the framework and in order to collect all the required data, a manual data collection system is selected to capture the history of the project. Tse (1989) named certain criteria to prepare site reports; he suggested that report completion time should be less than 15 minutes and the requested fields need to be as specific as possible. Also, he mentioned that the report must be prepared based on the studied project, must be easy to analyze, and it should include comments whenever required.

So in this system, the project inspector documents daily information in a pre-defined format and sends it to a central office to be transferred to a Microsoft Access database. Three different approaches were tested for three tunnel construction projects by the City of Edmonton. In one project a Microsoft InfoPath form was created to collect site data; in this system, a user fills out the form in an offline mode and submits it via email. The data gets transferred to the Microsoft SQL server database through a connection provided for this purpose. In the second project, data was collected through an online form. In this case, by submitting a form, data can be transferred to an SQL database and stored in a predefined table. Finally, in the other project, a site superintendent filled out a PDF form that was designed for the project and sent it to a project planner on a daily basis. Figure 4-2 illustrates the InfoPath form designed to collect information from the construction site.

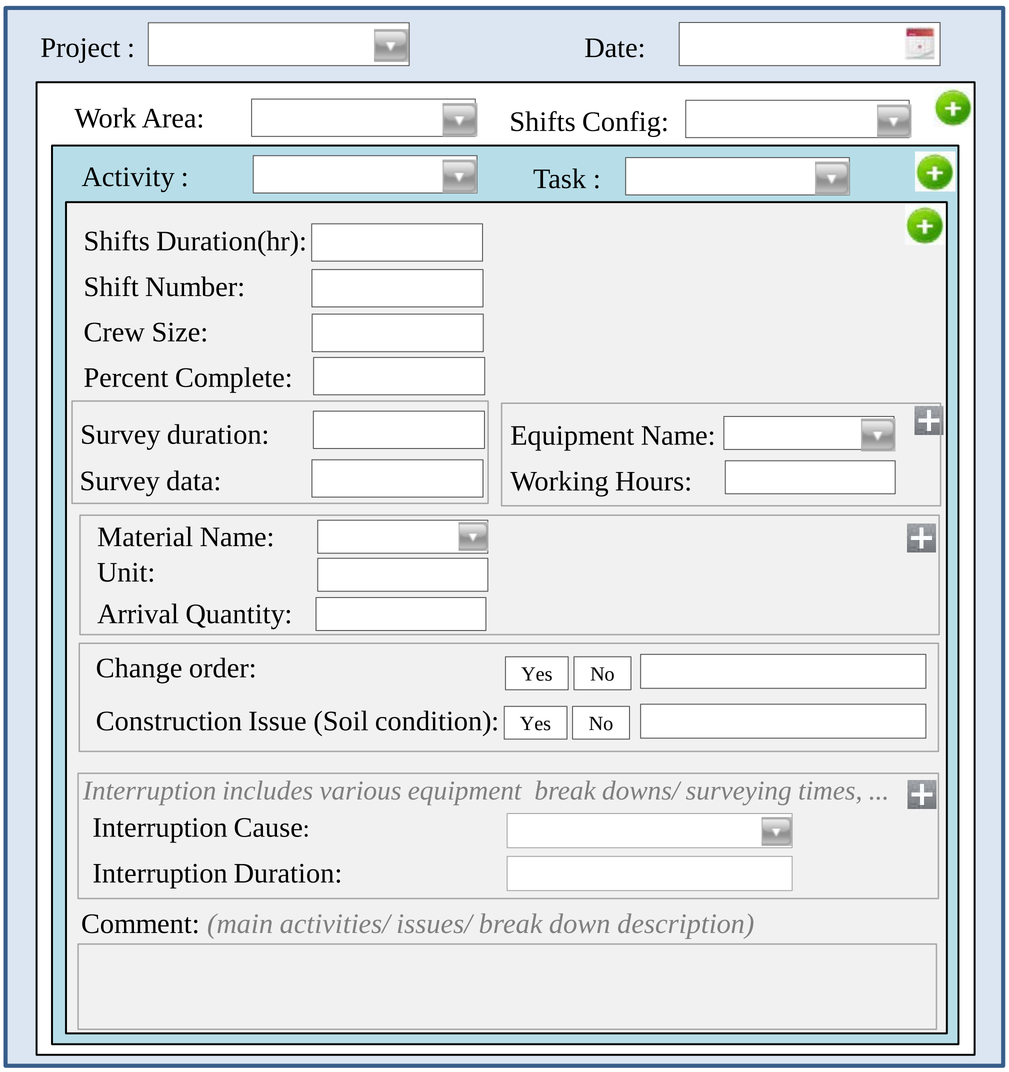


Figure 4-2: InfoPath Form for Daily Site Data Collection

The other methods also collect the same data from the project site. All the forms are designed based on the required information for updating the construction process model. The level of detail proposed for the data collection is specified based on the availability and accessibility of the data at the site, and the time and effort that it takes to collect and document it. The details should be considered based on the minimum amount of data that needs to be captured to generate accurate as-built processes, which produce valuable outputs for management use. The process of choosing the right data for creating and updating the process model is discussed further in Chapter 5.

4.2.2 Database Design and Structure

To manage and control a project, project managers usually use information captured from a construction site on a regular basis. Therefore, preparing an organized system for the data storage helps users easily access the data. In the construction industry, a database is usually utilized to organize and store project data. In this research, a Microsoft Access database is chosen for data storage and retrieval. In the database, construction data is stored in a table called “Daily Records”, and based on the requirements of a simulation, different queries were designed to provide input values for the framework (Figure 4-3).

In addition to the captured data from the site, a database may store information about plan and as-built product model specifications, such as original alignments or dimensions, and may also contain additional data about site conditions (e.g. geotechnical information).

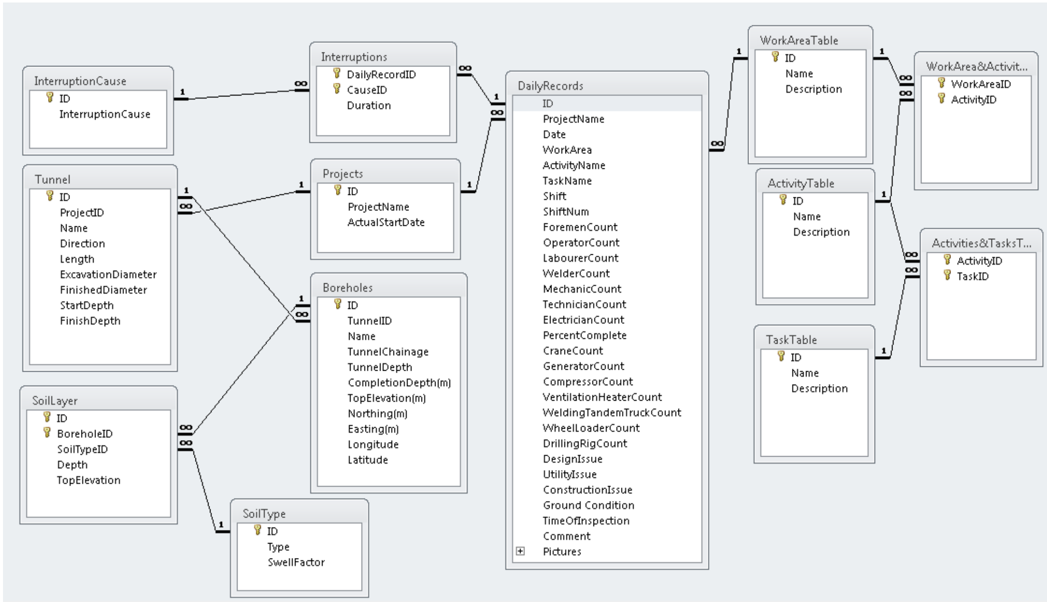


Figure 4-3: Database Structure and Entities Relationship

The main table in the project database is the "DailyRecords" table. In this table, all the data is documented on a daily basis and stored based on the area that work is being done, and the activity that the task belongs to. Figure 4-4 shows various fields of this table; crew size, percent completed, number of shifts per day, equipment, and different site conditions are recorded in this table. To provide a generic structure for a database, separate tables are created such as "Project," "WorkArea," "ActivityTable," and "TaskTable." To facilitate the use of information, various queries are developed to store information for a specific work area or for certain use of simulation, which are shown in Figure 4-4.

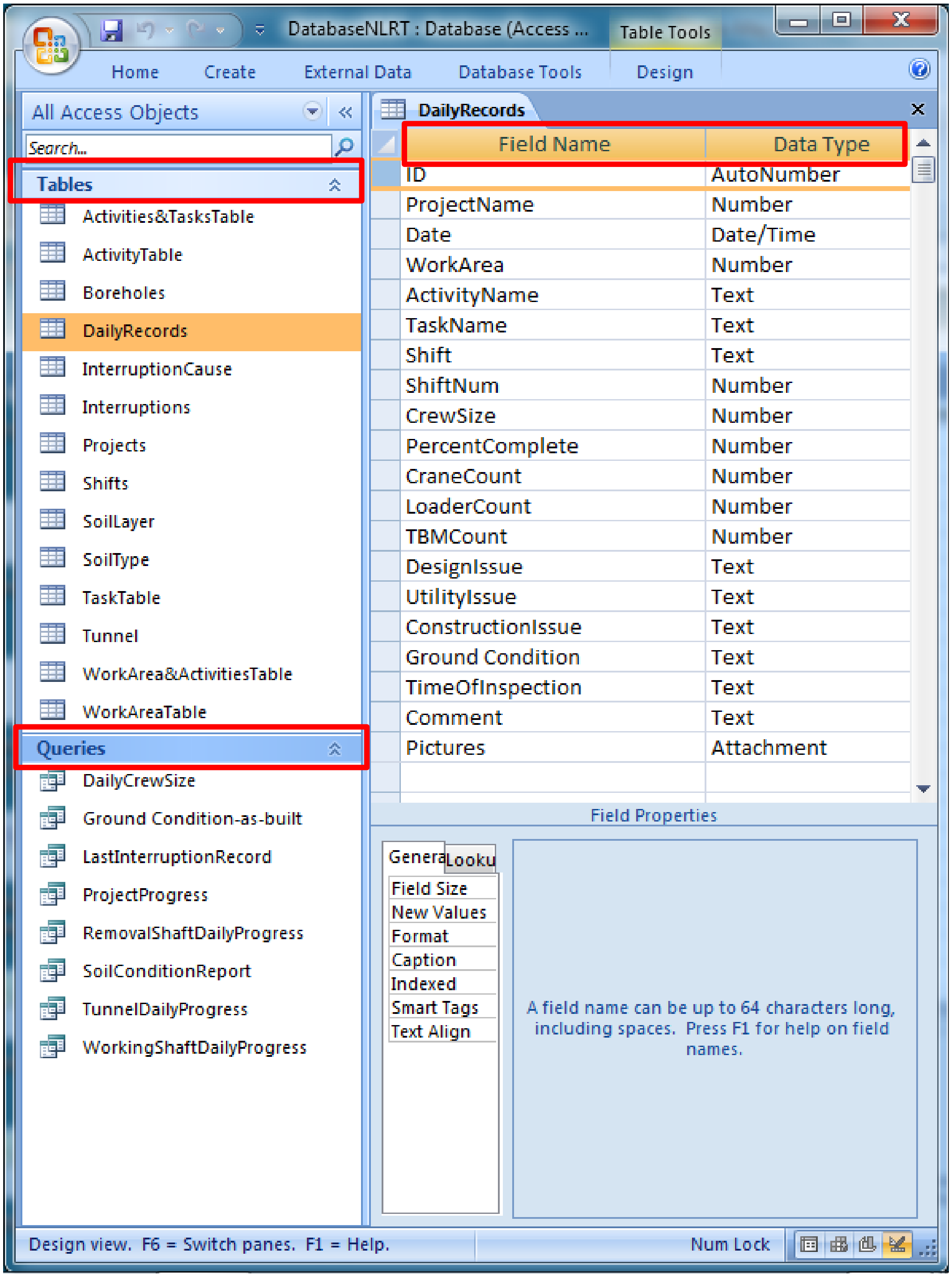


Figure 4-4: Database Tables and Queries List and Fields of Daily Record Table

4.2.3 Simulation Controller Component

Raw data from the construction site usually requires processing, analysis, and formatting to be ready to use either by different project management software applications, or just for reporting purposes. This module is responsible to connect to the database and collect the information required for updating different inputs of the process model and analyzing them; it then sends an update to the process model through the COSYE RTI. The data analysis depends of the type of input that the process model requires; it may perform calculations on the raw data to get the results and update them. This component is also responsible for regulating the frequency of sending updates for the process model in the framework.

4.2.4 Process Model Simulation Components

More than one instance of a process model simulator is considered in the framework; the plan process model is created during the planning stage of the project for various decision-making purposes, such as scenario-based planning, time and cost estimation, and resource allocation analysis. During this stage, all the simulation model inputs are defined based on expert judgements and historical data from previous similar projects. Another process model that is part of the application is the updated process model; this model is similar to the original model, but it varies in the input values as it is updated based on the site data updated by the simulation controller component.

It is common in the construction industry for a change to occur during the construction phase of the project. In many cases, the change order either adds new work to the project or changes the activity flow in the existing process. In the case of any changes to the construction process, there is a need for a new process model component that represents the as-built process of the project.

The framework is designed in such a way that in the presence of as-built information in the database, the as-built process model will be updated up to the point that the information is available in the database. From this point forward, the simulation model will continue to simulate the process based on the original model and original inputs from the planning stage. If changes happen to the project scope, a new process model will be created and joined to the framework to reflect the changes in the process. In this case, the original as-built model will no longer receive updates, but an as-built model which presents the new process will be updated with the available as-built data. Figure 4-5 illustrates the updating procedure for a process model in the proposed framework. This process is detailed in Chapter 6.

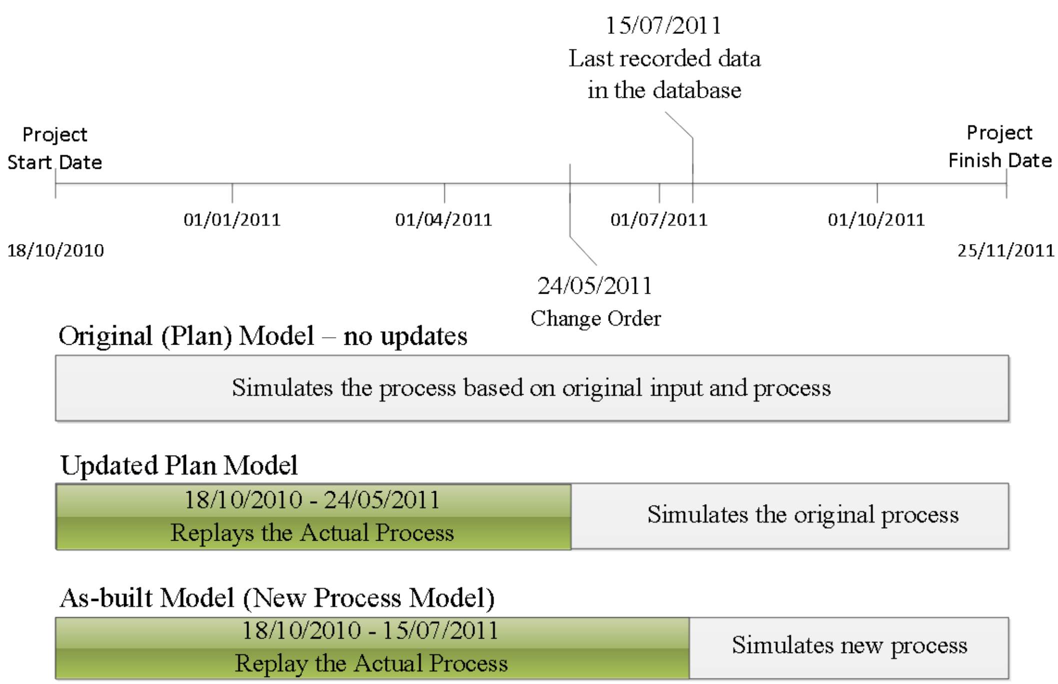


Figure 4-5: Updating the Process Model Based on the Actual Data

The purpose of having more than one instance of the process model is to give project managers an option to conduct a complete comparison analysis based on the outputs that each model provides.

4.2.5 Reporting Component

Industry application of the proposed framework would be easier if it featured a report component that pictured the project performance, demonstrated the history of what happened during the construction phase, and displayed comparison results between the original plan and the actual process to the project managers or decision makers. It should also facilitate the interpretation of results and reduce the time spent in explaining information.

In the proposed framework, the reporting component covers different information from the product information (similar to what as-built drawings usually provide) to the process information (such as resource utilization, activity flow, equipment breakdown, cost estimation, and any interruption that happens because of weather conditions or other external factors).

After determining the necessary components of the proposed framework, the methodology to develop the system is specified. Based on this methodology, the development of the framework involves three major phases:

1. Utilize an appropriate simulation architecture which allows integration of a process model, product model, and simulation systems in the framework.
2. Research and develop an information integration system to include as-built and as-planned process and product information. To produce an efficient system, a

complete understanding of construction activities and required site data is necessary.

3. Provide a structure to update inputs of a simulation model based on actual data captured from a construction site, and modify the process model simulation based on changes and new works added to the project scope, without recreating the entire simulation.
4. Develop a procedure to update reports as the simulation progresses to include the latest project information.

4.3 Implementation of Proposed Framework in COSYE

Simulation is known as a powerful technique in the construction industry which aids project managers in different decision-making processes by modeling construction activities and examining various scenarios. However, initial studies revealed that current discrete event simulation techniques are not an efficient solution for this framework; therefore, the proposed framework is developed in an HLA-based distributed simulation environment to exploit its capacity for integrating different components and facilitating data collection and information flow into the system.

As previously discussed in Chapter 2, distributed simulation enables integration of several models or software applications, while allowing interoperability and component reuse (Fujimoto 1999). By dividing a complex model into sub-models, many developers can participate in the creation of simulation models using various simulation techniques. Distributed simulation also enables the execution of sub-models by multiple computers.

A Construction Synthetic Environment (COSYE) is a distributed simulation framework developed at the University of Alberta based on HLA rules, for construction simulation (AbouRizk, 2006). Study of the applications developed in the construction domain utilizing COSYE assured that services provided by COSYE facilitate the development of the proposed framework in this research. In this section, the development of the framework in the COSYE environment will be discussed.

4.3.1 Proposed COSYE Federation Structure and Design

The use of HLA aims to provide a reusable, interoperable and extensible framework. Modularity in this research can be achieved by encapsulating a construction project and its information in a structured manner, and reusability can be attained by developing a simulation model and data management structure in a generic way. The simulation model will be extensible if it is constructed in a structured manner that allows for future improvements. The federation should be distributed in such a way that each component has its own special purpose that allows it to be reusable in future development, or easily replaced by a federate with a similar functionality. Figure 4-6 pictures the proposed system developed in the HLA-based COSYE environment. Different federates can join the federation and communicate through the messages that are being exchanged via the COSYE RTI server. The RTI facilitates time management and data transfer among federates. The Federation Object Model (FOM) needs to be developed in a comprehensive format to include all federation object classes and their attributes.

To facilitate reusability of the system, the process model should be encapsulated so that it can easily be replaced with the process model of a different construction project. Three different process model simulators may join the federation to show the original process, the updated process and the as-built process model if any changes are made to the process model or to the construction method. The as-built federate acts as a simulation controller; it communicates with a central database and provides as-built data for the federation on a daily basis during the construction phase of a project. The database not only stores information regarding the activities in the project, it also stores the product specification and external environment conditions such as geotechnical information, project geometry, and the surveying results, which may not be captured in the process model. The reporting federate produces project management reports for comparison of the project performance with the original plan, based on the results provided by the process model federates. In addition to the main components, various federates are considered as support federates to produce accurate results and make the simulation model more realistic (i.e. supplier federate). The visualization federate is also a part of the framework to enable the user to validate the model and visually track the process for monitoring and control purposes.

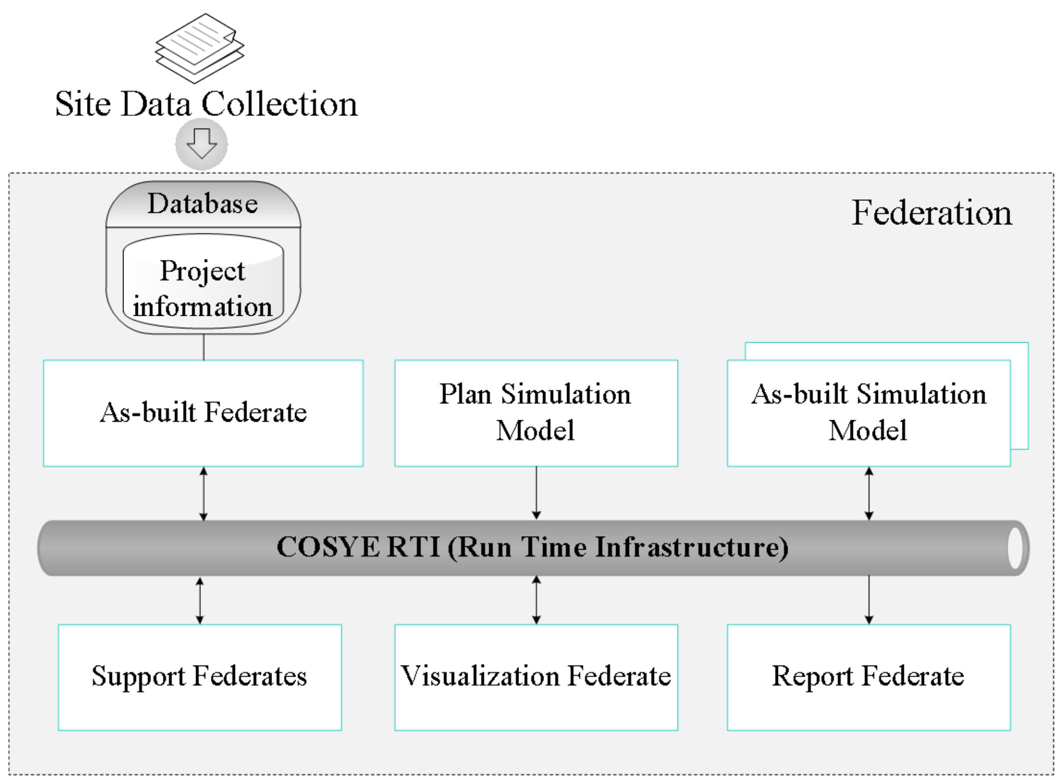


Figure 4-6: The Proposed COSYE Federation Architecture

4.4 Case Study: Tunnel Construction Federation Development

To implement the framework, a utility tunnel construction project in Edmonton, Alberta, Canada was selected as a case study. Since simulation modeling is the most practical method for projects with repetitive construction processes, of which the tunnel construction project is a good example, many simulation models have been utilized for modeling tunnel construction projects (AbouRizk et al. 1999; Fernando et al. 2003; Al-Bataineh 2008; Marzok et al. 2008). The proposed structure for the tunneling construction federation is presented in Figure 4-7.

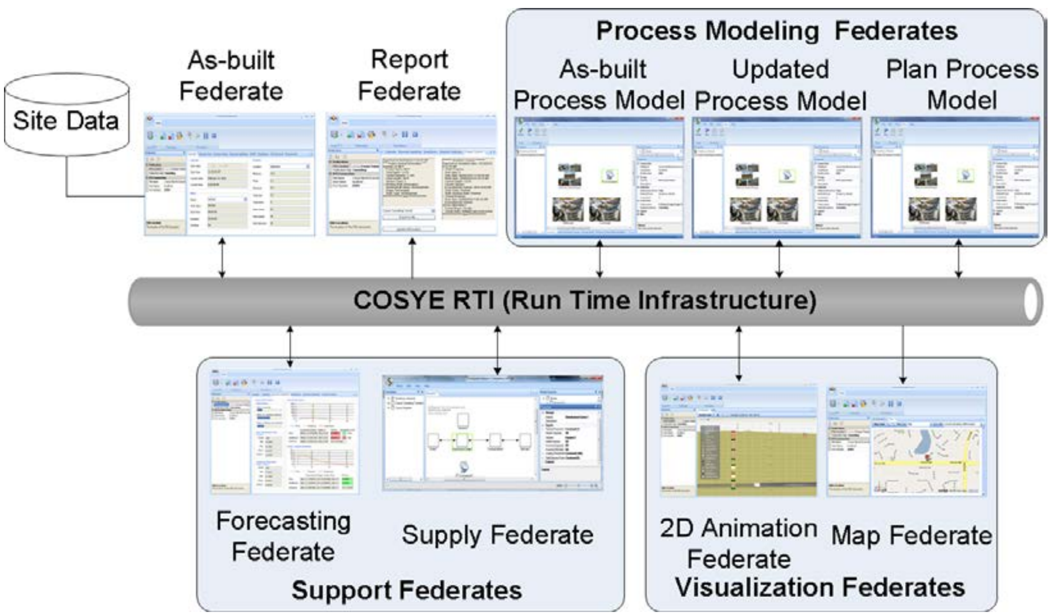


Figure 4-7: Tunnel Construction Federation

4.4.1 Federation Object Model

In an HLA federation, each federate is modeled independently so there must be a standard mechanism to facilitate data representation and exchange in the federation. Based on HLA standard, all the object and interaction classes and their attributes are defined in an object model, called Federation Object Model (FOM), developed with the compliance of the Object Model Template (OMT) (IEEE 2000. Std 1516.2 2000). HLA defines OMT as the interface language between federates which provides a common framework for object model documentation with a standard format and syntax. During simulation run, federates interoperate by exchanging data for the objects and interactions. Therefore, the object classes and interactions that each federate is providing or expecting from other federates should be specified in the FOM (Straßburger 2001).

To create a standard and generic FOM for a specific domain, a standard data exchange format is required; available standards for data exchanges in the construction domain, such as IFC (Industry Foundation Classes), mainly provide common classes and attributes for buildings, so they are not comprehensive enough to be applicable in other kinds of projects, like tunnel construction. So as a first step, a comprehensive study needs to be done to identify all the object and interaction classes that can be used in this domain. For that reason, a reusable high level project model was defined (Figure 4-8) as a foundation to create the FOM for various construction project federations in the COSYE environment (Moghani and AbouRizk 2010). The project model is intended to increase the interoperability and reusability of the components in various applications in COSYE (AbouRizk et al. 2010).

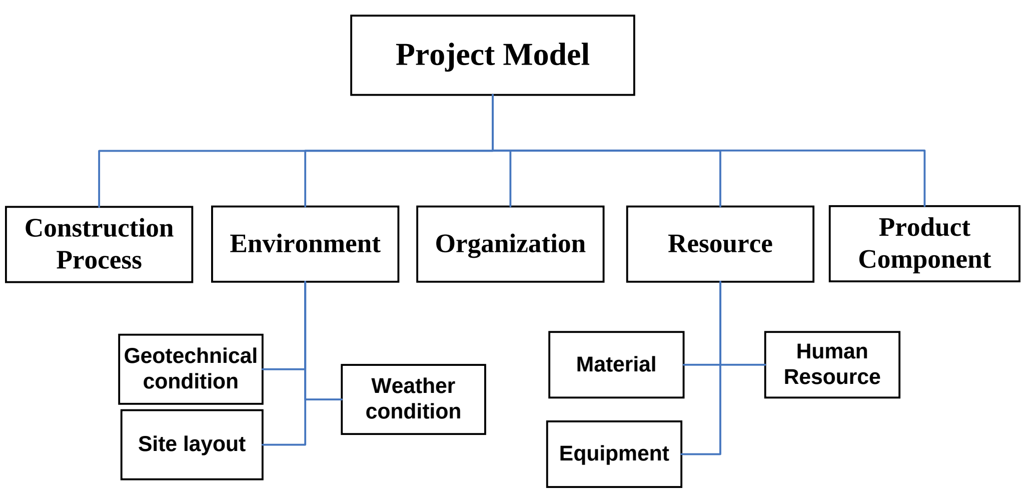


Figure 4-8: The High Level Project Model of Construction Operations

To create a tunnel federation object model, conceptual models for the construction process, product, environment, resource, and management were developed. To develop the models, the tunnel construction method for utility facilities was investigated, and all the information regarding activities and required resources was collected. In order to develop a product model, all the CAD drawings were reviewed, and details were added to the conceptual model. It is important to mention that the following models were created for the case study and they focus on the tunnel construction method using Tunnel Boring Machine (TBM). Figure 4-9 illustrates the project model and Figure 4-10 shows the product and process model for the utility tunnel construction project using TBM for tunnel excavation.

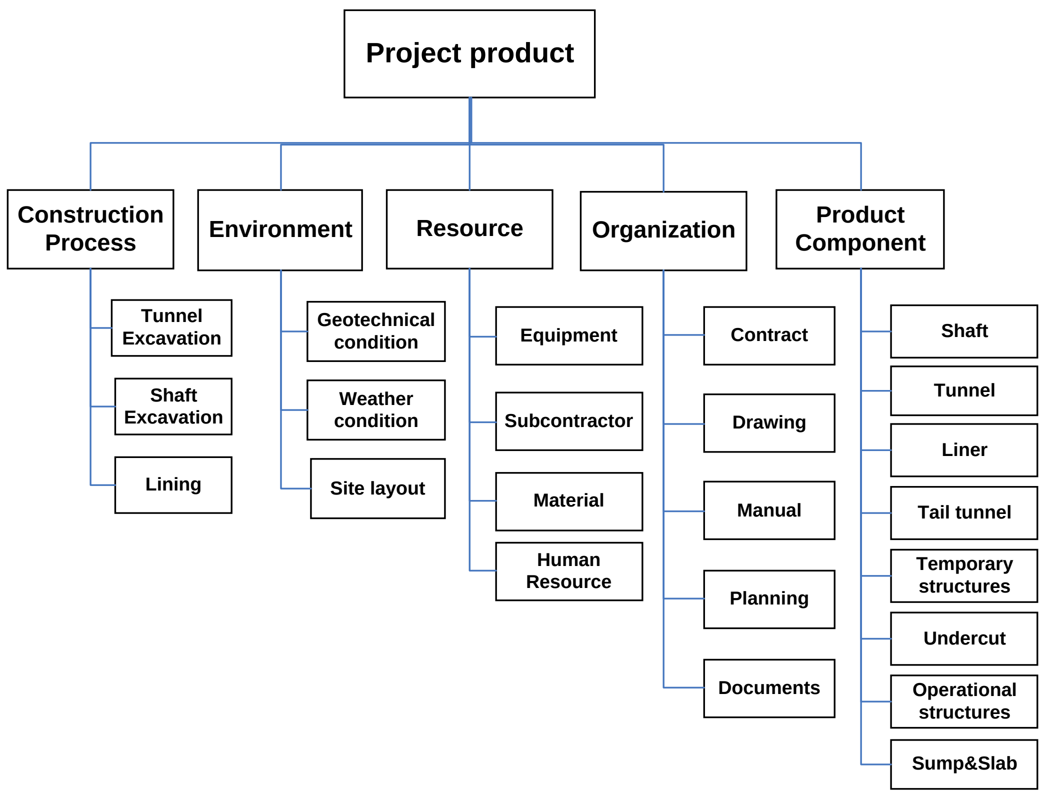


Figure 4-9: The Project Model for a Tunnel Construction Project

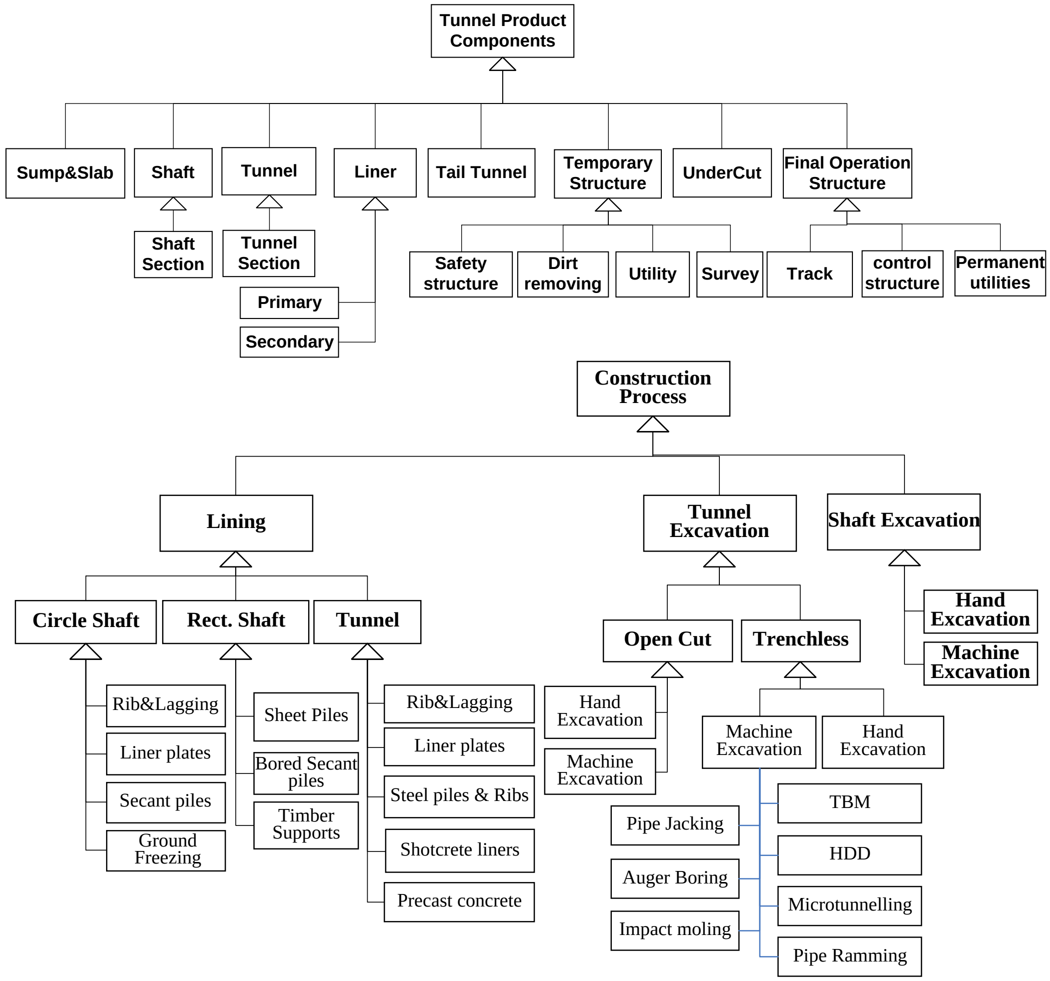


Figure 4-10: (Top) TBM Tunnel Construction Product and (Bottom) Process Model

The next step is the development of the FOM based on the conceptual models. The FOM is created using a C# or Visual Basic class in Visual Studio. It has a hierarchical structure using a tree-type interface. Figure 4-11 depicts the first version of the tunnel federation object model. In the development phase, each federate was designed based on an existing object model. By evolving the model, the FOM has been updated and also expanded to be more generic and reusable in various tunnel projects.

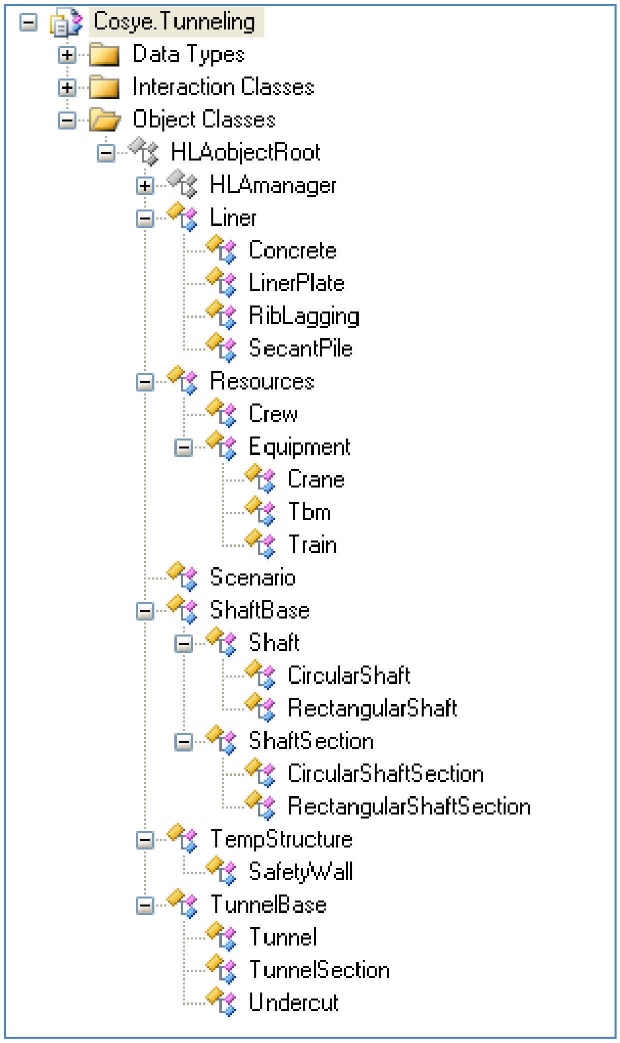


Figure 4-11: Tunnel Construction Federation Object Model (FOM)

4.4.2 Process Modeling Federate

Simphony.NET 4.0, a stand-alone Discrete Event Simulation application, was utilized to create a process model (Moghani et al. 2011). Simphony.NET 4.0, as a flexible modeling tool that can be easily integrated into other simulation environments, was chosen to be part of the framework. To be able to record the as-built construction process and replay it in the future, the simulation model should contain many details of a construction project. Also, additional services must be available in Simphony to allow Simphony model to join the federation as a component of the entire system. A generic element called "RTI

Connection" is created in Simphony and can be added to the model. This element connects a model to the COSYE RTI and controls joining and execution of the model, as well as sending and receiving updates through RTI. The process model development and its integration into a COSYE environment is described in detail in Chapter 5.

4.4.3 As-Built Federate (Simulation Controller)

The as-built federate is one of the major components of the framework. This federate provides updated information from the construction site to the Simphony federates. The as-built federate connects to the Microsoft Access database to read the construction process information on a daily basis. The information includes the progress of an activity for a certain work area in the site and the related data regarding interruptions, working hours, equipment breakdowns, crew size, material usage, and the list of equipment employed for that specific activity. As it is shown in Figure 4-12, the as-built federate user interface demonstrates daily reports and traces the detail of the simulation process. In Chapter 6, the communication of the as-built federate with other federates and sending updates to the federation are described in more details.

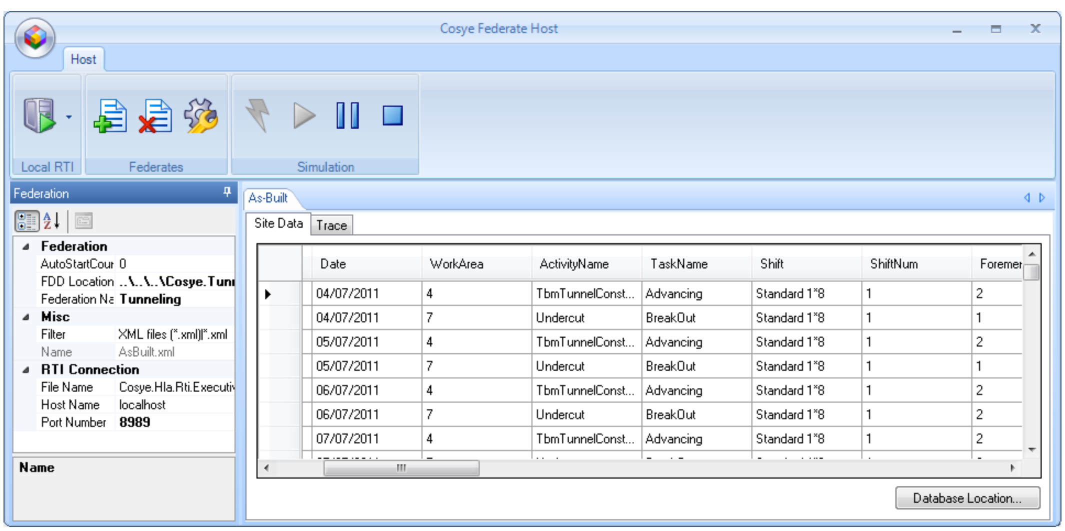
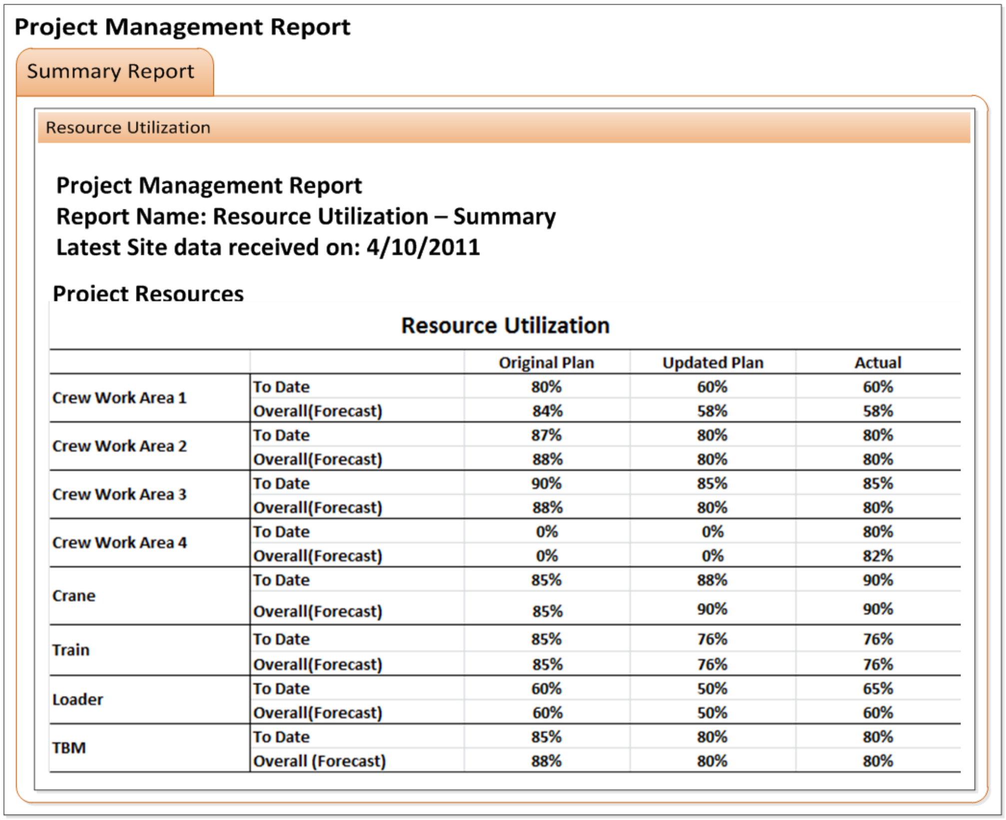


Figure 4-12: As-Built Federate User Interface

4.4.4 Report Federate

The objective of this federate is to provide simulation results to a user. During a simulation run, outputs from all joined Simphony federates in the federation will be sent to the report federate. This federate utilizes the Crystal Report application to generate reports. The application can be bounded to the Microsoft Visual Studio as a reporting tool, and since the federation is developed in Visual Studio 2010, the report generator application integrates nicely into the system. Using Crystal Reports, a developer can benefit from the graphical design capability of the tool.

At the end of each day, in the simulation, outputs are published in the report federate from cost, project progress, resource utilization, and daily working hours. Reports are designed to show the latest state of the project on a daily basis, and also the projected status regarding project performance, cost, schedule, resources, and change order reports in both a summary and in detailed reports for each section. Since information is updated during the simulation, a user can easily access the project results by opening reports during the simulation run. It also allows saving and printing of the reports. Figure 4-13 and 4-14 demonstrate two examples of the reports generated in the application. The process of creating and updating reports during the simulation is described in Chapter 6.



Project Management Report

Summary Report

Resource Utilization

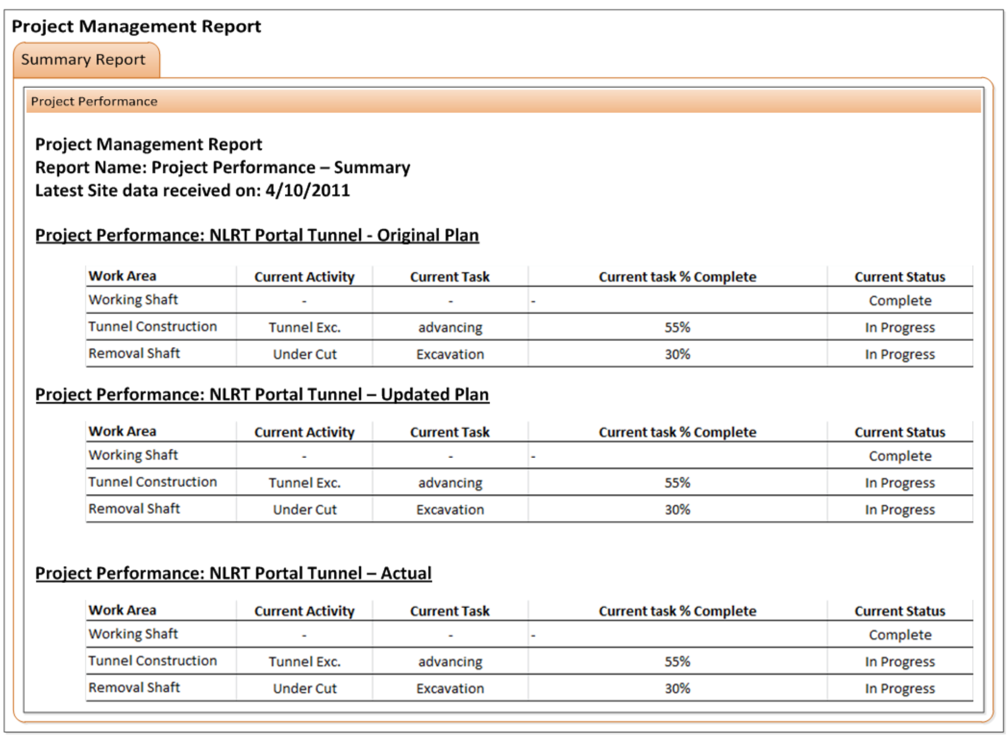
Project Management Report
Report Name: Resource Utilization – Summary
Latest Site data received on: 4/10/2011

Project Resources

Resource Utilization

		Original Plan	Updated Plan	Actual
Crew Work Area 1	To Date	80%	60%	60%
	Overall(Forecast)	84%	58%	58%
Crew Work Area 2	To Date	87%	80%	80%
	Overall(Forecast)	88%	80%	80%
Crew Work Area 3	To Date	90%	85%	85%
	Overall(Forecast)	88%	80%	80%
Crew Work Area 4	To Date	0%	0%	80%
	Overall(Forecast)	0%	0%	82%
Crane	To Date	85%	88%	90%
	Overall(Forecast)	85%	90%	90%
Train	To Date	85%	76%	76%
	Overall(Forecast)	85%	76%	76%
Loader	To Date	60%	50%	65%
	Overall(Forecast)	60%	50%	60%
TBM	To Date	85%	80%	80%
	Overall (Forecast)	88%	80%	80%

Figure 4-13: Report Layout for the Resource Utilization Comparison



Project Management Report

Summary Report

Project Performance

Project Management Report
Report Name: Project Performance – Summary
Latest Site data received on: 4/10/2011

Project Performance: NLRT Portal Tunnel - Original Plan

Work Area	Current Activity	Current Task	Current task % Complete	Current Status
Working Shaft	-	-	-	Complete
Tunnel Construction	Tunnel Exc.	advancing	55%	In Progress
Removal Shaft	Under Cut	Excavation	30%	In Progress

Project Performance: NLRT Portal Tunnel – Updated Plan

Work Area	Current Activity	Current Task	Current task % Complete	Current Status
Working Shaft	-	-	-	Complete
Tunnel Construction	Tunnel Exc.	advancing	55%	In Progress
Removal Shaft	Under Cut	Excavation	30%	In Progress

Project Performance: NLRT Portal Tunnel – Actual

Work Area	Current Activity	Current Task	Current task % Complete	Current Status
Working Shaft	-	-	-	Complete
Tunnel Construction	Tunnel Exc.	advancing	55%	In Progress
Removal Shaft	Under Cut	Excavation	30%	In Progress

Figure 4-14: Report Layout for the Project Performance Comparison

4.5 Support Federates

Different support federates are designed to join the federation in order to enhance the functionality and accuracy of the simulation results for different construction projects. The visualization federate animates the construction process in a 2D and 3D environment. The visualization federate was built using the Microsoft XNA game engine, which is a framework and set of tools that facilitate the creation of animated 3D environments. The 3D models of TBM, train, shaft, and tunnel sections were developed in Autodesk™® 3ds Max™® 2010 and then converted to .x files, which can be loaded by the Microsoft XNA game engine. The animation and simulation work concurrently in the application; the TBM's movement is controlled by the current chainage, provided by the simulation during run-time. A TBM and crane breakdown can be initiated from either the simulation federate or the visualization federate and it will reflect in the other one immediately. The federate helps developers to validate the simulation model, and assists managers to monitor and control the project during the construction stage when the simulation is running based on actual data (Zhang et al. 2012). Figure 4-15 depicts the visualization federate in the tunnel construction federation.

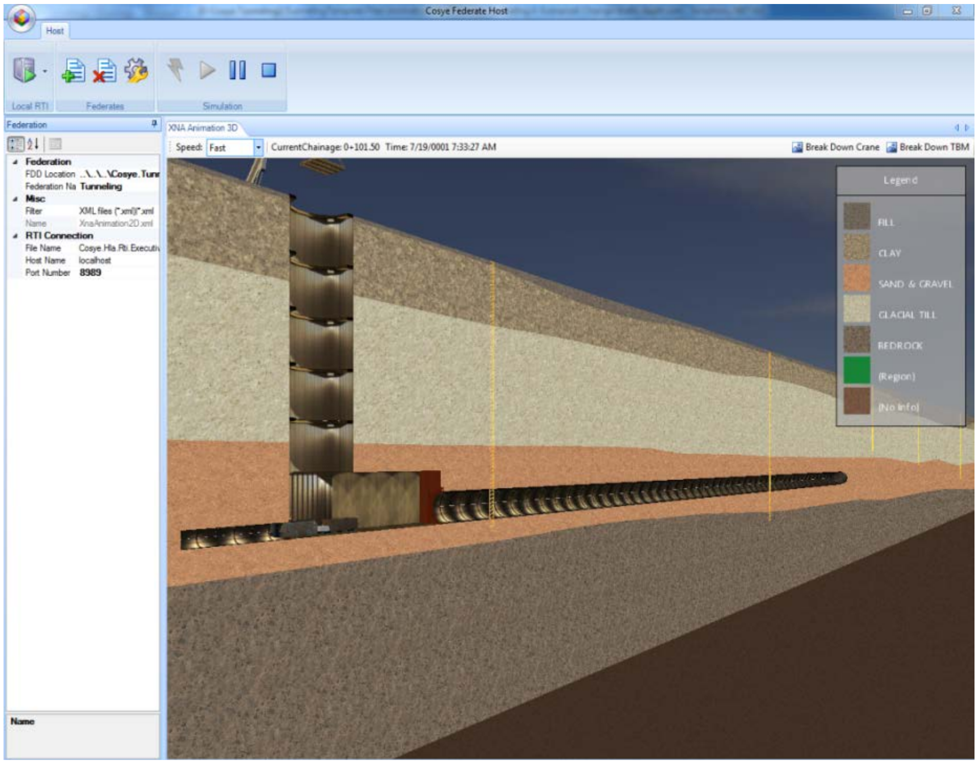


Figure 4-15: Visualization Federate in the Tunnel Construction Federation

As simulation advances, a map federate shows the TBM position on a street map provided by Google map™ and Google Earth™ application programming interfaces (APIs). This federate provides different types of maps such as normal, satellite, and 3D maps (Zhang et al. 2012). Figure 4-16 illustrates a map federate in the tunnel construction federation.

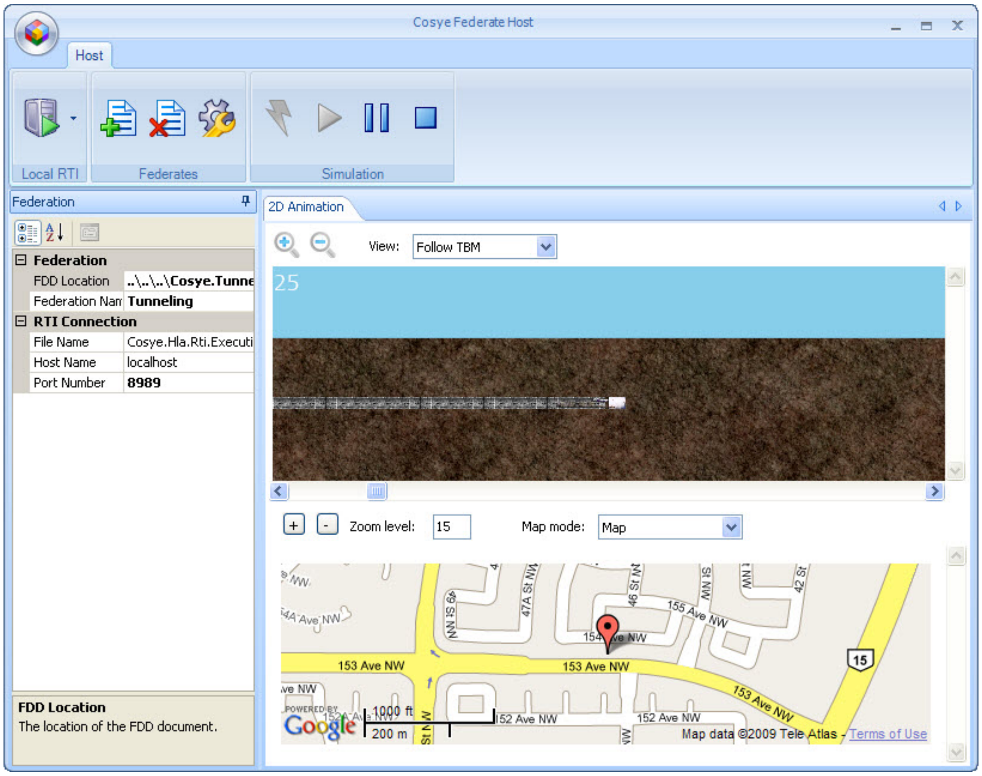


Figure 4-16: Map Federate in Tunnel Construction Federation

Supplier federate is a Special Purpose Simulation template developed in Simphony. Five different elements were created to model a supply chain process and logistic services required by a construction project. A federate is created as a generic federate that can join federation and support the material supply of any construction project. In this federation, the supply federate listens to the material request from Simphony federates and responds by sending interactions through the RTI to the Simphony federates. Two of the elements, namely the order element and the transport element, have the functionality required to communicate with the construction process model. Figure 5-17 shows a supply federate in the tunnel federation.

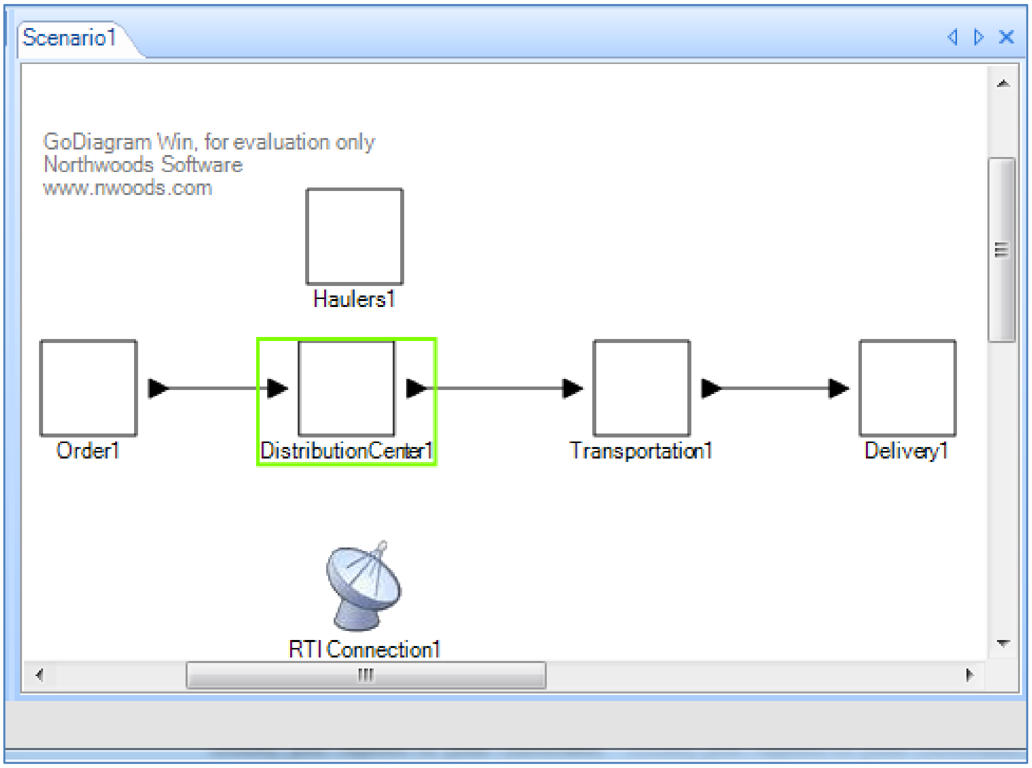


Figure 4-17: Supply Federate in the Tunnel Construction Federation

4.6 Tunnel Construction Database

In the tunneling simulation model, there are many different kinds of inputs; shaft and tunnel geometry such as diameter, depth, and length is obtained from CAD drawings, geotechnical information is extracted from borehole documents, the unit cost of equipment, material, crew, and overhead costs are estimated using historical data, duration of various tasks (e.g. surveying, assembling liners, TBM installation), or data such as train speed, and TBM advance rate are entered as probability distributions based on previous experiences. During construction phase and for documentation of construction process and product information, a database with multiple tables was created to store necessary information required for tunnel construction simulation. For the simulation inputs, especially during the design phase, the data stored in a database is tunnel coordinates and borehole data. For as-built information, a database records the actual data received from the site including site information, activities progress, and surveying data. Figure 4-18 provides a number of screenshots of the database.

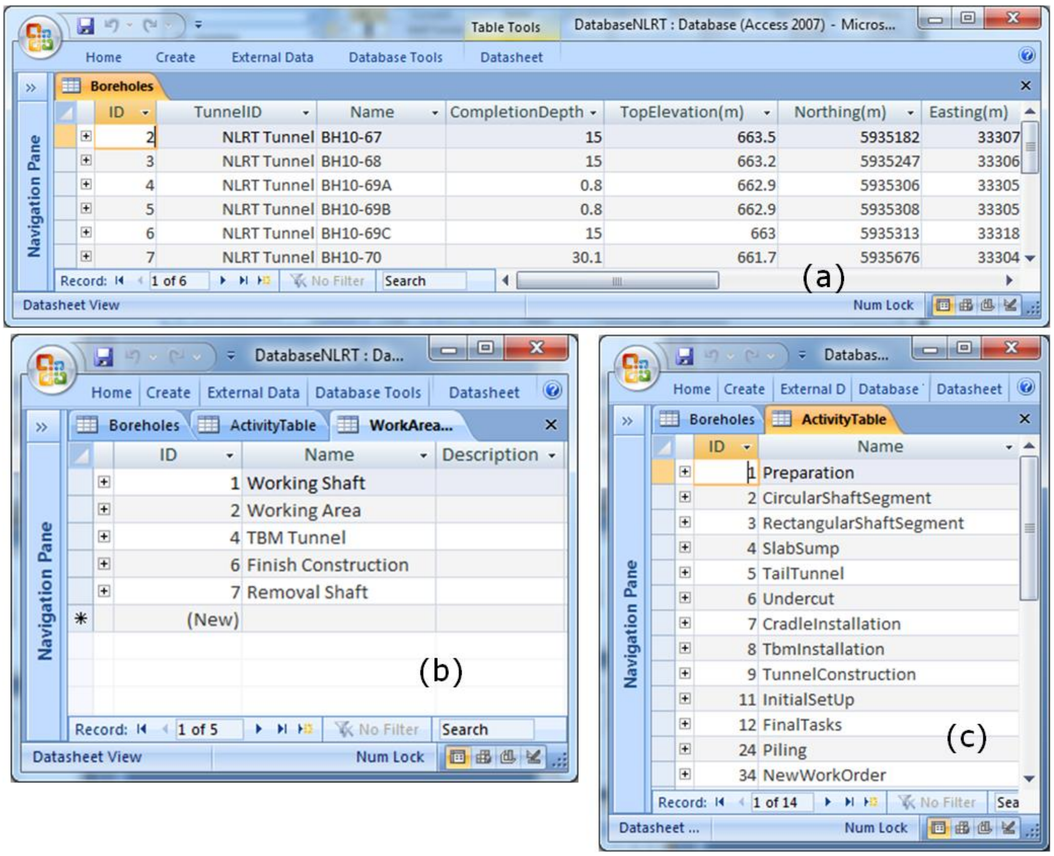


Figure 4-18: (a) Borehole Data Table, (b) Work Area Table, and (c) Activity Table

4.6.1 Updating Simulation Model Based on As-Built Information

In this research, the process of updating the simulation model is divided into two main steps; in the first step, it is assumed that no changes have happened to the simulation process, which means that the plan and the actual construction processes are similar. In this case, changes should apply to the simulation inputs to reflect the real progress and project state at any given time during the project. The update process happens on a daily basis; an as-built federate checks the arrival of data for each day of the project, processes the raw data, transforms it into meaningful inputs for the simulation federate, and sends updates to the Simphony model through the RTI. To implement the updating procedure, this research proposes a new hybrid simulation approach that uses Discrete Event Simulation (DES) and Time Step Simulation (TTS) methods to regulate and control process model updates. This process will continue until the last entry in the database; henceforth, simulation will continue to use the original values for the inputs.

The next step is updating the simulation process based on new work and scope changes that have happened in the project. The proposed solution is that if any changes occur in the activity list, a new simulation model will be added to the framework with a new process model. In this case, a user needs to change the original model based on the change order and adds it as a new federate to the federation. The new as-built process model will be updated based on the site data until the project ends, or to the point that as-built data ends in the database. This part of the research is described in more detail in Chapter 6.

4.7 Conclusion

In this chapter, the general structure of the proposed framework is defined based on the objectives set in this research. In the first step, a conceptual model is created to determine how the framework should operate to fulfill all the requirements of the research, and then the framework structure and its components are specified. In the next stage, the architecture of the framework in the distributed simulation environment is demonstrated, and the requirements for developing the proposed framework in the form of COSYE federation are presented through a tunnel construction case study. In this case study, all the federates that are required for the as-built documentation and all the supporting federates in the federation are introduced. In the next chapter, the integration of

Simphony into the federation is presented. In Chapter 6, the details of communication between federates and the process of updating Simphony inputs and the construction process model are covered.

Chapter 5 - Design, Development and Integration of a Discrete Event Simulation Model into a COSYE Federation [1]

5.1 Introduction

In this research, the proposed framework for the documentation of construction information employs COSYE, the HLA-based simulation environment, to overcome the limitation of stand-alone simulation application, and to benefit from modularity, reusability, and extensibility features of a distributed simulation system in order to create a generic federation for industry application. Since the main federate of this framework is the process model, Simphony, a Discrete Event Simulation tool, was selected to model the construction process and to integrate into the COSYE federation. By this approach, any construction process can be modeled in Simphony and attached to the proposed federation to create a tool for documentation purposes during the construction phase.

This chapter is divided into three sections. The first section describes why Simphony was selected as part of the proposed framework to simulate a construction process model, then, a generic solution is presented to integrate Simphony into COSYE. This section demonstrates how the integration of Simphony in the federation architecture can significantly facilitate the development and use of the proposed framework for different construction projects, especially for demonstrating scope changes in the process model (Moghani et al. 2011a). In the second section, the development of an SPS template for the purpose of project documentation is presented. To demonstrate the development process,

[1] Part of this chapter has been previously published (Moghani et al. 2011a – See Appendix B; Moghani et al. 2011b- See Appendix C).

a tunnel construction project simulation was selected as an example for the template development. Different case applications are presented in this section; the 137 Avenue tunnel project (described in Chapter 3) is reviewed to specify the inputs and parameters that need to be considered in the simulation template in order to document and replay the as-built process more accurately. The existing tunnel template for TBM tunneling method was modified based on this case study. The second case study is the drainage tunnel project which was used to verify and validate the template. And the Edmonton North Light Rail Transit (NLRT) project was used to extend the application of the tunnel template to include the Sequential Excavation Method (SEM) with either a ribs and lagging or shotcrete lining system (Moghani et al. 2011b). Finally, this chapter presents the integration of Simphony into COSYE through a tunnel construction simulation case study. The process of studying a construction project and developing a Simphony template for as-built documentation is similar for different types of projects. Once the template is developed, the integration process into COSYE federation will be the same. Therefore, the entire process presented in this chapter can be generalized for other construction projects.

5.2 Integration of Simphony Model into COSYE

One of the advanced implementations in COSYE is the tunnel construction simulation (Xie and Moghani 2009). The first version of the federation consists of several federates (Figure 5-1).

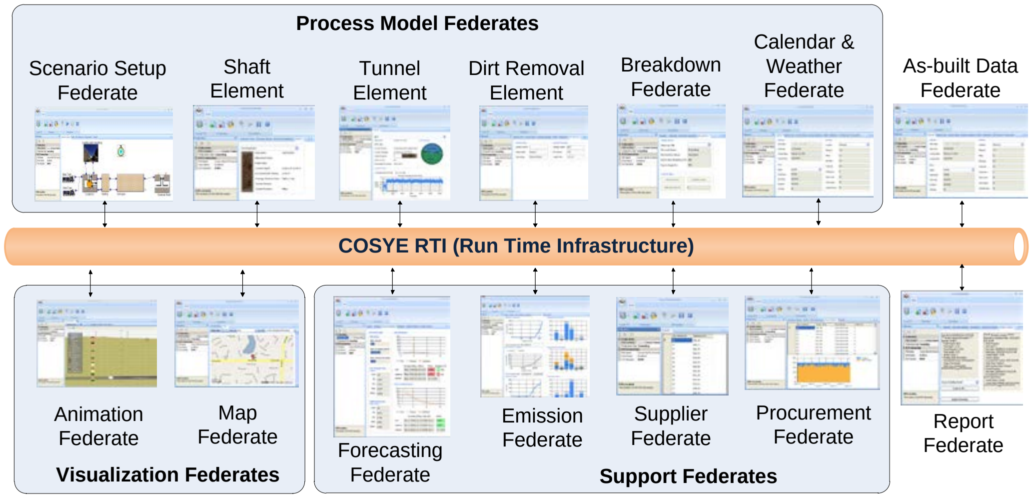


Figure 5-1: Tunnel Federation Structure in COSYE

In the first version, six different federates were developed to demonstrate the entire process of a City of Edmonton utility tunnel construction project. Shaft construction, tunnel excavation, dirt removal process, equipment breakdowns, weather, calendar, and scenario set up were each modeled separately, and communicated through the RTI in the federation.

In the distributed simulation models in COSYE, each federate is developed as a Visual Studio class coded in C# or Visual Basic and the user interface does not allow users to modify or change codes or logic of the simulation. This places restriction on the application of a COSYE federation in industry since programmers would have to be hired to revise and update the process as the project progresses. This could also limit application of the proposed framework for different construction projects since it requires development of new federates by writing new programming codes and then updating them for as-built documentation.

Early studies show that Simphony is a proven application for process modeling, and its new features facilitate integration into other simulation environments. Since the main concern for the framework was creating a reusable application, it was decided to integrate

Simphony as one federate to replace process model federates in the federation and represent the construction process model in one Simphony model (with COSYE-compatible simulation elements) and use its visual capabilities to update the construction process instead of writing new codes for every changes happen in the process. This would facilitate more efficient development of the framework and create a COSYE-aware Simphony model to be a part of a collaborative and comprehensive model.

As previously mentioned, Simphony.NET 4.0 was developed particularly to be used in an integrated environment; new features have been added to Simphony to facilitate communication between COSYE and Simphony models. A Simphony modeling element called “RTI connection” was created as a generic element to be added to all Simphony models to enable them to join federation. The responsibility of this element is to provide a connection to the RTI, manage join and execution of a Simphony model, and to send and receive messages through RTI and time management in the Simphony federate. Every federate in a federation has two objects called RTI Ambassador and Federate Ambassador which send and receive messages from the RTI. These two objects are encapsulated in the RTI connection element to allow communication between Simphony and other federates, through the RTI.

For elements to become COSYE-aware, they need to utilize the services provided by the RTI connection element to communicate with the RTI. Figure 5-2 demonstrates the connection of the Simphony model through RTI Connection to the COSYE RTI.

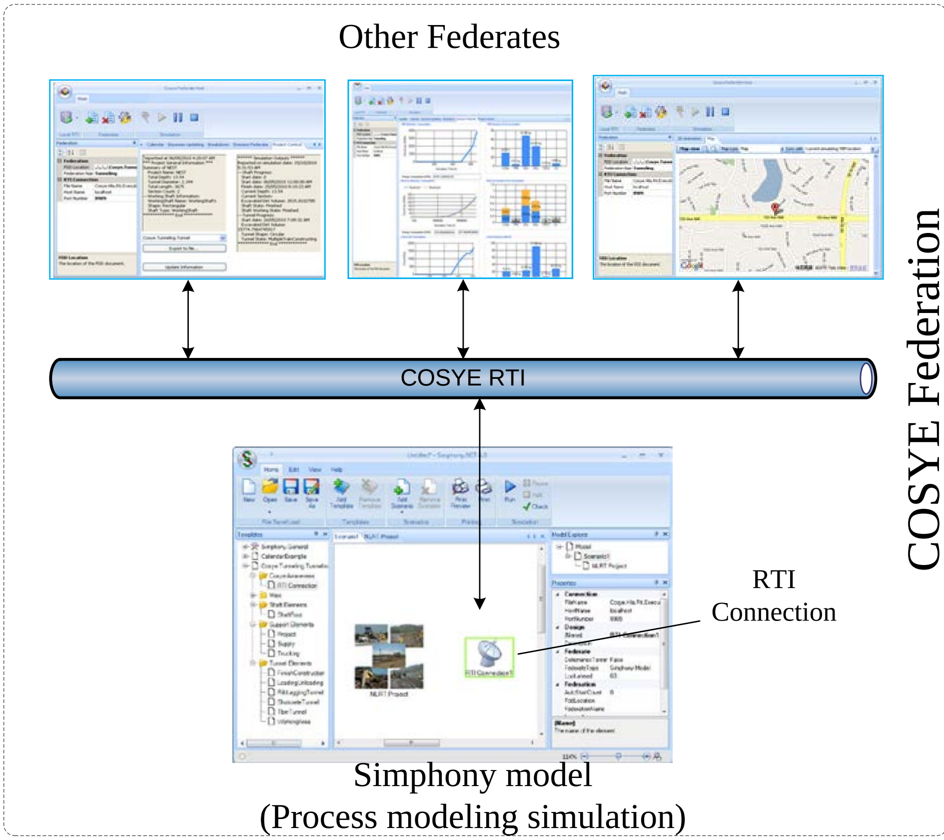


Figure 5-2: Integration of Simphony Model into COSYE using RTI Connection Element

As a test model, a tunnel construction simulation template was created in Simphony 4.0 and its elements were modified to send and receive updates from different federates. These elements are known as COSYE-aware elements in the template, and by adding RTI connection to a Simphony model, the elements will start communicating with other federates in the tunnel federation.

5.3 Development of a Simphony Template to Document the As-Built Construction Process

To develop a simulation model for a specific domain in construction, a good knowledge of the domain is required. A user needs to gather general project information from available sources such as project planning documents, expert knowledge, and historical data from previous similar projects. The simulation model must process specific levels of detail to be considered acceptable by the industry, and provide valuable outputs in order to be used for decision-making purposes (Hajjar and AbouRizk 2002). This is even more critical if the simulation model is intended to document as-built history of a project; it may require more detail than the simulation models that are utilized in other areas, such as productivity analysis, cost estimation, and risk analysis. This chapter provides a guideline through a case study to determine the type of simulation inputs and the level of detail that requires for accurate as-built documentation.

In this section, as a case study, the development of a Special Purpose Simulation template for as-built documentation in tunnel construction projects is discussed. The analysis of the collected site data for the 137 Avenue tunnel project (previously discussed in Chapter 3) defines the basic requirements that the template should have in order to be successfully employed, both for decision making during the planning phase, and documentation of the project during the construction phase. This case study specifies the various elements, their input parameters, and their interaction in the model. To develop a template, the first step was to upgrade the existing tunnel template created in an older version of Simphony (Al-Bataineh 2008) to a new version (Simphony 4.0). Then, the findings from this case study were applied to enhance the functionality of the existing TBM tunneling template.

In addition to the TBM tunneling, the template was modified extensively to include more construction methods for tunneling projects, including Sequential Excavation Method (SEM) with a ribs and lagging lining and shotcrete lining as a temporary lining system for tunnel projects (A user Guide for a new Tunnel template is provided in Appendix D). Each modification was based on the real tunnel project constructed by the City of Edmonton.

5.3.1 Tunnel Construction Simulation in Simphony

Tunnel construction projects are one of the many categories of projects that have been the subject of research for the application of simulation for project planning, enhancing productivity, and scenario comparison. AbouRizk and Dozzy (1993) created a General Purpose Simulation model for specific tunnel projects. Touran and Asai (1987) developed a CYCLONE model to control schedule by predicting the advance rate of TBM in soft rock. Ruwanpura (2001) describes the development of a Special Purpose Simulation model for tunnel construction using Tunnel Boring Machine (TBM); his Simphony template was developed to optimize equipment use by predicting soil profile along the tunnel to reduce the cost and duration of the project. Likhitruangsilp and Ioannou (2003) presented a stochastic methodology for evaluating tunnelling performance by using discrete event simulation; they use STROBOSCOPE to provide probability distributions for tunnel advance rate and tunnelling unit cost for all possible alternatives to determine optimal excavation and support methods for the project. Chung (2007) proposed a simulation-based productivity model for utility tunnel construction operations. He implemented Bayesian updating application in the simulation to update project schedule and predict tunnel productivity in the TBM tunnelling method. Al-Bataineh (2008) developed a scenario-based planning template in Simphony.NET for TBM tunnelling, employing discrete event simulation technique and applying the concept of distributed simulation to create modular elements. His model considers various project information including cost, schedule, weather effects, material supply, and dirt handling to select the best scenario. Xie et al. (2011) developed a progress monitoring system for TBM tunneling to capture progress data and provide real-time project progress input into the simulation and revise the input values of the simulation model.

The focus of the aforementioned studies was the development of simulation models (or templates) for utility tunnel construction projects being undertaken in Edmonton, Alberta, Canada, where, based on geotechnical conditions and availability of equipment, a Tunnel Boring Machine (TBM) is the most common tunnel construction method, specifically for the long tunnels. Hand tunneling or Sequential Excavation Method (SEM) is used for short tunnels with a smaller diameter. In SEM method, either ribs and lagging or shotcrete are used as primary linings, and cast-in-place concrete lining or precast pipes with grouting are considered as permanent linings.

5.3.2 Desaign and Development of the Tunneling Template for the As-Built Documentation

In this section, the data collected for the project performance analysis during the PIR process for the 137 Avenue tunnel project is reviewed for the design of the TBM tunnel construction template in Simphony for the purpose of project documentation. The template is intended to be utilized during both planning and construction phases, specifically for the documentation of the project. Since the existing tunnel template in the previous version of Simphony was designed for productivity analysis and scenario-based planning, the same template is converted to Simphony 4.0, and the additional features are added based on the findings from this study.

5.3.2.1 Overview of the TBM Tunneling Method

The selection of the appropriate method depends on several factors such as ground condition, tunnel length, availability of materials and equipment, local experience, and surface limitation. In the City of Edmonton, based on the ground condition and availability of the Tunnel Boring Machines (TBMs), this method has been used in many tunnel construction projects, especially for utility tunnels in urban areas.

The tunnel construction process starts by excavating a working shaft to reach the tunnel depth. The working shaft provides access to the tunnel during the construction phase for lowering down the material and removing dirt from the tunnel. Excavation and lining of the working shaft usually happens in different segments based on existing soil layers. Excavation of each segment can be done both by machine and hand depending on the segment depth and geotechnical condition. A retrieval shaft is usually constructed at the end of a tunnel for TBM removal. A crane is located at the top of the working shaft and retrieval shaft for lifting the loads. A tail tunnel and undercut, expanded areas at the beginning of the tunnel, are constructed using hand excavation and ribs and lagging for a lining to facilitate TBM positioning and material handling during the operation.

Tunnel construction starts after lowering and installation of the TBM. In tunnel construction with TBMs, the boring cycle includes TBM excavation and advancement, installation of concrete segments, and resetting the TBM for the next cycle. In each cycle, TBM advancement is equal to the concrete liners' length. The excavated muck for each cycle is transferred from a tunnel face through a conveyer to the muck carts. A train consists of two or more muck carts and a material cart for moving out the dirt and providing liner segments to the TBM. This process is repeated until the end of the tunnel excavation length. Other activities such as utility extension or surveying are also done in certain intervals; during these activities, the tunnel process stops to give enough space to perform the task.

Based on the construction method, the working shaft may be constructed in the middle of the tunnel. A two-way tunnel starts either by utilizing two TBMs to excavate both tunnels simultaneously, or by excavating and lining one tunnel and then removing the TBM from the removal shaft and reinstalling it for the second tunnel operation.

5.3.2.2 Site Data Evaluation for 137 Avenue Tunnel Project

The 137 Ave. Tunnel project is a two-way tunnel project with one rectangular shaft (working pit) in the middle. Because of the large size of the working pit, there was no need to construct undercut and tail tunnel. For this project, one TBM machine was utilized to construct both west and east tunnels. Several manholes and connections were also constructed in this project, which will not be discussed in this section.

Chapter 3 mentions that the daily site data and different documents were reviewed to record the actual process of the project. In addition to that, the as-built drawings, along with the surveyor's report, were collected to document final product information. This experience provided valuable information on how the simulation model should be developed, what kind of activities and in what level of detail should be included, and what type of information relating to resources, materials, and external factors should be considered to make simulation a useful application for the as-built documentation of projects. Ailland et al. (2010) specified two aspects of projects that need to be captured and described in the process simulation; the first is the data related to the process, such as durations, work flow, dependencies, and resources utilized during the construction process. The second feature includes the project current progress and other information such as material inventory, site condition, and human resources.

In this section, the data analysis for the 137 Ave. tunnel project is presented to determine the components and parameters of the simulation model.

1) Work Packages, Activities and Tasks

In the construction project, all project activities are divided into work packages, usually by area or project component. This concept is created to integrate construction schedule, cost estimation, risk analysis, project forecasting, material supply, safety, quality, permitting, etc. (COAA 2007). Within each work package, different activities can be defined; these activities are usually specified in the project schedule and they provide information in a higher level, which may not be helpful for detail documentation of a project. Chin et al. (2005) noted that to prepare a daily site report for project control, the project should be documented at the task level, which has more detail than activities, and can be managed on a daily basis.

The same concept was used to document daily activities in the 137 Ave. tunnel project. Figure 5-3 illustrates the work packages, activities, and tasks in the project. If the simulation model follows the same structure, it will align with other construction documents and provide sufficient details for managing and recording the construction process.

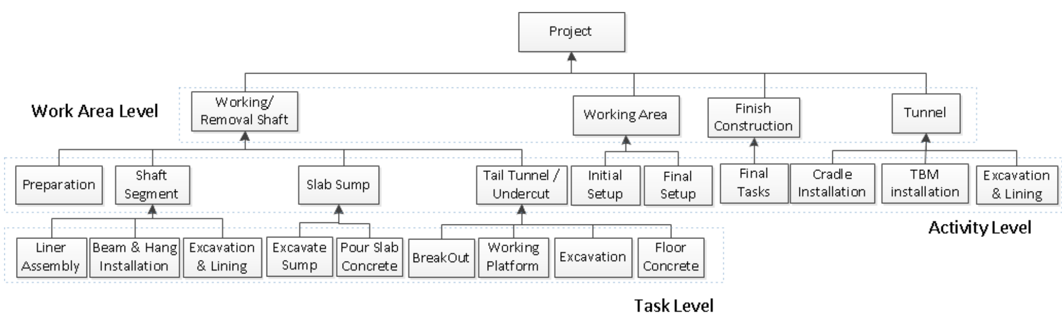


Figure 5-3: Work Packages, Activities and Tasks in TBM Tunnel Projects

2) Construction Resources and Materials

As shown in Figure 5-4, the major project cost is allocated to the construction crew, equipment, and material. So in this section, various data that needs to be collected and considered in a simulation model in each category is defined. Also, the type of equipment and crew that need to be considered in the model is specified. The project cost can be analyzed in other projects to determine what type of resources and materials need to be considered in the simulation model.

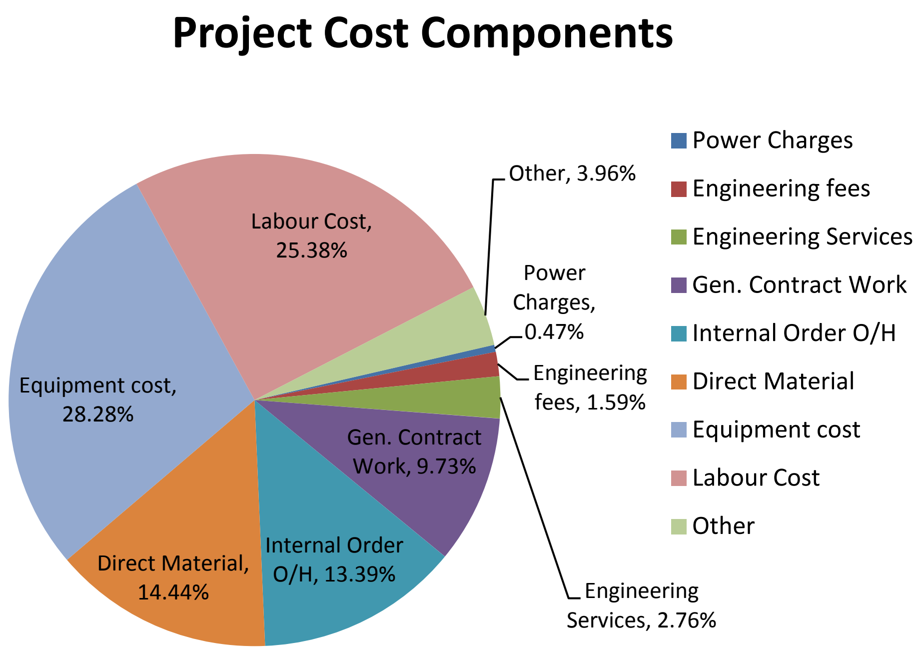


Figure 5-4: Cost Components for the 137 Ave. Tunnel Project

As mentioned before, in the base plan of the project, to achieve the expected completion date, three crews were assigned: tunnelling crew, shaft crew, and drilling crew. Based on the project information, construction crews were allocated at the work area level and the following information was collected for each crew on a daily basis:

- Related work area.
- Size.
- Working hours (working shifts) per day.
- Skill.

The main material in TBM tunneling is the concrete segments used for tunnel lining. In the 137 Avenue tunnel project, concrete segments cost around 60% of the total material cost. So, the liner segments were considered in the tunnel simulation and a single unit cost for all other materials was added to each of the work packages. The other data that was collected was the liner supply information such as delivery dates and the order sizes.

In this project, the cost of the TBM and cranes is around 58% of the total project equipment cost, and cost associated with trucks, loaders, and excavators is about 21% of the total (Figure 5-5). Based on this information, the TBM tunnel construction simulation template includes TBM, Crane, Truck, Loader, and Train (for the dirt removal process inside the tunnel) as resources in the process model.

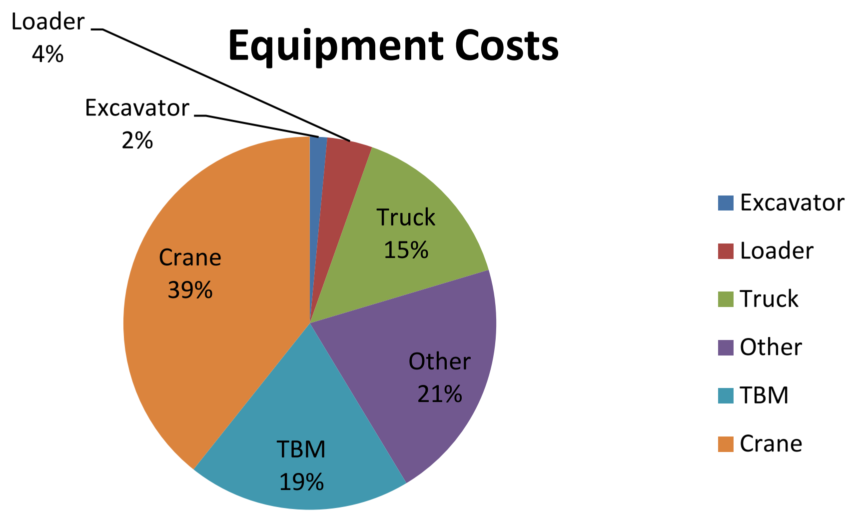


Figure 5-5: Equipment Cost for the 137 Ave. Tunnel Project

In addition to the equipment cost, daily site records include the following information:

- Related work area.
- Advance rate of TBM on a daily basis and for each shift.
- TBM breakdowns, which were not recorded properly.
- Crane daily working hours.
- Crane breakdowns, which were not recorded completely.
- The number of trucks and their capacity.

3) Interruption Causes

The interruptions in the process were recorded on a daily basis in the following categories:

- Equipment breakdowns.
- Weather conditions.
- Extending track, utility, gantry, and conveyor (they are part of the process but the excavation process stops until the tasks are completed).
- Surveying.
- Delays in material supply.
- Preparation and extra activities during the project.

Interruptions were documented separately for each work area.

4) Calendar, Daily Working Hours and Shifts

All working days and non-working days were defined in the project calendar, created in Microsoft Project. During the construction phase, working and non-working days were documented in the daily site records and used to update the project calendar. The working hours and shifts were also recorded on a daily basis for each working area.

5.3.2.3 Overview of the TBM Tunnelling Template in Simphony 4.0

Al-Bataineh (2008) divided the elements into three groups: shaft operation elements (shaft template), tunnel operation elements (tunnel template), and support elements (Figure 5-6). Calendar, project, trucking, supply, and shift control elements are considered as supporting elements in the template and can be reusable for other construction projects. “Project” element reports different cost items (crew, material, equipment, dirt removal, and indirect), total cost, start date, and finish date of the project.

“Shift control” element can be added to the shaft model and tunnel model separately to reflect shift differences in different work packages in the project. The “Supply” element controls supply of liners and TBM parts for the project and “Trucking” deals with dirt removal in the site.

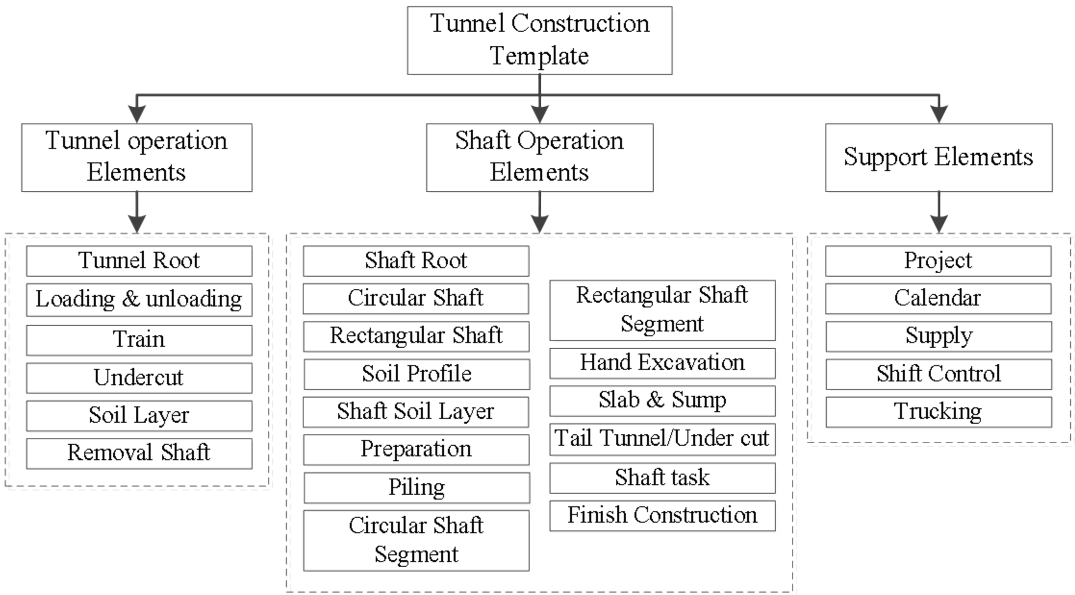
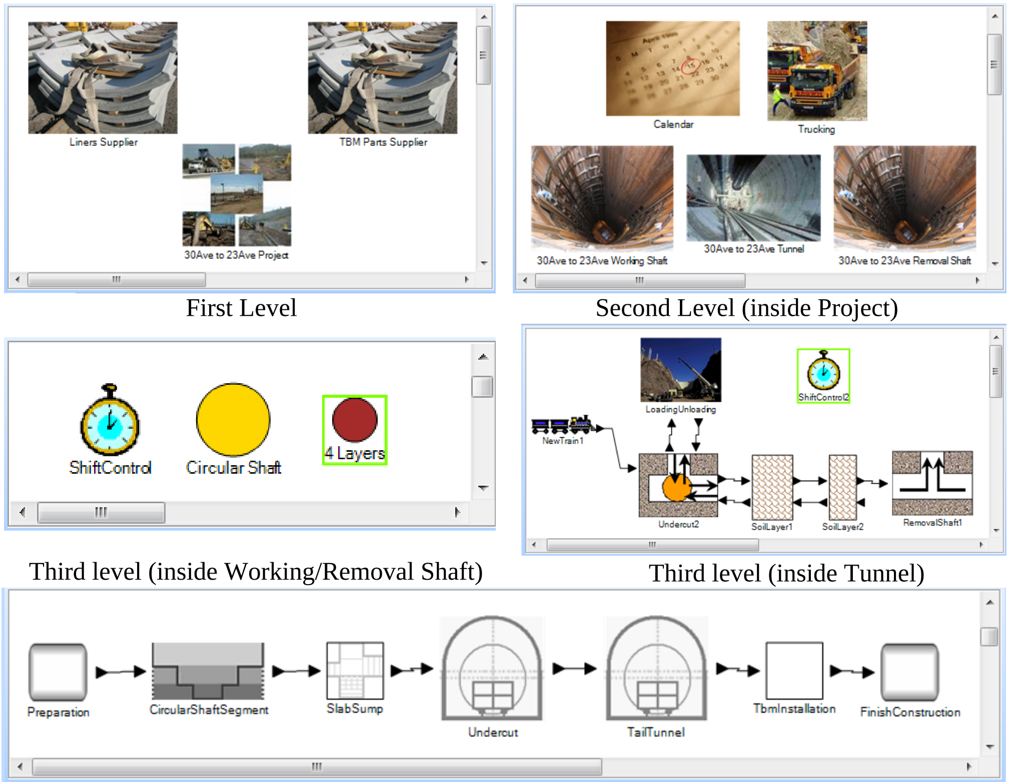


Figure 5-6: TBM Tunnel Construction Elements in the Old Version of the Template

In his tunneling template, the shaft template consists of fourteen elements; all the tasks in the shaft construction process are encapsulated in these elements. So, each element may contain more than one task scheduled as an event in the internal code. The shaft model takes into account the soil layers for the shaft excavation.

The tunnel template consists of six elements that simulate loading and unloading activities on the top of the shaft, different geotechnical tunnel sections along the tunnel, TBM removal, surveying and extending utilities as internal events, and the activities inside the undercut such as track installation. The entire template has a modular and hierarchical structure; each element is located in specific level, encapsulates certain tasks, and communicates with other elements on the same, higher, and lower levels. An example of a tunnel construction model is illustrated in Figure 5-7.



Forth level (inside Circular Shaft)

Figure 5-7: Hierarchical Tunnel Simulation Model in the Old Version of the Template

5.3.2.4 The Enhancement of the TBM Tunnel Template

Based on the data evaluation for 137 Avenue tunnel project, the following improvements were proposed and implemented for the existing tunnel template:

- Number of Elements was reduced to the major work areas in the TBM tunnel construction, including: Shaft (working shaft, removal shaft, and access shaft), Loading Unloading, Work Area (represents undercut and tail tunnel), Tunnel and Finish Construction Elements (Figure 5-8). The Project, Supply, and Trucking elements are the support elements in the template.

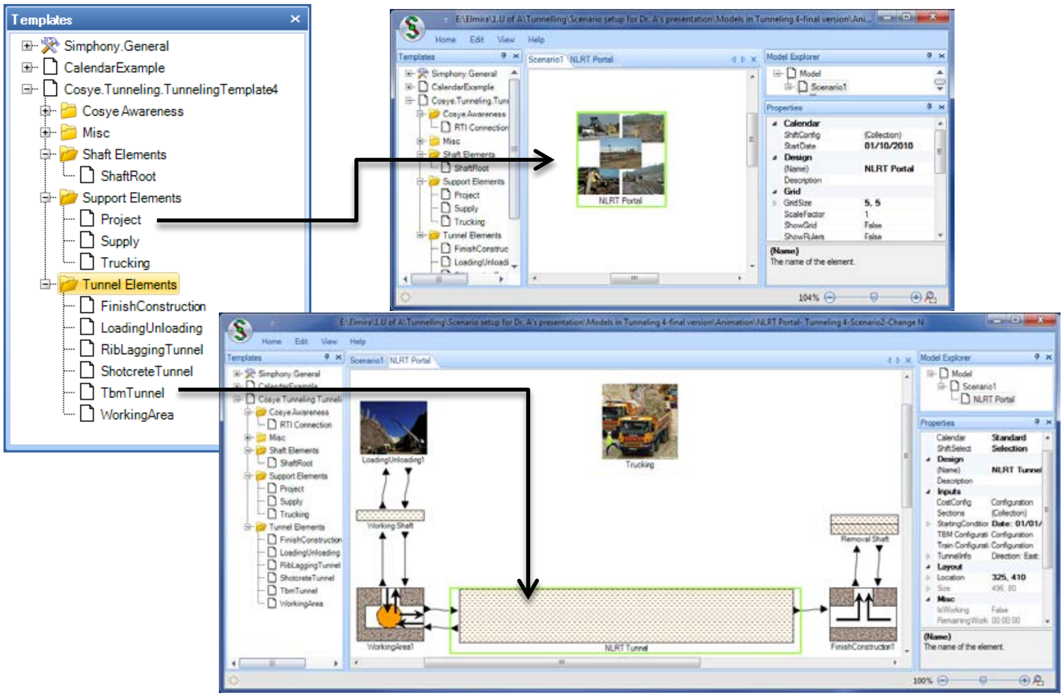


Figure 5-8: New Structure for the TBM Tunnel Template

- In the new version, one element represents shaft construction, soil profile, and all the construction activities and tasks for the shaft construction. An input parameter called "Tasks" is added to the element property grid that enables the user to select a list of activities from a predefined list of shaft construction activities. This increases the flexibility of the shaft element to model any new activities added to the list. Figure 5-9 depicts shaft element and its properties.

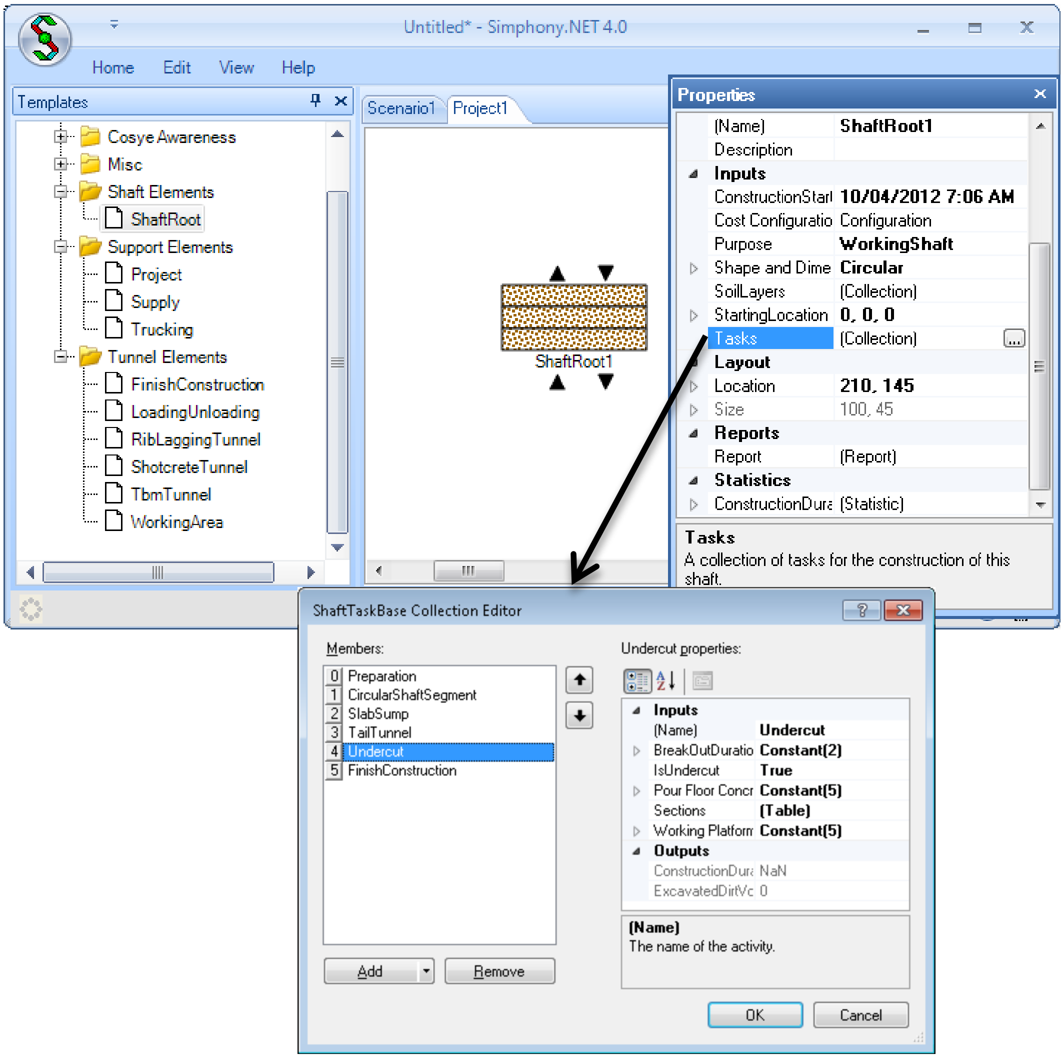


Figure 5-9: Shaft Construction Activities in the Tunneling Template

- To be able to model working hours in the project, a calendar and shift properties are added to the project, shaft, and tunnel elements. In the project element, the user can specify a collection of calendars for the project. A calendar specifies shifts, working hours, working days, and non-working days in the project. Also, a user can add exceptional days to reflect the changes to the working hours for specific days. This parameter is designed to replicate the Microsoft Project calendar; the functionality and even the user interface are similar in order to make it easier for the industry to use, as they are usually familiar with Microsoft

Project. Since working hours and shifts can vary in different work areas (e.g. the Tunnel and Shafts may have different working hours or shifts), a user is able to select a calendar separately for shafts and tunnel elements from the collection of project calendars. The other elements usually have the same working time as either shaft or tunnel. For instance, crane in Loading Unloading element, should work during shaft or tunnel excavation. So Loading Unloading element would work with shaft calendar during shaft construction and with tunnel calendar during the tunnel excavation. The Working Area and Finish Construction elements work based on the tunnel calendar. Figure 5-10 presents the calendar and shift collection in the template.

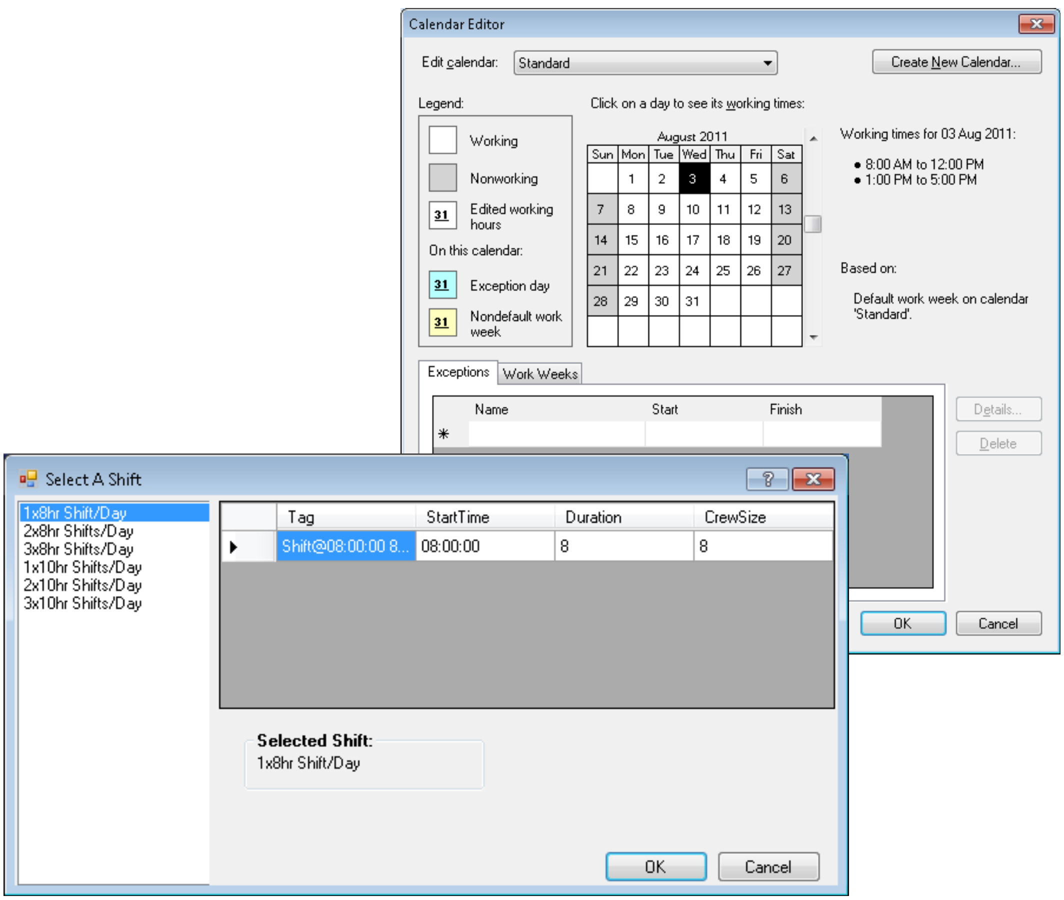


Figure 5-10: Calendar and Shift Collection in the Tunnel Template

- For each work area, a crew size is added in the shift configuration; this helps to assign shifts to the crews to enable calculation of crew utilization for the day and night shift crews separately.
- A start date property is added to the project, shaft, and tunnel elements to control the start date condition for each element. In this case, a user can consider any delays to the start of each work area, if required.
- As it is shown in Figure 5-11, a cost configuration is considered for each work area to provide cost information for each work area individually.

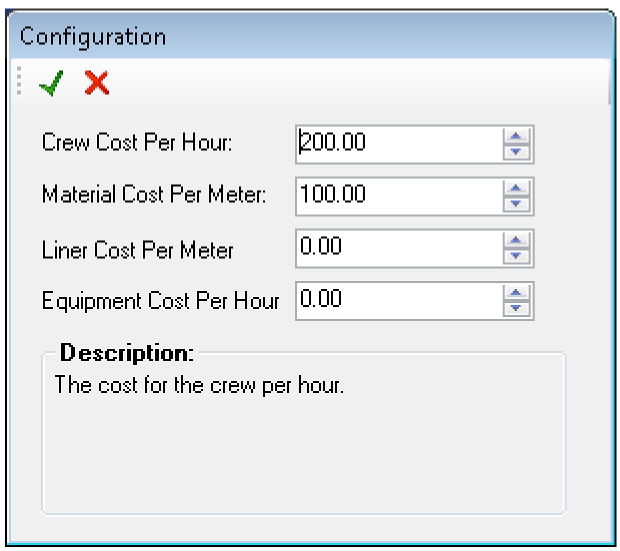


Figure 5-11: Cost Configuration for the Tunnel Element

- For each work area (element), a report is produced to present the start and finish date, and the construction cost, in total and in detail for that area. Figure 5-12 illustrates a project and tunnel report produced by the template.

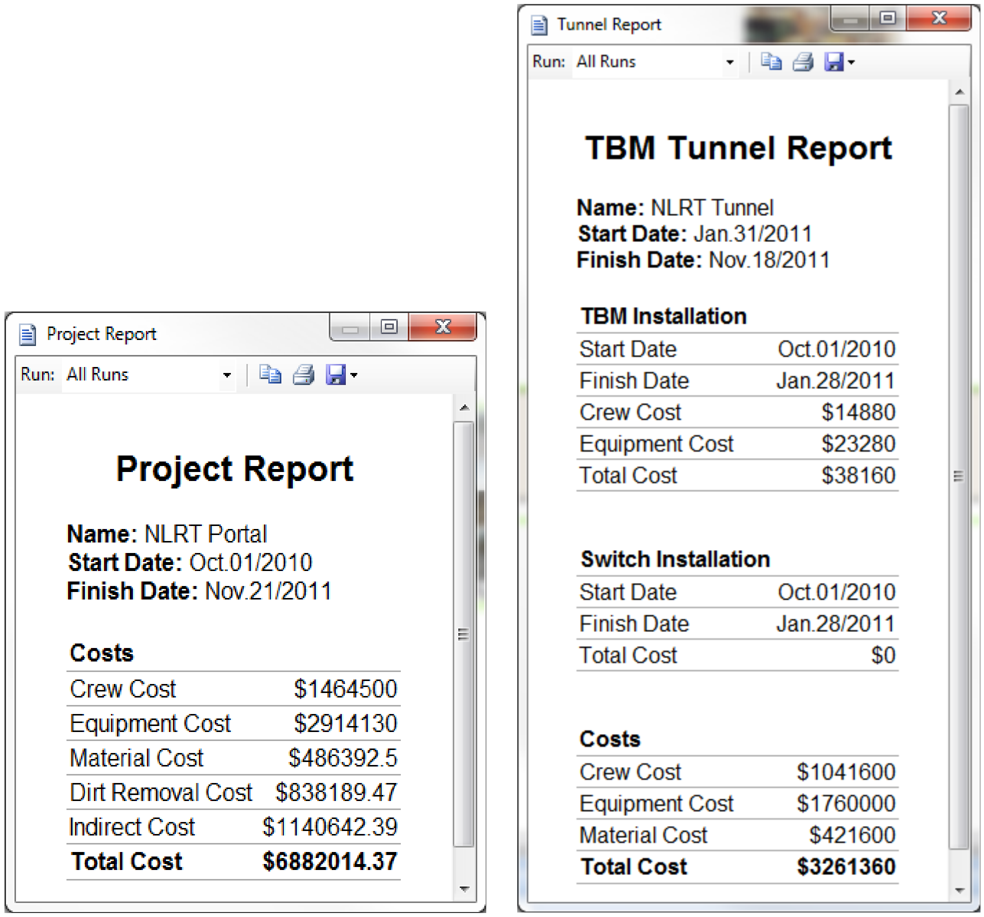


Figure 5-12: Report Examples Generated by the Tunnel Template

In addition to the changes to elements and their properties, the process model was also updated. New activities and tasks are added to the shaft, work area, and finish construction elements, and a tunnel process is slightly modified to match the real process in the site. All the changes are implemented in the new version of the TBM template to make it a suitable application for project documentation. To validate the new template, a TBM tunnel project in City of Edmonton is selected as a case study in this research.

The case study is a drainage tunnel for the underground section of the Edmonton North Light Rail Transit (NLRT) extension project located in Edmonton, Alberta, Canada. This tunnel is a 500 m tunnel with an inside diameter of 2340 mm. A one-way TBM tunneling method was selected as the construction method in this project; a TBM with a diameter of

2540 mm was utilized for the excavation and lining process, and a working shaft and retrieval shaft were constructed at two ends of the tunnel. A Simphony model was developed (Figure 5-13) to calculate the productivity, cost, and total duration of the project and compare them with actual values in the project.

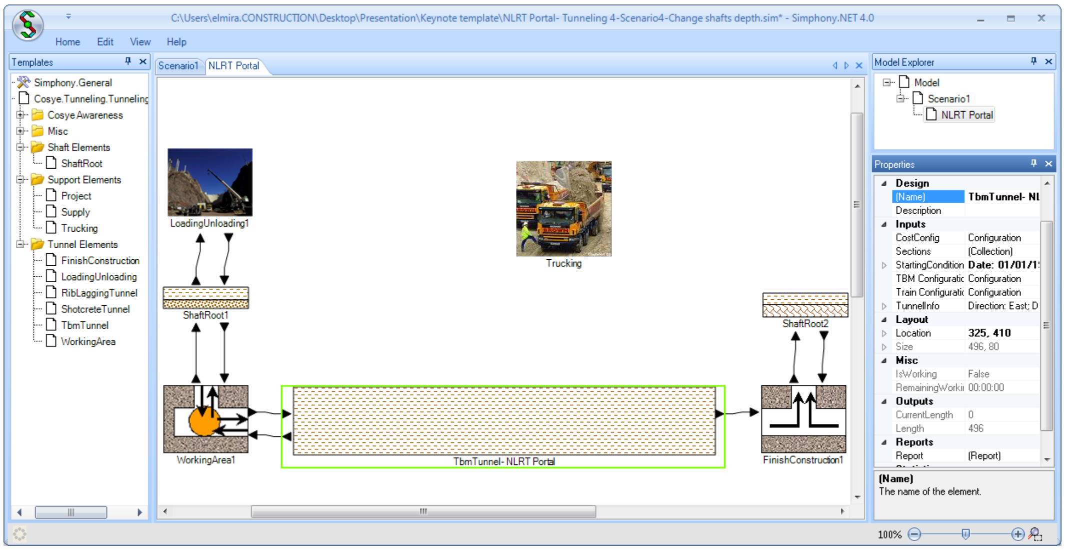


Figure 5-13: NLRT Portal Drainage Simulation Model in Simphony 4.0

In addition, different simulation outputs, in the form of reports are used to compare simulation results with the actual values.

5.3.3 Development of the Tunnel Template for the Sequential Excavation Method (SEM)

The North LRT (Light Rail Transit) Tunnel project was studied to develop new elements in the tunnel template to model Sequential Excavation Method (SEM) for tunnel construction (Moghani et al. 2011 b). The original study was conducted to select a construction strategy and the best scenario for the tunnel construction project in Edmonton, Alberta, Canada that choose from shotcrete and ribs and lagging lining systems. This project is a part of the LRT extension including two parallel tunnels with a total length of 764 m, extending from the existing underground station to the street level station (MacEwan station). The construction will start from tunnel portals located at MacEwan station and will be continued up to the pre-built section. The remaining sections will be constructed from the end of the pre-built section to Churchill station.

Sequential Excavation Method (SEM) was selected as the excavation method in both construction strategies under study. In the sequential excavation process, a cross section is divided in smaller segments based on tunnel size; these segments are excavated in sequence for one meter advancement of the tunnel. In this project, the cross section is divided in heading and bench sections with approximately similar areas, as shown in Figure 5-14(a). An excavator, a loader, and a truck are considered as a set of equipment for excavation and mucking processes. Two assumptions were made for the construction sequence:

- Excavation and lining of two one-meter heading sections is followed by the excavation and lining of a two-meter bench section. This process is illustrated in Figure5-14(b).
- Excavation and lining of a heading is followed by the excavation and lining of a bench for a specific length of the tunnel.

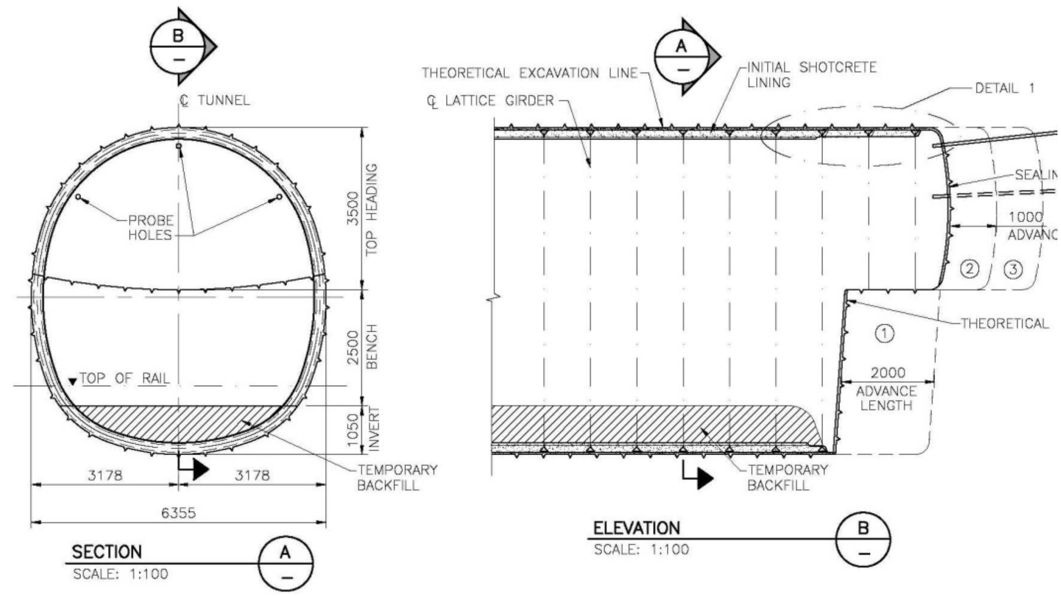


Figure 5-14: Tunnel Cross Section & Excavation Sequence (IFL Consultants, Inc., 2008)

5.3.3.1 Sequential Excavation with Rib and Lagging Lining System

In the tunnelling method which uses a rib and lagging lining system, the entire tunnel is lined using the same materials and lining arrangements. The steel ribs are installed every one meter at the end of excavated sections. Laggings, or wooden pieces, are inserted between ribs and soil to support the excavated area. Ribs usually come in one or two sections and are bolted together when they are fixed in their location. A timber spreader and steel rods are used to fix the position of ribs. Based on the tunnel alignment, surveying will be performed every 6-15 m, and utilities are installed as the tunnel advances.

5.3.3.2 Sequential Excavation with Shotcrete Lining System

In the shotcrete lining method, the preliminary lining differs along the tunnel based on geotechnical condition. The designers for this considered a standard support system for both tunnels which includes installation of different layers of shotcrete and wire mesh. In this project three layers of shotcrete with two layers of wire mesh in between are specified as a standard support.

In addition to the standard support, designers defined five types of additional supports based on soil conditions, and then divided tunnels into longitudinal sections based on their needs for additional supports. All the sections with their additional supports are listed in Table 5-1.

Table 5-1: Tunnel Longitudinal Sections based on Additional Supports for Shotcrete Lining

NLRT - North bound					NLRT - South bound				
Section	Start point	End Point	Length	Extra Supports*	Section	Start point	End Point	Length	Extra Supports*
North1	768	758	10	E	South1	755.5	745.5	10	B+E
North2	758	651	107	-	South2	745.5	629	116.5	E
North3	651	642	9	A	South3	629	619	10	A+E
North4	642	475	167	Pre-Built	South4	619	466.9	152.1	Pre-Built
North5	475	466	9	B+C	South5	466.9	457.9	9	B+C
North6	466	328	138	C	South6	457.9	423	34.9	C
North7	328	256	72	-	South 7	423	335.6	87.4	C+E
North8	256	236.6	19.4	A	South8	335.6	325.3	10.3	C
					South9	325.3	223	102.3	-
					South10	223	203.58	19.42	A

*All the sections have the standard support

*All the sections have the standard support

Since the lengths of each section and the installation durations for the additional supports are the only important factors for simulation purposes, the design characteristic of each support will not be considered in the template.

5.3.3.3 SEM Elements Development

Based on the case study, two tunnel elements are developed to model the SEM method with the shotcrete and ribs and lagging lining systems. The Element defines the excavation sequence for each section, and also the heading to bench advancement ratio. For the tunnel cross section, different tasks can be specified for each section (heading or bench), and various materials can be defined for each lining sequence. As well as TBM tunnel element, material, crew, equipment, and unit costs are also considered for the development of the element. To develop the elements and integrate them in the existing template, changes have been made to create generic work area and finish construction elements that work for all three tunneling methods. Also, the dirt removal simulation was modified to consider trucks as dirt removal means in the tunnel template. Figure 5-15 depicts the properties of ribs and lagging tunnel element.

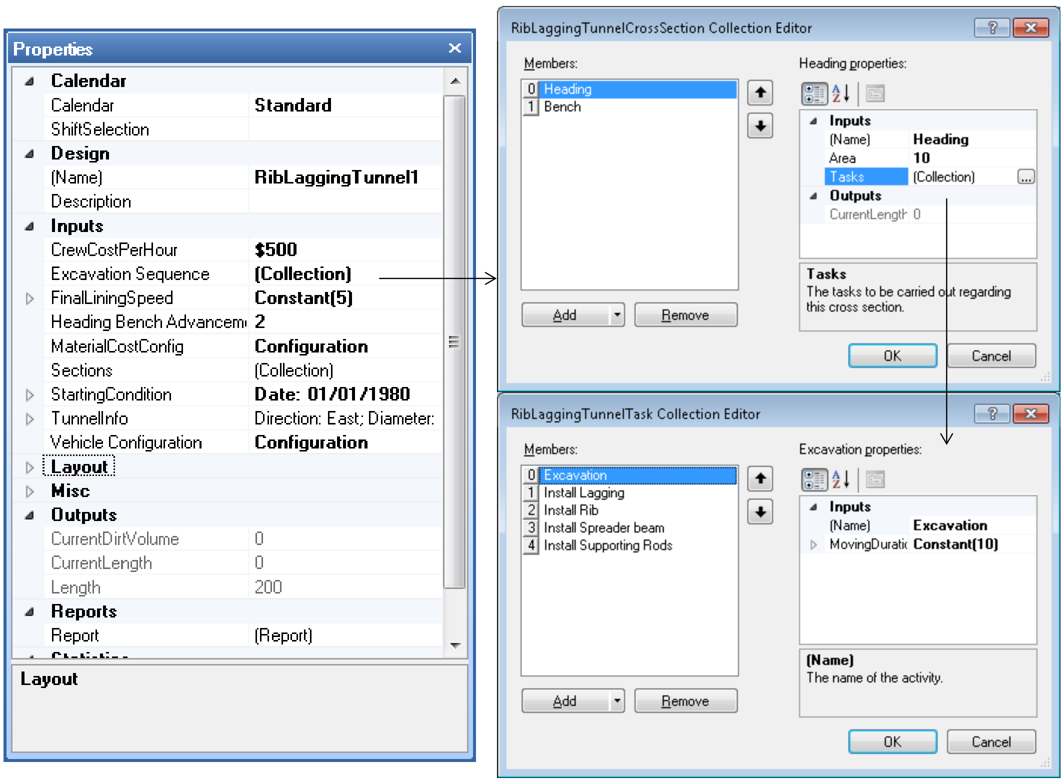


Figure 5-15: Properties of Ribs and Lagging Tunnel Element in Simphony 4.0

The NLRT project is simulated using the tunnel template for the SEM method with both lining systems. Different scenarios are developed within each system and the best option, with a minimum cost and duration, is selected for the construction of the tunnel. Figure 5-16 shows the model used for this case study.

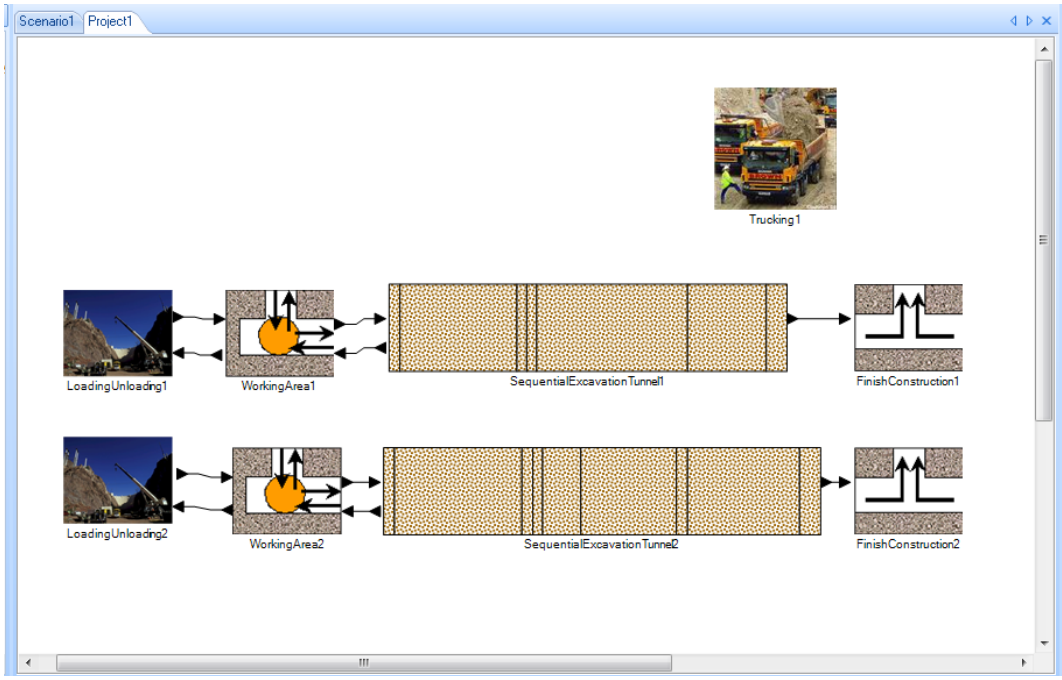


Figure 5-16: NLRT Project Simulation Model in Simphony 4.0

5.4 Integration of Tunneling Template into the COSYE Federation

To integrate a Simphony model into the COSYE federation, four major steps should be taken:

- Create, test, and validate a tunnel template in Simphony.NET 4.0 to simulate the process model.
- Update FOM to consider all the classes and attributes that are related to the process model.
- Transform elements into COSYE-aware elements to enable communication of these elements with other federates in the federation. This can be achieved by changing the codes in Visual Studio class that represent each element.

- Connect the Simphony model to the existing tunnel federation to test the compatibility of the Simphony model with other federates.

The development of the template for the tunnel construction case studies has been demonstrated in the previous section. In the next section, the integration of the tunnel template in the tunnel federation will be discussed.

5.4.1 Updating of the Tunnel Federation Object Model (FOM)

The first step in developing a federation in COSYE is to create a Federation Object Model (FOM) that includes all the object classes, attributes, and interactions that will be required in the federation execution. The FOM classes and attributes should be comprehensive enough to be reusable in the future. As mentioned before, in the first version of the federation, the process model was distributed within different federates that were communicating through RTI. Having the entire process model in one Simphony federate reduces the network traffic, since many communications happen internally and within the Simphony model. Due to this change, the FOM is updated accordingly and all the object classes, interactions, and their attributes that were originally used for communication purposes among the process model templates are removed from the FOM if no other federates are "interested" in them. By revising the old FOM, many existing federates required modification to be able to connect to the federation.

Figure 5-17 illustrates the object classes in the final version of FOM for the tunnel federation and the proposed framework. In the FOM, object classes are defined in seven categories; Process (Construction Activity), Environment, Project, Resources, Scenario, Inventory, and Inventory Item.

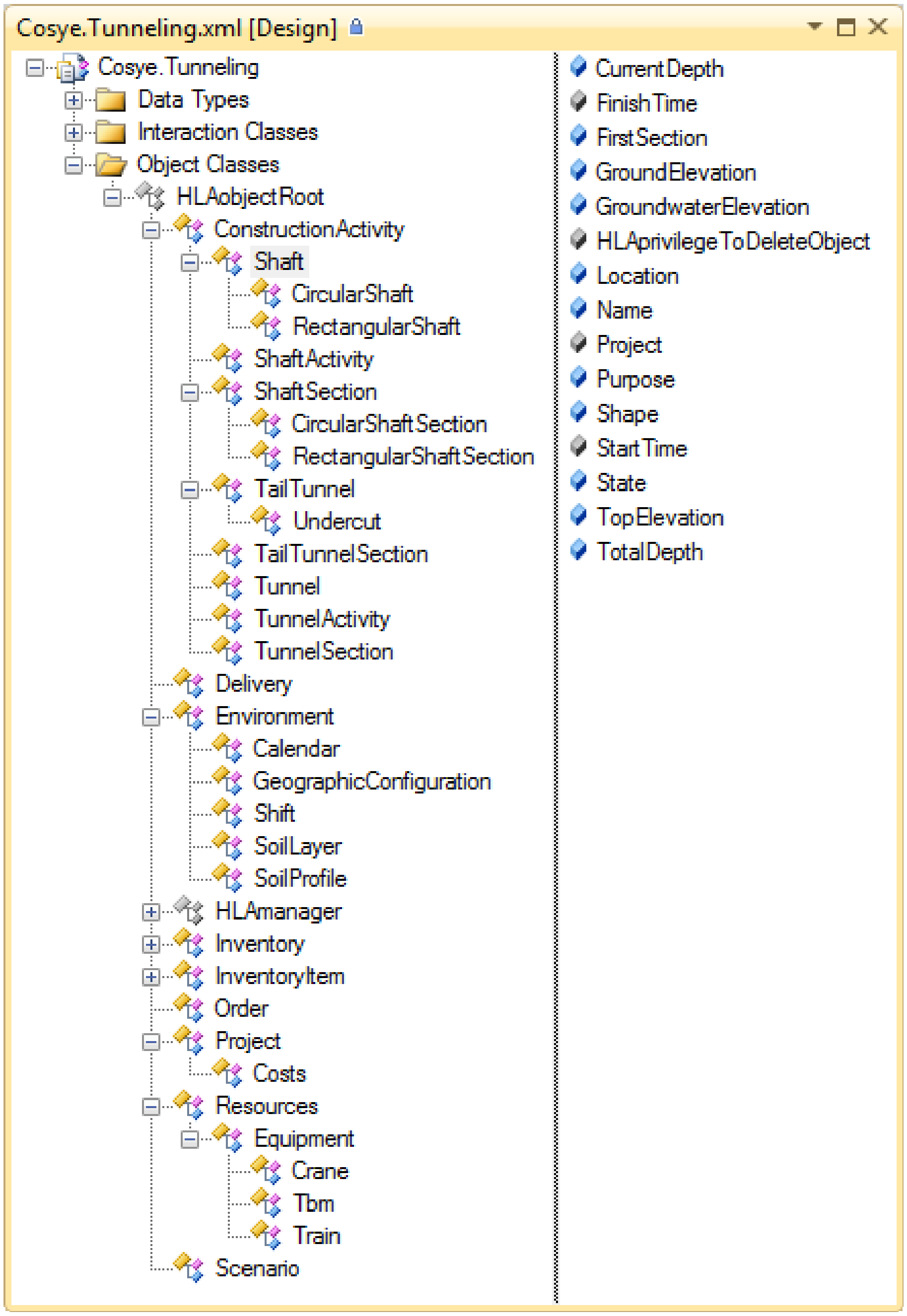


Figure 5-17: Federation Object Model (FOM) for the Tunnel Federation

The Construction Activity Category includes all the objects related to the construction process, and the Inventory and Inventory Items classes include all the object classes related to the Supply federate. Other object classes in the FOM are mainly used for communication between Simphony and the as-built, visualization, and report federates. In addition to object classes, Interaction classes and enumerated data types are also defined in the FOM (Figure 5-18).

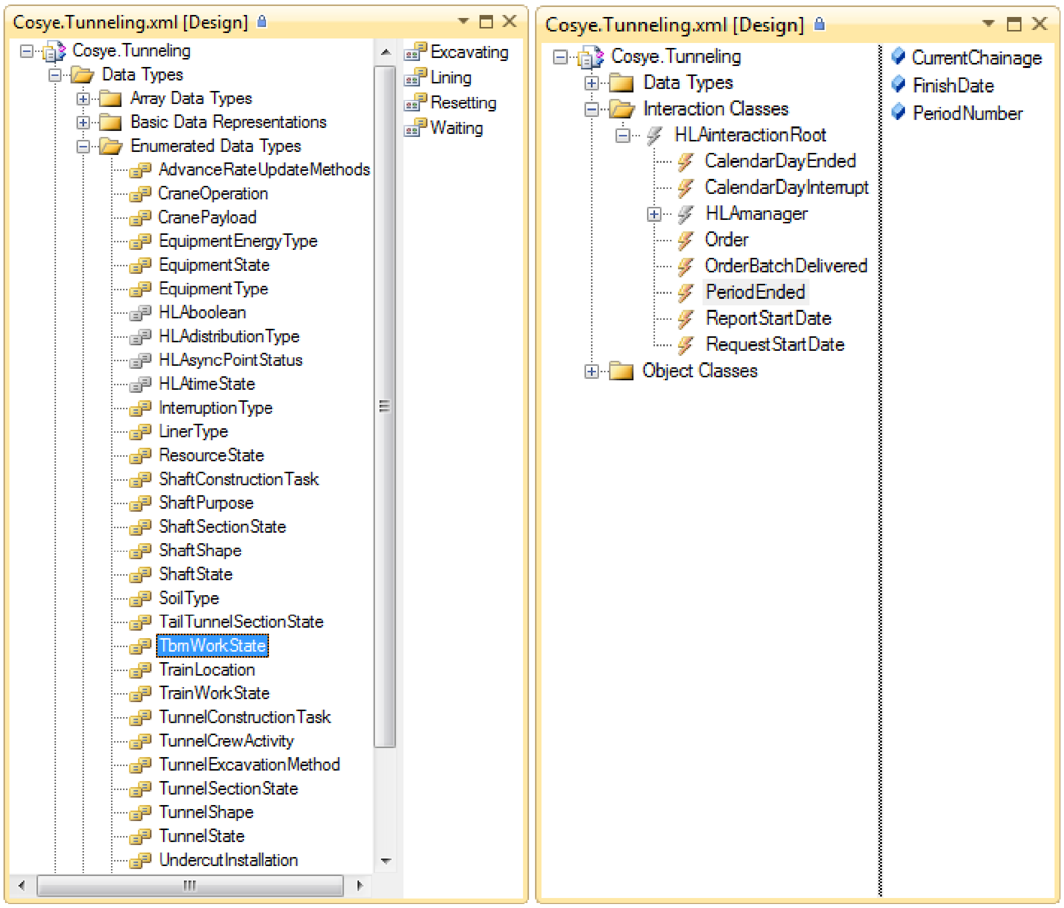


Figure 5-18: (Left) Enumerated Data Type and (Right) Interaction Classes in the Federation

5.4.2 Development of COSYE-Aware Elements in the Tunneling Template

For the tunnel construction template, a generic Visual Studio class called ICOSYEAwareElement was developed as an interface that each element must implement to become a COSYE-aware element. The ICOSYEAwareElement includes all the methods that RTI connection calls such as Start Execution, Initialize Object Instances and Make Declarations. Another generic class called FederationManager, developed in the tunnel template, works as an RTI connection with additional built-in features. Whenever an event occurs in the RTI, the Federation Manager will call all the elements that implement ICOSYEAwareElement interface to receive updates and communicate with other federates. Since, a Simphony template should be designed in such a way that allows users to run a model on its own or as part of a federation execution, and as a single federate, a developer needs to include an indicator to check if Simphony joins a federation. To be able to track it, a Boolean variable is added in the written code of a project element. This variable becomes true if it finds that the RTI connection was added to the model, otherwise it will remain false. This public variable can be accessible from other elements in the model, so each element can check the value of the project, and if it returns true, then all the related codes for the COSYE aware situation will be executed. Figure 5-19 shows an example of code that needs to be added to the element to make it COSYE-Aware.

```
ShaftRoot.cs
Cosye.Tunneling.TunnelingTemplate4.ShaftRoot        LoadingUnloadingInputPointName

#region InitializeRun(int)
/// <summary> ...
protected override void InitializeRun(int runIndex)
{
    base.InitializeRun(runIndex);

    if (this.Project.IsCosyeAware)
    {
        this.fedAmb = this.GetService<IFederateEvents>();
        this.rtiAmb = this.GetService<IRTIambassador>();

        this.fedAmb.DiscoverObjectInstance += this.DiscoverObjectInstance;
        this.fedAmb.FederationSynchronized += this.FederationSynchronized;
        this.fedAmb.ReceiveInteraction += this.ReceiveInteraction;
        this.fedAmb.ReflectAttributeValues += this.ReflectAttributeValues;
    }

    this.calendarEntity = new CalendarEntity();
    this.CrewCost = 0D;
    this.CurrentSoilLayerIndex = 0;
    this.EquipmentCost = 0D;
    this.FinishTime = double.PositiveInfinity;
    this.hasStarted = false;
    this.InnerCalendar = this.Scenario.Calendars[this.InnerCalendarType].Clone();
    this.StartTime = double.PositiveInfinity;

    this.ShiftAwareList.Clear();
    this.SoilLayers.InitializeRun(runIndex);
    this.Tasks.InitializeRun(runIndex);

    Trace.WriteLine(this.Name + ".Subscribed to Calendar");
    this.Engine.SubscribeCalendar(this.calendarEntity, this.InnerCalendar, this.CalendarWorkChange);
}
#endregion
100 %
```

Figure 5-19: Inserted Code to a COSYE-Aware Element

5.4.3 Adding Simphony Federate to the Tunnel Federation

After updating the FOM and adding proper codes to the COSYE-aware elements, a Simphony model can join the federation through the "RTI Connection" element that is added to the model at the project level. Now Simphony can work as a federate and send and receive updates through the RTI.

5.5 Conclusion

The focus of this chapter is the development of a Simphony template that can be utilized as part of the proposed framework for the as-built documentation, especially for the actual construction process. Also, the integration of Simphony as a stand-alone discrete event simulation tool into COSYE federation is described through a tunnel construction case study.

After the integration of Simphony into COSYE, the next phase of the framework development is to update the Simphony model by sending updates from the as-built federate based on actual data stored in a Microsoft Access database. The process of updating the Simphony model is presented in Chapter 6.

Chapter 6 - Implementation of the Framework for As-Built Documentation in Construction Projects

6.1 Introduction

As previously mentioned in Chapter 4, the proposed framework consists of four major elements: a database that stores all the as-built data, an as-built federate that connects to the database and acts as a simulation controller to send data and update the Simphony inputs and process model, a Simphony federate that simulates a construction process, and a reporting federate that provides comparison results for the planned and actual project performance. In this chapter, the processes of reading data from a database and sending it to the RTI by the as-built federate and updating the Simphony model and the report federate based on as-built daily information are discussed. To implement the framework, a TBM tunnel construction project was selected as a case application and then the North LRT Portal Drainage Tunnel was selected as the specific case study to test and validate the framework. As previously described in Chapter 5, a simulation model for this project is created using the TBM tunneling template. Even though a tunneling project is selected for the implementation process, the structure of the database and the updating procedure, from a design and programming perspective, are generic and can be applied for different types of construction projects.

In this Chapter, the process of updating the Simphony model is discussed in five sections; the first section provides a brief overview of Discrete Event Simulation (DES) modeling in Simphony, describes how it works for a case application, and introduces new features of a Simphony template developed specifically for this research. The implementation of a calendar in the Simphony model and its effect on scheduling events and the process of scheduling breakdowns and various interruptions in the process model is discussed using tunnel and shaft construction elements in the TBM tunneling template in Simphony.NET 4.0.

The next section describes the proposed hybrid simulation approach for updating simulation progress. In this section, the processes of reading data from a database and then updating the project state on a daily basis are covered. The third section provides details of the changes that should be made in a Simphony model to enable modeling both plan and as-built process models in terms of activities, equipment breakdown, and scheduling various interruptions. Next, the implementation of construction process changes and adding a new instance of the Simphony model to the federation in order to document changes in the construction process are described. This chapter also provides detailed information on updating resource information, shifts, and project calendars. And finally, the updating process for product information and the implementation of the report federate and the process of information updates in different reports are covered.

6.2 Analyzing Discrete Event Simulation Modeling in Simphony

Simulation technique has become a promising tool for planning and management of construction projects over the last few decades. Early studies show that the majority of simulation applications for modeling construction projects are based on Discrete Event Simulation methodology. “A discrete-event simulation is one in which the state of a model changes at only a discrete, but possibly random, set of simulated time points, called event times” (Schriber and Brunner 2010). In discrete event systems, entities initiate and respond to events and moves from one state to another state during a simulation run. Entities usually capture resources to do activities in the simulation.

Resources are usually capacity-limited, so entities compete for their use and sometimes must wait to use them, experiencing delay as a result.

Simphony.NET is a Special Purpose Simulation (SPS) tool that consists of a simulation engine which is a discrete event processor responsible for scheduling and processing the generated events as well as managing entities, resources, and waiting files. In a basic discrete event engine class in Simphony, various methods and properties are provided that help implementation of a template. In addition to the event processing features, other services such as math library, statistical collection and analysis service, and the random number generator are also provided in Simphony (Hajjar and AbouRizk 2000). Modelling services such as scenarios and various types of modeling for template development are also part of this software. In addition to all common components of discrete event systems, Simphony provides particular features such as calendar and user-defined lists, to make entities calendar aware, that have been added during this research development. In this section, some of the modeling approaches in Simphony, and specifically in the tunnel template, are illustrated to give a clear idea of how the template has been developed using discrete event scheduling.

6.2.1 Scheduling Events in the Tunnel Template

In this section, event scheduling in the tunnel and shaft elements in the tunnel template as a case application is discussed. In the template, shaft construction consists of a series of activities that happen sequentially; based on the list of activities entered in the "Tasks" (a shaft input variable), the program checks through the list and schedules events until the shaft construction completes. It is important to mention that both in the codes and in the input variable, a user enters a list of activities which is incorrectly named as tasks. Each activity consists of one or more tasks; for instance, “CircularShaftSegment” includes liner assembly task, section excavation task, and beam and hang installation task.

Each shaft activity is coded in a separate C# class in the Visual Studio project which represents a template in Simphony, so there are seven different classes that model activities for circular and rectangular shafts, including: preparation, circular shaft segment, rectangular shaft segment, piling, slab and sump, finish construction, and tail tunnel/undercut construction. At the start of shaft construction simulation, a “ShaftRoot” class creates an entity and transfers it into the first activity (class) in the list. In the shaft element, except for the breakdown entities, only one entity flows throughout the entire shaft construction process, so after modeling each activity, the entity is transferred out. Then “TaskBase” class checks for the next activity in the list to transfer the entity to that activity, and this continues until it reaches the “FinishConstruction” activity which determines the end of the shaft construction. In this process, the start of each task depends on the completion of its predecessor tasks and each task can have its own progress rate independent from other tasks in the model. Figure 6-1 shows the process of scheduling events in the shaft element.

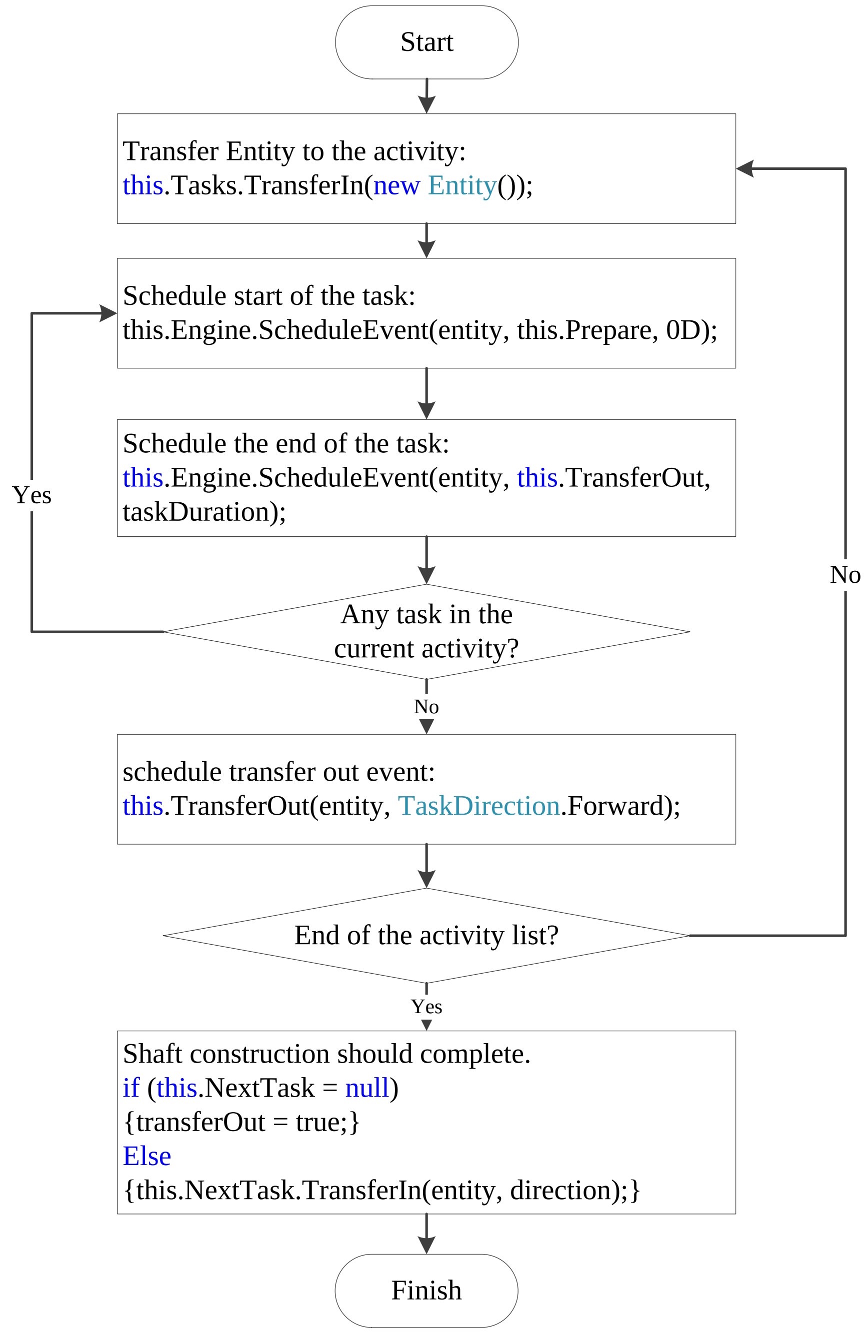


Figure 6-1: Event Scheduling in the Shaft Element with Sequential Activities

On the other hand, in the tunnel element, a set of repetitive activities such as excavation, lining, and TBM resetting, along with the dirt removal activities, occur to construct one tunnel section. In the tunneling process, events can be scheduled concurrently; for instance, while TBM is lining a section, train or truck should remove dirt to the undercut or outside of the tunnel. Figure 6-2 illustrates tunnel excavation process in the TBM tunneling method.

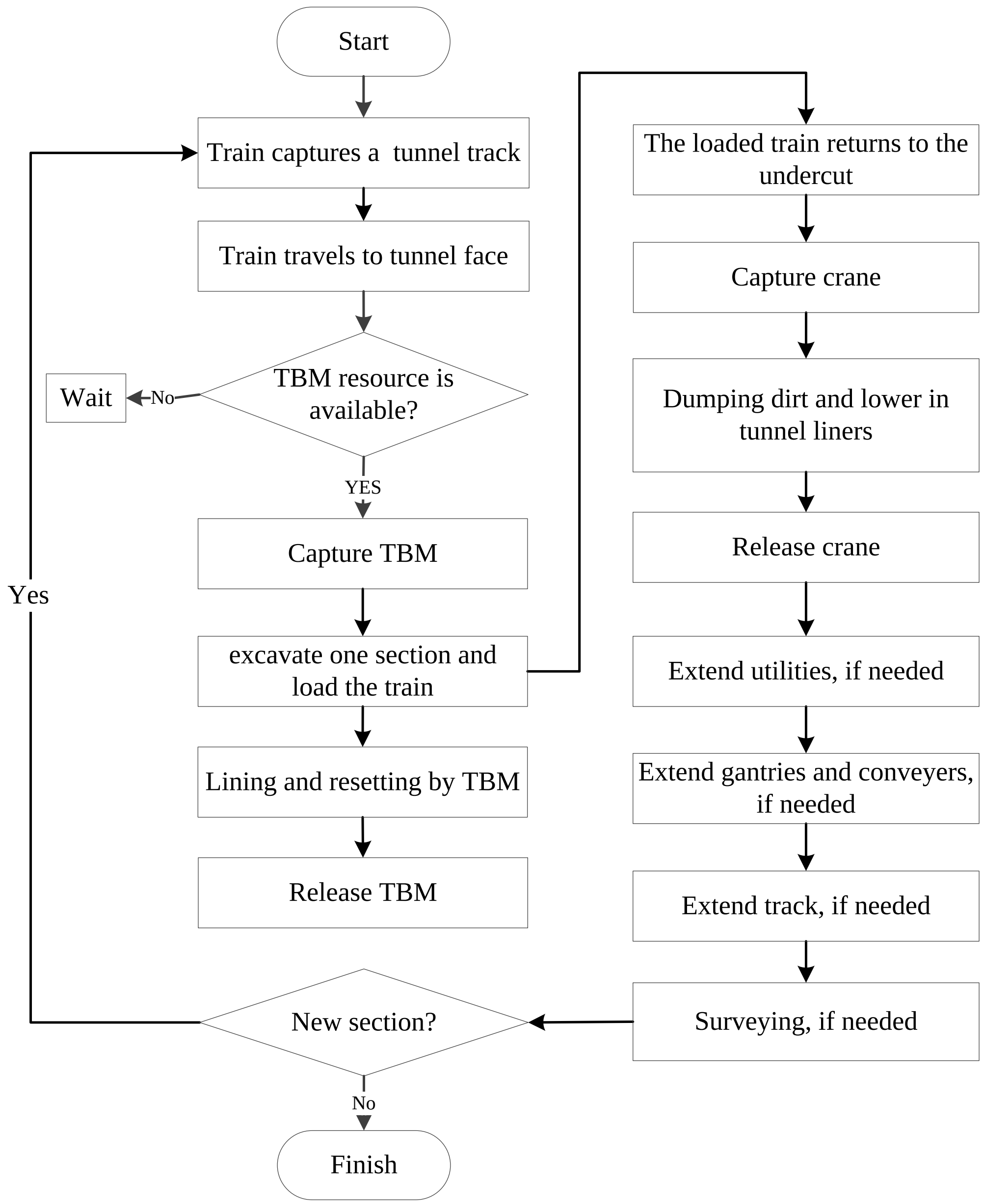


Figure 6-2: Tunnel Construction Process in TBM Tunneling Simulation Model

Unlike the shaft element, different entities are created in tunnel element to go through the tunnel construction process; the dirt entity is used for scheduling tunnel excavation, dirt removal to the undercut, moving dirt to the ground level, and moving the train back to the tunnel face. Also, before sending the train inside the tunnel, it checks for any other tunnel tasks, such as surveying, that need to be scheduled before starting the next section. The other entity, called tunnel entity, is responsible for scheduling initial activities that happen before tunnel excavation, such as install cradle, install TBM, and drive TBM into the tunnel, as well as lining and resetting activities of the TBM. The reason for this is that when the dirt entity is used for scheduling dirt removal events, at the same time the tunnel entity is responsible for scheduling lining and resetting events.

The progress of tunnel construction is usually calculated based on the current length of a tunnel that is completed, so in this case, there is no separate progress rate for the activities that happen during the construction of one tunnel section.

In the construction project, activities happen either in sequence or in a repetitive cycle, so shaft and tunnel can represent all kinds of construction processes.

6.2.2 Implementation of Breakdowns and Interruptions in Simphony

In the construction simulation, equipment breakdown is modeled based on historical information from previous similar projects. Usually two input parameters, “time between breakdowns” and “repair time”, are required to model equipment breakdown in the construction simulation. Due to the uncertainty in the breakdown occurrence, input values are mainly entered in the form of random variables that follow certain distributions. To model breakdown and any other interruptions in the tunnel template, separate entities should be created and assigned for simulating breakdowns in the TBM, crane, train, and loader. Figure 6-3 demonstrates an example of train breakdown simulation in the tunnel element.

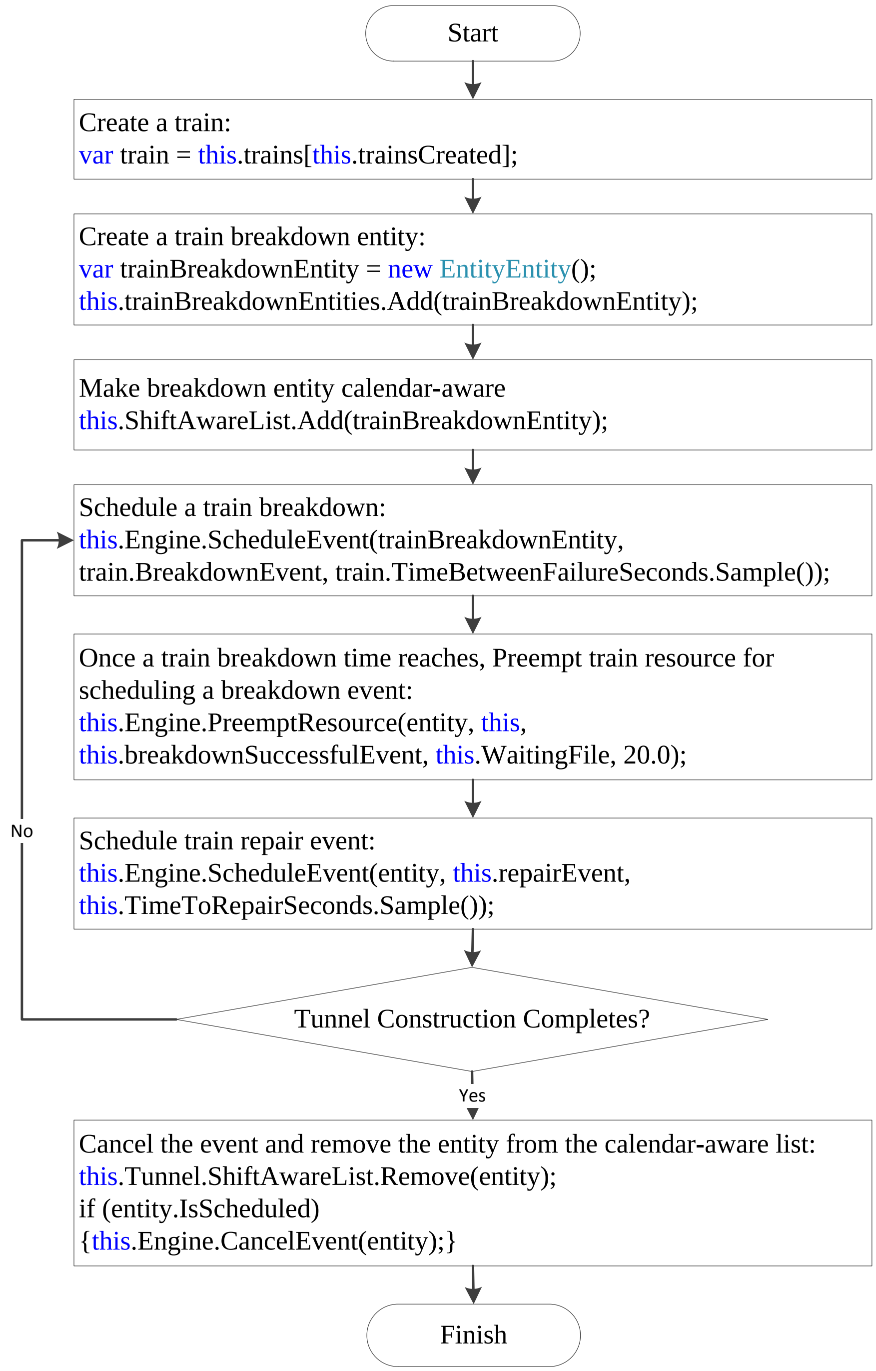


Figure 6-3: Train Breakdown Simulation Process in Tunnel Simphony Element

After creation of an instance of equipment resource, a new breakdown entity is created and added to the calendar-aware list associated with the calendar of the work area that the equipment belongs to. At this point, an event is scheduled to start the first breakdown after a duration sampled from a distribution entered as an input value for the breakdown interval. Another event will be scheduled after equipment successfully breaks down to schedule start of the repair activity, and then another event occurs to finish it. This process will continue until the completion date of the work area that the equipment is assigned to.

6.2.3 Implementation of Calendar in Simphony

In Simphony.NET 4.0, a user can specify different scenarios for one model; each scenario can have its own properties (Figure 6-4) such as a set of user-defined calendars, a start date, and a time unit for the simulation clock. In the tunnel template, in a scenario level, a user can define different calendars for a project and assign them to each work area in the model. The start date of the scenario is equal to the start date of a project, and the time unit is defined as second to make the template work in the COSYE environment.

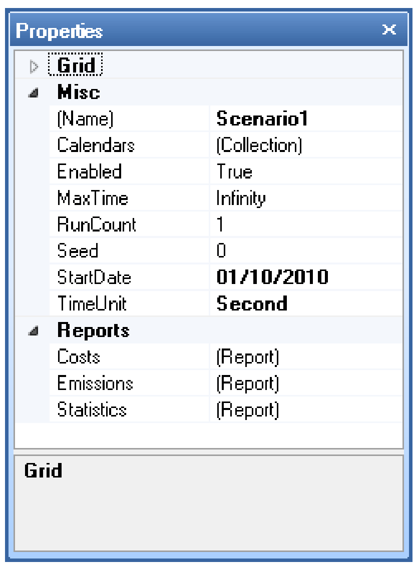


Figure 6-4: Property Grid for Scenarios in Simphony

To implement calendar in a Simphony template, new methods were added to the discrete event engine in Simphony such as “SubscribeCalendar” and “UnsubscribeCalendar.” For each work area, such as working shaft, tunnel, and removal shaft, a calendar entity is created and then subscribed to the associated calendar of that work area. For each calendar, a calendar-aware list is created to store a list of entities that flow through the model. During a simulation run, a calendar entity responds to the calendar events. Based on the calendar non-working and working status, all the entities in the list will be suspended or resumed, respectively. Figure 6-5 shows the process of implementation of calendar in the tunnel template.

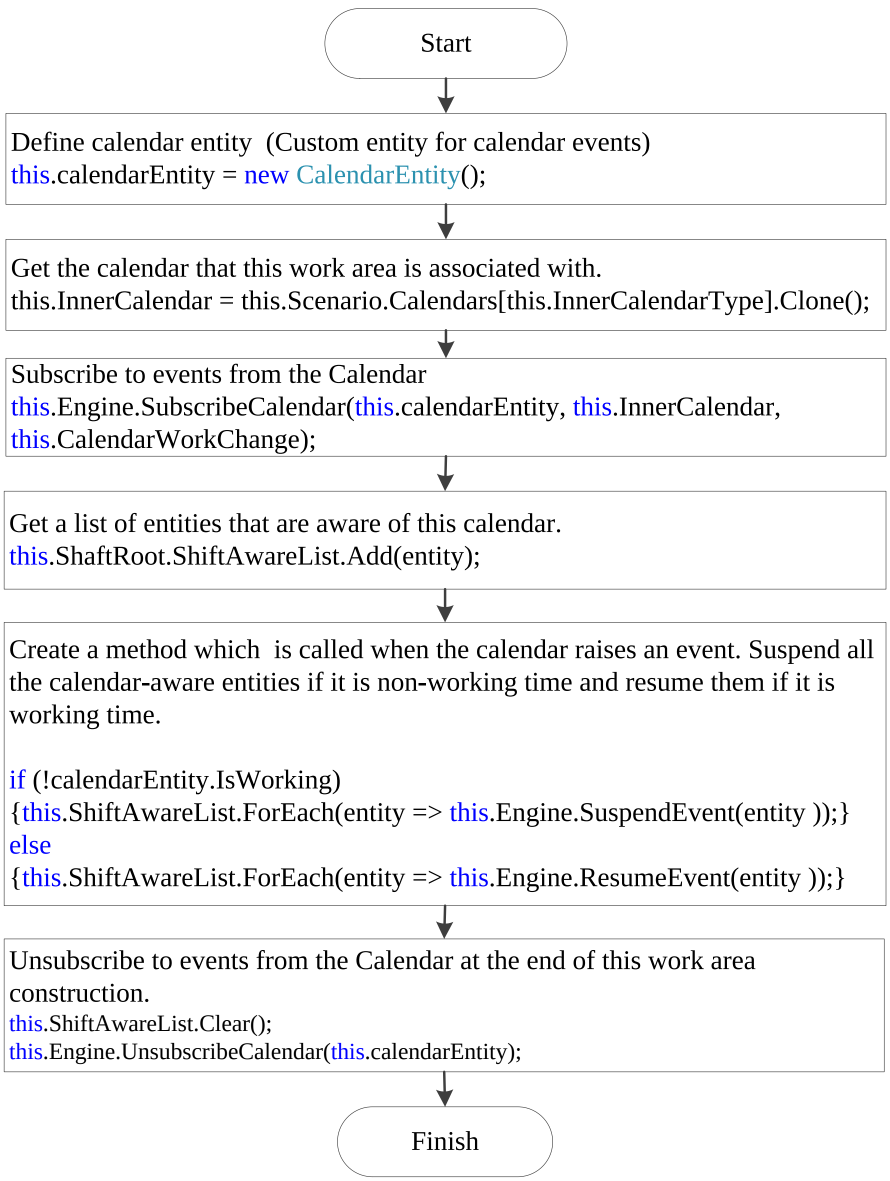


Figure 6-5: Implementation of Calendar in the Tunnel Template

6.3 Hybrid Simulation Approach for Updating Construction Process

Recording of as-built data and actual project process is a means to control construction progress and document history of a project for future use. The frequency of data collection in the construction phase is subject to various factors such as the complexity of the work, the cost and time required for a data collection in the job site, the speed at which the project is advancing, and how often a project control group requires monitoring of project progress. In this research, a daily data collection system is utilized to provide more detail of the actual project progress. Figure 6-6 illustrates a query in the Microsoft Access database which includes five fields of daily information of the working shaft.

WorkingShaftDailyProgress

Date	Name	ActivityName	TaskName	Percent Complet
11-Nov-10	Working Shaft	CircularShaftSegment	Excavation	92.50%
12-Nov-10	Working Shaft	CircularShaftSegment	Excavation	92.50%
15-Nov-10	Working Shaft	CircularShaftSegment	Excavation	98.00%
16-Nov-10	Working Shaft	CircularShaftSegment	Excavation	100.00%
17-Nov-10	Working Shaft	CircularShaftSegment	BeamAndHangShaftInstallation	15.00%
18-Nov-10	Working Shaft	CircularShaftSegment	BeamAndHangShaftInstallation	35.00%
19-Nov-10	Working Shaft	CircularShaftSegment	BeamAndHangShaftInstallation	50.00%
22-Nov-10	Working Shaft	CircularShaftSegment	BeamAndHangShaftInstallation	70.00%
23-Nov-10	Working Shaft	CircularShaftSegment	BeamAndHangShaftInstallation	70.00%
24-Nov-10	Working Shaft	CircularShaftSegment	BeamAndHangShaftInstallation	70.00%
25-Nov-10	Working Shaft	CircularShaftSegment	BeamAndHangShaftInstallation	100.00%
26-Nov-10	Working Shaft	SlabSump	SumpExcavation	100.00%
29-Nov-10	Working Shaft	SlabSump	SlabPouring	100.00%
30-Nov-10	Working Shaft	TailTunnel	WorkingPlatformConstruction	100.00%
1-Dec-10	Working Shaft	TailTunnel	BreakOut	100.00%
2-Dec-10	Working Shaft	TailTunnel	Excavation	10.00%
3-Dec-10	Working Shaft	TailTunnel	Excavation	19.40%
6-Dec-10	Working Shaft	TailTunnel	Excavation	29.00%
7-Dec-10	Working Shaft	TailTunnel	Excavation	38.70%
8-Dec-10	Working Shaft	TailTunnel	Excavation	67.70%
9-Dec-10	Working Shaft	TailTunnel	Excavation	83.90%
10-Dec-10	Working Shaft	TailTunnel	Excavation	100.00%
13-Dec-10	Working Shaft	TailTunnel	PourFloorConcrete	100.00%
4-Jan-11	Working Shaft	Undercut	WorkingPlatformConstruction	18.00%

Record: 34 of 88 No Filter Search

Num Lock

Figure 6-6: Microsoft Access Query for the Working Shaft Daily Progress Data

As it is shown, due to various factors that affect construction progress, the values of percent complete for one task alter day by day. Therefore, a simulation model that records the actual project state should be able to update simulation inputs especially the construction progress on a daily basis, based on the real data collected from the site. This causes a problem in simulation modeling in Simphony since it works differently; as previously mentioned, in Discrete Event Simulation modeling, events happen at the start and end of each task, and occurrence of each event causes a model to change from one state to another. For example, a tail tunnel excavation (Shown in Figure 6-6) is modeled as a single task in the template which means that Simphony schedules two events; one event for the arrival of entity at the excavation task at time (t) and the other for the completion of the task at time (t + d) in which (d) represents the total duration of the excavation. So if task duration extends more than a day, knowing that no event happens in the middle of the task, a daily update cannot be achieved unless a simulation mechanism provides daily update capability in the application.

In the current tunnel template, if no interruption or equipment breakdown happens during a task, as long as the number of working hours per day remains the same, a task will have equal progress each day, which is calculated by:

$$\textit{Daily Progress for a task}\ \% = \frac{\textit{Woking hours per day}}{\textit{Task total duration}} \times 100 \qquad \textit{Eq. (6-1)}$$

Therefore, there is a linear increment in the percent complete of a task on a daily basis.

To model actual project progress, the proposed solution in this research is a hybrid simulation approach that combines time-step simulation with a discrete event modeling technique. In this method, when one task is completed, Simphony schedules an event to start the next task, but at the end of each day, the state of a task should be updated based on the actual data captured from the site, which means that every 24-hour time interval, an update should apply to the state of a task. In this method, scheduling tasks is based on a discrete event concept while task progress is simulated in a time-step fashion. An as-built federate sends updates to a Simphony model through RTI in a time-step fashion, so COSYE helps integration of time-step and discrete event modeling components in one system. The process of updating activity progress based on actual data is discussed later in this chapter.

6.4 Documentation of As-Built Information in COSYE Federation

In a construction project, due to various factors and variables, a project may not be constructed as planned; changes vary from small ones in the scope of work to complete modifications of design and construction method, and all these variations from the original plan may result in claims and disputes. Therefore, the proposed framework not only should reflect real construction progress and product information, it also needs to demonstrate all the scope changes in the construction process. To implement the framework and to be able to document all the as-built information in Simphony including the process changes, two basic steps were followed; as first step, it was assumed that no changes happen in the process model, which means that a Simphony model remains unchanged from what was originally defined during the planning stage. In this phase, the process of updating activity duration, project progress, and input variables such as resources, working hours, and shift information was determined. In the next step, a solution for updating process model due to scope changes was investigated.

In this section, a first step of the updating process for a simulation model in Simphony is elaborated. In this stage, a wide range of project information such as: start date of a project, duration of activities, shifts and working hours, product information, project resources, and environment conditions should be updated based on actual data collected and stored in the database.

6.4.1 Documentation of Actual Project Start Time in the Simphony Model

At a start of a simulation run and before sending any updates from the as-built federate to the Simphony model, the first task is to create a connection between the Microsoft Access database and as-built federate. This should happen before federation execution and after all federates join the federation, so the as-built federate can publish all the information that other federates are interested in. Before execution of the federation, the program allows users to select a project database by clicking a "Database Location" button provided on the user interface of the as-built federate.

The next task before federation execution is to update actual start date of a project in Simphony, which was one of the challenging tasks in this phase. When the Simphony model runs in a stand-alone mode, the start of a simulation time is set based on the time entered as a scenario start time. In the tunnel template, the start time of a project is equal to the scenario start time and this facilitates the calculation of start and finish date of activities and the entire project; for instance, to calculate project finish date, the following code is used:

```
"Project.ActualStartDate.AddSeconds (this.FinishTime).Date";
```

This code basically adds the total simulation time to the start time of the project (scenario) and presents it as a calendar date.

Based on HLA time management strategy, there should be a start time for the entire federation; in HLA, two options are available for setting federation start time:

- A particular “master federate” can announce the start time after synchronization at “ReadyToInitialize” by sending an interaction to the RTI, or,
- Every federate (that wishes) announces the starting time using an interaction after synchronization point at “ReadyToInitialize” and federation start time becomes the minimum of these dates/times.

In COSYE federation, two interaction classes are added to set project start time: “Report Start Date” and “Request Start Date”. These two methods are also implemented in the RTI connection element in Simphony. In the tunnelling federation, in the absence of an as-built federate, the start time for the federation is set as the start time of a project. But if an as-built federate joins the federation, the project start date should be modified based on the actual project start date that is recorded in the database. This should happen before running the federation and before any federate starts advancing time and processing any event. Therefore, before simulation execution, the as-built federate reads project actual start date from the database and reports it to other federates. Then, other federates reflect the value and update project start date.

6.4.2 As-Built Documentation of Construction Process

The assumption in this section is that there is no scope change in the project so the project is constructed as it was planned; however, the progress rate, and therefore the finish date of activities and the entire project, can vary. The proposed method to update the activity duration is based on the daily progress report for a process or certain activity. This method updates duration of the activities in the project based on the daily progress update. Two different procedures are proposed to update activity duration, project progress, and input variables, as described below:

1. The first approach is used to update construction progress and input values for the shaft construction where a process consists of non-repetitive activities. In this case, the duration and progress rate (% complete) can be obtained for each activity or each task separately.
2. The second method is developed to update progress rate and input values for repetitive processes such as tunnel construction which require execution of a set of tasks to achieve certain progress. Therefore, the progress is not recorded for each activity (or task) in a cycle but for the entire process. But, the same as the previous approach, a duration of each activity can be calculated in each cycle.

6.4.2.1 As-Built Documentation of Construction Progress for Non-Repetitive Activities

This method is used to record as-built documentation for the shaft construction in the tunneling template. After processing data, the as-built federate requests ownership of attributes for each task, updates their values, and then terminates the ownership. These attributes are:

- Name, which represents task name, related activity, and work area.
- Percent Complete, which presents total percent complete of the activity (or task) on a daily basis.
- Duration, which calculates the total working hours minus interruption durations.
- Last Entry, which represents the last data in the database.

The as-built federate reads data and updates attribute values at the start of each day (12:00AM). At the same time (after one second as a lookahead time, which is the minimum time stamp the RTI allows each federate to place on an event it sends, as a function of wall clock time), Simphony receives data, but waits until the start of the first shift of the day to simulate current data. This procedure will continue until the end of shaft construction, or until the last entry in the database.

In the Simphony template, changes were made in the event processing method to make the template work for three conditions:

1. When the as-built federate is not part of the federation, or Simphony is used as a stand-alone application.
2. When as-built data is available for part of a shaft construction, which means that it may end in the middle of any task in the process.
3. When as-built data is documented for the entire shaft construction process.

The solution proposed in this section is to duplicate an event that schedules completion of each task. Based on the availability of as-built data, one or more events may be fired for each task in the process. Figure 6-7 demonstrates the event scheduling algorithm in this element.

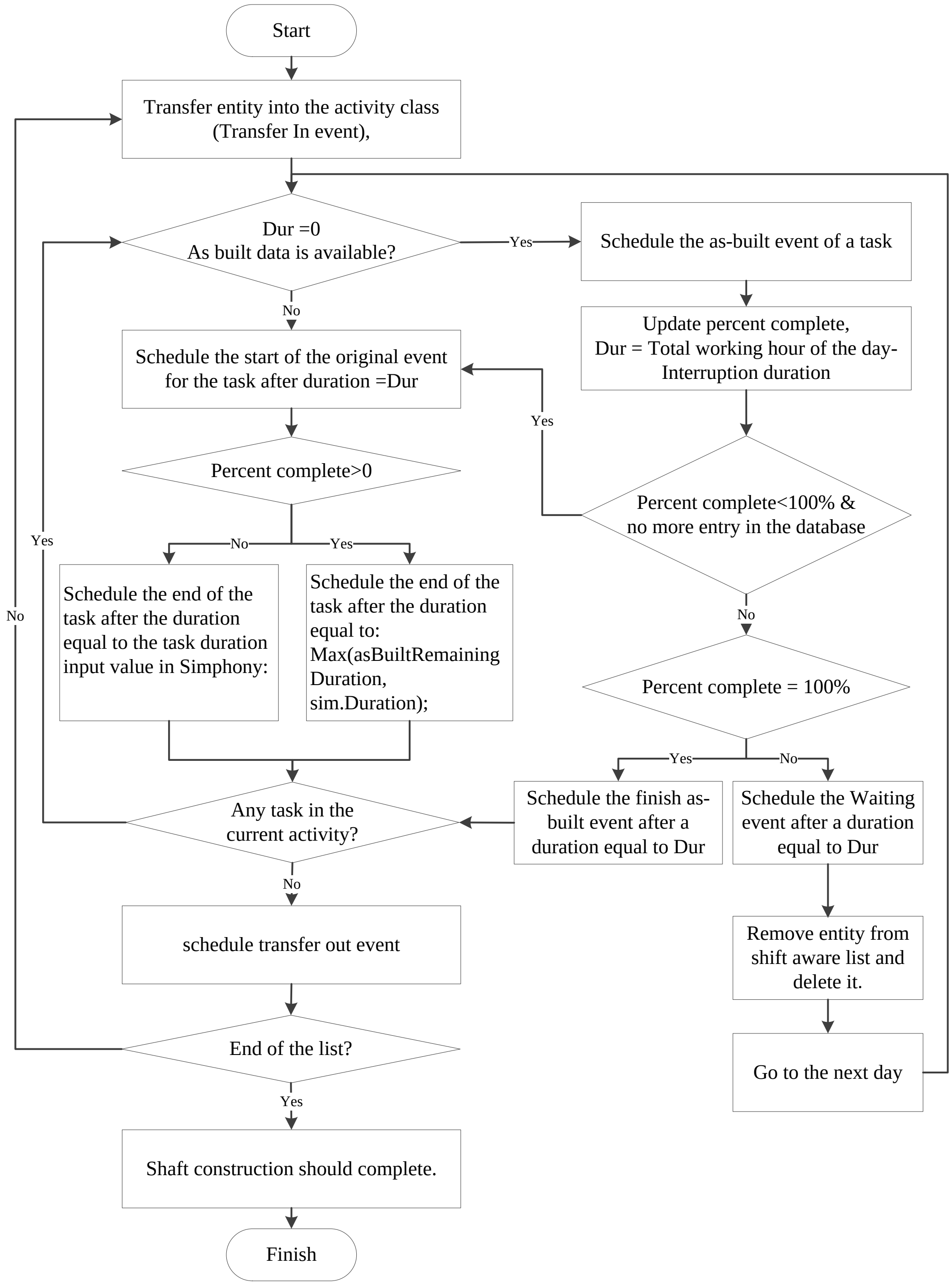


Figure 6-7: As-Built Event Scheduling in the Shaft Element

As it is illustrated, if as-built data is available for the current task, then an as-built event is scheduled. Then, another event fires after a duration equal to the working hours minus interruption duration to schedule the end of the as-built event. If the entry in the database was the last one, then a program will continue to simulate the rest of the model without updating any values based on actual data, but if there are still entries in the database for the following day, the entity will be removed from the calendar-aware list. By receiving new updates for the following day, a new entity will be created and added to the list for that day. If there is no entry in the database for the following day and the percent complete for the task is less than 100%, then from that point forward, the program will simulate the remaining time of the task and the rest of the model. At this point, an event will be fired at the end of the current day to schedule start of the original event, and after that, another event will be fired to schedule the end of the event after the remaining duration for the task, which is calculated as follows:

$$\textit{As-Built Remaining Duration} = \frac{\textit{1-\% Complete}}{\textit{\% Complete}} \times \textit{Total As-Built Duration so far} \qquad \textit{Eq. (6-2)}$$

$$\textit{Sim. Duration} = \left(\textit{1-\% Complete}\right) \times \textit{User Input for the Task Duration.Sample} \qquad \textit{Eq. (6-3)}$$

$$\textit{Remaining Duration} = \textit{Max (Asbuilt Remaining Duration, Sim. Duration)} \qquad \textit{Eq. (6-4)}$$

And, the remaining tasks will be simulated without scheduling as-built events. All the tasks in the shaft template were modified to include as-built event and additional required features to make the element work with as-built data.

6.4.2.2 As-Built Documentation of Construction Progress for Repetitive Activities

The proposed method is utilized in a TBM tunnel construction element to update tunnel progress, as well as duration of repetitive activities in the tunneling process. Tunnel advancement, which is calculated based on the completed segments of a tunnel, is the most common data collected on a daily basis in the construction site. This usually helps project managers to predict tunnel progress and soil profile changes along the tunnel alignment.

As previously mentioned, in the tunnel simulation template, construction of one segment consists of various tasks such as: excavation, dirt transfer to the working shaft, unloading dirt, loading train with liners, transferring liners to the tunnel face, lining a segment, and resetting TBM for the next cycle. Ourdev et al. (2007) divided a tunnel operation in two groups; production operations, and support activities, and defined a cycle time as the total of operation time (excavation time) and support time. A support time may consist of lining and resetting duration plus train travel time to and from the tunnel face, and unloading and loading duration. Depending on the current chainage (constructed length) of a tunnel and number of trains deployed for dirt removal, operation and support activities and their sequence may vary as described below:

1. At the beginning of a tunnel construction, there is no space for operation of a train which usually consists of one or more muck cart and one material cart, so crew use one muck cart to remove dirt from the tunnel face and one material cart to deliver segments to the tunnel face. Also, the TBM machine should advance into the tunnel to provide required length for installing its remaining parts (conveyor belt and gantry) to have more productivity in each cycle. So, at the beginning, tunnel advancement is determined based on capacity of a muck cart. Usually, the crew uses a muck cart with a capacity equal to half of the excavation volume of one segment, which means that excavation happens in two cycles while lining and resetting of TBM happens after the second one, where enough length is available to install segments. Here, a total duration to complete one segment is calculated by:

   $$Cycle\ Time\ (t_c)=2t_{ex(half\ section)}+t_s \qquad Eq.\ (6\text{-}5)$$

   Where: $t_s=\left(t_{Lining}+t_{Resetting}\right)+t_{Load}+t_{Unload}+2\ t_{travel}$

2. Once the TBM is fully installed, a train can be utilized for the dirt removal and a tunnel advancement can increase to one segment and. In this case, a total cycle time is equal to:

   Cycle Time $(t_c) = t_{ex\,(full\ section)} + t_s$ *Eq. (6-6)*

   Where: $t_s = Max\ [(t_{Lining} + t_{Resetting}), (t_{Load} + 4t_{Unload} + 2\ t_{travel})]$

3. If crew decide to use two trains to expedite the dirt removal process especially in a long tunnel, a switch should be installed in the working shaft area. By using a switch, one train can leave the undercut and take liners to the tunnel face while the other train is unloading the dirt. After dumping the dirt, the crane lowers down the liner blocks for the next segment of the tunnel. This completes one cycle of tunnel operations. Here, the cycle time is calculated by:

   Cycle Time $(t_c) = t_{ex\,(full\ section)} + t_s$ *Eq. (6-7)*

   Where: $t_s = Max\ [(t_{Lining} + t_{Resetting}), (2\ t_{travel})]$

Usually, due to many uncertain parameters in the underground construction, a tunnel element is designed to let users select a probability distribution for duration of each task in the model, and then during simulation run, Simphony samples a single value from the assigned distribution to schedule a task finish time.

According to the above-mentioned points, a proposed solution should be totally different from the one proposed for the shaft element; the updating solution is described in the following steps:

- On a daily basis, the as-built federate reads working time (excluding breakdown and other interruptions duration) and actual advancement data from a database and calculates actual cycle time of tunnel construction:

$$\textit{Actual Cycle Time } (t_{ac}) = \frac{\textit{(Total working time-total interruption durations)}}{\textit{Today Advancement}} \qquad \textit{Eq. (6-8)}$$

- At the end of the previous working day, the Simphony template sends an update to the federation for the operation and support activities' original duration, and then the as-built federate calculates "Plan Cycle Time" using either *Eq.(6-5)*,. *Eq.(6-6)*, or *Eq.(6-7)*.
- In the next step, the as-built federate calculates an actual duration for each activity based on the original duration of the activity in the Plan Cycle Time. For instance, lining duration is calculated as follows:

$$\textit{Actual Lining Duration } (t_{aL}) = \frac{\textit{Sampled lining duration}}{t_c} \times t_{ac} \qquad \textit{Eq. (6-9)}$$

- At the end of the previous day, the as-built federate updates the actual duration for each activity in the form of constant value instead of probability distribution.
- After Simphony receives updates at the start of the working shift for the next day, it simulates tasks with constant durations so at the end of the day, simulation results are similar to actual values.
- At the end of each day, all durations are reset in Simphony to the original distributions and this process will continue until the construction completion date.
- If actual data is not available for the entire tunnel, Simphony will start simulating a process based on original values.

Figure 6-8 demonstrates the process of updating tunnel advancement on a daily basis and based on actual data.

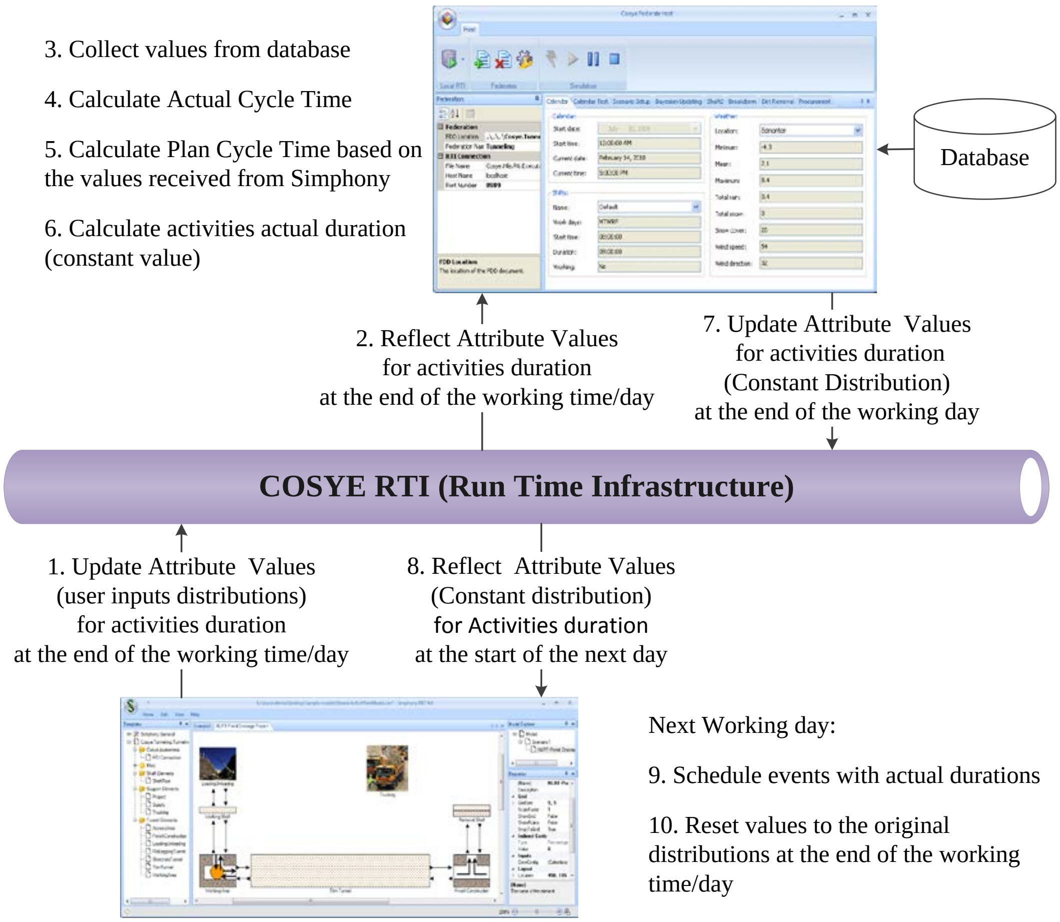


Figure 6-8: Process of Updating Tunnel Construction Operation on a Daily Basis

6.4.3 As-Built Documentation of Interruptions and Equipment Breakdowns

Common interruptions in the construction process can be a result of equipment breakdown, weather condition, and delay in material delivery. In addition, there are certain tasks (special tasks) that affect construction progress; for example, in the tunnel construction, tasks such as gantry and conveyor installation, surveying, and track and utility extension interrupt tunnel advancement, so in this section, they are treated as interruptions in the tunnel construction.

As mentioned earlier in this chapter, interruptions such as equipment breakdown and surveying are periodic activities that are scheduled in user-defined intervals. But this would be different in a real situation since interruptions can happen at any time during the construction phase, so the simulation approach to model interruption won't be practical in recording the actual project. Besides, there are interruptions in the real situation that are not considered in the simulation model, which makes process model documentation more complex. To document interruptions in the model, the solution is described as follows:

- For each day in the database, the as-built federate refers to two tables; the Interruption table, to retrieve causes and durations for the interruptions, and the DailyRecord table, to find the corresponding work area for each interruption. It is possible that more than one interruption occur in a day in one work area in the project.
- At the beginning of a working day, for any interruption recorded in the database, except for equipment breakdowns, the as- built federate sends an interaction to send interruption type and duration to the Simphony model. For equipment breakdown recorded in the database, the as-built federate changes the "Equipment State" to down and updates the attribute value by sending an update message to the RTI.
- For interruptions that are not currently scheduled in the Simphony template, it is suggested to treat them as one type of interruption and simulate them similar to equipment breakdowns by creating a resource called OtherInterruption. The entity which flows through the process captures this resource at the beginning of the tunnel construction process and keeps it through the entire process. At the start of each day, if an element receives any update from RTI, the interruption entity is created and

added to the calendar-aware list, then it pre-empts the interruption resource, suspends current events, and schedules end of interruption so no event will be scheduled during that interruption. At the completion of the interruption event, the interruption entity will be removed and deleted from the calendar-aware list.

- To schedule special tasks such as surveying: at the beginning of each day if the element receives an interaction related to a special task, a variable called "shouldDoAsBuiltSurveying" is set to true for the task. Then, during a simulation run, when the entity reaches the point where special tasks, including conveyor and gantry installation, track and utility extension, and surveying, need to be scheduled, and if the variable returns true, the task will be scheduled using a duration value received from the as-built federate; otherwise, it will pass the events without scheduling them.

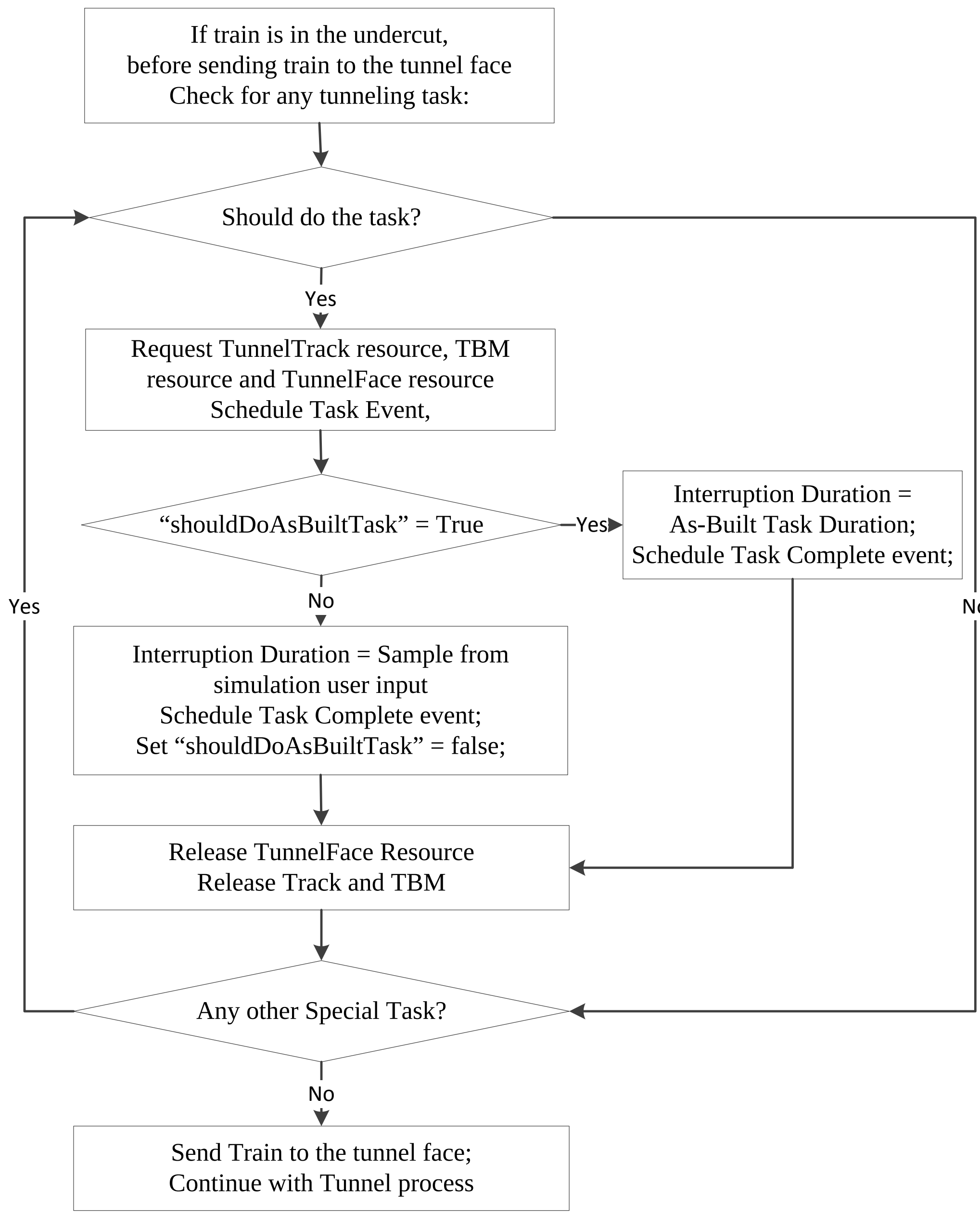


Figure 6-9: Special Tasks Scheduling in the Plan and As-Built Tunnel Process

- For interruptions such as equipment breakdowns, the updating process is complicated. In the existence of the as-built federate, no breakdown event is scheduled based on simulation input variables. When the as-built federate reads recorded breakdown in the database, it changes the state of equipment to "down" and sends an update attribute value to the RTI. When a Simphony element, subscribed for that equipment class and its attributes, receives the update, a breakdown event will be scheduled at the beginning of a day, and a repair time will be set to the duration recorded in the database. This will continue until the point that as-built data is available in the database. But after that, there should be a method to reschedule the occurrences of equipment breakdown as previously modeled in the original Simphony template. To find last Interruptions recorded in the database, a Microsoft Access query called "LastInterruptionRecord" was created to find the last occurrence of each interruption in the as-built records (Figure 6-10).

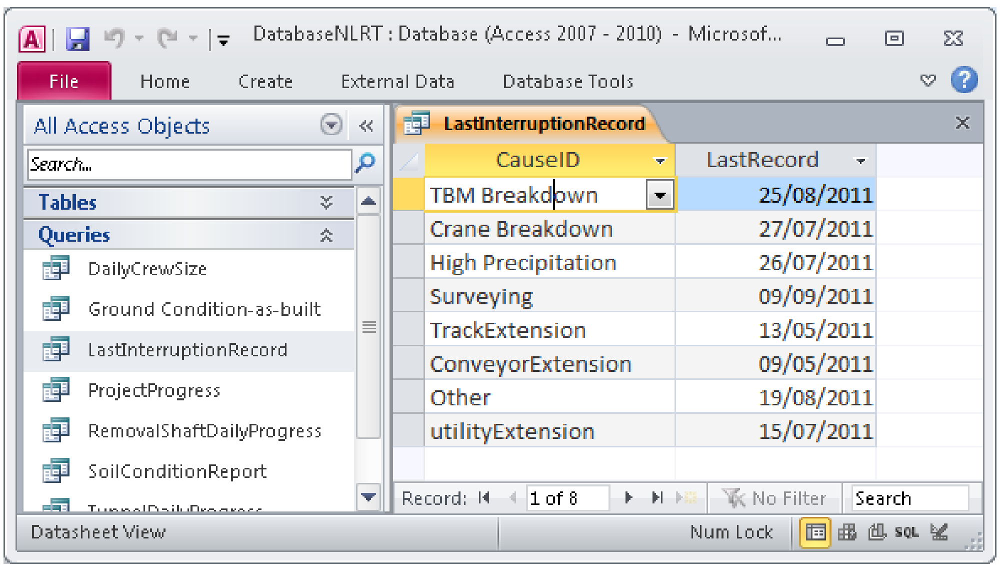


Figure 6-10: Last Interruption Records in the As-Built Database

If today's record of a breakdown is the last one in the database, the as-built federate calculates the time difference between TimeNow and the last date that as-built data was recorded in the database, and compares it to a sampled value from a random distribution assigned to "TimeBetweenFailure" attribute, then it updates the interval value to the maximum of these two numbers. The same procedure is designed for scheduling special tasks in the tunnel element.

- In the case of multiple interruptions in one day, Simphony will receive them at the beginning of a day and schedule them sequentially, except for the special tasks which are scheduled after excavation and dirt removal of the first tunnel section in that day.

The implementation of as-built interruptions in the Simphony template was one of the most challenging parts of this research, and proposed methods are developed in a generic way that can be used for any other Simphony template.

6.4.4 As-Built Documentation of Resource Information

In this study, crew size and equipment quantity are the only simulation inputs that will be updated based on daily records. In a construction project, crew information is often recorded through daily timesheets; this information is usually used for estimating project crew cost, calculating payments, and issuing monthly paycheques. In the tunnel template, crew size and unit cost of crew are two input variables that assist users in calculating crew cost for each work area, and for the entire project. Updating crew size can provide a clear idea of how work progresses in regards to the number of workers on the site, so it helps with resource planning in the project. In the simulation model, each work area has an attribute for crew size, which is a fixed number that remains the same during the construction period for that work area, but in the as-built records, this value will be changed on a daily basis, so a simulation model should be capable of updating crew size every day. The process of reading data from a database is similar to the previous records:

- For every working day, the as-built federate reads crew size separately for each task in the database.
- Then the as-built federate sends the updated value for the “size” attribute of the crew class in the FOM.

- At the start of the next working day, tunnel, shaft, and other elements that are interested in the crew size, will receive an update.

These changes are mainly reflected in the report federate, where crew size and utilization, along with a progress rate, are reported on a daily basis to help the project manager improve resource management in the project, or for future planning purposes.

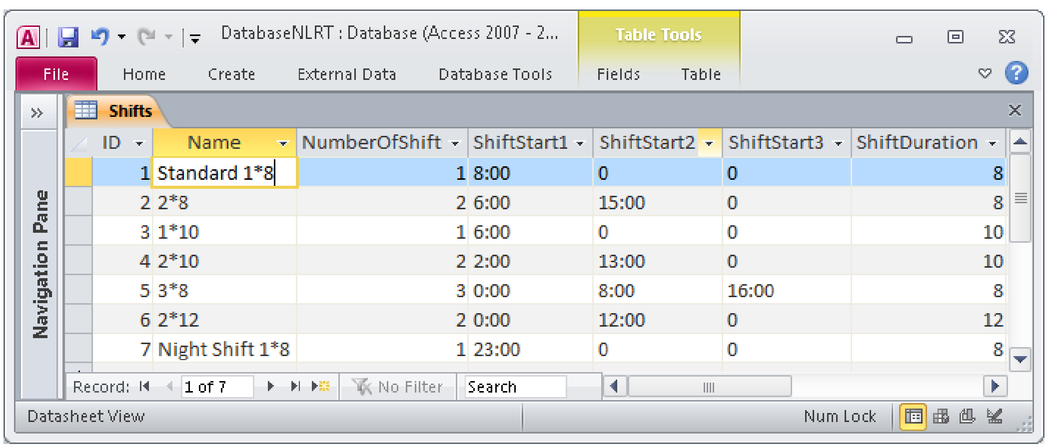
In addition to the crew size, number of different types of equipment is also recorded in a daily site report, which assists in cost estimation and future project planning. As described in Chapter 5, only certain types of equipment in the project are modeled in the template. For instance, in the tunnel template, based on a real case study, around 80% of equipment cost was related to the crane, TBM, loader, and truck operation. So availability of these machines, and their utilization and breakdowns, affect productivity in the project. Therefore, in the tunnel template, five different equipment types including crane, TBM, truck, loader, and train are modeled as resources in the process. The updating process for the as-built records is the same as for crew size; equipment number is updated on a daily basis and used mainly for reporting purposes.

6.4.5 Documentation of Shifts in the As-Built Process Model

In the planning stage, the project management team defines the project calendar, with all holidays and exceptional days, to use as a base calendar, and specifies working hours and number of shifts per day for different work areas on a copy of the base calendar. In the tunnel template, each work area has its own calendar (a copy of the project calendar), which defines working hours during regular working days, non-working days, and exceptional days. So, in the tunnel template, a user can create different calendars and manually enter information based on the project calendar, and then assign calendars to main work areas.

During the actual construction phase, because of various factors such as delays, the project manager can add extra shifts, extend working hours per day, or ask workers to work during the weekend. In case of changes in working days, shifts, and working hours (changes to the calendar in general), the Simphony model should be able to accommodate changes.

In the construction site, along with other site information, it is common to document number of shifts, working hours, and crew information per shift for each work area in the project. In this research, the same data is collected, but to easily store and retrieve it from a database, a table was created at the beginning of the project and possible shift configurations were added to the table (Figure 6-11), so in the daily reports, one can select from a drop-down list developed based on the information stored in this table. At any time in the construction phase, a user can add a new shift to this table and assign it to any work area. In a “DailyRecords” table, there is a column called “shift” which contains the shift name. A shift name assigns to each record (row) in the table.



ID	Name	NumberOfShift	ShiftStart1	ShiftStart2	ShiftStart3	ShiftDuration
1	Standard 1*8	1	8:00	0	0	8
2	2*8	2	6:00	15:00	0	8
3	1*10	1	6:00	0	0	10
4	2*10	2	2:00	13:00	0	10
5	3*8	3	0:00	8:00	16:00	8
6	2*12	2	0:00	12:00	0	12
7	Night Shift 1*8	1	23:00	0	0	8

Figure 6-11: Shift Table in the As-Built Database

In the next step, the as-built federate reads shifts information for the current day and converts it to a meaningful format accepted by Simphony calendar such as time spans, and then sends an update to the RTI. In order to send shift information through RTI, a shift object class is created and added to the FOM as it is shown in Figure 6-12.

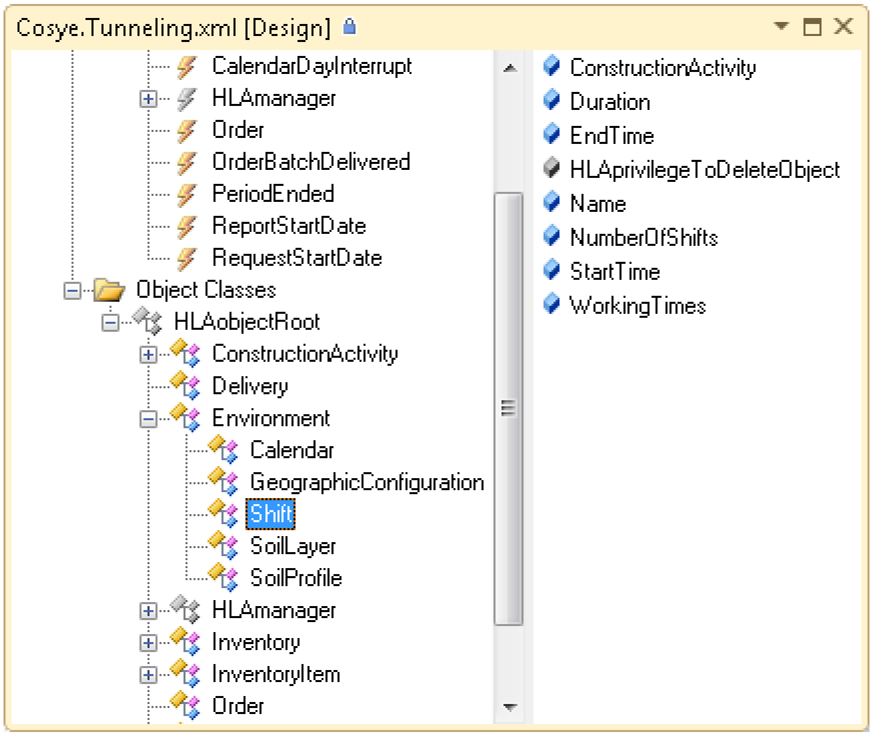


Figure 6-12: Shift Class and Attributes in the Federation Object Model (FOM)

For each working day, shift information including name, working times, number of shifts, duration of each shift and a related work area are being updated by as-built federate. The "WorkingTimes" attribute is an array of time spans that sets start and end of each shift in a day. Figure 6-13 illustrates an example of how values are being set for "WorkingTimes" attribute in a 2×8 hrs. shift.

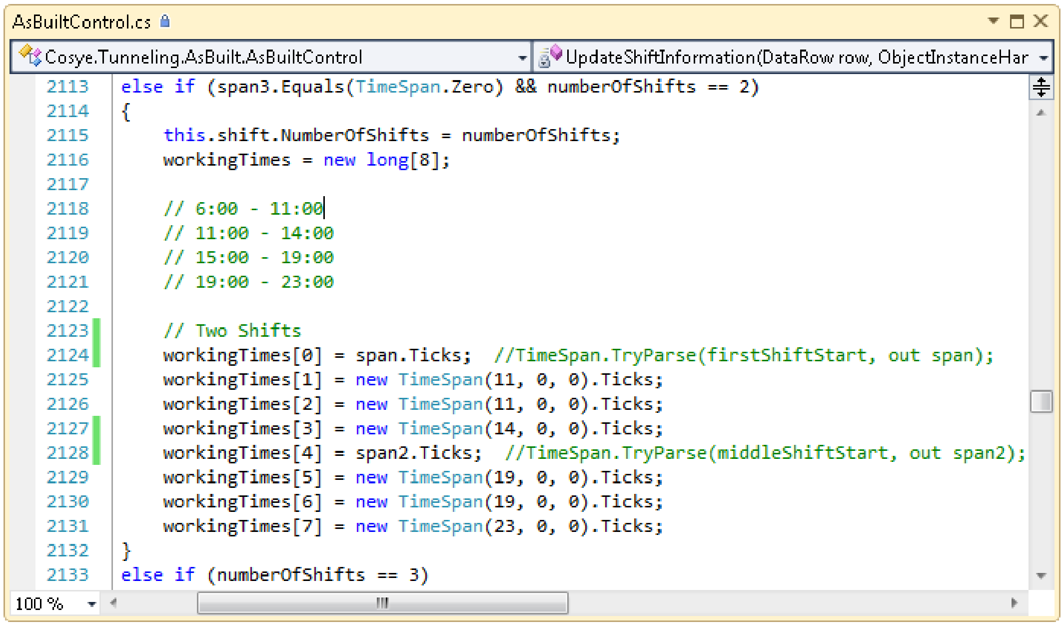


Figure 6-13: Setting Working Times for a 2×8 hrs. Shift in the As-Built Federate

After a Simphony element receives updated values at the start of the day, it will update its calendar and working time for that day.

6.5 As-Built Documentation of Construction Process Changes in COSYE

Scope changes are a normal and expected part of the construction process. The need for scope changes arises from different uncertainty factors in the construction site including site condition, constructability issues, design faults, material availability, and many others. One of the main reasons to document the as-built process is to record all scope changes during the construction stage in case of future need (e.g. dispute resolution).

Hence, one of the objectives of this research is to document changes in construction projects using the simulation approach. Since documentation of the as-built process and product information in a distributed simulation environment requires extensive changes in discrete event simulation modeling, during the first stage of the development phase, it was assumed that no process changes are documented in the database. After managing the process of updating the Simphony model in the COSYE federation, the next step focuses solely on documentation of process changes in the project. As described in Chapter 4, the proposed solution is to consider multiple instances of Simphony in the federation that model the planned process, as-built process with no changes, and the revised model which incorporates scope changes. The distributed simulation environment facilitates implementation of the proposed method, allows running three different models in one system, and provides an integrated report on the project performance. To have Simphony federates produce results associated with the process model they are representing, a control mechanism is put in to allow Simphony models to accept or ignore messages/updates received from the RTI. The federation is designed in such a way that a message sent from a federate is tagged with a specific, user-defined name so that other federates can decide whether to accept or ignore the message. For instance, in a Simphony federate, when a message is received with a user tag that does not match the project element name, Simphony ignores the message. This means that when the as-built federate reads information from the database and sends update messages to the RTI, it can specify a user tag to the message to determine which Simphony template should reflect updated values. Messages from Simphony are either tagged with the project name, or with "Report." As such, messages that are tagged with the project name are meant for the as-built federate, or any other federates which accept messages with this tag.

As it is shown in Figure 6-14, when the as-built federate receives a reflect attribute value message, or receives an interaction from RTI, it checks the user-supplied tag to match it with a project name in the database. The same method is used in Simphony when an element in the template receives a message from the as-built or another federate.

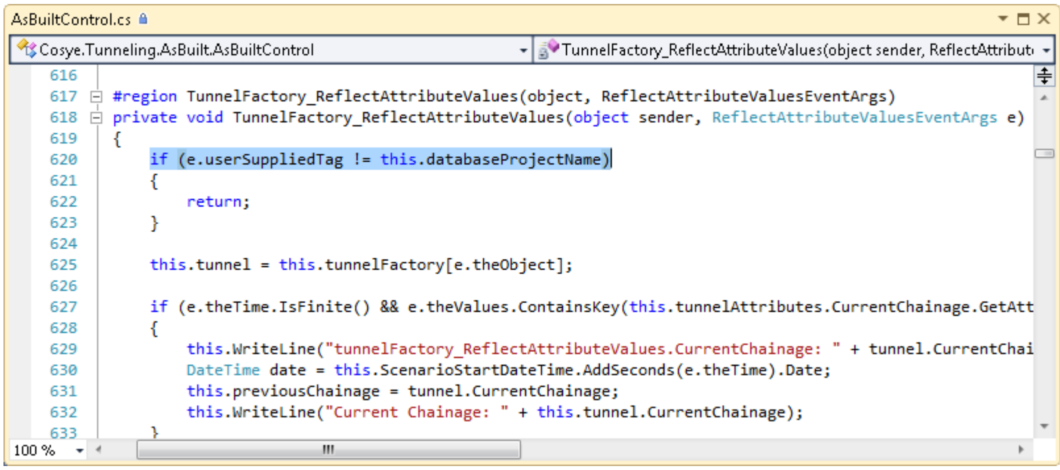


Figure 6-14: Implementation of User-Supplied Tag in the As-Built Federate

Before any scope changes happen in the project, two Simphony models may join the federation to model original and as-built process models. In this case, during the federation execution, only the as-built model should reflect changes, while the plan model runs with the original inputs (Figure 6-15).

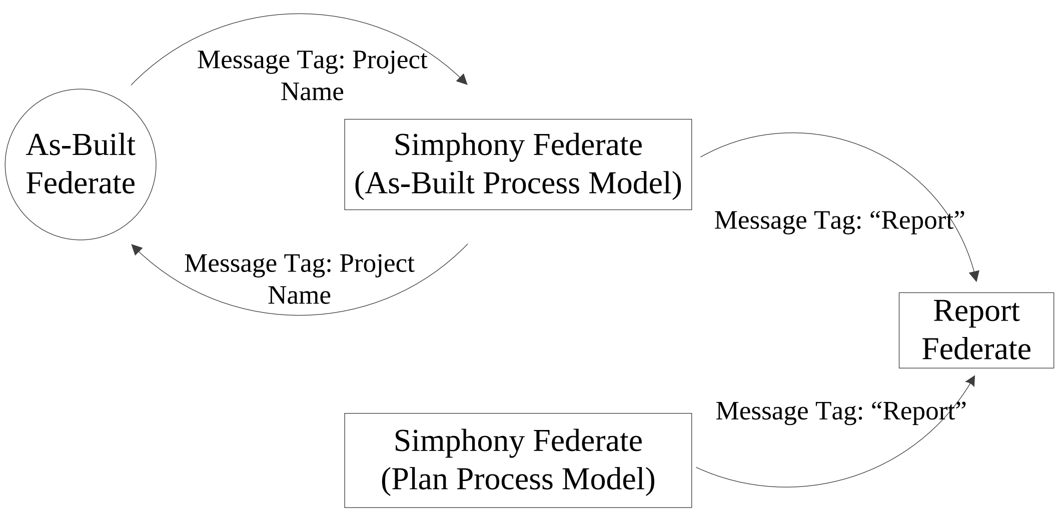


Figure 6-15: The Information Flow to Update Simphony Model in the Federation

After a scope change is recorded in the database and a Simphony federate with a new process model joins the federation, the updating procedure will become more complicated. For example, if a scope change happens today, a modeller creates a new model, and then at the start of the federation execution, adds it to the federation as a new Simphony federate. When a federation runs up to the current date, both as-built models receive updates from the as-built federate, so they will demonstrate the same results and same process; however, from this point forward, since there is no site data stored in the database, two as-built models (with different processes) should start simulating based on user-defined input variables. But if documentation of as-built data continues, then the as-built Simphony model that does not include changes should stop getting updates, while the revised process model should update the process based on actual data recorded in the database, which means that the only model that should reflect updated values is the new as-built model with the process changes implemented.

To document a process change in Simphony, the first step is to store its information in the database. Therefore, a table called “ChangeOrders” is created to store name and start date of the change. This is the date that the as-built model needs to recognize to start simulating based on original user inputs, and the revised as-built model should either simulate based on original inputs, or receive updates from the as-built federate. Figure 6-16 illustrates the structure of federation after a revised model is added. In this situation, messages for the purpose of as-built documentation can be divided into three main formats: updates meant for the as-built federate should be tagged with the name of a project element which contains the name of the project in the as-built database, updates meant for the report federate should be tagged with “Report,” and updates meant for Simphony are those which exactly match the name of the project element.

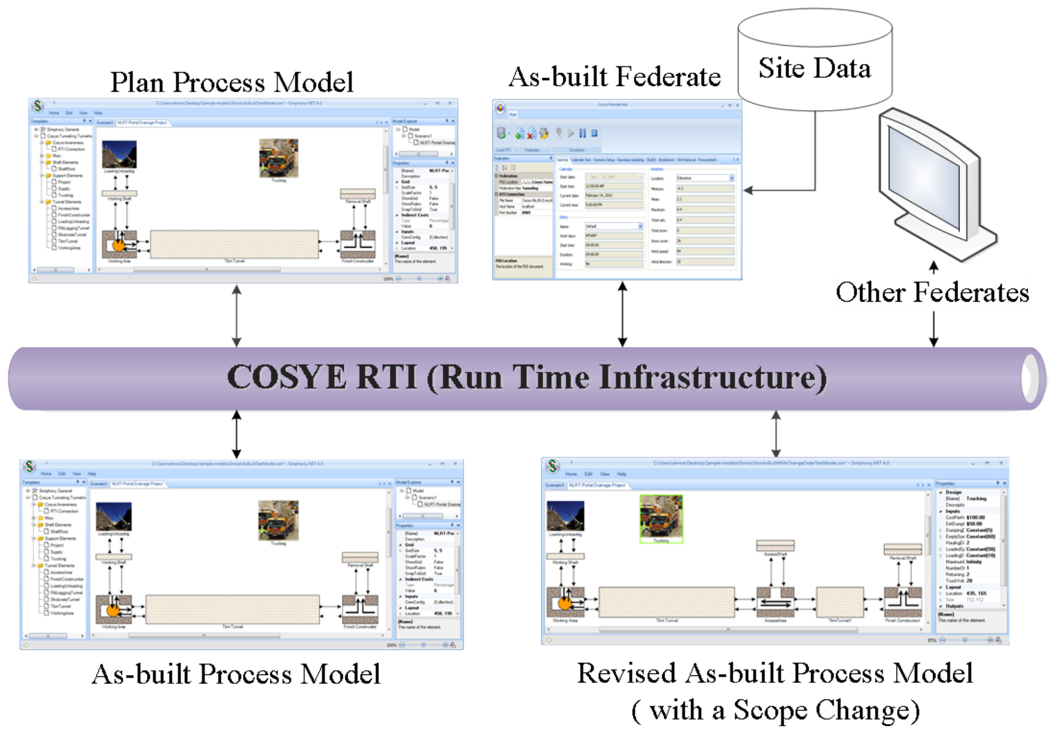


Figure 6-16: As-Built Process Documentation using Three Instances of Simphony in COSYE

If as-built data is recorded after the change happens in the project, the name of the scope change is used in the "Work Area" field in the "DailyRecords" table. The updating process will progress the same as described in section 6.3.

6.6 Implementation of Report Federate

The report federate is basically a listener in the federation; it collects information about all the process models simulated in Simphony federates. It is a federate that only subscribes to the objects and their attributes, and does not publish any additional information to the federation.

The report federate only responds to the messages tagged with a user-supplied tag equal to "Report," and based on the updates it receives, the report federate creates certain types of records. In the federation, only Simphony federates send updates with user-supplied tags equal to "Report," so all other updates will be ignored by the report federate.

The report federate utilizes Crystal Reports, a general reporting tool that is bundled with Visual Studio.Net to generate report layout, tables, and charts. Crystal Reports is capable of grouping data, creating sub reports, and developing interactive graphs.

To create different types of reports for management purposes, records are organized and grouped together internally; for instance, data can be recorded based on the work area. In this federation, the report federate can provide the following type of reports:

- Project Performance Report that includes current state and percent complete of the tasks for each work area and activity within that work area.
- Project Progress Report that reports all the details of project progress on a daily basis including percent complete, interruption cause and duration, shift information, and working hours for each task in a work area, and for each activity that this task is part of.
- Project Schedule Report that provides the summary and detailed reports of task start and finish dates (estimated finish date for the tasks that are in progress) in a table or graph format.
- Resource Utilization Report that includes utilization of each resource up to the current date, and the overall forecasted utilization for each Simphony federate.
- Resource Allocation Analysis that includes details of resource allocation such as number of each resource for each work area in the project, on a daily basis.
- Project Cost Report that reports equipment, crew, material, and indirect cost up to the current date, and the overall forecasted cost for each item based on simulation final results.
- Project Earned-Value Analysis Report that provides a graph and a table to show the total cost of the project up to the current date, and the overall forecasted cost for the plan and as-built Simphony federates.

- Change Order Report that reports changes, and the detailed and overall cost of changes in the project.

All these reports change dynamically during the simulation run. A user can open and print a report from the report federate or export it to the Excel file.

In this research, additional features were developed in the Simphony template to be able to generate and display specific records in the reports. For instance, the calculation of task progress on a daily basis, especially when Simphony is not receiving as-built updates, would be a challenging task in DES modeling. A “RemainTime” attribute was introduced for each event in the Simphony to calculate the remaining duration of an event, the elapsed duration, and the percent complete at the end of each working day. Also, for calculation of resource utilization on a daily basis, especially for the TBM, crew, train, and crane, a unique class was developed in the template that basically has three functions: Collect, Reset, and GetUtilization. The Collect function collects observations of whether or not the resource is currently in use. If the resource is in use, the elapsed time from when it was last used to the current time is accumulated. The Reset function resets any values previously collected, and the GetUtilization function returns the utilization of the resource at the specified time by taking the duration of how long the resource has been in use divided by the amount of time elapsed since the resource was first in use (total duration).

6.7 Documentation of Product Model

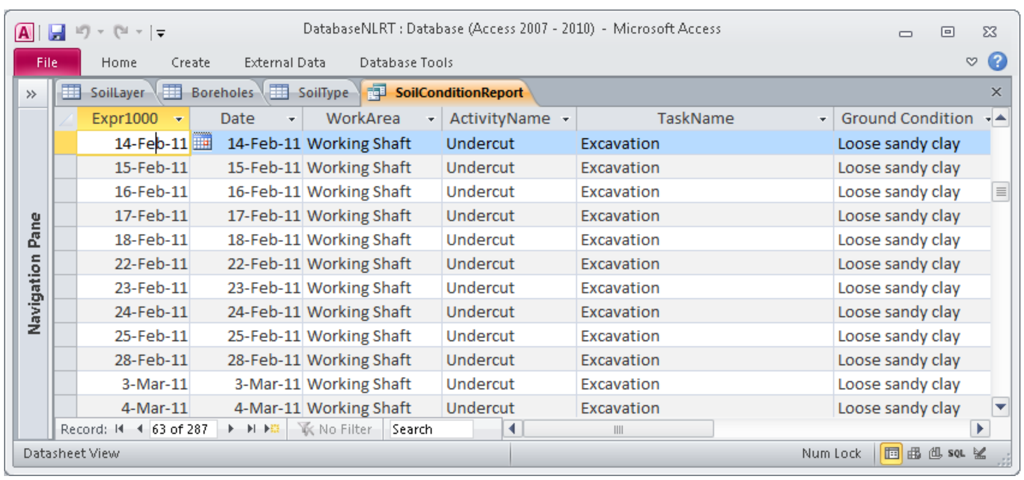
In addition to the documentation of the as-built process model in the distributed simulation framework, the proposed application aims to document the as-built product model to provide an integrated documentation system that assists project managers in different decision-making processes.

In this research, the main focus is to create the integrated system through utilization of distributed simulation in the COSYE environment. The ability of COSYE to integrate various simulation components and software applications such as Simphony and CAD (Computer-Aided Design) systems facilitates as-built documentation of process and product models in the federation. To document product model, the as-built database in the federation can store all the product information in different tables and queries and send them through the as-built federate to the Simphony or other federates. It is important to mention that product data can vary from project to project; so tables and queries in the database should be specifically designed according to the project type. For instance, in this research, the tunnel alignment and soil profile changes along the tunnel are the most important data that should be collected during construction phase.

6.7.1 Documentation of Soil Profile Changes Along the Tunnel

In tunnel construction projects, geotechnical investigation is conducted to provide geotechnical subsurface information along the proposed tunnel alignment. To study soil characteristics in the construction site, a number of vertical boreholes should be drilled along the proposed alignment. These boreholes provide soil information in the locations that they have been excavated and it is a geotechnical engineer's responsibility to define soil profile between the boreholes. In the construction phase, soil characteristics can affect construction in different aspects; the type of excavation equipment is determined based on soil condition at the tunnel location, and an unpredicted and adverse soil condition can decrease tunnel advance rate compared to the assumptions made during the planning stage.

In the Simphony tunnel template, a tunnel element is designed to consider soil section properties including name, swell factor, and excavation advance rate in that soil type. In the tunnel element, a total length can be divided into multiple sections based on the soil layers that tunnel passes through. In this element, the user can specify different sections with associated soil type along the tunnel. In the planning stage, soil layers are modeled based on the geotechnical report. If soil types are identified in the borehole locations, then it is a modeller's responsibility to define soil profile and calculate the length of tunnel sections accordingly. Different methods and geotechnical software are available to predict soil types along the tunnel direction (Ruwanpura and AbouRizk 2001), but during the construction period, a contractor is able to examine and specify soil type as excavation progresses. Soil characteristics can be documented in a site report on a regular basis. In this research, a soil condition is recorded and stored in the "DailyRecord" table in the database, and then a query is created to extract soil data in a separate table, as it is shown in Figure 6-17.

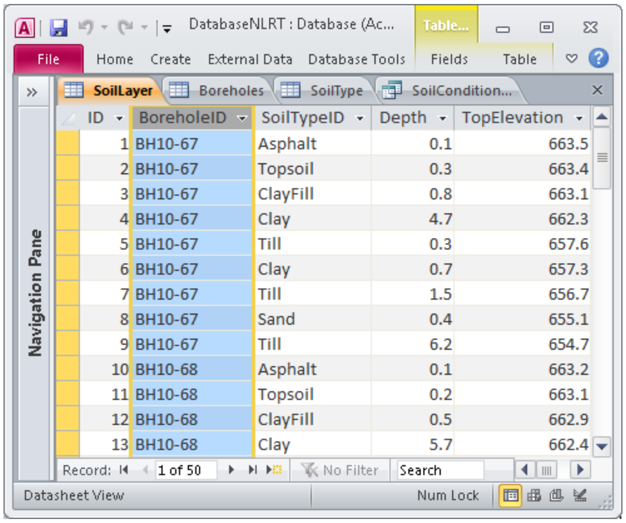


Expr1000	Date	WorkArea	ActivityName	TaskName	Ground Condition
14-Feb-11	14-Feb-11	Working Shaft	Undercut	Excavation	Loose sandy clay
15-Feb-11	15-Feb-11	Working Shaft	Undercut	Excavation	Loose sandy clay
16-Feb-11	16-Feb-11	Working Shaft	Undercut	Excavation	Loose sandy clay
17-Feb-11	17-Feb-11	Working Shaft	Undercut	Excavation	Loose sandy clay
18-Feb-11	18-Feb-11	Working Shaft	Undercut	Excavation	Loose sandy clay
22-Feb-11	22-Feb-11	Working Shaft	Undercut	Excavation	Loose sandy clay
23-Feb-11	23-Feb-11	Working Shaft	Undercut	Excavation	Loose sandy clay
24-Feb-11	24-Feb-11	Working Shaft	Undercut	Excavation	Loose sandy clay
25-Feb-11	25-Feb-11	Working Shaft	Undercut	Excavation	Loose sandy clay
28-Feb-11	28-Feb-11	Working Shaft	Undercut	Excavation	Loose sandy clay
3-Mar-11	3-Mar-11	Working Shaft	Undercut	Excavation	Loose sandy clay
4-Mar-11	4-Mar-11	Working Shaft	Undercut	Excavation	Loose sandy clay

Figure 6-17: Daily Soil Data during Construction Phase

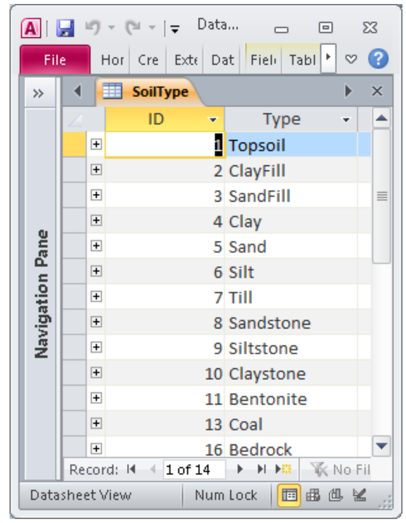
Three different tables that contain information from the geotechnical report are also created in the as-built database; a "Borehole" table contains all the borehole information from the geotechnical report, a "Soil Type" table lists all the possible soil types in the construction site, and a "Soil Layer" table specifies soil layers in all the boreholes. Figure

6-18 shows screenshots of the three tables. This data is also utilized to develop soil profile in the 3D visualization federate in the tunnel federation.



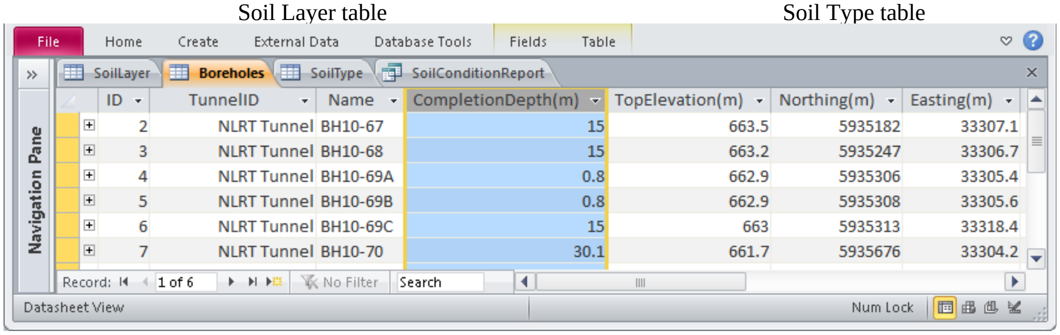
ID	BoreholeID	SoilTypeID	Depth	TopElevation
1	BH10-67	Asphalt	0.1	663.5
2	BH10-67	Topsoil	0.3	663.4
3	BH10-67	ClayFill	0.8	663.1
4	BH10-67	Clay	4.7	662.3
5	BH10-67	Till	0.3	657.6
6	BH10-67	Clay	0.7	657.3
7	BH10-67	Till	1.5	656.7
8	BH10-67	Sand	0.4	655.1
9	BH10-67	Till	6.2	654.7
10	BH10-68	Asphalt	0.1	663.2
11	BH10-68	Topsoil	0.2	663.1
12	BH10-68	ClayFill	0.5	662.9
13	BH10-68	Clay	5.7	662.4

Soil Layer table



ID	Type
1	Topsoil
2	ClayFill
3	SandFill
4	Clay
5	Sand
6	Silt
7	Till
8	Sandstone
9	Siltstone
10	Claystone
11	Bentonite
13	Coal
16	Bedrock

Soil Type table



ID	TunnelID	Name	CompletionDepth(m)	TopElevation(m)	Northing(m)	Easting(m)
2	NLRT Tunnel	BH10-67	15	663.5	5935182	33307.1
3	NLRT Tunnel	BH10-68	15	663.2	5935247	33306.7
4	NLRT Tunnel	BH10-69A	0.8	662.9	5935306	33305.4
5	NLRT Tunnel	BH10-69B	0.8	662.9	5935308	33305.6
6	NLRT Tunnel	BH10-69C	15	663	5935313	33318.4
7	NLRT Tunnel	BH10-70	30.1	661.7	5935676	33304.2

Boreholes table

Figure 6-18: Soil and Boreholes Information Tables Created During Planning Stage

6.7.2 Documentation of Tunnel Alignment

In construction projects, as-built documentation usually refers to creating as-built drawings in CAD systems. A traditional way to capture project as-built geometry is through the surveying data, but new methods such as GPS, Image processing, and video capturing are also utilized to capture as-built geometry of a project and update CAD drawings accordingly. In this project, surveying information is used to document tunnel as-built geometry. The surveying data is captured in daily records, then transferred to the as-built database and stored in the "DailyRecord" table.

In the tunnel project, 3D points along the tunnel can provide the actual alignment of the tunnel, so if the original alignment is captured from 2D drawings, then both original and as-built alignments can be presented in the records or in the CAD system. Once as-built data is captured, then the actual alignment can be created using surveying information. This will help in detecting changes in the product.

6.8 Verification and Validation of the Proposed Framework

In simulation, verification determines whether a model is performing as intended (Law and Kelton 1991). In this project and during the programming phase, various methods were applied to verify the simulation model and check if the programming codes contained any errors (known as bugs). In the federation, and especially in the Simphony template and as-built federate, the simulation logic, entity flow, timing, and distribution sampling were checked on a regular basis by reading through the written codes and watching the behaviour of the model against the real system (Robinson 1997). Also all the messages and interactions in COSYE were checked to ensure they send the right information to the right federate at the right time.

Two other techniques were used for simulation verification (Klenijnen 1995; Lucko and Rojas 2010); the simulation trace file and construction process visualization. Checking of intermediate simulation outputs through the use of trace commands both in the Simphony model and the as-built federate was really helpful to compare simulation results with the user expected values, and the development of the 3D visualization federate in the tunnel federation helped to find programming errors in the Simphony template by checking the animated process and comparing it with an actual construction process.

In the simulation system, validation ensures that "the model is sufficiently accurate for the purpose at hand" (Carson 1986). For validation of the framework, a face validation approach through a case study is applied (Lucko and Rojas 2010). In this method, a real project is simulated using a developed system and to validate the model, it is expected that the simulation outputs "be sufficiently similar to the real project outcomes" (Robinson 1997). In this research, since the proposed framework aims to document the as-built project information through a Simphony model, the outputs of the Simphony model should be similar to the actual project outcomes. This will happen if the process model in Simphony accurately reflects all the changes that happen in the real project.

To validate the framework, different tunnel construction projects can be utilized as case studies. Therefore, the NLRT drainage tunnel project, an ongoing tunnel construction project in Edmonton, Alberta, Canada, was selected to validate the framework. The main reason for this selection was that the as-built data were precisely documented during the construction phase of the project, which meant that more information was available from the real world project to compare with the simulation results to make validation of the system easier and more accurate.

The NLRT drainage tunnel project consists of a 500m tunnel that is constructed using TBM tunneling method. It is a one-way tunnel in which the TBM starts tunneling from the working shaft and advances to the end of the tunnel where it is retrieved from the removal shaft. As discussed in Chapter 5, a Simphony model was developed for this project for validation of the Simphony template. Therefore, in this phase, a planned process model previously created in Simphony can join the federation.

During the construction phase, a daily site report in the form of a PDF file was developed to collect site information on a daily basis. Figure 6-19 illustrates a sample report that was filled in and sent to the project manager.

Edmonton ASSET MANAGEMENT AND PUBLIC WORKS DESIGN AND CONSTRUCTION

DAILY PROGRESS REPORT

A. Project Description

Project Name:	NLRT – Portal Drainage Project
Date:	March 04, 2011
Contractor:	Drainage Design & Construction, AMPW Drainage Services, COE
Site Inspected By::	Nalaka Morugama
Time Of Inspection:	9.30 AM
Weather:	Forecast: Mixed Sun & Clouds — Low Temperature: -23° C; High Temperature: -14° C

B. Resource Summary

Task. Location & Activity	*Manpower*	*Equipment*
103 St/105 Ave North Tunnel undercut	Top man : 1 Tunnel Crew : 4 Operator : 1	Mobile Crane : 1 Generator : 1 Compressor : 1 Heater/Ventilator : 1

C. Current Project Issues

Design Issues:	N/A
Utilities :	N/A
Construction::	Loose sandy clay

D. Task Summary

Task	*Completed (%)*	*Pictures (No)*	*Progress Description*
1. Tail tunnel construction	100%		Tail tunnel complete by Dec 13, 2010
2. Undercut for main tunnel	83.3%		Crew has installed Rib#8 (upper half). Today Crew is digging to install legs for R#6. Up to date lower half 5.0m (of 7.5m) and upper half 8.0m (of 7.5m) complete. Ground is loose sandy clay.

Figure 6-19: Daily Site Report for the NLRT Drainage Tunnel Project (City of Edmonton, 2011)

A Microsoft Access database was created to document the information collected from the construction site. In addition to the daily reports, soil profile and tunnel alignment data from the planning stage were also documented in separate tables in the database.

The as-built process and product model were documented for this project utilizing the tunneling federation in COSYE (Figure 6-20). Both models simulate the same construction process, but the as-built process simulates the actual state of a project based on daily reports that are documented in the database.

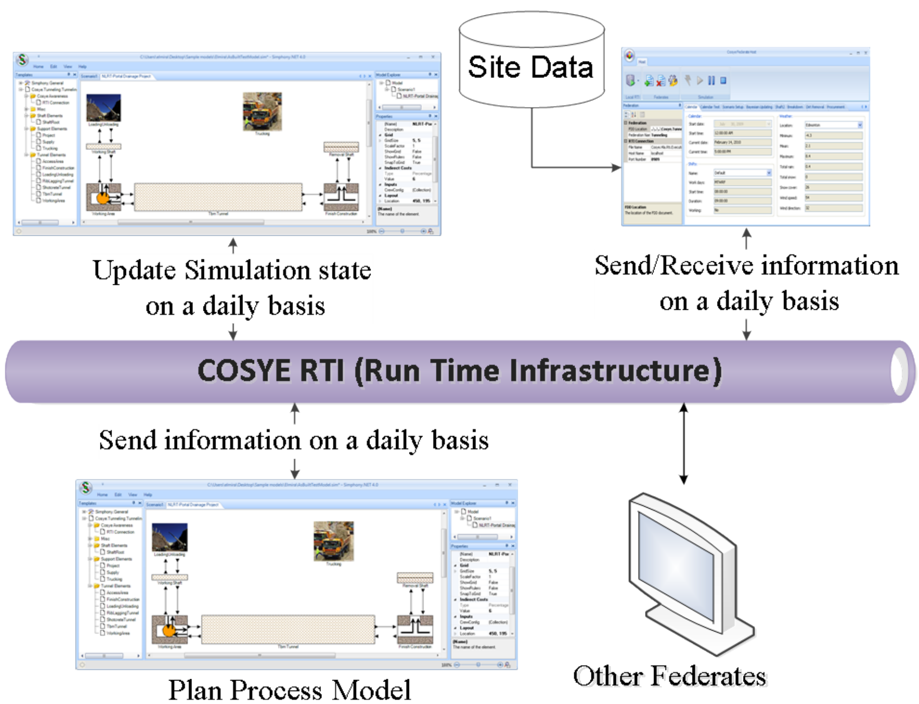


Figure 6-20: As-Built Documentation of Construction Project with No Scope Changes

After implementation of this phase of the project, and validation of simulation results based on report analysis, the next phase was to document scope changes in the project. It is assumed that a major breakdown happens for the TBM when it reaches the constructed chainage of 361m, and to fix it, the crew should remove the TBM from the tunnel. Since TBM machines are not able to move backwards in the tunnel, the solution is a construction of an access shaft and an enlarged area to provide access in front of the TBM to fix it in its current location, or remove it. A template was modified to model the access shaft and access area in the model. The access shaft is created using the shaft element, and the access area is modeled the same as finish construction element, with an attribute that defines a duration (delay) that includes the TBM repair time. The remaining section of the tunnel would start after the access shaft construction and delay duration. Figure 6-21 shows a new simulation model with changes implemented in the process.

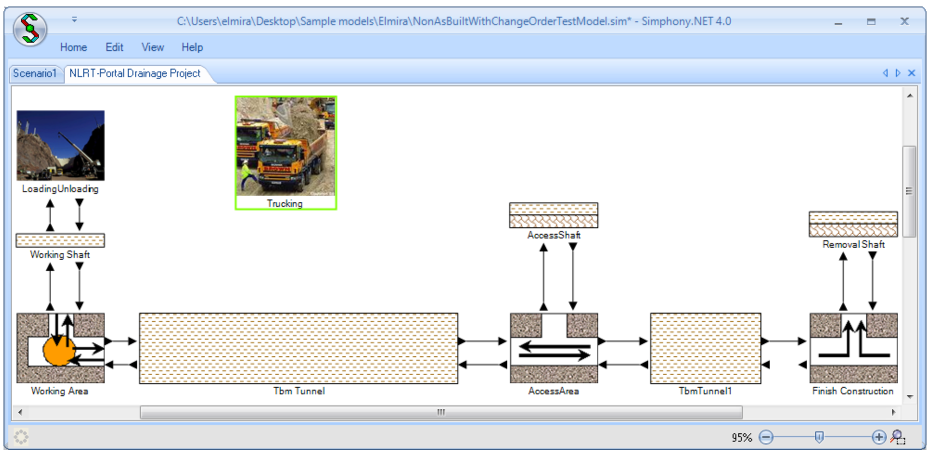
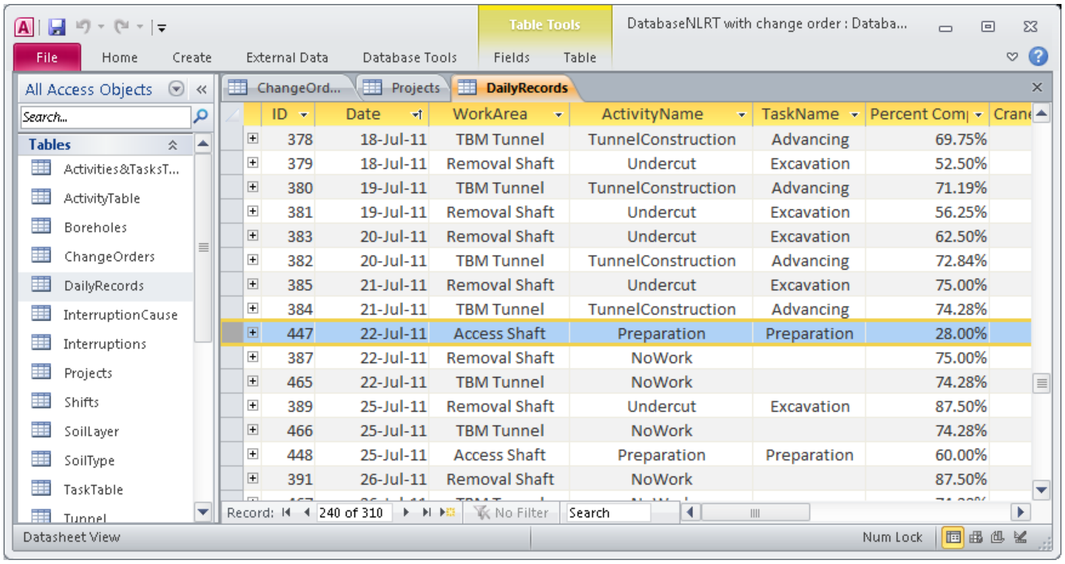


Figure 6-21: A Tunnel Project Simulation with a New Process Model

To document a scope change in Simphony, the first step is to store its information in the database. In this project, a table called “ChangeOrders” is created to store name and start date of the change. This is the date that the as-built model should start simulating based on original user inputs, and the revised as-built model should either simulate based on original inputs or receive updates from the as-built federate.

In the case of recording as-built data after the change happens in the project, the name of the scope change is used in the “DailyRecords” table, in the “Work Area” field, and during daily documentation of project state. Figure 6-22 shows a snapshot of a database table with changes that are documented on a daily basis. Based on the type of change that happens in a project, a user can select different ways of documenting it. The updating process will continue as described previously.



ID	Date	WorkArea	ActivityName	TaskName	Percent Com	Cran
378	18-Jul-11	TBM Tunnel	TunnelConstruction	Advancing	69.75%	
379	18-Jul-11	Removal Shaft	Undercut	Excavation	52.50%	
380	19-Jul-11	TBM Tunnel	TunnelConstruction	Advancing	71.19%	
381	19-Jul-11	Removal Shaft	Undercut	Excavation	56.25%	
383	20-Jul-11	Removal Shaft	Undercut	Excavation	62.50%	
382	20-Jul-11	TBM Tunnel	TunnelConstruction	Advancing	72.84%	
385	21-Jul-11	Removal Shaft	Undercut	Excavation	75.00%	
384	21-Jul-11	TBM Tunnel	TunnelConstruction	Advancing	74.28%	
447	22-Jul-11	Access Shaft	Preparation	Preparation	28.00%	
387	22-Jul-11	Removal Shaft	NoWork		75.00%	
465	22-Jul-11	TBM Tunnel	NoWork		74.28%	
389	25-Jul-11	Removal Shaft	Undercut	Excavation	87.50%	
466	25-Jul-11	TBM Tunnel	NoWork		74.28%	
448	25-Jul-11	Access Shaft	Preparation	Preparation	60.00%	
391	26-Jul-11	Removal Shaft	NoWork		87.50%	

Figure 6-22: Scope Change Documentation in the Daily Record Table

After the simulation execution for this project, different reports are created in the report federate that are used for validation of the federation. Figure 6-23 shows various graphs created in different reports for this project.

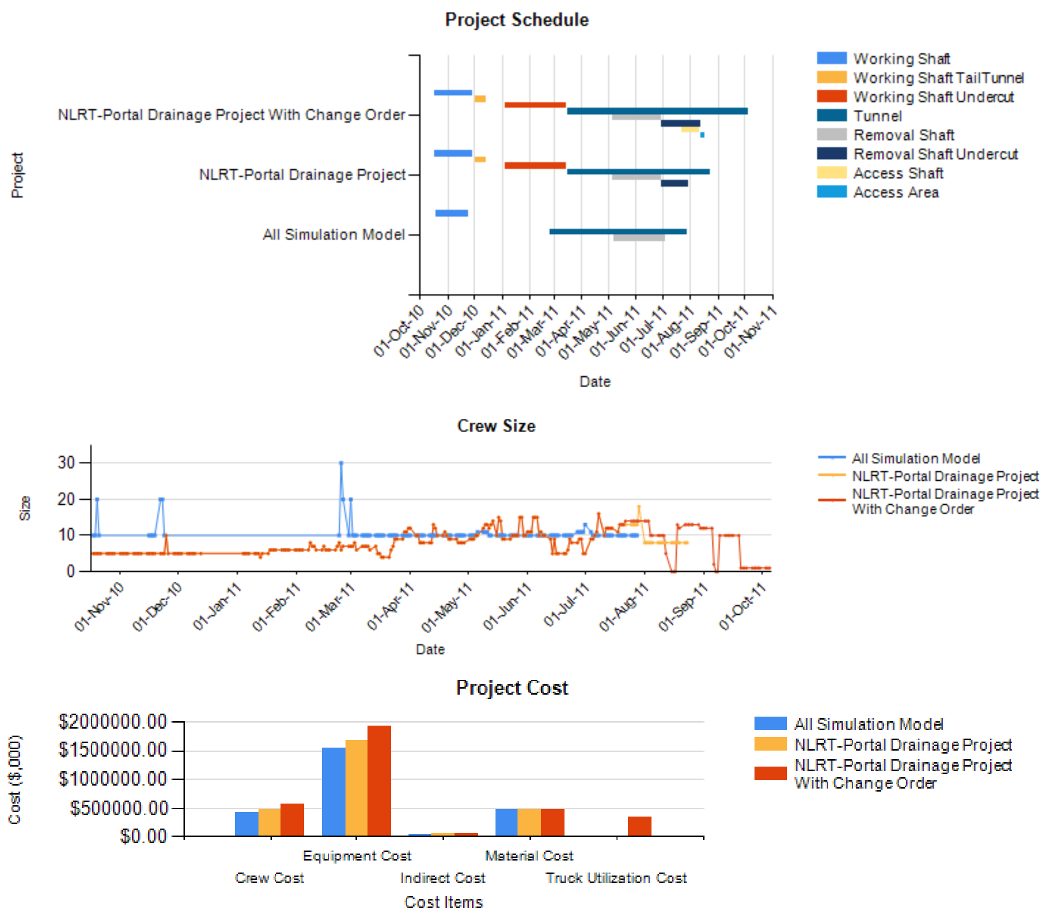


Figure 6-23: Sample Graphical Reports for the NLRT Drainage Tunnel Project

6.9 Conclusion

This chapter discusses the development of the framework in the COSYE environment. A tunnel construction project was selected for the implementation process. Two main phases were defined to facilitate federation development and as-built documentation of the tunnel project. In the first stage, it was assumed that the process model remains the same as planned and only Simphony inputs including tasks duration, number of resources, and project progress should be updated based on daily records, which are collected from the construction site. To document as-built information, a Simphony template was modified extensively to record the actual construction process and its detailed information including project progress, resource utilization, process interruptions, and project cost, on a daily basis. After completion of this phase and validation of the framework, in the second phase, the documentation of scope changes in the project was the main focus. The framework structure was modified to include multiple Simphony federates that need to be updated based on the messages being sent by the as-built federate. After implementation of this phase of the research, the ongoing project was used to validate the entire framework.

Chapter 7 - Conclusions and Recommendations

7.1 Research Summary

As-built information may refer to the documentation of construction production through a CAD system to show the final condition of the work as it was actually constructed. In this research, as-built documentation refers to the collection of various as-built documents including construction process information, product data, site daily records, and all information related to the project that can be captured and stored, especially during the construction phase. This information can be utilized during the construction phase of a project for project monitoring and control purposes, and it can be stored for post-project use such as lesson-learned studies, claim analysis, and dispute resolution, as well as future project planning. Studies show that complete and accurate as-built documentation is essential in project disputes, as it helps in the reconstruction of project history (Kangari 1995).

A case study, conducted for a post implementation review of a City of Edmonton tunnel construction project (presented in Chapter 3), was a motivation for this research project, as it shows that common as-built documentation does not provide sufficient detail of the construction process and its dynamic nature. Literature review also demonstrated that methods available for documenting as-built process or work flow through the CPM network are not capable of capturing resource interaction and detailed information about activity progress, delays, and resource utilization in an integrated and dynamic fashion and on a regular basis (daily). Therefore, the objective of this research was to find a method to record all aspects of a project during the planning and construction phases so that in the future, project managers can reconstruct an accurate and complete story of a project for possible future claims and other management purposes.

Based on this overall objective, the primary objective of this research was to record as-built project information in an integrated manner that includes the product data, site condition, dynamic aspects of a project, and all the information regarding work flow, activity progress, and resource interaction. The proposed framework and its components are described in Chapter 4 of this document.

To document the entire project history, especially the process data, a simulation technique was defined as a promising solution; this technique can easily demonstrate all the features of process-related information in a dynamic and integrated format, and it was successfully utilized to model the construction process in several studies in the past. A Simphony application was chosen as a discrete event simulation tool to model the construction process for both planned and as-built situations. Chapter 5 outlines the development of the Simphony template for tunnel construction projects for the purpose of as-built documentation.

To develop a generic framework in this research, a distributed simulation method was introduced to implement the entire system to allow the reusability and extensibility of the framework in future and for other types of construction projects.

A COSYE environment, developed based on HLA rules for distributed simulation development, was chosen to implement the framework. In the proposed federation, a Simphony model can join as a component of the framework, and a simulation controller is responsible for updating the Simphony model based on as-built information documented in a Microsoft Access database. After the framework implementation, which is described in detail in Chapter 6, an ongoing tunnel project was selected as a case study for validation of the proposed framework.

7.2 Research Contributions

The main contribution of this research is the development of a generic framework that allows users to document as-built process information including activity sequence, resources, and external factors, as well as product model and static project information, in a single simulation system, to demonstrate the full story of what happened in the site, and compare it with the original plan. As discussed earlier, although a considerable amount of work has been done in academia in the area of information documentation for different project management purposes, most work has focused on capturing the product information, or developing information management systems; the work is usually limited in scope and does not document the dynamic aspect of projects in an integrated manner. This research provides a new tool that helps managers to access the detailed information of construction processes for various purposes such as quality control, claims and disputes, process improvement, lesson-learned studies, and future project planning.

High Level Architecture technology is utilized to standardize the integration process between simulation systems and different computer software, to support interoperability, reusability, and extensibility of the framework, and to provide a generic solution.

7.2.1 Academic Contribution

From the academic perspective, this research proposed a new system that utilizes simulation technique to enhance as-built documentation by integrating actual data with simulation data. In this system, simulation can be utilized during both the planning and construction phases to not only help managers in decision-making, but also to assist them in recording what is actually happening during the construction stage. In addition to the documentation of static project information, the proposed framework provides valuable dynamic information of a project such as work flow, resource utilization, and the effect of external factors in an integrated format, which can only be achieved through the simulation modeling approach.

In this framework, a distributed simulation environment allows users to run multiple instances of a process model and compare the results in a report format generated by a separate component of a system. Therefore, a management group receives the outcome of a comparison without requiring programming and simulation knowledge.

This research also utilizes a hybrid simulation modeling technique that combines the time-step simulation approach with discrete event simulation. Using a hybrid model enables control of the advancement of a system on a daily basis to update the state of a project according to the data received from the construction site.

7.2.2 Industrial Contribution

The actual framework developed during this research can be implemented in the involved company's new projects to help project managers control the actual progress, resource allocation, and process workflow, and use the documented information later on for lesson-learned studies, claims, and future project planning in the organization.

Some of the components of the proposed framework, including the data collection system, and a simulation template for tunnel construction projects, are already being used for City of Edmonton projects.

Study of the available commercial information management applications shows that in these systems, the entire module usually works as a warehouse for different documents such as payroll, meeting minutes, request orders, activity information, resources, costs, inventories, etc., but collected data is presented in separate tables and is not integrated. Therefore, if decision makers need to derive new information such as resource interaction, which is not collected and calculated, it can be challenging and complicated. Also, current applications do not capture the dynamic nature of the construction process, so they are not able to provide the full story of a project when it is completed.

7.3 Limitations

Implementation and validation of the proposed framework through different tunnel construction projects demonstrated that the simulation-based framework for as-built documentation can be successfully utilized in the real environment and for various tunnelling projects. However, the ultimate goal of this research is to develop a generic framework that can be employed in other types of construction projects. Therefore, the distributed simulation environment was selected as a development environment to facilitate interoperability of multiple components of a simulation system while supporting the reusability and extensibility of each component for future use. However, due to reasons listed below, it may require additional effort to utilize the framework in other types of projects.

- The existence of any sort of data collection system in an organization would help in the implementation phase of the framework, and show the value that this system adds to the as-built data collection and storage process. Since the as-built data collection takes a lot of time and effort, the organization should consent to utilize resources both in the site and the offices, and spend time and money as required, based on the technology and procedures used in the projects. In the City of Edmonton tunnel projects, the data collection was part of a project management process, so the implementation of the framework was easier, and the information that it provided for the project managers was in detail and included site information.

- Development of a Simphony template for the purpose of as-built data documentation requires thorough knowledge of a construction project such as activities, work flow, resources, external factors, and site data. This requires a detailed study of the construction project and the capture of expert knowledge through interviews or questionnaires. In addition, Simphony should be able to support the development of the template considering all the details regarding the project as-built information. In this research and in the template development phase, Simphony could not support development of particular details such as shift documentation, resource utilization, and other as-built data that needed to be captured by a simulation model. Therefore, it was time consuming to add new features to Simphony, or to add details through programming.
- During this research project, COSYE was still in the development process, and there were limitations in COSYE regarding development of the framework. Therefore, many features were added to COSYE during framework implementation. In some cases, simplified solutions were considered to model the as-built and report federates. While such factors are still missing in COSYE, the framework implementation process may take a lot of time and effort.

7.4 Recommendations for Future Research and Development

This research demonstrated that the simulation-based framework is a powerful and effective technique for as-built documentation that can capture process and product information, as well as site data and resource interaction during the construction phase.

The proposed framework mainly focused on the dynamic aspects of the project, and as-built process documentation, and there are still issues that need to be considered for comprehensive implementation of this methodology in the future. Recommendations for future study are as follows:

- There are several studies in the field of automated data collection in different construction projects that demonstrate implementation of different systems in construction sites, depending on the type of project, and the information that needs to be captured and stored for various purposes. In this research, since the main focus was the documentation of the as-built process and product model through simulation technique, the automation of data collection was not the focus of the research. So for future research, the integration of automated data collection systems within the framework could provide information in a shorter time and in a more organized manner.
- This framework can be utilized for other kinds of construction projects. The current distributed simulation framework enables users to replace the tunnel construction simulation models with different models created in Simphony, while the structure of the framework remains the same. To extend this methodology to other types of projects requires detailed study of the construction domain and creation of a comprehensive simulation template in Simphony that can capture as-built information.
- The documentation of product as-built information in this project was performed in a simple way based on data stored in the database. It would be of great benefit if the product could be illustrated visually to give a clear idea of exactly what has been constructed compared to plan data. This would provide a control tool for project managers to accurately monitor the as-built product model and discover

any deficiencies as compared to the plan. Development of the framework in the COSYE environment gives the opportunity to demonstrate product information in a separate component such as a CAD federate that utilizes 2D/3D CAD software applications. This development could increase the tendency toward employment of the system in the industry.

- Although the proposed framework is designed to document the actual project story for future use, it can be effectively employed during the construction phase to help project managers monitor, control and predict project performance. The ability of the framework to document as-built process and product information during the construction phase can be used for real-time project monitoring and early error detection in project performance. Also, implementation of the framework in the COSYE environment allows integration of other tools and software applications. Therefore, different methods such as expert systems and Artificial Intelligent techniques can easily be utilized in the framework to provide an effective project monitoring and control application.

The future vision for the proposed framework is that it may be used as a comprehensive tool that automatically captures as-built product and process information, including dynamic aspects of the project, and records it through a simulation model in the distributed simulation environment that provides management reports and visualization systems to demonstrate the planned and as-built product model. This application would generate the story of the project for lesson-learned studies, claims and dispute resolution, as well as project planning and control during the construction phase.

References

AbouRizk, S. M. (2006). “Collaborative simulation framework for multi-user support in construction.” *Discovery Grant Proposal*, University of Alberta, Edmonton, AB, Canada.

AbouRizk, S. (2010). “The role of simulation in construction engineering and management.” *J. Constr. Engrg. and Mgmt*, ASCE, 136(10), 1140-1153.

AbouRizk, S., and Dozzi, S. (1993). “Applications of computer simulation in resolving construction disputes.” *J. Constr. Eng. Manage.*, 119 (2), 355–373.

AbouRizk, S., and Hajjar, D. (1998). “A framework for applying simulation in the construction industry.” *Can. J. Civ. Eng.*, 25(3), 604-617.

AbouRizk, S., and Hague, S. (2009). “An overview of the COSYE environment for construction simulation.” *Proc., Winter Simulation Conference*, New York, NY, USA, 2624-2634.

Abourizk, S. M., Hague, S., and Moghani, E. (2009). “Developing a bidding game using high level architecture.” *Proc., ASCE International Workshop on Computing in Civil Engineering*, Austin, TX, USA, 346, 513-522.

Abourizk, S. M., Hague, S., Mohamed, Y., and Robinson, F. A. (2010). “An HLA-based bidding game with intelligent virtual player.” *Proc., Winter Simulation Conference*, Baltimore, MR, USA, 3056-3068.

AbouRizk, S., Mohamed, Y., Taghaddos, H., Saba, F., and Hague, S. (2010). “Developing complex distributed simulation for industrial plant construction using High Level Architecture.” *Proc., Winter Simulation Conference*, Baltimore, MD, USA, 3177-3188.

AbouRizk, S. M., Ruwanpura, J. Y., Er. K. C., and Fernando, S. (1999). “Special purpose simulation template for utility tunnel construction.” *Proc., Winter Simulation Conference*, Phoenix, AZ, USA, 948-955.

Ailland, K., Bargstädt, H. J., and Hollermann, S. (2010). “Construction process simulation in bridge building based on significant day-to-day data.” *Proc., Winter Simulation Conference*, Baltimore, MD, USA, 3250-3261.

Akinci, B., Kiziltas, S., and Pradhan, A. (2006). “Capturing and representing construction project histories for estimating and defect detection.” *13th EG-ICE Workshop*, Ascona, Switzerland.

Al-Bataineh, M. T. (2008). “Scenario-based planning for tunneling construction.” PhD thesis, University of Alberta, Edmonton, AB, Canada.

Alvanchi, A., Lee, S. and AbouRizk, S.M. (2011). “Modeling framework and architecture of hybrid system dynamics and discrete event simulation for construction.” *J. Computer-Aided Civil and Infrastructure Engineering*, 26(2), 77-91.

Azimi, R., Lee, S., AbouRizk, S., and Alvanchi, A. (2011). “A framework for automated and integrated project monitoring and control system for steel fabrication projects.” *J. Automation in Construction*, 20(1), 88-97.

Boer, C. A. (2005). “Distributed simulation in industry.” Ph.D. Thesis, Erasmus University Rotterdam, Rotterdam, The Netherlands, <http://hdl.handle.net/1765/6925> (Oct. 1, 2011).

Carson, J. S. (1986). “Convincing Users of Model's Validity is Challenging Aspect of Modeler's Job.” *Ind. Eng.*, 18 (6), 74-85.

Chang, D. Y., and Carr, R. I. (1987). "RESQUE: a resource oriented simulation system for multiple resource constrained processes." *PMI Seminar/Symposium*, Milwaukee, WI, USA, 4-19.

Chassiakos A. P., and Sakellaropoulos S. P. (2008). "A web-based system for managing construction information." *J. Adv. Eng. Software*, 39(11), 865–876.

Cheng, T. and Teizer, J. (2010). "Real-time data collection and visualization technology in construction." *Proc., Construction Research Congress*, Banff, AB, Canada, 339-348.

Chin, S., Kim, K., and Kim, Y. (2005). "Generate-select-check based daily reporting system." *J. Comput. Civ. Eng.*, 19(4), 412-425.

Chin, S., Yoon, S., Choi, C., and Cho, C. (2008). "RFID+4D CAD for progress management of structural steel works in high-rise buildings." *J. Comput. Civ. Eng*., 22(2), 74–89.

Chiu, C.-Y., and Russell, A. D. (2011). "Design of a construction management data visualization environment: A top–down approach." *Automation in Construction*, 20(4), 399-417.

Chung, T. (2007). "Simulation - based productivity modelling for tunnel construction operations." PhD thesis, University of Alberta, Edmonton, AB, Canada.

COAA. (2007). "Construction work packages." *Construction Owners Association of Alberta*, AB, Canada.

Cox, S., Perdomo, J., and Thabet, W. (2002). "Construction field data inspection using pocket PC technology." *Proc., Distributing Knowledge in Building*, International Council for Research and Innovation in Building and Construction, CIB W78 conference.

Davidson, I., Skibniewski, M., (1995). “Simulation of automated data collection in buildings.” *J. of Computing in Civil Engineering*, 9(1), 9-20.

Dawood, N., Sriprasert, E., Mallasi, Z. and Hobbs, B. (2002). “Development of an integrated information resource base for 4D/VR construction processes simulation.” *Automation in Construction*, 12(2002), 123-131.

El-Omari, S. (2008). “Automated data acquisition for tracking and control of construction projects.” PhD thesis, Concordia University, Montreal, QC, Canada.

Fayek, A., AbouRizk, S. M., and Boyd, B. (1998). “Implementation of automated site data collection with a medium-sized contractor.” *Proc., International Computing Congress on Computing in Civil Engineering*, Boston, MA, USA, 454-456.

Fernando, S., Er, K. C., Mohamed, Y., AbouRizk, S., and Ruwanpura, J. Y. (2003). “A review of simulation applications for varying demands in tunneling.” *Proc., ASCE Construction Research Congress*, Honolulu, HI, USA, 19-21.

Fujimoto, R. M. (1999). “Parallel and distribution simulation systems.” John Wiley & Sons Inc., New York, NY, USA.

Goedert, J. and Meadati, P. (2008). “Integrating construction process documentation into building information modeling.” *J. Constr. Eng. Manage*., 134(7), 509–516.

Hajian, H., Becerik, B. (2009). “A research outlook for real-time project information management by integrating advanced field data acquisition systems and building information modeling.” *Proc., ASCE International Workshop on Computing in Civil Engineering*, Austin, TX, USA.

Hajjar, D., and AbouRizk, S. (1999). “Simphony: an environment for building special purpose construction simulation tools.” *Proc., Winter Simulation Conference*, Phoenix, AZ, USA, (2), 998-1006.

Hajjar, D. and AbouRizk, S. M. (2000). “Application framework for development of simulation tools.” *J. Comput. Civ. Eng.*, 14(3), 160-167.

Hajjar, D., and AbouRizk, S.M. (2002). “Unified modeling methodology for construction simulation.” *J. Constr. Eng. Manage.*, 128(2), 174-185.

Hegazy, T., Elbeltagi, E., and Zhang, K. (2005). “Keeping better site records using intelligent bar charts.” *J. Constr. Eng. Manage*., 131(5), 513–521.

Hendrickson, C., and Au, T. (1989). “Project management for construction, fundamental concepts for owners, engineers, architects and builders.” Prentice Hall, Pittsburgh, PA, USA.

Kangari, R. (1995). “Construction documentation in arbitration.” *J. Constr. Eng. Manage.*, 121(2), 201-208.

Kleijnen, J. P. C. (1995). “Verification and validation of simulation models.” *Eur J Oper Res*, 82(1), 145-162.

Kratkiewicz, G., Fedyk, A., and Cerys, D. (2004). “Integrating a distributed agent-based simulation into an HLA federation.” *Proc., 2004 Spring Interoperability Workshop*, Simulation Interoperability Standards Organization.

Kuhl, F., Weatherly, R., and Dahmann, J. (1999). “Creating computer simulation systems: an introduction to the high level architecture.” Prentice Hall PTR, Upper Saddle River, NJ, USA, 1-85.

Law, A.M., and Kelton, W.D. (1991). “Simulation Modeling and Analysis.” McGraw-Hill, New York, NY, USA, 697-699.

Lees, M., Logan, B., and Theodoropoulos, G. (2007). “Distributed simulation of agent-based systems with HLA.” *ACM Transactions on Modeling and Computer Simulation*, 17(3).

Likhitruangsilp, V., and Ioannou, P. G. (2003). “Stochastic evaluation of tunneling performance using discrete event simulation.” *Proc., Construction Research Congress*, Honolulu, HI, USA, 1-8.

Lucko, G., and Rojas, E. (2010). “Research validation: Challenges and opportunities in the construction domain.” *J. Constr. Eng. Manage.*, 136(1), 127–135.

Martinez, J., and Ioannou, P. G. (1994). “General purpose simulation with stroboscope.” *Proc., Winter Simulation Conference,* San Diego, CA, USA, 1159-1166.

Marzok, M., Abdallah, M., and El-Said, M. (2008). “Tunnel_Sim: decision support tool for planning tunnel construction using computer simulation.” *Proc., Winter Simulation Conference,* Miami, FL, USA, 2504-2511.

Method 123 Ltd. (2006). “Project Management Methodology”, *MPMM™ Software (Educational Version),* <http://www.mpmm.com/project-management-courses-products.php> (Aug. 2010)

Minson, R., and Theodoropoulos, G. K. (2008). “Distributing RePast agent-based simulations with HLA.” *Concurrency and Computation: Practice and Experience*, 20, 1225–1256.

Moghani, E. and AbouRizk, S.M. (2010). “A simulation-based framework for construction project information management.” *Proc., International Conference on Modeling and Applied Simulation*, Fes, Morocco.

Moghani, E., AbouRizk, S. M., and Hague, S. (2011a). “The integration of a stand-alone discrete event modeling environment into a distributed construction simulation system.” *Proc., CSCE Annual Conference*, Ottawa, ON, Canada, V3, 2495-2504.

Moghani, E, AbouRizk, H., AbouRizk, S., and Sander, H. (2011b). “Analyzing transit tunnel construction strategies using discrete event simulation.” *Proc. Winter Simulation Conference,* Phoenix, AZ, USA, 3500-3510.

Navon, R. (2005). “Automated project performance control of construction projects.” *J. Automation in Construction*, 14(4), 467-476.

Ourdev, I., Abourizk, S. M., and Al-Bataineh, M. (2007). “Simulation and uncertainty modeling of project schedules estimates.” *Proc. Winter Simulation Conference*, Washington, DC, USA, 2128-2133.

Pederson, P., Dudenhoeffer, D., Hartly, S., and Permann, M. (2006). “Critical infrastructure interdependency modeling: a survey of U.S. and international research.” Report. INL/EXT-06-11464, Idaho Falls: Idaho National Laboratory.

Robinson, S. (1997). “Simulation model verification and validation: increasing the user's confidence.” *Proc., Winter Simulation Conference*, Atlanta, GA, USA, 53–59.

Russell, A. (1993). “Computerized daily site reporting.” *J. Constr. Eng. Manage.*, 119(2), 385–402.

Ruwanpura, J. Y. (2001). “Special purpose simulation for tunnel construction operations.” PhD thesis, University of Alberta, Edmonton, AB, Canada.

Ruwanpura, J.Y., and AbouRizk, S.M. (2001). “Design, development and application of soil transition algorithms for tunneling using special purpose simulation.” *Proc., Winter Simulation Conference*, Arlington, VA, USA, (2), 1512-1520.

Sardroud, J. M., Limbachiya, M.C. (2010). “Effective information delivery at construction phase with integrated application of RFID, GPS and GSM technology.” *Proc., The World Congress on Engineering*, Vol. I, London, U.K.

Sawhney, A. (1994). “Simulation-based planning for construction.” PhD thesis, University of Alberta, Edmonton, AB, Canada.

Schriber, T. J., and Brunner, D. T. (2010). “Inside discrete-event simulation software: How it works and why it matters.” *Proc., Winter Simulation Conference*, 151-165.

Scott, S. and Assadi, S. (1999). “A survey of the site records kept by construction supervisors.” *Construction Management and Economics*, 17(3).

Straßburger, S. (2001). “Distributed simulation based on the high level architecture in civilian application domains.” PhD thesis, University of Magdeburg, Germany.

Stumpf, A. L., Chin, S., Liu, L.Y., and Ganeshan, R. (1995). “Use of a relational database system to integrate product and process information during construction.” *CIB W78 workshop, Modeling of buildings through their life-cycle*, Stanford, CA, USA, 316-326.

Taghaddos, H., AbouRizk, S. M., Mohamed, Y. and Ourdev, I. (2008). “Distributed agent-based simulation of construction projects with HLA.” *Proc. Winter Simulation Conference*, Austin, TX, 2413 - 2420.

Touran, A., and Asai, T. (1987). “Simulation of tunneling operations.” *J. Constr. Engrg. and Mgmt.*, ASCE, 113(4), 554–568.

Tse, R. L. M. (1989). “Computerized daily construction site reporting system (DSRS).” MSc thesis, University of British Columbia, Vancouver, BC, Canada.

Wilcox, P. A., Burger, A. G., and Hoare, P. (2000). “Advanced distributed simulation: a review of developments and their implication for data collection and analysis.” *J. Simulation Practice and Theory*, 8 (3-4), 201-231.

Xie, H., AbouRizk, S. M., and Fernando, S. (2011). “Integrating real-time project progress input into a construction simulation model.” *Proc., Winter Simulation Conference*, Phoenix, AZ, USA, 3443-3454.

Xie, H. and Moghani, E. (2009). “Tunneling construction simulation using high level architecture.” *Proc.,16th Annual Canadian Construction Resarch Forum*, Banff, AB, Canada, 133-155.

Zhang, Y., AbouRizk, S., Xie, H., and Moghani, E. (2012). “Design and implementation of loose-coupling visualization components in a distributed construction simulation environment with HLA.” *J. Comput. Civ. Eng.*, 26(2), 248–258.

Appendix A- A Simulation-Based Framework for Construction Project Information Management[1]

Elmira Moghani [a], Simaan AbouRizk [b]

[a]PhD Candidate, Department of Civil and Environmental Engineering, University of Alberta, Edmonton, Canada

[b]Professor, Department of Civil and Environmental Engineering, University of Alberta, Edmonton, Canada

ABSTRACT

Modern capital projects generate large amounts of documentation in several different formats and controlled by many different project parties: everything from design drawings to cost estimates, schedules, equipment information, change orders, and work logs. Accessing this documentation can be both difficult and time-consuming. Project managers clearly need efficient methods to manage project information; the current methods focus on capturing and integrating information through a database and CAD drawings, but they are limited in scope, not easy to re-use in other projects, and do not capture the construction process. The solution is a well-structured simulation-based system that can dynamically capture, store, process, and access all project information, from the planning stage through the construction process to the completed project, including all changes to the original plan. The motivation for a simulation-based foundation for this approach stems from its ability to model the dynamic processes involved rather than just the static information.

Keywords: project information, construction process, High Level Architecture, Distributed simulation

[1] This paper is published at the proceeding of the International Conference on Modeling and Applied Simulation, October 13-15, 2010, Fes - Morocco

INTRODUCTION

Complex capital projects can involve thousands of workers, hundreds of millions of dollars, and many years of work. These types of projects generate large amounts of documentation in many different formats and controlled by many different project parties: everything from design drawings to cost estimates, schedules, equipment information, change orders, and work logs. During the construction period, a company generates reports in certain areas, such as workers' time cards and the actual as-built drawings of the project, and projects are also almost never completed exactly as planned; keeping track of changes in the building process is very complicated, adding yet another layer of documentation. However, even this level of documentation is incomplete, especially when it comes to recording the actual building process.

Accessing this documentation can also be both difficult and time-consuming. Because the existing documentation is stored based on the interest of each party (e.g., drawings may be kept with the design consultant, the foreman might have worker records, and the project manager might have the financial information), the project information is disorganized and reconstructing a record of how the project was completed may require hiring experts. This can be a problem, particularly when it comes to settling claims and dealing with quality control, both of which can mean losses of millions of dollars (Akinci 2004).

It should therefore be unsurprising that project managers and contractors are always looking for efficient ways to collect and store construction histories and retrieve them effectively (Hendrickson and Au 1989). Different studies have focused on developing a computer based information system to integrate the collection, processing and transmission of information (Lock 1993). A database management system (DBMS) has become a common solution to overcome some of the limitations of data sharing (Mazerolle and Alkass 1993; Bowler 1994; Dawood et al. 2002). But the current methods do not capture the construction process; existing methods usually focus on capturing and storing physical information from computer-aided design (CAD) drawings, or the scope of the project from as-planned and as-built documents, rather than capturing the actual building process. They are also limited in scope, focusing on small parts rather than the entire project. For example, project managers who are interested in equipment control try to capture information such as working hours, breakdown or fuel consumption of onsite equipment, but estimators are interested in collecting the workers' daily log or material cost (Navon 2005; Fayek et al. 1998). Even integrating the project information database using the Critical Path Method (CPM) does not support capturing the complete operation; a CPM network only describes tasks to a certain level of information, such as duration and resources.

The solution is a well-structured simulation-based system that can dynamically capture, store, process, and access all project information, from the planning stage through the building process to the completed project, including all changes to the original plan. Ideally, the system should integrate this information and present it to the user, providing an overview of the project at every stage for comparison with the project as planned. It should be able to incorporate process models, product models, resource models, and static information in one system. A user should be able to see the resources that completed a given scope of work in a given period of time under the influence of external factors (e.g. weather).

This kind of documentation will be helpful for claims, control purposes, reproducing actual product drawings, operation and maintenance, and planning for future similar projects. It will also increase the learning process in the involved organizations for future projects by enabling access to the information of previous projects (FIATECH 2009). The solution must be easy to use from a manager's perspective, and it must also be reusable for different types and sizes of construction projects.

One promising method for implementing the solution is computer simulation. Computer simulation was introduced to the area of construction research by Halpin (1977) with his proposed CYCLONE system. The proposed system and all derivatives were used in construction research to model real construction processes, leading construction managers to more efficient use of materials, manpower, and equipment (Paulson et al. 1987; Ioannou 1989; Martinez and Ioannou 1994; Hajjar and AbouRizk 2002). Computer simulation has been successfully implemented in the construction industry for many purposes (Sawhney 1994), and has already been effectively used in managing information in the construction domain by integrating database management systems with a simulation model (Moghani et al. 2009). In this approach, a construction manager can use the collected data to re-run a simulation process model and examine the effect of new data on project performance. However, current modeling approaches cannot integrate or assimilate information from different sources and by different participants in an efficient and organized manner. For a more complex project that needs different information from different participants or software, distributed simulation is a new simulation technique that can facilitate modeling effort and integration in the simulation environment.

AbouRizk introduced High Level Architecture-based distributed simulation to simulate construction projects (2006), facilitating integration, collaboration, and reusability of the simulation model. High Level Architecture (HLA) was developed in 1995 by the Department of Defense (DoD) as an advanced technique for integrating simulation models to support reuse and interoperation of simulation models and reduce the cost and effort of modeling in simulation projects (Fujimoto 2003). Under the HLA standard, different developers can build individual components (federates) of one system (a federation), maintaining interoperability between them. This approach allows us to standardize the integration process between different computer software, simulation systems, and from different users; it is therefore a promising implementation for the proposed information management solution.

This study aims to develop an interoperable and reusable simulation-based framework to integrate project information including as-planned, process, and as-built information to enable construction managers to create the real history of the project from planning to completion. This framework will utilize HLA and a software application framework approach (Froehlich et al. 1998) for its implementation to be a reusable, extensible and modular model, with flexibility in representing different modeling approaches and data forms in the same simulation. In this framework a 3D CAD model will be integrated with process simulation model to automatically capture design information and use it for simulation purposes, and a relational database will be the medium for integrating as-built information with the simulation model and updating the process accordingly. The project is under development using the Construction Synthetic Environment (COSYE) as its base; it is focused on repetitive construction such as tunnelling, which lends itself better to simulation planning (AbouRizk and Ruwanpura 1999; Fernando et al. 2003; Al-Bataineh 2008; Marzok et al. 2008). The case studies used or planned are existing tunnelling projects in Edmonton, Alberta, Canada.

HLA-BASED TUNNEL SIMULATION FRAMEWORK

COSYE (Construction Synthetic Environment), the simulation environment used as a base for this research, was developed based on HLA standards to facilitate modeling more complex projects, such as tunnel construction, which involves many activities, equipment, materials and human resources.

The HLA standards consist of three main components: the HLA rules (IEEE 1516 2000), the interface specifications (IEEE 1516.1 2000), and the Object Model Template (OMT) (IEEE 1516.2 2000). The HLA rules outline the creation of a federation and federates and cover all their responsibilities to ensure a consistent implementation and proper interactions. The interface specification defines the functional interfaces between federates and the run time infrastructure (RTI). RTI software provides interface services that support federates' interactions and federation management, such as transferring the responsibility of updating an attribute between federates, managing data distribution and assisting with time management in the federation. Any software can interact through the RTI as a single federate. The Object Model Template is a standard for defining and documenting the form, type, and structure of data shared within a simulation, and it consists of three different object models: the Federation Object Model (FOM), the Simulation Object Model (SOM) and the Management Object Model (MOM).

The COSYE Framework is a software application that supports development of federations in Microsoft Visual Studio. In this Environment, the RTI server is a .NET implementation of the HLA RTI and runs as a windows server, and the COSYE OMT Editor is used to develop and edit OMT documents in Visual Studio (AbouRizk and Hague 2009)

Based on HLA standards, COSYE enables the integration of different computer software and different simulation techniques in a single environment to model all the processes involved in large scale projects. HLA services in COSYE also enable collaboration of different experts to develop various parts of a simulation model. For instance, in the tunnel federation, one may simulate the tunnel construction process while the other focuses on material supply simulation.

In this project, the tunnel federation simulates the whole construction process of a utility tunnel including excavation and lining of the working shaft, the tunnel and the retrieval shaft. Using HLA requires division of the simulation into different federates and the HLA architecture thus helps to create modular, reusable, and extensible modeling elements.

Brief Overview of Tunnelling

Constructing a utility tunnel occurs in different phases. First, to get access to the tunnel excavation depth, a vertical shaft called a construction shaft is usually excavated. This shaft is the main access during the construction process for lowering the equipment into the tunnel or removing the excavated dirt. Shaft construction is usually done in sections with the depths dependent on the soil type and geometry of the shaft. The excavation and lining processes are done for each section sequentially. After finishing the excavation and lining for shaft sections, the tunnel construction process can start.

Tunnel construction methods vary depending on geotechnical information, availability of equipment, and the geometry of a tunnel. For long tunnels or tunnels with a large diameter, a Tunnel Boring Machine (TBM) is typically utilized for excavation and lining processes while for short or small tunnels, hand excavation is a better option. Before starting the tunnel construction, the tail tunnel and undercut area (an enlargement at the bottom of the shaft used for staging material handling and dirt removal operations) are excavated. The tunnel construction activities are divided into:

- Excavation of the tunnel

- Removing the dirt from the excavation area and transferring it to the construction shaft (using muck carts).
- Hoisting the dirt to the ground level (using a crane, clamshell bucket, gantry, etc.) and transfer it outside the construction area
- Lowering down the liners and transferring them to the excavation face.
- Lining the tunnel
- Extending construction and utility services
- Excavating and supporting the removal shaft in a case of using TBM.

Figure 1 shows a typical utility tunnel layout.

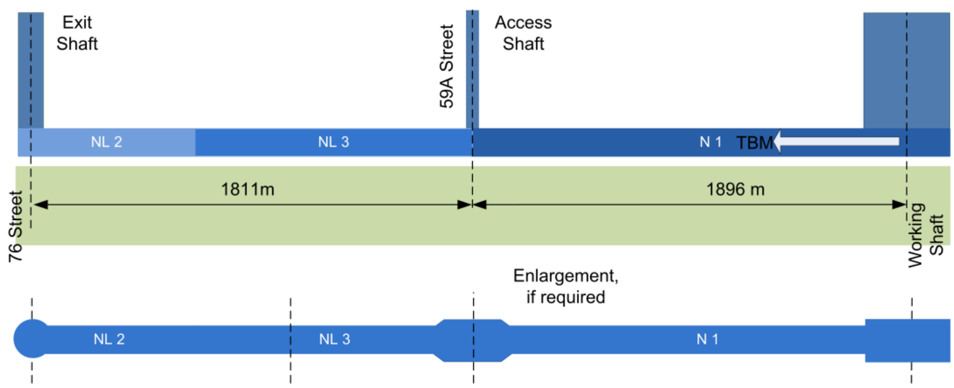


Figure 1: Layout of a Utility Tunnel

Developing the Tunnelling Product Model

In order to be able to create a generic federation and integrate the CAD model with the simulation, a conceptual project model consisting of product, process, management, environment, and resource models was developed (Figure 2).

The tunnel construction method for utility facilities was investigated, and all the information regarding activities and required resources was collected. In order to develop a product model, the CAD drawings were reviewed and the attributes were defined and added to the project model. It is important to mention that these models are for utility tunnels, and they mostly focus on the TBM (tunnel boring machine) tunnelling method.

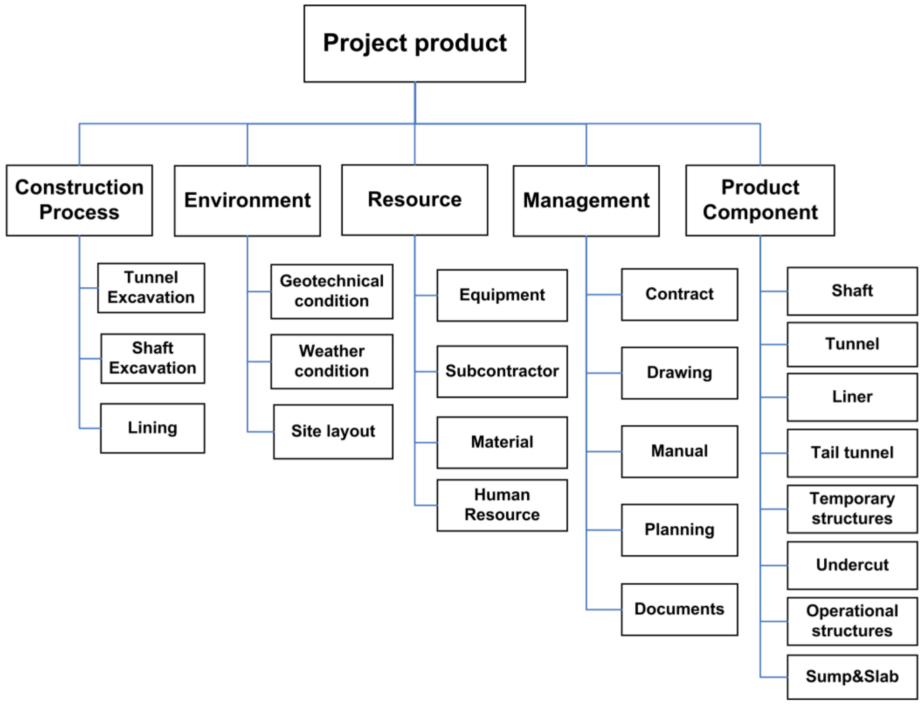


Figure 2: Tunnel Construction Project Model

Developing a Federation Object Model (FOM) for the Tunneling Federation based on the Product and Process Model

The first and most important task in developing the simulation model based on High Level Architecture (HLA) is to define the HLA federation object model. The FOM is composed of a group of interrelated components specifying information about classes of objects, interactions, attributes, and their parameters. The tunnelling federation object model was developed based on conceptual project models and is still being improved. Figure 3 shows the FOM in the tunnel federation.

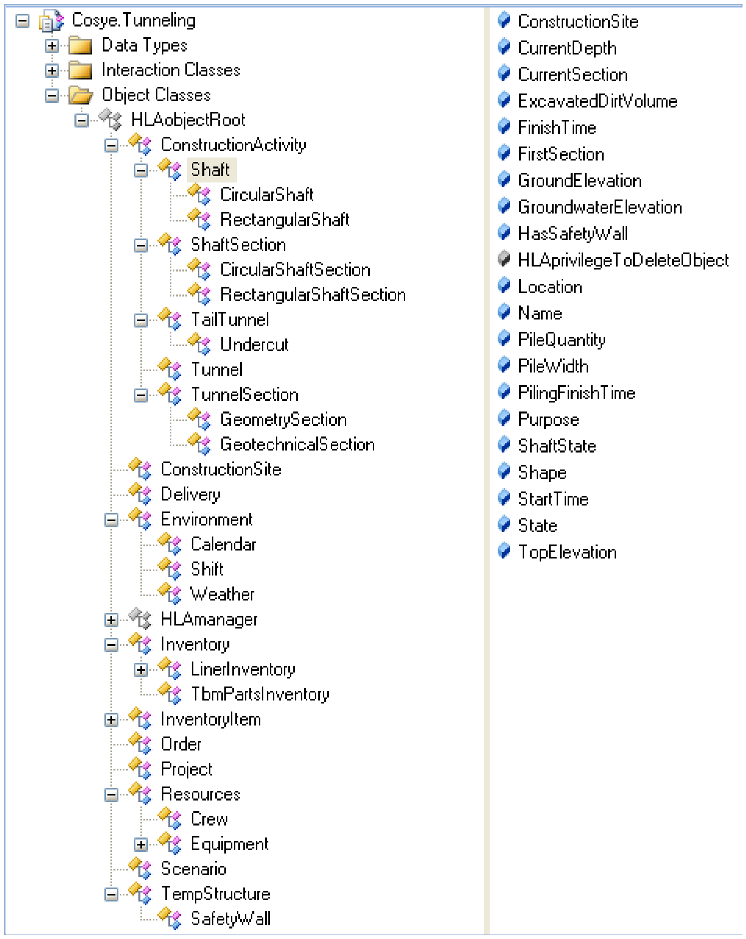


Figure 3: Tunnel Federation Object Model

Overview of Tunnelling Federation

Figure 4 shows the proposed framework for the simulation model, based on available studies and expert knowledge gathering. Most federates were developed based on a case study for tunnel construction in the City of Edmonton using a specific construction method, and will be used as a base for further development.

The scenario setup federate is designed for the user to configure different tunnel projects and scenarios. In the current federation, the user inputs all necessary parameters at this federate: shift length, shift start time, coffee break duration, lunch break duration, project start date, work status on weekends or holidays; project setting such as number of shafts, number of tunnels, and their attributes like tunnel length, section length, etc.; resource setting refers to equipment information and crew information. This information will be passed to other relevant federates through RTI.

The shaft federate simulates the shaft construction process including preparation, excavation, and lining. The tunnel federate is designed to simulate the complete process of constructing a tunnel including excavation, lining, resetting TBM, and TBM breakdown; it also covers other common activities, like extending utilities and surveying.

The dirt removal federate models the process of removing dirt from the tunnel face to the undercut, dumping dirt from undercut to ground, and loading carts with materials.

The supplier federate has the same responsibility as a supplier contractor: receive a new order from the contractor through the procurement federate, schedule for delivery of a new order and send a response to the procurement federate.

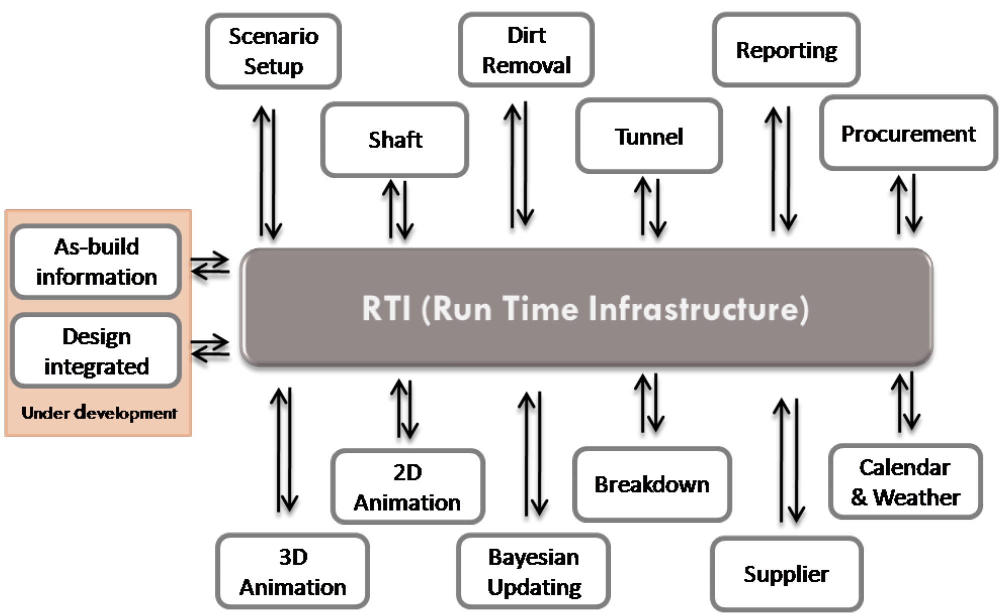


Figure 4: Tunnel Federation

There can be more than one supplier which can join the federation and receive orders from the procurement federate. The procurement federate plays the same role of the procurement office in the tunnelling contractor group. It tracks how many liners are in the inventory; if that level reaches a specific threshold, the procurement federate places an order for the concrete lining segments to the supplier. The Bayesian updating federate is designed to apply a Bayesian updating method for predicting machine breakdown, TBM advance rate, tunnel productivity, scheduling and cost. The visualization federate displays a 2D and 3D animation of the construction process as the simulation is running. It is a real time visualization of the entire process and shows the different states of the TBM during tunnel construction, the excavation and lining of the tunnel, as well as traveling muck carts in the tunnel.

In current federation, all the mentioned federates are fully functional for the NEST tunnel case study in Edmonton, Canada. Future development will involve connecting a 3D CAD drawing to the simulation model and extracting as-planned geometry and geotechnical information. This federate will remove the need for manual data entry, especially when new revisions of drawings are released. Capturing and storing all the as-built information will be based on automatic data transfer from the site and data will be stored in the SQL Server database. Figure 5 illustrates an example of relationship in the database.

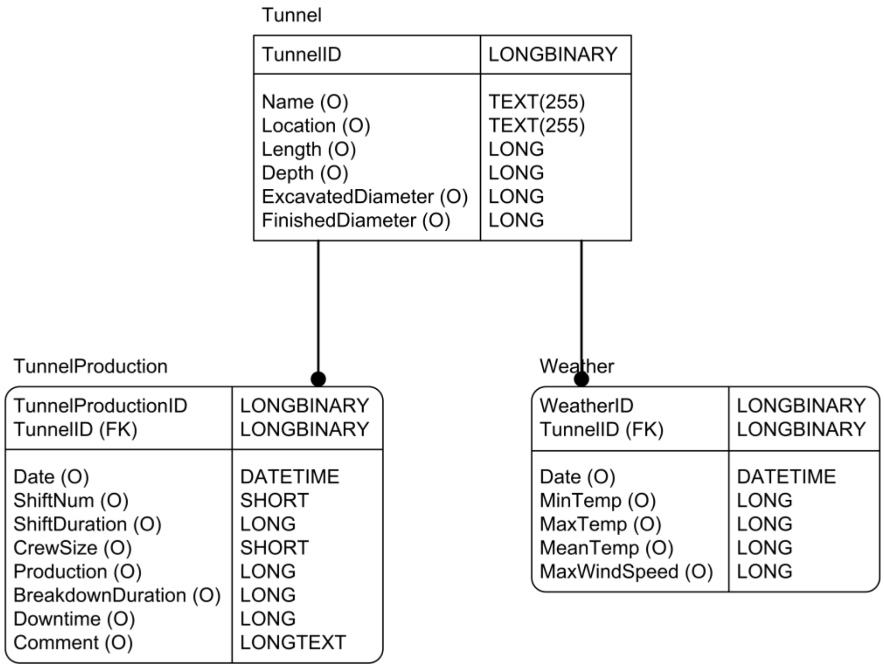


Figure 5: ER Diagram of Database

INFORMATION MANAGEMENT STRUCTURE

To fulfill the objective of this research, a vital part of the implementation process is to study the possible ways of representing information within the COSYE environment and create a conceptual structure for the data management plan. We determined the required information by answering the following questions:

- What kind of information will be useful for decision making process and different participants in the project?
- What kind of information from the planning stage will be required for developing the simulation model?
- What kind of information from the jobsite will change the inputs of the simulation model and how can we store it to make it exchangeable with original inputs?

- How can we make the history of projects reusable for future projects or simulation models?

Answering these questions determines the information required from a tunnelling construction project in order for the simulation to be representative. Finally, to collect and integrate on-site process information (as-built) and to be able to store the as-planned data and their relationships throughout a simulation model, an accurate knowledge management system needs to be developed. For this purpose, a comprehensive relational database is proposed to store planning data, actual as-built data, and the history of project process changes. A view of our vision is given in Figure 6.

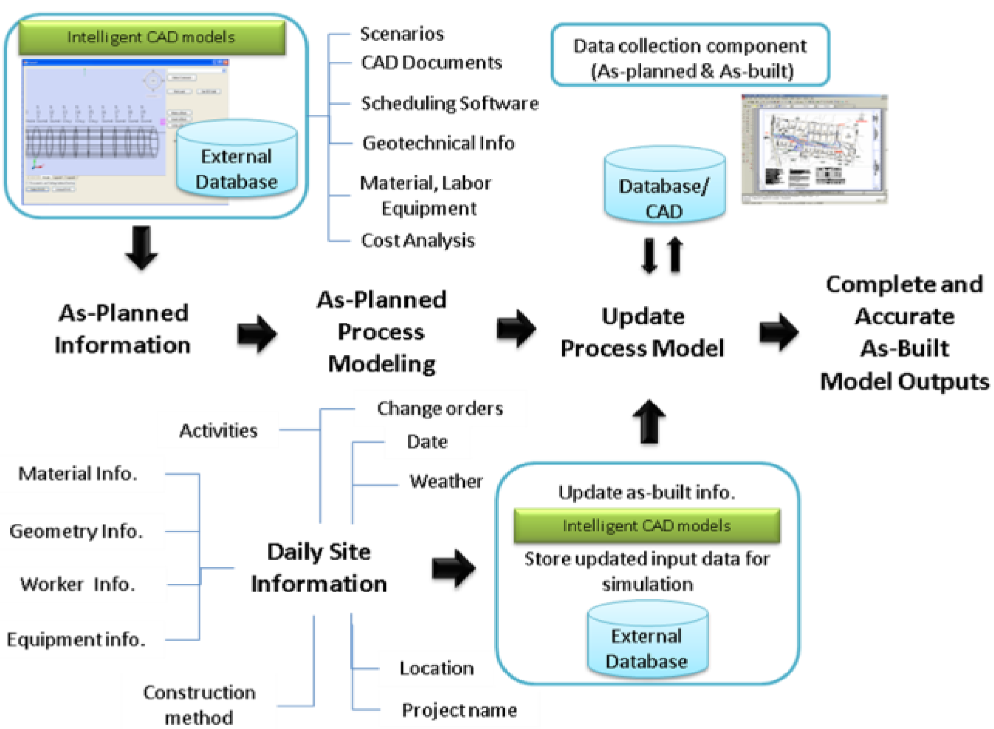


Figure 6: Integrated Framework for Tunnelling Construction

FUTURE WORK

In the future, new federates such as a design information federate will be developed to facilitate information transfer from CAD systems to the simulation model, and a relational database will be designed and connected to the federation to store as-built information as the project advances.

To make changes to the process model during the construction phase without a need to re-create the entire model (e.g., adding one removal shaft or changing the construction method of the tunnel), generic and reusable federates are required. To develop such federates; the current FOM needs to expand to provide all the object classes, attributes, and interactions for different tunnel construction methods, activities, equipments, and the required information from the planning and construction stage.

The simulation model and design inputs will then be updated based on new process and information (reconstructed from the beginning of the project but as if the project is going on now – i.e., simulate the simulation) to mirror any changes in design and construction, create an accurate model and verify it. The product/process data integration will be evaluated and adjusted if necessary.

Once the NEST case application is functional, the work will be repeated with another tunnel construction project to identify areas where the framework must be adjusted and an application framework approach as a basis of this research will be utilized to provide a generic framework with modularity, reusability and extensibility.

CONCLUSION

The ability to access detailed information of a project along with the actual construction process (as it was built) will be highly beneficial for the management process for different purposes such as quality control, claims, process improvement and future project planning. This paper proposes the development of a generic framework for information management in repetitive construction projects such as tunneling to incorporate process model, product model, resource model, and static information in one system. Deploying HLA-based distributed simulation and using a software application framework enables us to standardize the integration process between simulation systems, different computer software, and project models and supports interoperability, reusability, and extensibility of this framework. While all COSYE development to date has taken place in an academic environment and COSYE itself has not yet been commercialized, development of a full federation typically takes between one half to one full man-year (1,000 to 2,000 hours).

With this system, project managers will have full access to the project data. Moreover, they will have an accurate model at the end of the project for reviewing purposes, documentation for future work, and educational purposes. The system will save time and money, improve training, and increase operational efficiency.

ACKNOWLEDGMENTS

This study is supported by National Science and Engineering Research Council (NSERC) Industrial Research Chair in Construction Engineering and Management research.

REFERENCES

AbouRizk, S.M., Hague, S. 2009, An Overview of the COSYE Environment for Construction Simulation, *Proceedings of the Winter Simulation Conference*, pp. 2624-2634. Austin (TX, USA)

AbouRizk, S.M., 2006. Collaborative Simulation Framework for Multi-user Support in Construction. *Discovery Grant Proposal*.

AbouRizk, S.M., and Mather, K., 2000. Simplifying Simulation Modeling Through Integration with 3-CAD. *Journal of Construction Engineering and Management*, 126(6), 475-483.

AbouRizk, S.M., Ruwanpura, J.Y., Er. K.C., and Fernando, S., 1999. Special Purpose Simulation Template for Utility Tunnel Construction. *Proceedings of the 31st Winter Simulation Conference*, pp. 948-955. Dec. 05-08, Phoenix (Arizona, USA).

Akinci, B., 2004, Using Sensor Systems and Standard Project Models to Capture and Model Project History for Building Commissioning and Facility Management. *Facility Area Network Workshop*, Feb. 26-27, Civil Engineering Research Lab, Urbana-Champaign, IL

Al-Bataineh, M.T., 2008. *Scenario-Based Planning for Tunneling Construction*. Thesis (PhD). University of Alberta.

Dawood, N., Sriprasert, E., Mallasi, Z., and Hobbs, B., 2003. Development of an integrated information resource base for 4D/VR construction processes simulation. *Automation in Construction*, 12(2), 123-131.

Fayek, A., AbouRizk, S.M., and Boyd, B., 1998. Implementation of automated site data collection with a medium-sized contractor. *Proceedings of the International Computing Congress on Computing in Civil Engineering*, pp. 454–457. Boston (Massachusetts, USA)

Fernando, S., Mohamed, Y., AbouRizk, S.M., and Ruwanpura, J.Y., 2003. A review of simulation applications for varying demands in tunneling. *Proceedings of the ASCE Construction Research Congress*, Honolulu (Hawaii, USA).

FIATECH., 2009. *Element 9: Lifecycle Data Management & Information Integration.* Available from: http://fiatech.org/lifecycle-data-management-information-integration.html?showall=1 [Accessed Feb. 22nd, 2010]

Froehlich, G., Hoover, J., Liu, L., and Sorenson, P., 1998. Designing Object-Oriented Frameworks. *CRC Handbook of Object Technology*.

Fujimoto, R.M., 2003. Distributed simulation systems. *Proceedings of the Winter Simulation Conference*, pp. 124-134. New Orleans (LA, USA)

Hajjar, D., and AbouRizk, S.M., 2002. Unified Modeling Methodology for Construction Simulation. *Journal of Construction Engineering and Management*, 128(2) , 74-185.

Halpin, D.W., 1977. CYCLONE: Method for Modeling of Job Site Processes. Journal of the Construction Division, ASCE, 103(3) , 489-499.

Hendrickson, C., and Au, T., 1989. *Project Management for Construction*. Pittsburgh, PA: Prentice Hall.

IEEE Std 1516, *2000. IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Framework and Rules*

IEEE Std 1516.1, *2000. IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)* - *Federate Interface Specification*

IEEE Std 1516.2, *2000. IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Object Model Template (OMT) Specification*

Ioannou, P.G., 1989. *UM-CYCLONE User's Guide*. Ann Arbor, Michigan, Department of Civil Engineering, The University of Michigan.

Martinez, J., and Ioannou, P.G., 1994. General Purpose Simulation with Stroboscope. *Proceedings of the 26th Winter Simulation Conference,* pp. 1159-1166, Orlando (Florida, USA)

Marzok, M., Abdallah, M., and El-Said, M., 2008. Tunnel_Sim: Decision Support Tool for Planning Tunnel Construction Using Computer Simulation. *Proceedings of the 2008 Winter Simulation Conference,* pp. 2504-2511. Miami (Florida, USA)

Mazerolle, M., and Alkass, S., 1993. An Integrated System to Facilitate the Analysis of Construction Claims. *The Fifth International Conference on Computing in Civil and Building Engineering*, pp. 1509-1516. Anaheim (California, USA)

Moghani, E., Salehi, M., and Taghaddos, H., 2009. Simulation Based Schedule Enhancement of Tower Cranes. *Proceedings of the 2nd International/ 8th Construction Special Conference*. May 27-30, St. John's (Newfoundland and Labrador, Canada)

Navon, R. 2005. Automated project performance control of construction projects. *Automation in Construction*, 14(4), 467-476.

Paulson, B.C., Chan, W.T., and Koo, C.C., 1987. Construction Operation Simulation by Microcomputer. *Journal of Construction Engineering and Management*, 113(2) , 302-314.

Sawhney, A., 1994., *Simulation-based planning for construction.* Thesis (PhD). University of Alberta.

Appendix B- The Integration of a Stand-alone Discrete Event Modeling Environment into a Distributed Construction Simulation System [1]

E. Moghani[1], S. M. AbouRizk[1], S. Hague[1] and S. Fernando[2]

1 Hole School of Construction, University of Alberta, Edmonton, Alberta, Canada

2 City of Edmonton Asset Management and Public Works, Edmonton, Alberta, Canada

ABSTRACT

Several computer simulation tools have been developed to model construction processes for different management purposes such as productivity analysis, resource allocation optimization, and cost control. Although these systems are recognized as valuable management tools, limitations encountered when modeling the entire construction lifecycle, including the process model, product model, and external factors, have caused researchers to look for new techniques to facilitate integration in the simulation environment. An HLA-based distributed simulation technique introduced to the construction simulation domain to facilitate integration of various simulation techniques and software applications in a collaborative environment. The objective of this study is to integrate a Special Purpose Simulation tool as a component of a distributed simulation system to create a comprehensive modeling environment for a project. The system uses the potential capabilities of Simphony, a discrete event modeling tool, to create templates and update them through the user interface. For the implementation phase, the tunnel construction template is developed in Simphony.NET 3.5 and modified for integration in

[1] This paper is published at the proceeding of the 3rd International / 9th Construction Specialty Conference, June 14 – 17, 2011, Ottawa, Ontario, Canada

an HLA-based distributed model. A test federate is used to check the communication of the model with the COSYE RTI.

INTRODUCTION

Computer simulation has long been an effective technique for managing construction projects in the construction domain. Since the 1970s, several construction simulation tools have been developed to model construction process activities and assist project managers in examining possible alternatives, getting a clear idea of resource interactions in the project, and coming up with an optimal solution to implement in the real world.

Recently, researchers in the area of construction management work on the development of integrated project systems, mainly for the design and construction phase, to facilitate managing projects and be able to increase the collaboration between different organizations in a project and facilitate data transfer between various pieces of software (Latham 1994). Although simulation systems are recognized as valuable management tools, they are not suited for integration purposes or to model the entire project lifecycle, including the process model, product model, and external factors, which has caused researchers to look for new techniques that can facilitate integration in the simulation environment. High Level Architecture-based distributed simulation was introduced to overcome the limitations of existing methods by supporting integration of different computer applications, various simulation techniques and providing a multi-user environment (AbouRizk 2006). AbouRizk (2006) developed the Construction Synthetic Environment (COSYE) based on HLA rules to create integrated simulation models for industry users. In this environment, developers can divide a simulation into smaller components (federates) that can communicate and work together in one system (federation).

This research proposes the integration of a process modeling tool such as Simphony as a component of an HLA-based model, using the potential capabilities of the application while integrating it with various techniques and different series of software (e.g. AutoCAD, Microsoft Project, and estimating software) in one model. By this approach, any construction process can be simulated in Simphony or in other process modeling software and attached to a collaborative simulation system to create a comprehensive model for that domain. The proposed system can be a powerful tool for the management of the project through its life cycle. As a case study, an SPS template for tunnel construction projects is developed in a new version of Simphony (Simphony.NET 3.5) and then integrated in a COSYE federation.

CONSTRUCTION SIMULATION MODELING IN SIMPHONY

Various simulation applications have been developed to model construction processes; the early applications such as CYCLONE, RESQUE (Chang and Carr 1987), and Stroboscope (Martinez and Ioannou 1994) are known as General Purpose Simulation (GPS) tools. To create a model using a GPS tool, a user needs to have a deep understanding of simulation techniques which made it difficult to use for industry personnel. So, researchers tried to customize modeling tools for specific construction domains. They introduced Special Purpose Simulation (SPS) to facilitate modeling of specific type of projects. Hajjar and AbouRizk (1999) introduced Simphony, a Microsoft Windows-based SPS tool to develop a more flexible and easy to use application for industry. Simphony facilitates modeling the construction process through providing a graphical, hierarchical, modular and integrated modeling environment. In Simphony, a developer can use discrete event simulation programming to create a template, a collection of elements linked together and representing construction activities in one domain. Each element can encapsulate more than one activity in the process and its graphics represents the actual process in the site. This approach helps users, who can select elements based on their graphics, drag and drop them on modeling surface, and link them together to create a complete model, without a deep knowledge of programming. However, the template development requires broad knowledge of the construction domain to be able to include all the process details while keeping the flexibility of the model suitable for industry use. Several Simphony templates were created to model real projects and help construction managers in decision making and planning processes such as productivity measurement, risk analysis, resource allocation, site planning claim, and dispute resolution. A number of successful construction simulation models were created in Simphony including earth moving, pavement construction, concrete placement on high-rise buildings, tunnelling, underground pipe-jacking, tower crane utilization, equipment management, site layout optimization, etc.

A new version of Simphony called Simphony.NET 3.5 was developed to provide more flexible and extensible environment for modeling and integration purposes. Simphony 3.5 was developed utilizing the Microsoft.NET 3.5 framework under the object-oriented paradigm. It consists of the following modules:

- Services that facilitate discrete event simulation programming such as: Simphony Core Services (contains components such as the discrete event engine as well as math libraries, statistics and distribution) and Simphony Modelling Services (contains abstract versions of models, scenarios, and various types of modeling for template development),
- Simphony templates and Graphical User Interface.

Each modeling element in Simphony.NET 3.5 is a Visual Basic or C# class developed in one Microsoft Visual Studio project (which represents a template in Simphony). Unlike the previous Simphony, the development of a template is not tied to the user interface, which allows the use of the provided services in other simulation environments. Figure 1 shows a sample model and the coding environment in Simphony.NET 3.5.

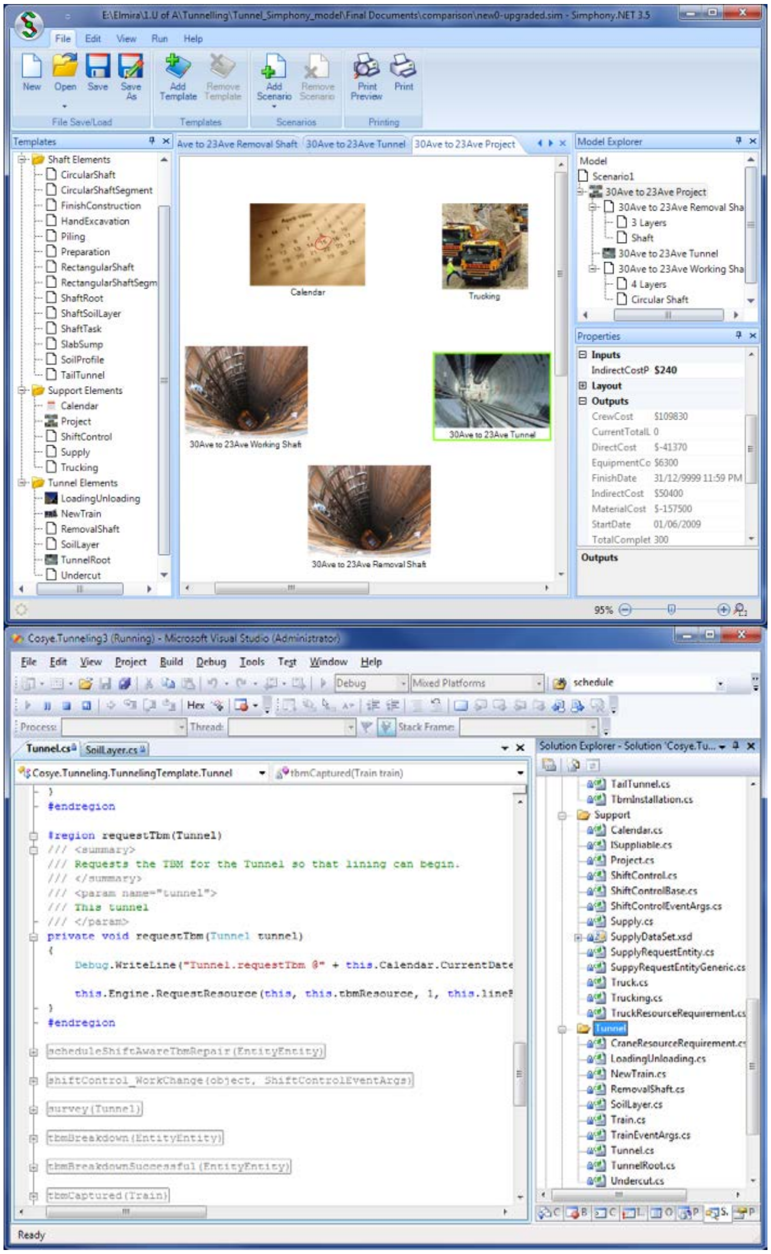


Figure 1: Simulation model and coding environment in Simphony.NET 3.5

AN HLA-BASED DISTRIBUTED SIMULATION OF CONSTRUCTION PROJECTS

Due to the complexity of construction projects and the recent trend of design and construction integration, current simulation techniques are unable to support this collaboration in the management domain, as their main focus is on the process modeling and resource interaction to calculate productivity or duration of the construction phase. More modern simulation tools are needed to model the project life cycle from the planning to the construction phase. They should also support integration of visualization tools (CAD systems, animation), databases, different simulation techniques (continuous, discrete event, real time, agent based, etc.), and software applications (Project schedule, estimation software, and control system) (AbouRizk 2010). To fulfill the recent needs in the construction simulation, the distributed simulation (DS) technique is selected as a suitable choice for this purpose.

Distributed simulation is developed in the military domain and is known as a technology "that enables a simulation program to execute on a computing system containing multiple processors, such as personal computers, interconnected by a communication network" (Fujimoto 1999). In a distributed system, modellers divide a complex simulation system into multiple subsystems that are interrelated to represent the entire system functionality and reduce the execution time of a system. The other objective is to decrease the time and cost of a model development by reusing of submodels in future similar projects The other advantage of a distributed model is that it enables the interoperability of different heterogeneous simulation techniques such as discrete event, real time, and continuous, in one system. It can also provide a collaborative simulation environment by enabling different parties to interact through a set of geographically distributed computers and reducing the time of communication and data transfer (Boer 2005). To fulfill all these requirements, a common framework and certain rules and standards were initiated to facilitate the development process. High Level Architecture (HLA), the most advanced standard available, developed in 1996 by the United States Department of Defense (DoD) to integrate simulation models (Kuhl et al. 1999).

In HLA terms, each submodel is called a federate and the entire simulation system is known as a federation. HLA rules seek two main goals; interoperability and reusability of the federates. HLA-based distributed simulation consists of three major components: HLA rules (IEEE 1516) to ensure proper federate interaction during execution; the Object Model Template (OMT) (IEEE 1516.2) defines a standard framework for recording the information contained in the required HLA object model for each federate and federation and consists of the federation object model (FOM) and simulation object model (SOM); and the HLA interface specification (IEEE 1516.1) describes the services provided to the federates by the Run Time Infrastructure (RTI). RTI provides services to federates' interactions and federation management. Figure 2 illustrates the functional view of HLA.

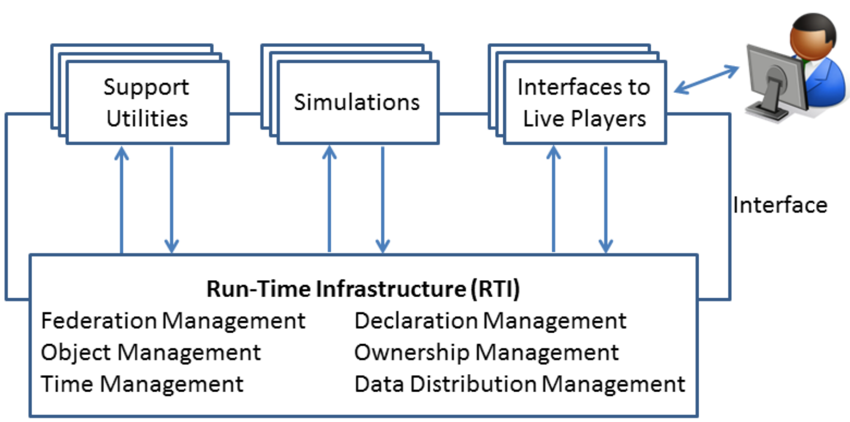


Figure 2: Structure of High Level Architecture (Wilcox et al. 2000)

In 2006, AbouRizk introduced an HLA-based distributed simulation environment called COSYE (the Construction Synthetic Environment) to the construction simulation domain to simulate more complex projects and facilitate interactions of different parties and various simulation components in a single system. The COSYE environment is a Microsoft.NET software application and supports the development of federations in Microsoft Visual Studio (AbouRizk and Hague 2009). COSYE supports:

- Modeling of different aspects of the project including process model, product model, resource interactions (equipment, material, crews) and external factors' influences (weather).
- Integration of different software applications to work together on multiple computers and different locations (e.g. consultant, project management or contractor offices) as one simulation system to facilitate collaboration in the project.

One of the advanced implementations in COSYE is the tunnel construction simulation (Xie and Moghani 2009) that consists of several federates (Figure 3).

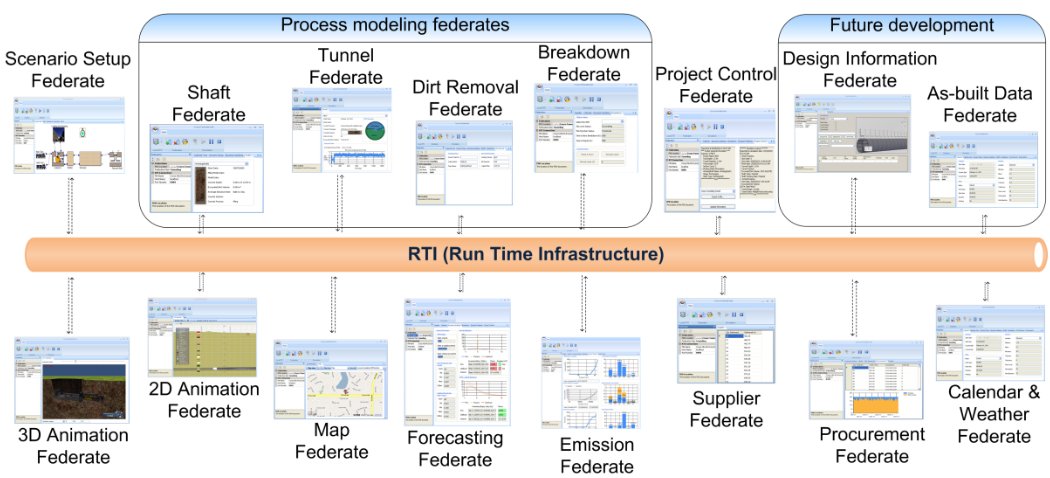


Figure 3: HLA-based Tunnel Construction Simulation

In the tunnel federation, 14 different federates are integrated to demonstrate the entire process of a utility tunnel construction in the City of Edmonton. The main advantage of this model over the existing Simphony templates is that all the federates are reusable for future projects. 2D/3D animation and map federates provide visualization capability for the construction manager. Managers can see a written report of the process from the project control federate and supplier and procurement federates can be run on the supplier's computer to show the material request. This model calculates emissions in the job site, provides scenario planning, and gives forecasting ability for users.

In the current distributed simulation models in COSYE, each federate is developed as a Visual Studio class coded in C# or Visual Basic and the user interface does not allow users to modify or change codes or logic of the simulation. Normally, in the planning stage, decision makers examine different scenarios of construction process and resource interaction, and during the construction phase, there is a need to update process based on the real situation. Thus, process modeling is the main part of the model which needs modification as the project progresses and the COSYE model is not capable of regularly updating the process model. This limitation can decrease industry interest in using the simulation model as they have to hire programmers to revise and update the model.

INTEGRATION OF SIMPHONY AS A DISCRETE-EVENT SIMULATION TOOL INTO COSYE

As the main concern in the management group is updating construction processes, and since Simphony is a well proven application for process modeling, a reliable solution is to integrate the Simphony model as one federate in a federation, represent the construction process model in the Simphony environment (with COSYE-compatible simulation models), and use Simphony's visual capabilities to update the process. This would facilitate more efficient development of COSYE federations, and create COSYE-aware

Simphony models to be a part of collaborative and comprehensive construction simulation models.

As previously mentioned, Simphony.NET 3.5 was developed particularly to be used in an integrated environment; new features have been added to Simphony to facilitate communication between COSYE and Simphony models. A Simphony modeling element called "RTI connection" is created as a generic element to be added to all Simphony models to enable them to join to the federation as one federate. The responsibility of this element is to provide a connection to the RTI and manage joining and execution of the Simphony model and sending and receiving messages through RTI. Other elements would utilize the services provided by the RTI connection element to communicate with the RTI. Figure 4 demonstrates the connection of Simphony model through RTI Connection to the COSYE RTI.

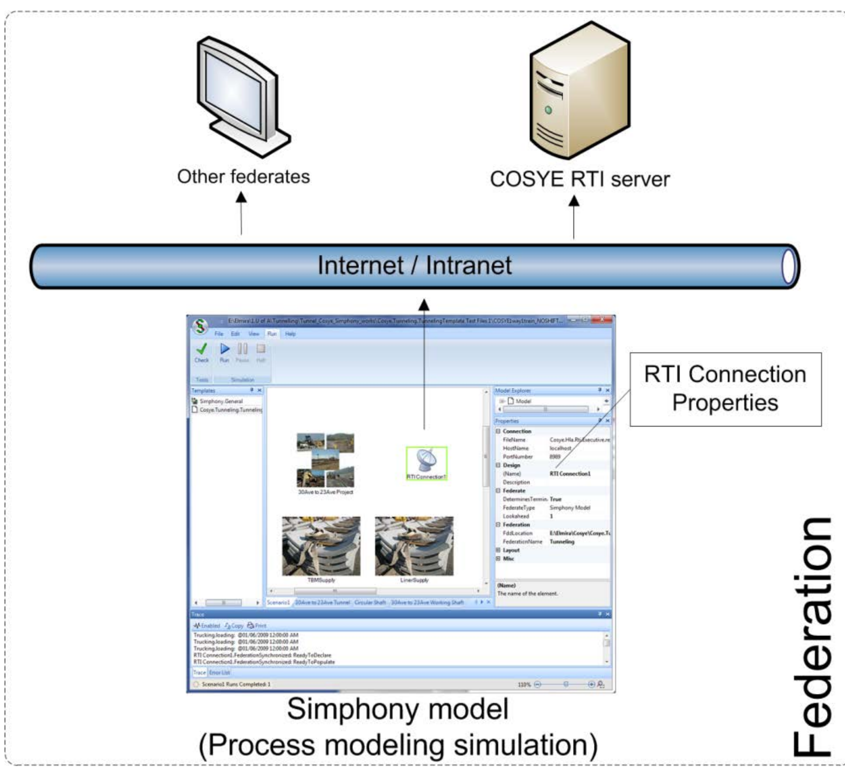


Figure 4: Connection of Simphony model to COSYE

As a test model, a tunnel construction simulation developed in Simphony 3.5 was connected to the COSYE as a process modeling federate in the tunnel federation.

CASE STUDY: TUNNEL CONSTRUCTION SIMULATION

Tunnel construction projects are one of the many categories of projects that have been the subject of researchers for the application of simulation to improve productivity in the system. AbouRizk and Dozzy (1993) created a general purpose simulation model for specific tunnel projects. Touran and Asai (1987) developed a CYCLONE model to control schedule by predicting the advance rate of TBM in soft rock. Runwanpura (2001) describes the development of a special purpose simulation model for tunnel construction by Tunnel Boring Machine (TBM). His Simphony template was developed to optimize the equipment use and predict cost, schedule and resource utilization. Al-Bataineh (2008) developed a scenario-based planning for tunnelling construction based on discrete-event simulation in Simphony.NET. His model considers various project information, including cost, schedule, weather effect, material supply, and dirt handling to select the best scenario. The model uses the concept of distributed simulation in Simphony to create modular elements.

The focus of the aforementioned studies was the development of simulation models (or templates) for utility tunnel construction projects being undertaken in Edmonton, Alberta, Canada, where, based on geotechnical conditions and availability of equipment, a Tunnel Boring Machine (TBM) is usually utilized for tunnel construction purposes. The same construction method is selected in the current project to simulate the construction process of utility tunnels. The development of the proposed system is divided in three major phases:

- Create a tunnel template in Simphony.NET 3.5 to simulate process modeling. The template is the recreation of an existing template that previously developed in older version of Simphony by Al-Bataineh (2008). The new template was enhanced to cover more details of construction process.
- Once the Simphony template was tested and validated based on real case study, all the elements was transformed into the COSYE-aware elements to enable the connection of the model to the COSYE to act as a process modeling federate in the federation.
- Connect the Simphony model to the existing tunnel federation to test the compatibility of the Simphony model. As an initial step, the connection of the Simphony model with a test federate through COSYE RTI is tested.

Tunnel Construction Template in Simphony.NET 3.5

In this study, the existing tunnel template in the previous Simphony (Al-Bataineh 2008) was selected as a baseline for the development of a new template in Simphony.NET 3.5. Al-Bataineh divided the elements into three groups: shaft operation elements (shaft template), tunnel operation elements (tunnel template), and support elements. The entire tunnel construction process can be simulated by shaft and tunnel elements. Calendar, project, trucking, supply and shift control elements are considered as supporting elements in the template and can be reusable for other construction projects. "Project" element reports different cost items (Crew, material, equipment, Dirt removal, and indirect), total cost, start date, and finish date of the project. "Shift control" element can be added to the shaft model and tunnel model separately to reflect shift differences in different work packages in the project. The "Supply" element controls supply of liners and TBM parts for the project and "Trucking" deals with dirt removal in the site.

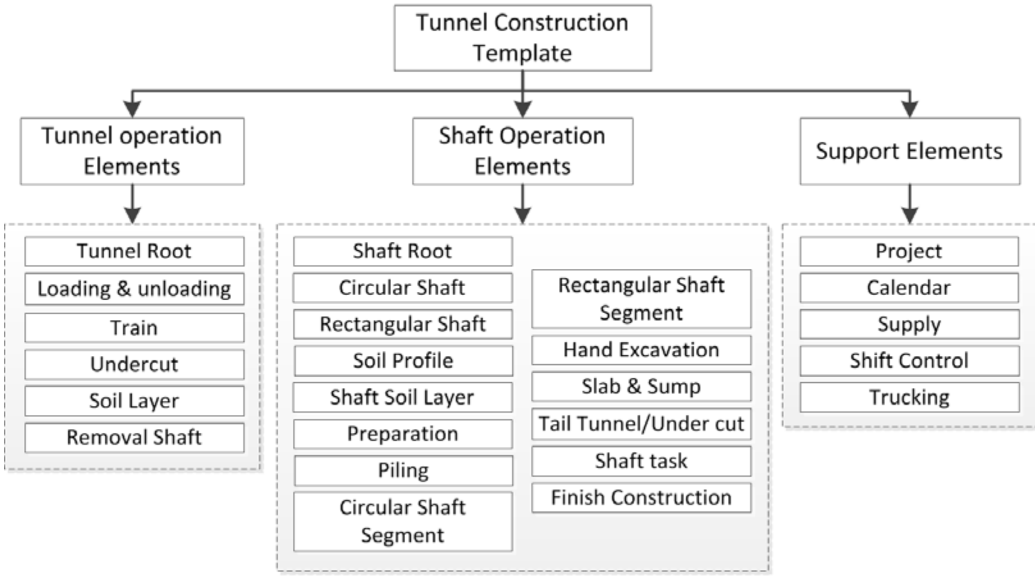


Figure 5: Simphony Tunnel Construction elements

The shaft Template consists of 14 elements; all the tasks in the shaft construction process are encapsulated in these elements. So, each element may contain more than one task scheduled as an event in the internal code. The Shaft model takes into account the soil layers for the shaft excavation. The tunnel template consists of 6 elements that simulate loading and unloading activities on the top of the shaft, different geotechnical tunnel sections along the tunnel, TBM removal, surveying and extending utilities as internal events and the activities inside the undercut such as track installation. The template is also capable of modeling 2-way tunnelling. The entire template has a modular and hierarchical structure; each element is located in specific level, encapsulates certain tasks and communicates with other elements in the same level or other levels. An example of a tunnel construction model is illustrated in Figure 6.

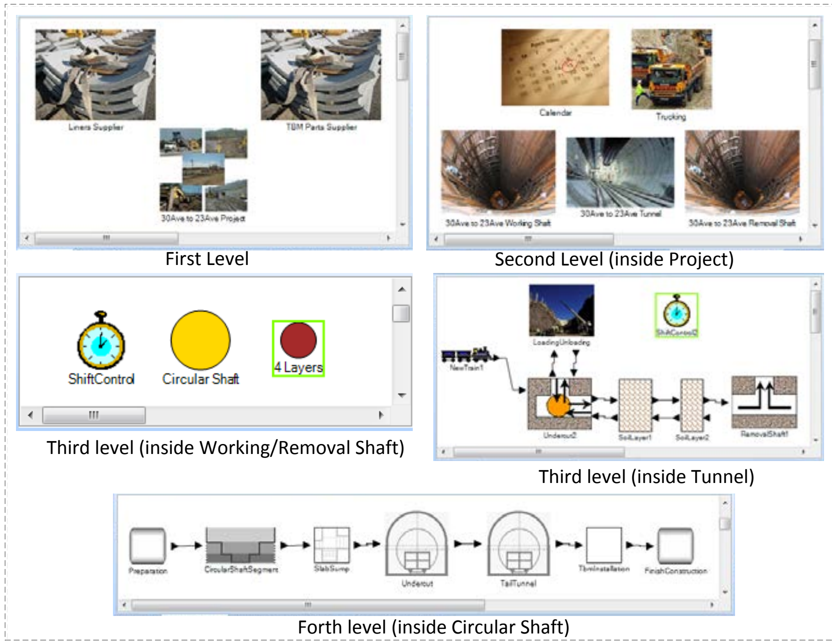


Figure 6: Hierarchical Tunnel Simulation Model in Simphony.NET 3.5

Based on this model, the tunnel construction process starts by excavating a vertical shaft to reach the tunnel invert. The vertical shaft, called the working shaft, provides access to the tunnel during the construction phase for lowering down the material and removing the dirt from the tunnel. Excavation and lining of the working shaft usually happens in different segments based on existing soil layers. Excavation of each segment can be done both by machine and hand depending on the segment depth and geotechnical condition. Usually for TBM installation and material handling during the operation, undercut and tail tunnel as two enlarged areas in the bottom of the shaft are constructed. Tunnel construction starts after lowering down and installation of the TBM. Main activities in tunnel divided into: excavation of a tunnel, removing dirt from tunnel face to the undercut and to the ground level, and lining the section and this will continue until the end of the tunnel. Other activities such as utility extension or surveying are also done in certain time intervals. A retrieval shaft is excavated at the end of the tunnel for TBM removal.

For validation of the model, an actual tunnel construction project that was created and validated in previous Simphony was recreated in the new Simphony and the results were compared. As the new template considers more details of tunnel construction, the results were more accurate.

COSYE- aware elements in tunnel template

To make a COSYE-aware Simphony model, the “RTI Connection” element was added to the model and the related codes are inserted in the related class of elements that need to be COSYE-aware. These elements usually interested in publishing or subscribing to the attributes of the objects. Figure 7 shows an example of code inserted into the elements.

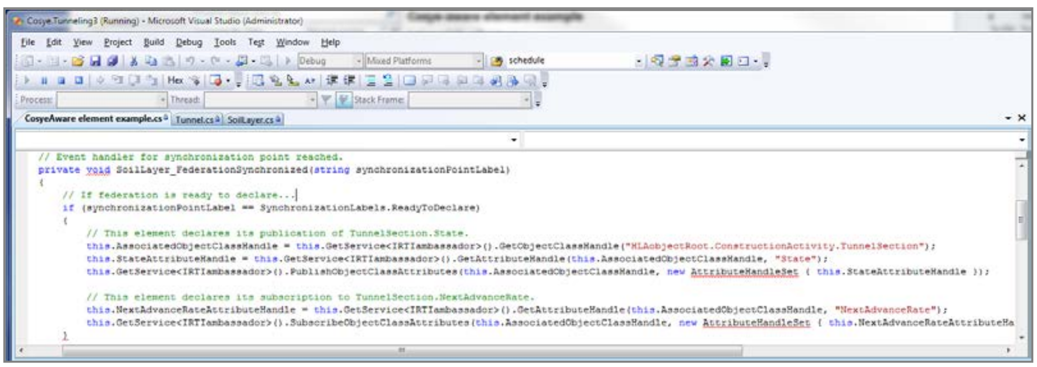


Figure 7: Inserted code for making COSYE-aware element

Tunnel Construction Federation

The first step in developing a federation in COSYE is to create a Federation Object Model (FOM) that includes all the object classes, attributes and interactions that will be required in the federation execution. The FOM classes and attributes should be comprehensive enough to be reusable in the future. In this project, a new FOM is created by modifying the existing tunnel federation FOM. All the classes and attributes in the tunnel template which are shared within the federation were added to the FOM and those classes and attributes related to the previous tunnel, shaft, dirt removal, and supplier federates were deleted if no other federates were "interested" in them. Due to changes in the old FOM, many existing federates required modification to be able to connect to the federation. To test the functionality of the tunnel template in the COSYE federation, a "Test Federate" was developed to communicate with the template. The "Test Federate" basically prints all the updates received from a tunnel Simphony model. Figure 8 shows the test federate and printed results.

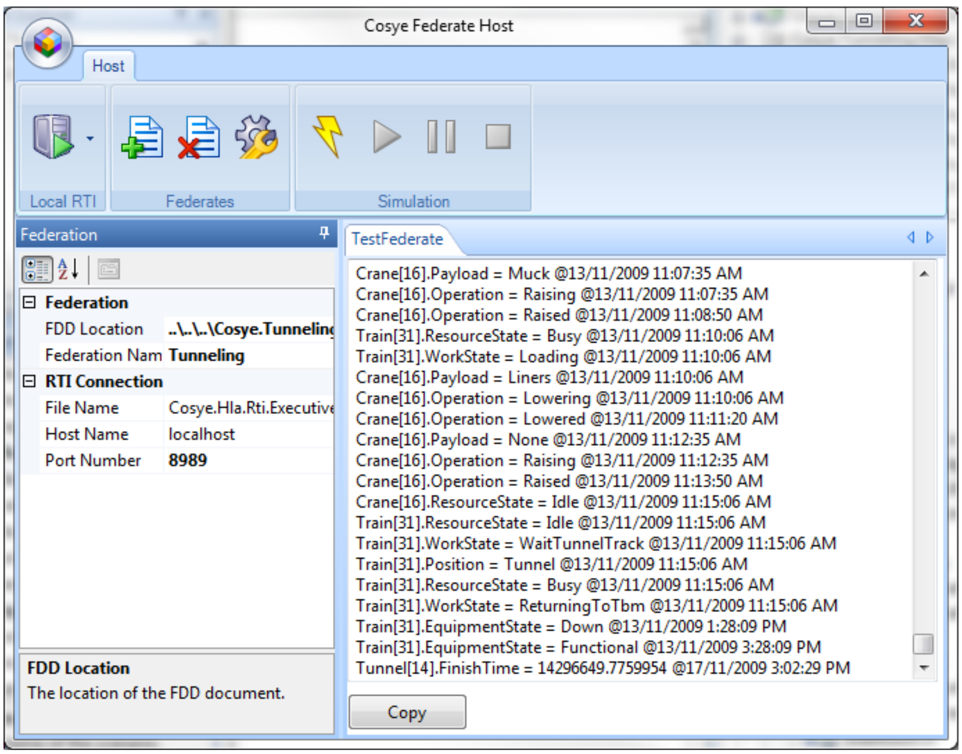


Figure 8: Inserted code for making a COSYE-aware element

FUTURE WORK

Visualization federate and as-built information federates are two important federates that need to be integrated with the tunnel template into the federation. The COSYE framework also enables the connection of CAD models and scheduling software to the federation and communication of different software application with tunnel template.

This will help to automate data transfer from the software to the model and reduce the time of data entry to the system.

CONCLUSION

Existing construction simulation tools usually focus on modeling activities and associated resources to facilitate optimum decision making for the project. These tools are not able to model the entire project life cycle and usually are not developed to work in the integrated system. An HLA-based Distributed simulation environment (COSYE), a more powerful simulation tool, is introduced to provide a collaborative environment to model the different aspect of the project. But the system was developed based on hard-coded simulation components and makes updating the model more challenging as it requires programing skill and regular code updates when changes happen. The limitation of this system in its user interface, as well as the ability of the system to integrate various software applications, encouraged the authors to integrate Simphony into COSYE as a component of the model, create the process model in this software and use its capabilities of graphical modeling and provided services for easier programming. The integration of Simphony is discussed in this paper and the implementation process is examined through the development of SPS modeling for a tunnel construction project. This approach can be easily implemented for other construction simulation models in Simphony and COSYE.

ACKNOWLEDGEMENTS

This project is funded by the National Science and Engineering Research Council (NSERC) Industrial Research Chair in Construction Engineering and Management research. The authors also wish to thank the City of Edmonton for their extensive collaboration.

REFERENCES

AbouRizk, S. 2010. The role of simulation in construction engineering and management. *J. Constr. Engrg. and Mgmt*, ASCE., 136(10): 1140-1153.

AbouRizk, S., and Hague, S. 2009. An overview of the COSYE environment for construction simulation. *2009 Winter Simulation Conf.*, Association for Computing Machinery, New York, NY, USA, 2624-2634.

AbouRizk, S. 2006. Callaborative Simulation Framework for Multi-user Support in Construction. *Discovery Grant Proposal*, Edmonton, Alberta, CA.

AbouRizk, S., and Dozzi, S. 1993. Applications of computer simulation in resolving construction disputes. *J. Constr. Eng. Manage.*, 119 (2): 355–373.

Al-Bataineh, M. T. 2008. *Scenario-Based Planning for Tunneling Construction*. PhD Thesis . Edmonton, Alberta, Canada: University of Alberta.

Boer, C. A. 2005. *Distributed Simulation in Industry*. ERIM Ph.D. Series Research in Management 65, Erasmus Research Institute of Management, born at Satu Mare, Romania.

Chang, D. Y., and Carr, R. I. 1987. RESQUE: a Resource Oriented Simulation System for Multiple Resource Constrained Processes. *PMI Seminar/Symposium,* Milwaukee, Wisconsin, USA, 4-19.

Fujimoto, R. M. 1999. *Parallel and Distribution Simulation Systems* .1st ed., John Wiley & Sons, Inc., New York, NY, USA.

Hajjar, D., and AbouRizk, S. 1999. Simphony: an environment for building special purpose construction simulation tools. *Winter Simulation Conference*, Phoenix, AZ, USA, 2:998-1006.

Kuhl, F., Weatherly, R., and Dahmann, J. 1999. *Creating Computer Simulation Systems: An Introduction to the High Level Architecture*, Prentice Hall PTR, New Jersey, USA.

Latham, M. 1994. *Constructing the Team, Final Report of the Government/Industry Review of Procurement and Contractual Arrangements in the UK Construction Industry*. HMSO, Department of the Environment, London, England.

Martinez, J., and Ioannou, P. G. 1994. General Purpose Simulation with Stroboscope. *Winter Simulation Conference*, ASCE., San Diego, CA, USA, 1159-1166.

Ruwanpura, J. Y. 2001. *Special purpose simulation for tunnel construction operations*. PhD thesis, Edmonton, Alberta, Canada: University of Alberta.

Touran, A., and Asai, T. 1987. Simulation of tunneling operations. *J. Constr. Engrg. and Mgmt.*, ASCE, 113(4): 554–568.

Wilcox, P. A., Burger, A. G., and Hoare, P. 2000. Advanced distributed simulation: a review of developments and their implication for data collection and analysis. *J. Simulation Practice and Theory*, 8 (3-4): 201-231.

Xie, H. and Moghani, E. 2009. Tunneling Construction Simulation Using High Level Architecture. *16th Annual Canadian Construction Resarch Forum*. Banff, Alberta, Canada, 133-155

Appendix C- Analyzing Transit Tunnel Construction Strategies Using Discrete Event Simulation [1]

Elmira Moghani
Hala AbouRizk
Simaan M. AbouRizk

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

Heiner Sander

ILF Consultants, Inc.
3911 Old Lee Highway, Suite 42E
Fairfax, Virginia 22030, USA

ABSTRACT

Selection of an appropriate construction strategy for a project is one of the challenges faced in the planning stage. It is essential to choose a suitable method that can reduce cost, time, and any disruption in the area, especially for projects in urban areas. The management group must consider possible techniques, test various scenarios using those techniques, calculate the associated cost and time, and determine the most desirable solution. In this research, a simulation based approach was used to assist the management group in choosing the best strategy for construction of a transit tunnel project in Edmonton, Alberta, Canada. A discrete event simulation tool was developed to model a Sequential Excavating method using either shotcrete or rib and lagging as preliminary supporting systems. The tool enables users to create simulation models for different methods and calculate total duration, resource utilization, and cost of the project. The results comparison is demonstrated in this paper.

INTRODUCTION

This paper is published at the proceeding of the 2011 Winter Simulation Conference

, Dec 11-14, 2011, Phoenix, AZ, USA

Selection of a construction strategy among possible alternatives at the beginning of a project is a challenging task which requires analysis of potential scenarios, equipment and material utilization, and human resource allocation to select the best solution, based on estimated cost and total duration of each scenario. A simulation technique can support the decision making process as it enables the modeling of an actual construction process on a computer, and lets managers examine various options, compare their results, and select a desirable solution. In many studies in the construction domain, simulation tools have been used for comparing alternatives. For example, Ioannou and Martinez (1996) used STROBOSCOPE to develop simulation models to compare two construction methods for a tunnel project, the conventional method and the New Austrian Tunneling Method (NATM).

In this project, a simulation approach is applied to compare construction strategies for a transit tunnel project in Edmonton, Alberta, Canada. Based on preliminary studies, the management group has decided that two different construction methods can be implemented in this project: a sequential excavation method with either rib and lagging or shotcrete as primary liners. They also defined different sequences for the excavation process, which created new alternatives to compare.

Simulation models were developed in Simphony.NET 3.5, a simulation tool developed for the construction domain. Simphony is known as a powerful construction simulation tool that provides both general-purpose and special purpose simulation tools and allows users to model construction activities, connect them together, and assign them related resources to represent a real construction process. Simphony provides a graphical user interface; developers can easily modify the process, change the parameters of the model, run the simulation, and get new results. Simphony has been successfully employed in many construction management studies, especially in the area of tunnel construction simulation (Mohamed and AbouRizk (2001), Ruwanpura (2001), Fernando et al. (2003), Al-Bataineh, (2008), Marzok et al. (2008))

In this paper, the process of developing simulation models for two construction strategies for the transit tunnel project is discussed in detail, and analysis of simulation outputs and comparison of results is illustrated.

THE NORTH LRT (LIGHT RAIL TRANSIT) TUNNEL PROJECT

Project Description

The City of Edmonton is constructing a 3.3 km LRT extension from an existing station (Churchill station) to the northwest of the city. This study focuses on a part of this project: two parallel tunnels with a total length of 764 m, extending from the existing underground station to the street level station (MacEwan station).This is the only portion of the project that will be constructed as an underground tunnel. The construction will start from tunnel portals located at MacEwan station and will be continued up to the pre-built section. The remaining sections will be constructed from the end of the pre-built section to Churchill station, as shown in Figure 1.

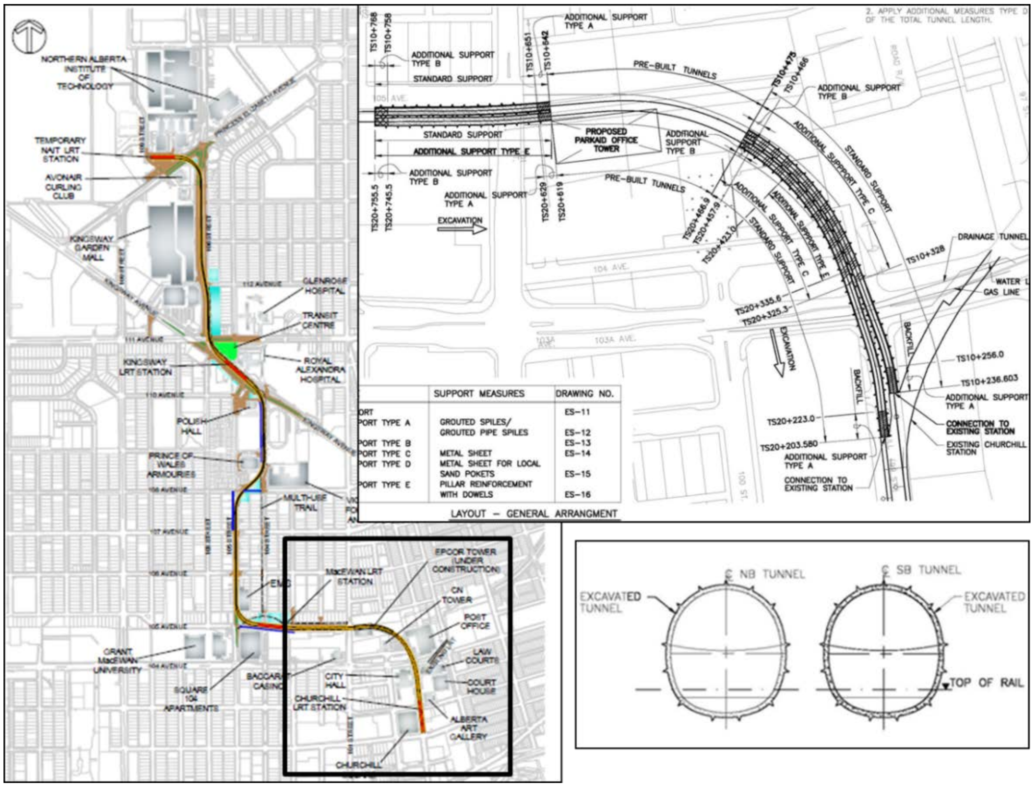


Figure 1 - NLRT Project Layout (IFL Consultants, Inc. (2008))

Sequential excavation was selected as the excavation method in both construction strategies under study. In the sequential excavation process, a cross section is divided in smaller segments based on tunnel size; these segments are excavated in sequence for a one meter advancement of the tunnel. In this project, the cross section is divided in heading and bench sections with approximately similar areas, as shown in Figure 2(a). An excavator, a loader and a truck are considered as a set of equipment for excavation and mucking processes. Two assumptions were made for the construction sequence:

- Excavation and lining of two one-meter heading sections is followed by the excavation and lining of a two-meter bench section. This process is illustrated in Figure 2(b).

- Excavation and lining of a one-meter heading is followed by the excavation and lining of a bench for a specific length of the tunnel.

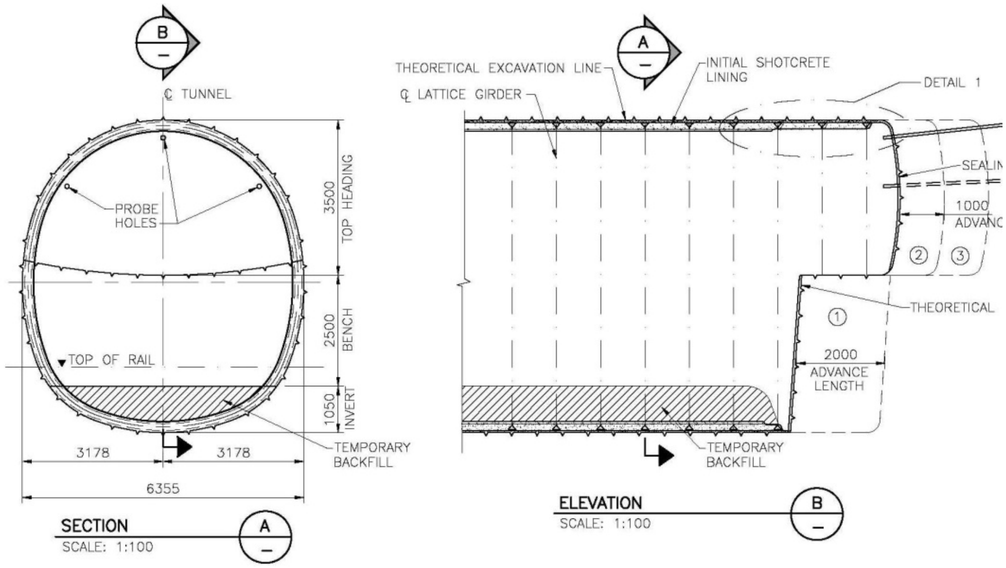


Figure 2 - Tunnel cross section and excavation sequence (IFL Consultants, Inc. (2008))

Sequential Excavation with Rib and Lagging Lining System

In the tunnelling method which uses a rib and lagging lining system, the entire tunnel is lined using the same materials and lining arrangements. The steel ribs are installed every one meter at the end of excavated sections. Laggings, or wooden pieces, are inserted between ribs and soil to support the excavated area. Ribs usually come in one or two sections and are bolted together when they are fixed in their location. A timber spreader and steel rods are used to fix the position of ribs. Based on the tunnel alignment, surveying will be performed every 6-15 meters, and utilities are installed as the tunnel advances.

Sequential Excavation with Shotcrete Lining System

In the shotcrete lining method the preliminary lining differs along the tunnel based on geotechnical condition. The designers for this considered a standard support system for both tunnels which includes installation of different layers of shotcrete and wire mesh. In this project three layers of shotcrete with two layers of wire mesh in between are specified as a standard support.

In addition to the standard support, designers defined five types of additional supports based on soil conditions, and then divided tunnels into longitudinal sections based on their needs for additional supports. All the sections with their additional supports are listed in Table 1. Because for simulation purposes the lengths of each section and the installation durations for the additional supports are the only important factors, the design characteristic of each support will not be discussed in this paper.

Table 1 - Tunnel longitudinal sections based on additional supports for shotcrete lining

NLRT - North bound				
Section	Start point	End Point	Length	Additional Supports*
North 1	768	758	10	E
North 2	758	651	107	-
North 3	651	642	9	A
North 4	642	475	167	Pre-Built
North 5	475	466	9	B+C
North 6	466	328	138	C
North 7	328	256	72	-
North 8	256	236.6	19.4	A

*All the sections have the standard support

NLRT - South bound				
Section	Start point	End Point	Length	Additional Supports*
South1	755.5	745.5	10	B+E
South2	745.5	629	116.5	E
South3	629	619	10	A+E
South4	619	466.9	152.1	Pre-Built
South5	466.9	457.9	9	B+C
South6	457.9	423	34.9	C
South 7	423	335.6	87.4	C+E
South8	335.6	325.3	10.3	C
South9	325.3	223	102.3	-
South10	223	203.58	19.42	A

*All the sections have the standard support

NLRT TUNNEL CONSTRUCTION SIMULATION IN SIMPHONY

In this research, simulation models were developed in Simphony.Net 3.5, a Microsoft Windows-based construction simulation tool to model a discrete event simulation system. This software is the latest version of Simphony, which was originally developed at the University of Alberta (Hajjar and AbouRizk 1999). Simphony provides a framework for developing General Purpose Simulation (GPS) and Special Purpose Simulation (SPS) templates to help users in academia and industry create construction process models, based on their knowledge of simulation as well as the construction domain. Using GPS template, users build models utilizing abstract elements such as activities, queues, and resources. SPS templates provide a set of elements related to a particular construction domain, which makes simulation more accessible for industry. Simphony has a graphical user interface and hierarchical modeling capability. Users can drag and drop elements into the Simphony modeling interface and connect them based on the logic of a given process. Resources are assigned for different activities in the process and statistical results are available for every resource in the process.

In Simphony, a simulation model can have any number of runs for Monte Carlo simulation purposes if required. More than one scenario can be modeled in one simulation file, which allows a user to run scenarios and compare them at the same time. Simphony includes statistical outputs and different kinds of reports, such as cost and resource utilization, which are useful for comparison purposes.

Model Development

Four different models were created for the NLRT tunnel project using the GPS template in Simphony.NET 3.5, one for each alternative, as listed in Table 2.

Table 2- Tunnel Simulation Model Alternatives

Alternative	Construction method	Excavation process
Alternative 1	Shotcrete Lining	Excavate Heading and Bench in sequence
Alternative 2	Rib & Lagging Lining	Excavate Heading and Bench in sequence
Alternative 3	Rib & Lagging Lining	Excavate Headings (for the entire tunnel), then Bench sections
Alternative 4	Rib & Lagging Lining	Combination of Alt. 2 and Alt. 3 processes

In Alternative 4, because the soil condition of the tunnel in the middle section (Sections with additional support type C shown in Figure 3) is not good, experts suggested that the excavation of these sections be the same as alternative 3 and the rest constructed the same as alternative 2.

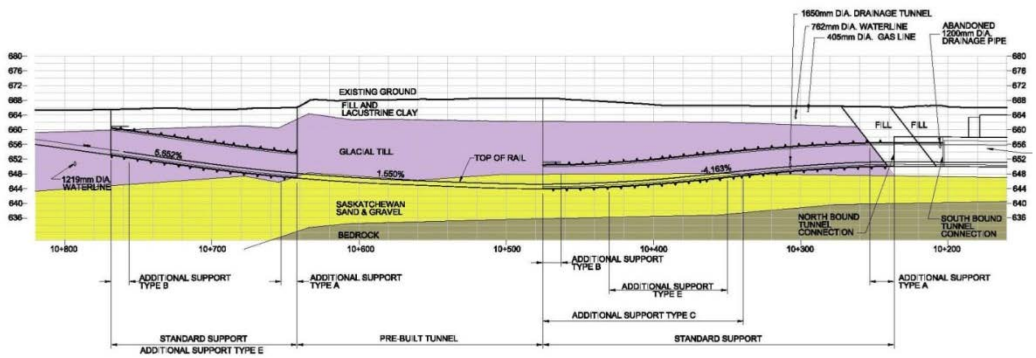


Figure 3 – Tunnel Longitudinal Sections (IFL Consultants, Inc. (2008))

The simulation algorithm for Alternative 1 is shown in Figure 4. The same algorithm with a different list of activities for heading and bench construction was considered for the rib and lagging lining in alternative 2. In alternative 3, the process is simpler as the construction of headings is completed before construction of bench sections.

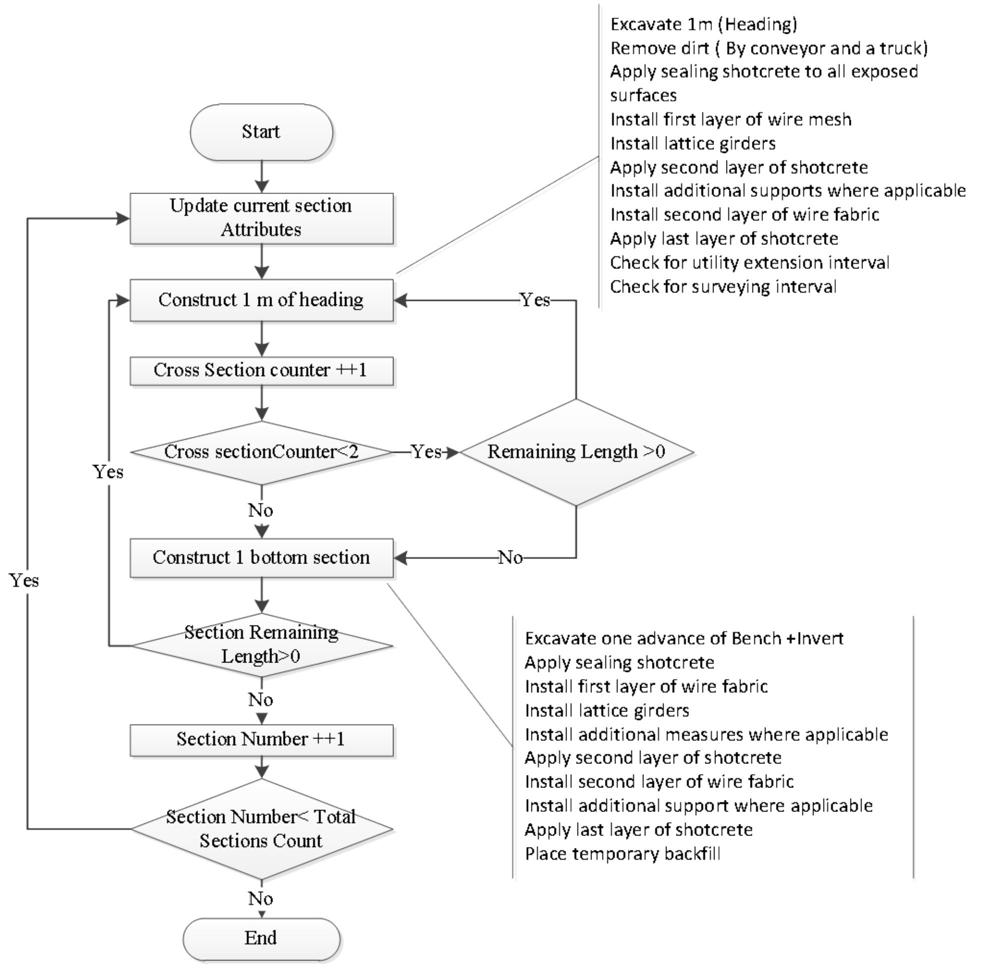


Figure 4 – Simulation Process Algorithm

To create graphically similar models and ease development, the simulation model for the rib and lagging method was created in a similar manner to the shotcrete lining method and also considered the longitudinal sections. This did not affect the simulation process and outputs but resulted in models with similar appearances in the Simphony user interface. Figure 5 shows the simulation model for both construction methods. As it is shown in this figure, in both models the south and north bound tunnels are simulated in the same Simphony file so the limitation of start time for the north bound tunnel can be considered in the model (via a valve element at the beginning of the north bound process model which controls an activator element in the south bound model; when the length of south bound construction reaches 30m, the activator opens the valve and allows the process in the north bound tunnel to start.)

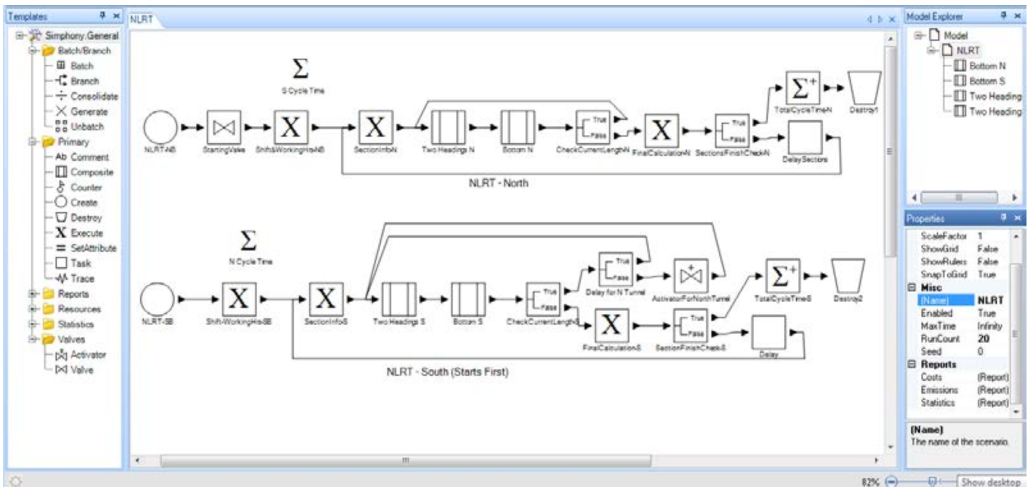


Figure 5 – Simulation model for different construction strategies for NLRT project

In order to make the model more organized and understandable for the user, the hierarchical feature of Simphony was used; all activities for the excavation and lining of the heading and bench are encapsulated in two composite elements (i.e., an element that has no simulation behaviour and is used for grouping elements). Inside of each composite element two cycles are modeled; the main cycle simulates the excavation and lining process depending on the type of construction method and the other one simulates the trucking cycle for removing dirt to the outside of the tunnel, which is similar for both methods. Figure 6 shows the process of excavation and lining of one heading.

The sequence of tasks and their duration are obtained from historical data and expert opinion. The durations are presented as minimum, maximum, and most likely values (i.e., a triangular distribution) for each activity. The statistical elements record the duration of each cycle for different simulation runs.

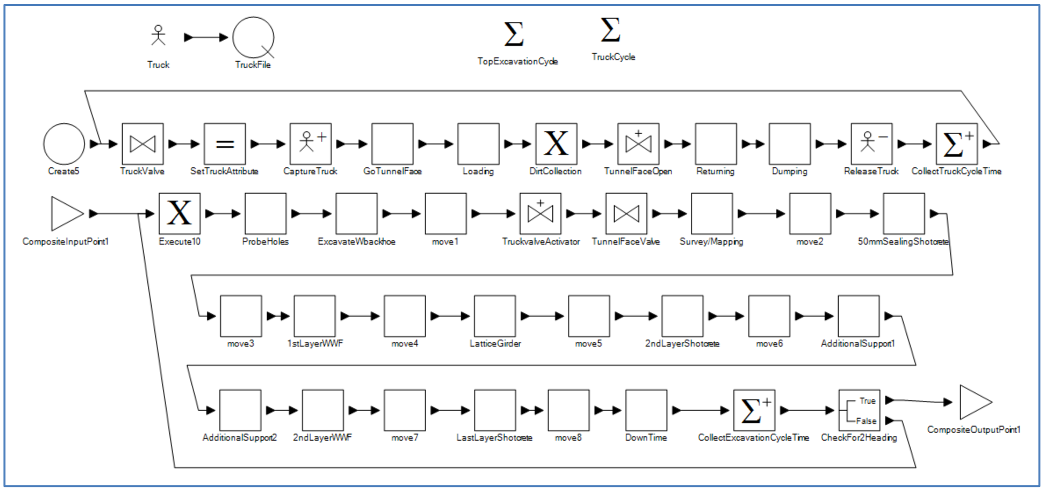


Figure 6 – Simulation model for the excavation and lining of one heading section

SIMULATION RESULTS AND ANALYSIS

Each simulation model was executed for 100 runs, to perform Monte Carlo simulation analysis and to provide users with statistical output such as resource utilization and cycle times. Also, for every simulation model, different scenarios were defined by changing inputs such as resources, shift length, number of shifts per day and number of working days per week. This allows users to test possible situations, compare outputs and select the best solution. As an example, the results for Alternative 1 are included in Table 3.

Table 3 – Simulation result for Alternative 1 - Shotcrete Lining

		Mobilization (Week)	Shift Length (H)	Shift Per Day	Days Per Week	working weeks	Total weeks
North	Scenario 1	5	10	2	5	44	53
	Scenario 2	5	12	2	5	36	45

		Mobilization (Week)	Shift Length (H)	Shift Per Day	Days Per Week	working weeks	Total weeks
South	Scenario 1	5	10	2	5	51	66
	Scenario 2	5	12	2	5	40	55

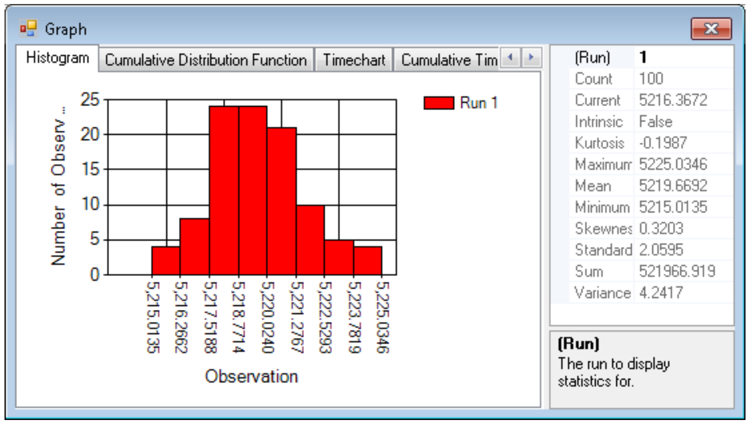
5-week and 11-week mobilization durations were considered for the north bound and south bound portions of the project, respectively. A delay after the pre-built section was also taken into consideration, to transfer equipment and mobilize tunnel parts. The results of simulation models for other alternatives and their associated scenarios are summarized in Table 4.

Table 4 - Simulation Result for Alternative 2 – Rib & Lagging

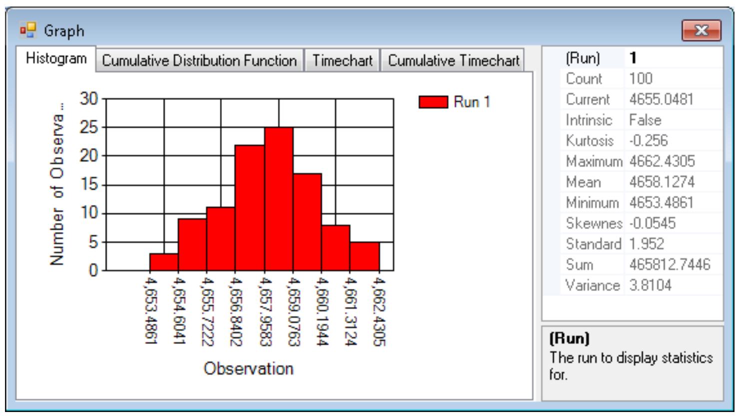
		Mobilization (Week)	Shift Length (H)	Shift Per Day	Days Per Week	working weeks	Total weeks
North	Scenario 1	5	10	2	5	75	84
	Scenario 2	5	12	2	5	62	71

		Mobilization (Week)	Shift Length (H)	Shift Per Day	Days Per Week	working weeks	Total weeks
South	Scenario 1	5	10	2	5	80	95
	Scenario 2	5	12	2	5	66	81

Two other scenarios gave results which were relatively close to alternative 2. This shows that shotcrete lining takes less time than rib and lagging lining systems, which would mean less interruption of traffic in the downtown area. Based on construction duration, assumptions for the equipment list, and number of crews available, decision makers should decide which alternative would be the best solution according to cost and duration.



(a)



(b)

Figure 7 – Cycle time of (a) north bound and (b) south bound tunnels of the project, using the shotcrete lining method

The Simphony model also provides statistical results for the cycle time of tunnel construction; since 100 runs were considered for each model, the statistical results contained a range of numbers with a minimum, maximum and mean value of the range. Figure 7 and Figure 8 illustrate the cycle time (hours) for construction of the north and south bound tunnels of the NLRT project for shotcrete lining and rib and lagging methods.

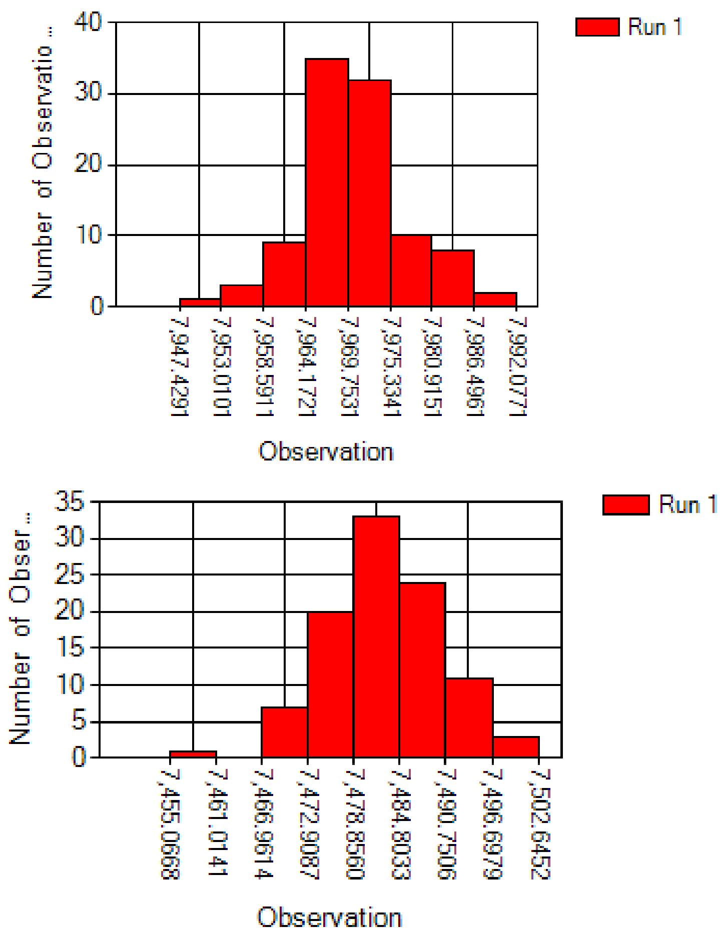


Figure 8 – Cycle time of (a) north bound and (b) south bound tunnels of the project using the rib & lagging method

The cost for the construction processes was divided into equipment cost, crew cost, and material cost. Table 5 lists the approximate cost of each method, without considering overhead cost or any costs related to final lining. The estimation shows similar values for both methods; however, the overhead cost of the project (not shown) will be much higher for the rib and lagging method because of the longer duration.

Table 5 - Simulation Result – Cost estimation

Method	Length (m)	Duration (Weeks)	Shift (h)	Crew Cost	Equip. Cost	Material Cost	North/South Tunnel Cost	Tunnel Operation Cost
Shotcrete (North)	364	36	12	6,120,000	2,941,120	2,847,936	11,909,056	25,067,000
Shotcrete (South)	399.8	40	12	6,800,000	3,230,384	3,128,035	13,158,419	
R&L (North)	364	62	12	8,370,000	910,000	910,000	10,190,000	21,099,000
R&L (South)	399.8	66	12	8,910,000	999,500	999,500	10,909,000	
Shotcrete (North)	364	44	10	7,480,000	2,948,400	2,847,936	13,276,336	28,305,000
Shotcrete (South)	399.8	51	10	8,670,000	3,230,384	3,128,035	15,028,419	
R&L (North)	364	75	10	10,125,000	910,000	910,000	11,945,000	24,744,000
R&L (South)	399.8	80	10	10,800,000	999,500	999,500	12,799,000	

Based on the time and cost analysis, the sequential excavation method with shotcrete lining seems to be the better solution. However, the final decision rests with the project managers, who will also consider availability of equipment and crew and time and cost limitations.

SUMMARY AND CONCLUSION

This paper presents an application of simulation for the selection of a construction method for a transit tunnel project. Simulation, a powerful tool in construction management, helps decision makers to choose the optimum construction scenario, considering limitations of time, money and space. This research utilizes the General Purpose Simulation template in Simphony.NET 3.5 to develop models for two construction strategies suitable for the tunnel project under study. Since the main concern of the client was the project duration, the output results were designed to demonstrate the cycle time for the tunnel construction. Four alternatives for the construction process and two scenarios for each alternative were analyzed for this project. Based on the simulation results, the sequential excavating method with shotcrete lining had the shortest duration, which means it can save time and decrease traffic disruption in the area.

ACKNOWLEGEMENT

The authors would like to thank the National Science and Engineering Research Council (NSERC) for financial support and the City of Edmonton for their extensive collaboration.

REFERENCES

Al-Bataineh, M. T. 2008. "Scenario-Based Planning for Tunneling Construction." Doctoral dissertation, University of Alberta, Edmonton, Alberta, Canada.

Fernando, S. , Er, K. C. , Mohamed, Y. , AbouRizk, S. and Ruwanpura, J. 2003. "A review of simulation applications for varying demands in tunneling." In Proceedings of *the Construction Research Congress;* ASCE, Honolulu, Hawaii, USA, pp. 19–21.

Hajjar, D., and AbouRizk, S. 1999. "Simphony: an environment for building special purpose construction simulation tools". In Proceedings of *the Winter Simulation Conference*, Phoenix, AZ, USA, 2:998-1006.

IFL Consultants, Inc. 2008, "North LRT Extension–Downtown to NAIT, SEM Tunneling." Technical Report ILF/USA: C055 AD-0002 Rev.0.01 081015, Fairfax, Virginia, USA

Ioannou, P. G., and Martinez, J. C. 1996. "Comparison of construction alternatives using matched simulation experiments." *J. Constr. Eng. Manage*., 122(3), pp. 231–241.

Marzok, M., M., Abdallah, and M., El-Said. 2008. "Tunnel-sim: Decision support tool for planning tunnel construction using computer simulation." In Proceedings of *Winter Simulation Conference,* Miami, Florida, USA, pp. 2504-2511.

Mohamed, Y., and AbouRizk, S.M. 2001. "Optimizing tunneling operations using simphony's special purpose simulation library." In Proceedings of *the 8th Canadian Construction Research Forum,* Kananaskis, Alberta, Canada.

Ruwanpura, J. Y. 2001. "Special purpose simulation for tunnel construction operations". Doctoral dissertation, University of Alberta, Edmonton, Alberta, Canada. 2, 1512-1520.

Appendix D- User Manual for the Tunnel Construction Template in Simphony 4.0 [1]

OBJECTIVE

The tunneling template is a Special Purpose Simulation template which is meant to be used for modeling entire tunnel construction operations, including shaft and tunnel excavation, with an adequate level of detail and short implementation time. The template allows users to compare different scenarios, detect problems that may happen during the simulation, develop the optimum solution in order to increase the productivity, and reduce cost for the construction and operation of a tunnel. A tunnel template is divided into three categories: Shaft Elements, Tunnel Elements, and Support Elements. Shaft Element is the only element that includes all the activities of shaft construction. The Tunnel category consists of 7 different elements that enable modeling of TBM tunnel construction, as well as sequential excavation tunneling. The Support category consists of 3 elements. More detailed explanation of each element and the functionality of each element in the template will be discussed later in this document. This report explains different aspects of the template in the latest version of Simphony (Simphony 4.0) in detail.

TUNNEL CONSTRUCTION OVERVIEW

Tunnel construction occurs in different phases; if the tunnel is located in lower depths, in order to get access to the tunnel, a vertical shaft called a construction shaft (working shaft) is usually excavated. This shaft is the main access for lowering the equipment into the tunnel or removing the excavated dirt to the ground level during the construction

[1] This manual was created in Dec. 2010 and revised in May 2012

process. Shaft construction is usually done in sections with the depths dependent on the soil type and geometry of the shaft. The excavation and lining processes are done for each section sequentially. After finishing the excavation and lining of shaft sections, the tunnel construction process can start.

Tunnel construction methods vary depending on geotechnical information, availability of equipment, and the geometry of a tunnel. For long tunnels or tunnels with a large diameter, a Tunnel Boring Machine (TBM) is typically utilized for excavation and lining processes, while for short or small tunnels, hand excavation (Sequential Excavation) is a better option. Before starting the tunnel construction, the tail tunnel and undercut area (an enlargement at the bottom of the shaft used for staging material handling and dirt removal operations) may be excavated.

Tunnel Excavation with TBM

TBM tunneling construction activities are divided into:

- Excavating the tunnel.
- Removing the dirt from the excavation area and transferring it to the working shaft (using muck carts).
- Hoisting the dirt to the ground level (using a crane, clamshell bucket, gantry, etc.) and transferring it outside the construction area.
- Lowering down the liners and transferring them to the excavation face.
- Lining the tunnel.
- Extending construction and utility services.
- Removing the TBM.

A removal shaft is constructed at the end of the tunnel to remove the TBM. In this method, the excavation and lining of the tunnel is done by the TBM. The linings are concrete segments placed by the TBM. Figure 1 shows a typical tunnel layout.

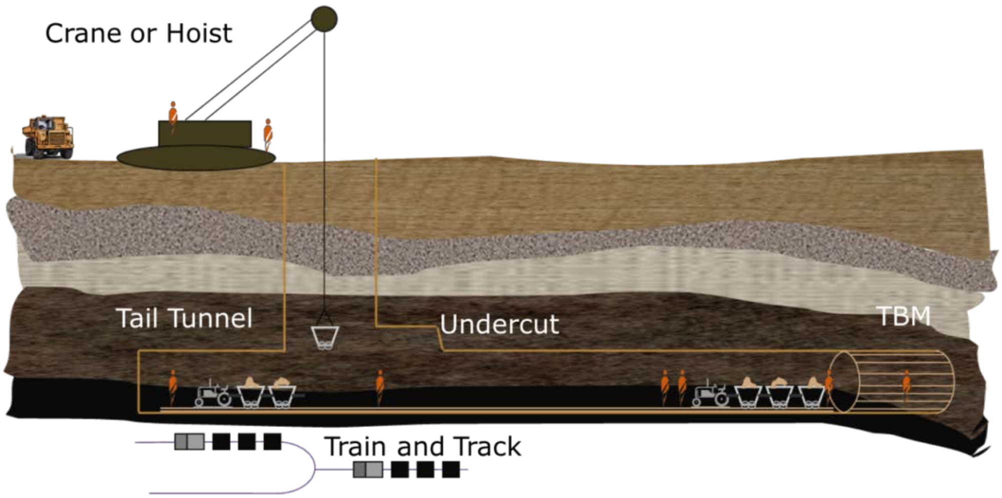


Figure 1- Tunnel Construction using TBM

Figure 2 illustrates a simulation model for this type of construction method.

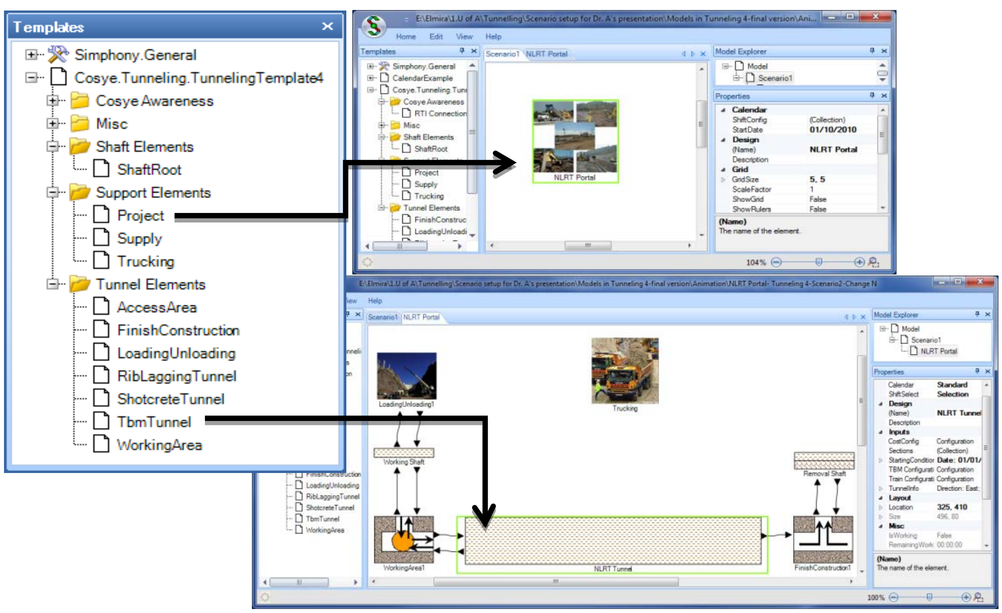


Figure 2 - A Simphony Model for TBM Tunnelling

Tunnel Construction using Sequential Excavation Method (SEM)

The difference between the two methods is the tunnel construction process that models excavation and lining. The excavation of a working shaft (if required) is the same as the previous method, and there is no need to construct a removal shaft unless an access is required at the end of the tunnel. For the tunnel lining, two options are considered; a ribs and lagging lining and a shotcrete lining as a temporary lining are modeled in the template. To be able to model both methods, two different tunnel construction elements were developed while other elements of the template are the same.

In the Sequential Excavation method, the excavation area (tunnel cross section) is divided into smaller segments based on tunnel size. The segments are then excavated and supported sequentially. The excavation can be done by hand or by excavating equipment such as backhoes and roadheaders. In medium-sized tunnels, usually a cross section is divided into heading and bench segments as it is shown in Figure 3. The sequence of excavation can be different based on the ground condition:

- Excavation of heading and bench happens sequentially; the length of excavation can vary based on the design requirements. For instance, the excavation and lining of the heading segment can proceed two meters, and then the bench segment can be constructed. Usually, the heading segments are ahead of the bench segments to provide enough space as a construction platform for the heading part.
- The heading segment can be excavated throughout the tunnel before the bench excavation starts.

Figure 3 – Tunnel Sequential Excavation Method

In this excavation method, the tunnel is usually lined using rib and lagging or shotcrete materials, based on the geotechnical condition (Figure 4).

Figure 4 – Tunnel Lining in Sequential Excavation Method

Figure 5 shows a Simphony model for Sequential Excavation tunneling.

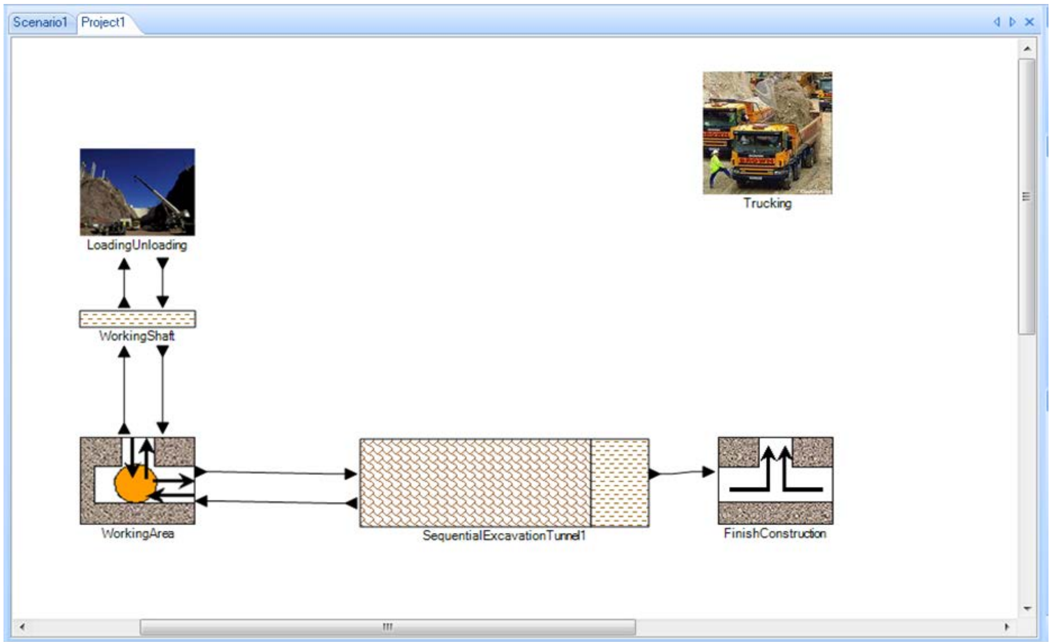


Figure 5 – Simphony model for Sequential Excavation Method

SHAFT CONSTRUCTION ELEMENT

This category consists of one element that encapsulates all the tasks and related resources required for the shaft construction.

Shaft Root

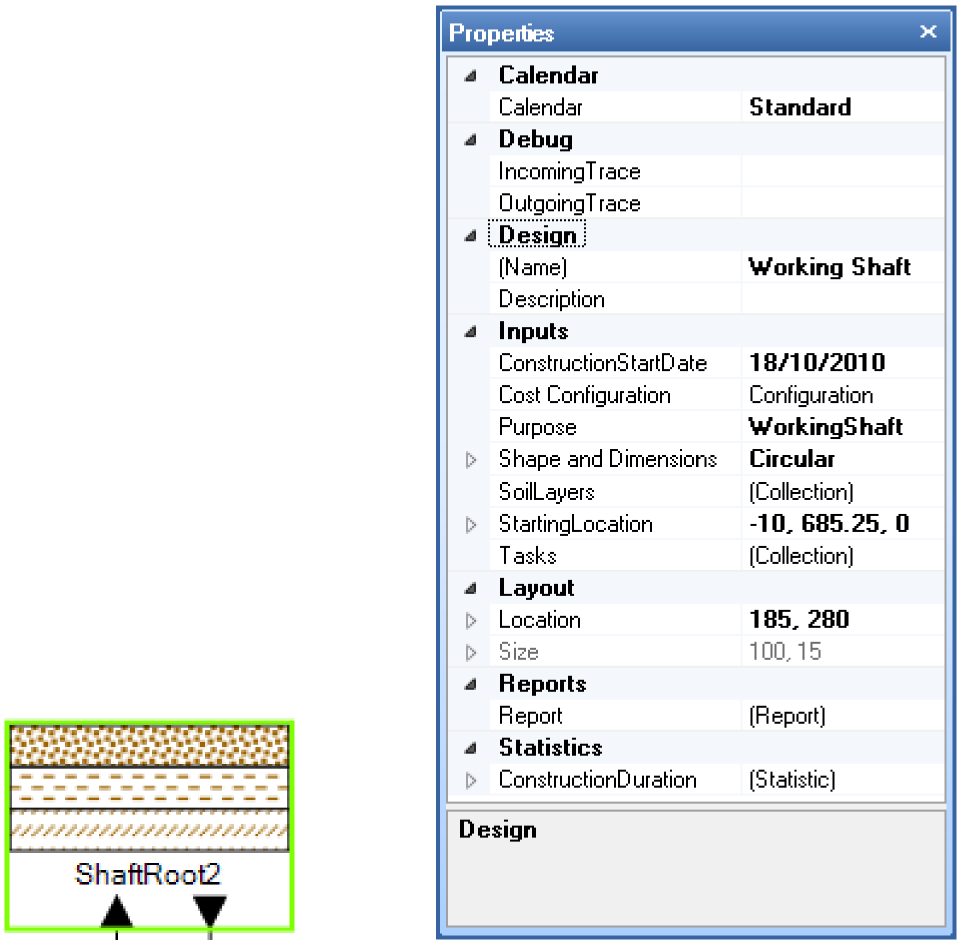


Figure 6 – Shape and Properties of the Shaft Element

Input Parameters:

Calendar:

Attribute Name	Type	Default value	Description
Calendar	Enumeration	Standard	The Name of the Element

A parameter in this section specifies a shaft calendar including working and non-working days, shifts per day, and crew size per shift. A user should specify different calendars in the scenario level and select one for each work area such as working shaft, tunnel, and removal shaft.

Design:

Attribute Name	Type	Default value	Description
(Name)	String	ShaftRoot1	The Name of the Element
Description	String		The Description of the element

Inputs:

Attribute Name	Type	Default value	Description
Construction Start Date	Date Time	-	The date that the shaft construction begins including site preparation
Cost Configuratio n	Custom data structure	0	The Unit cost of crew, equipment (crane) and material consumed during construction of the shaft
Purpose	Enumeration	0	The purpose that the shaft is used for; it can be working shaft, removal shaft, access shaft, or manhole
Shape and Dimension	Custom data structure	0	The shape and dimension of the shaft (supports circular and rectangular shaft sections)
Soil Layers	List <Soil Layer>	-	A collection of soil layers in the area which shaft is being constructed
Starting Location	Vector 3D	0,0,0	The shaft start point in 3D coordinate
Tasks	List <Task>	-	A collection of tasks for the construction of the shaft

Report:

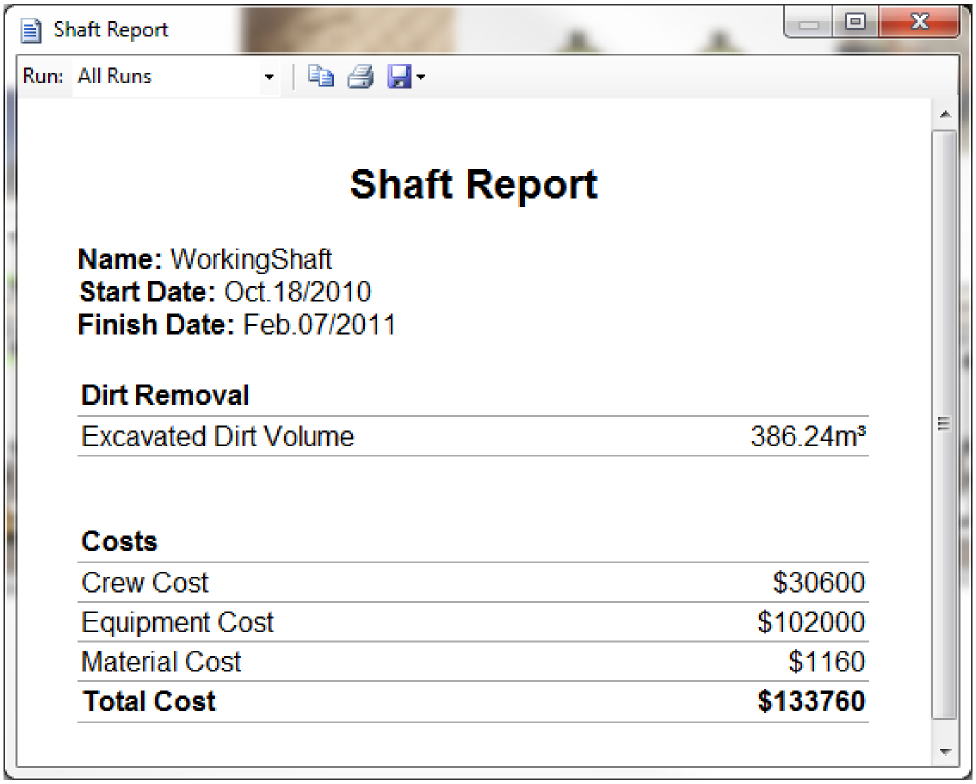


Figure 7 – Shaft Element Report and its Properties

Attribute Name	Type	Default value	Description
Start Date	DateTime	-	The simulated date on which the preparation stage of shaft construction began
Finish Date	DateTime	-	The simulated date on which the shaft construction was completed
Excavated Dirt Volume	Double	-	The volume of dirt excavated so far, measured in cubic metes
Crew Cost	Double	-	The part of the cost of construction activities attributable to wages, especially for direct labour
Equipment Cost	Double	-	The part of the cost of construction activities attributable to equipment (including crane) usage
Material Cost	Double	-	The cost of materials used or consumed during construction of the shaft
Total Cost	Double	-	The sum of Crew, Equipment, and Material costs

Statistics:

Attribute Name	Type	Default value	Description
Construction Duration	Double	0	The duration of shaft construction in days

In the following section, some input parameters will be discussed in more detail along with the screen shots.

- Cost configuration: if a user clicks on a small button on the right-hand side of this input parameter, the following window will pop up. In this window, a user can provide the crew hourly cost, as well as the hourly cost of equipment such as crane and material cost used for one meter of the shaft.

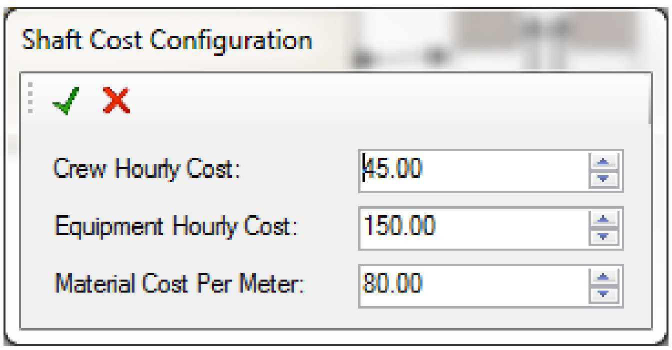


Figure 8 – Cost Configuration in the Shaft Element

- Shape and dimension: shafts can be constructed as circular or rectangular. A user has two options to choose. The key difference between the circular and rectangular shaft is the definition of area. The area for the circular shaft is defined by its diameter, while for the rectangular shaft, length and width of the opening is used. By selecting a particular shape, the parameters will automatically change from diameter to length and width.

Figure 9 – Shape and Dimension Properties in the Shaft Element

- Soil layers: when a user clicks on a button at the right-hand side of the "Soil Layers" property, a window will pop up which includes a collection of soil layers. When the user clicks the "Add" button, a new soil layer will be added to the collection. In the property box of each soil layer, a user can choose a soil type from 15 available types and specify other parameters of the soil. The order of soil layers can be change by clicking the "up" or "down" arrows in the "SoilLayer Collection Editor" window. A shaft can be in one or more soil layers based on the shaft depth and defined soil layer depths in this section.

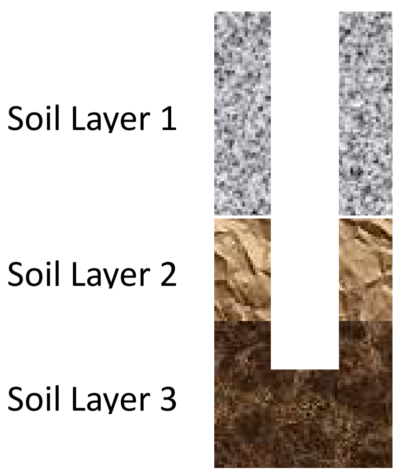


Figure 10 – Soil Layers in the Shaft Location

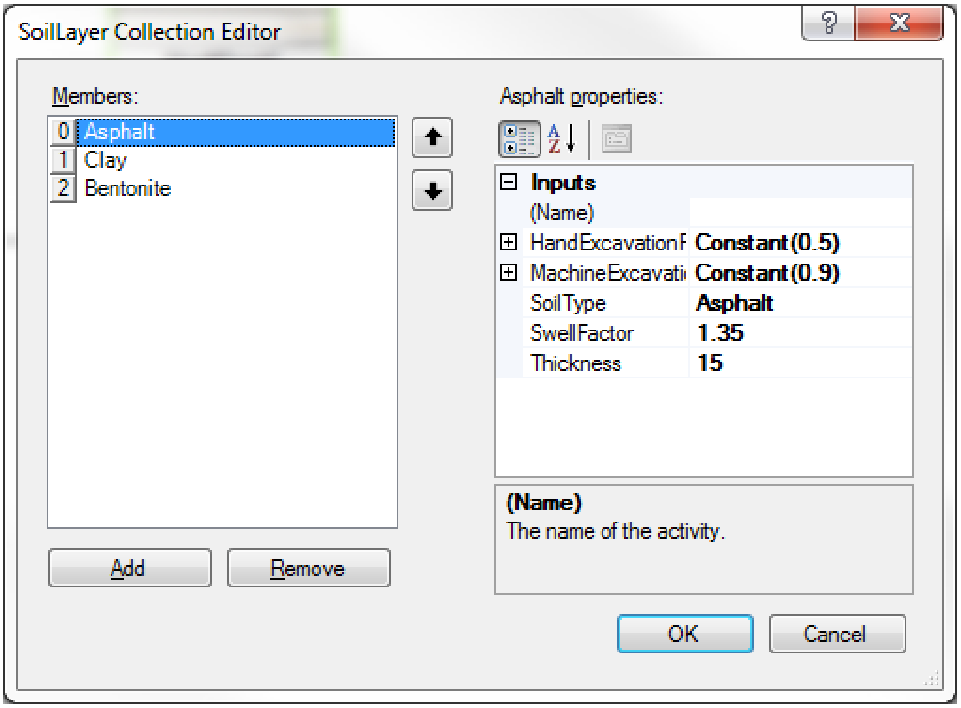


Figure 11– Soil Layers Properties in the Shaft Element

- Tasks: by clicking the button next to the "Tasks" parameter, a window called "Shaft TaskBase Collection Editor" will be opened. By clicking on a little arrow next to the "Add" button, a user can see and select activities for the shaft construction and enter the inputs for each activity in the property grid. The order of the tasks can be changed using the up and down arrow ("Finish Construction" task should be always added as the last task in this collection for all kinds of shafts).

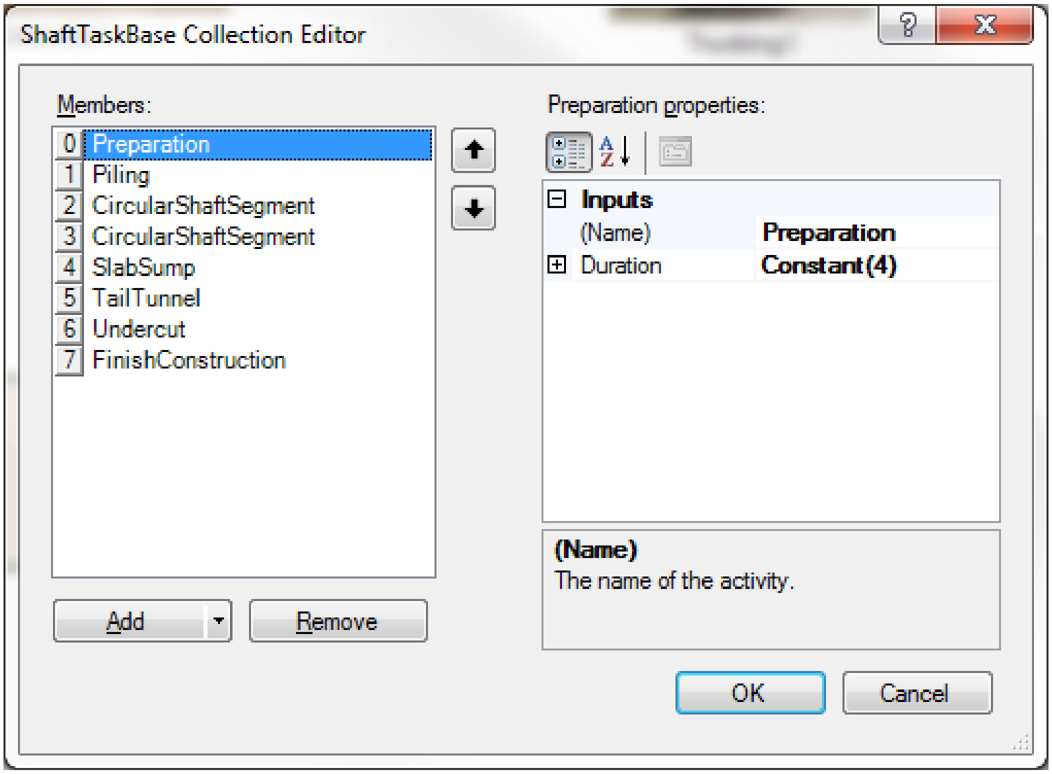


Figure 12– List of Shaft Construction Activities in the Task Property of the Shaft Element

The shaft root element can simulate the following tasks:

- Preparation
- Piling
- Circular shaft segment/rectangular shaft segment
- Slab and sump
- Tail tunnel
- Undercut
- Finish construction

Description:

This element represents the site preparation stage of the shaft construction.

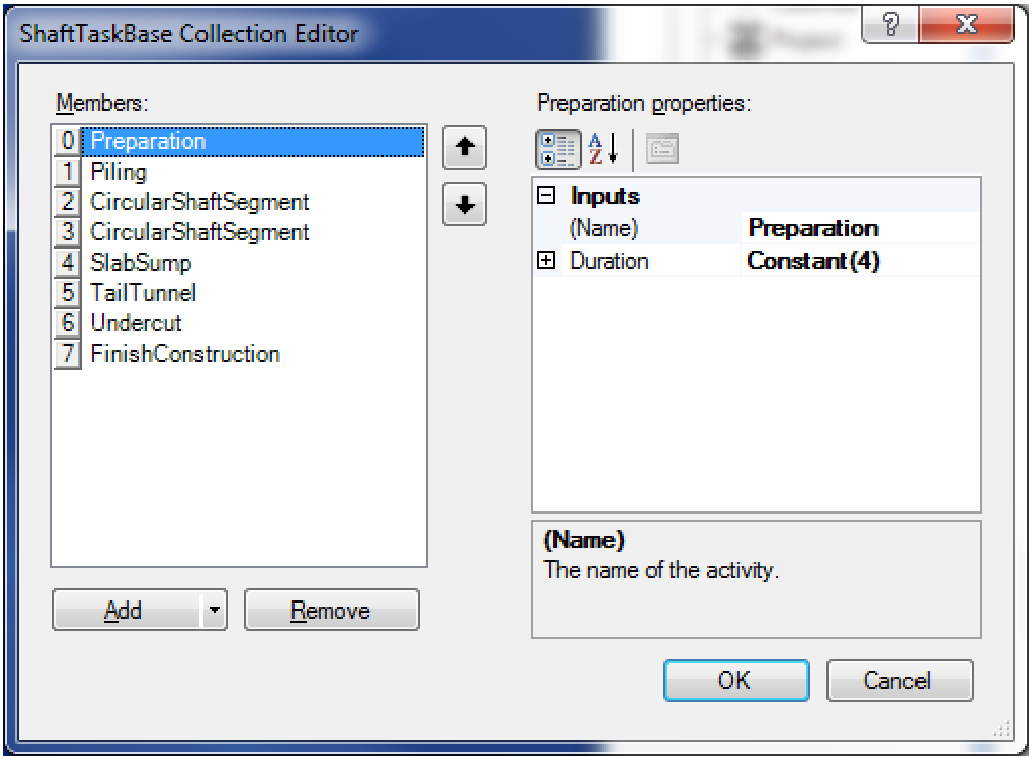


Figure 13– Preparation Task and its Properties in the Shaft Element

Inputs:

Attribute Name	Type	Default value	Description
(Name)	String	Preparation	The Name of the Element
Duration	Distribution	Constant(4)	The duration of the site preparation activities, in days

Output Parameters: None

Statistics: None

Piling

Description:

This element represents the piling activity of shaft construction.

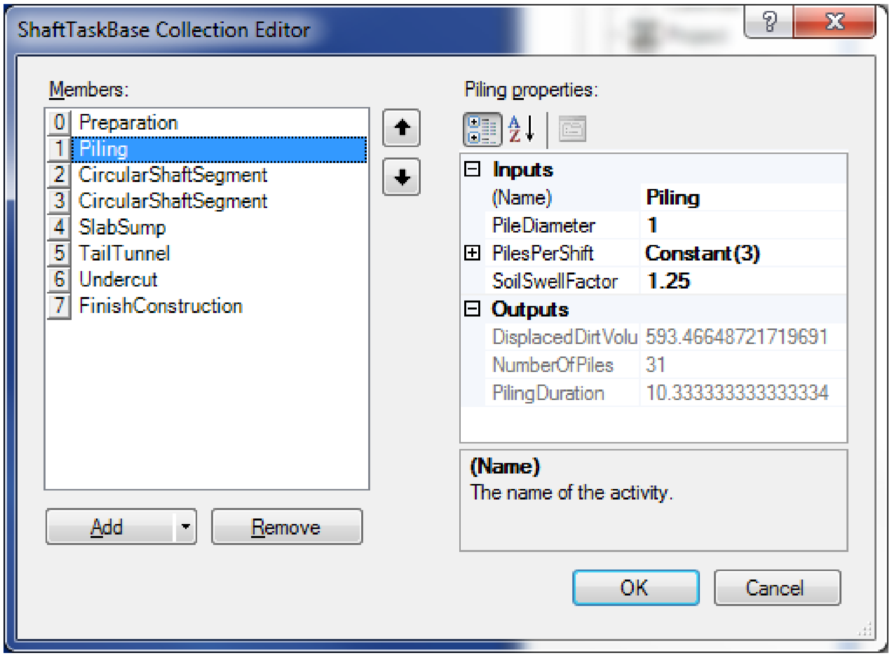


Figure 14– Piling Task and its Properties in the Shaft Element

Input Parameters:

Attribute Name	Type	Default value	Description
(Name)	String	Piling	The Name of the Element
Pile Diameter	Double	1	The diameter of the piles, in meters
Piles per Shift	Distribution	Constant(3)	The number of piles that can be driven in a single shift
Soil Swell Factor	Double	1.25	The swell factor of the soil around the shaft

Output Parameters:

Attribute Name	Type	Default value	Description
Displaced Dirt Volume	Double	0	The volume of dirt displaced by the piling process, measured in cubic meters after piling
Number of Piles	Integer	0	The number of piles necessary, given the pile diameter and shaft dimensions
Piling Duration	Double	0	The duration of the piling activity, in days

Statistics: None

Circular Shaft Segment/Rectangular Shaft Segment

Description:

The Circular Shaft Segment represents a circular shaft segment and encapsulates the liner assembly, machine excavation, and beam and hang shaft installation processes of that segment. The Rectangular Shaft Segment encapsulates the machine excavation dimensions and Waler Installation processes of that segment. If the machine excavation dimensions are less than the shaft dimensions, the remaining area will be excavated by hand.

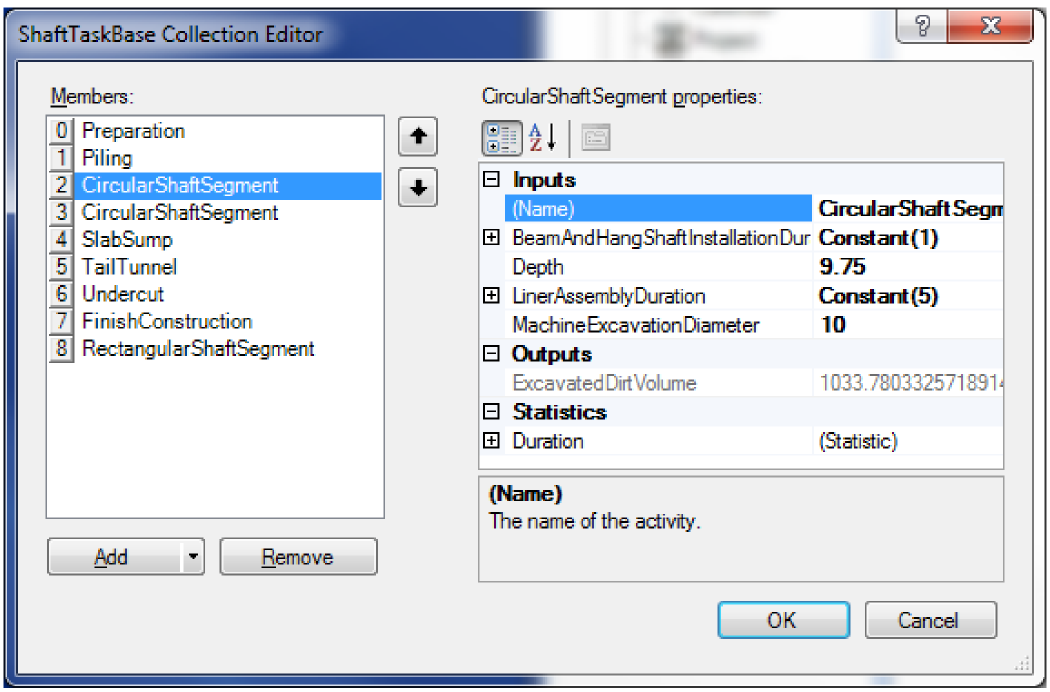


Figure 15– Circular Shaft Segment Construction Tasks and Properties in the Shaft Element

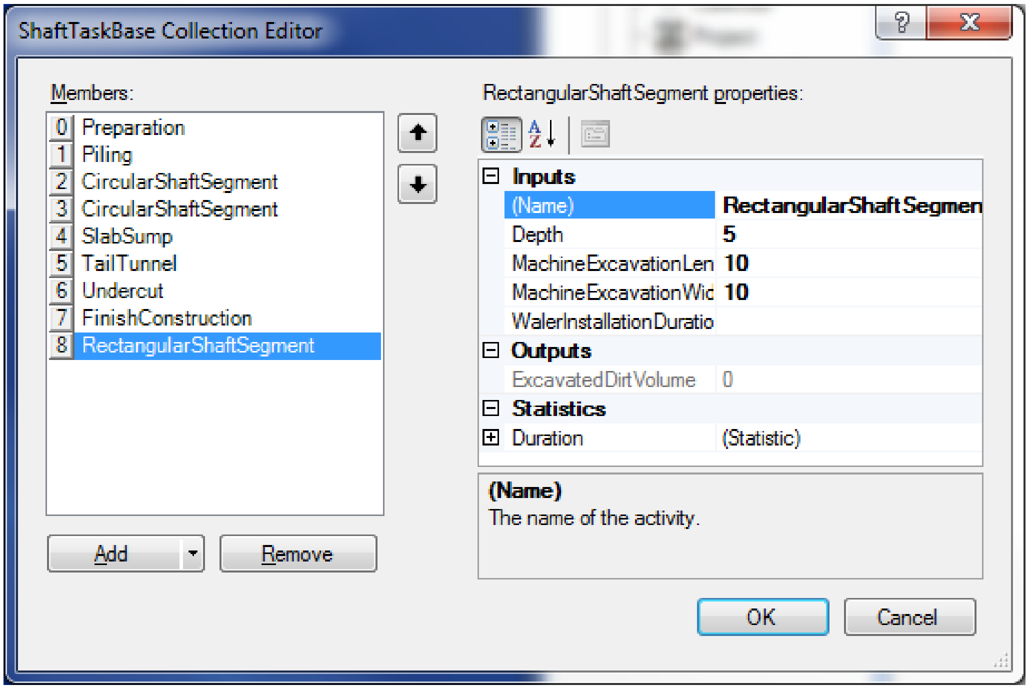


Figure 16– Rectangular Shaft Segment Construction Tasks and Properties in the Shaft Element

Input Parameters:

Attribute Name	Type	Default value	Description
(Name)	String	CircularShaft Segment / RectangularShaftSegment	The Name of the Element
Depth	Double	9.75	The distance between the top and bottom of the segment, in meters
Bean and Hang Shaft Installation Duration (Circular)	Distribution	Constant(1)	The time that it takes to install the bean and hang shaft for this segment, in hours
Liner Assembly Duration (Circular)	Distribution	Constant(5)	The time that it takes to assemble the liner for this segment, in shifts
Machine Excavation Diameter (Circular)	Double	14.67	The diameter of the area of this segment that will be excavated by machine, in meters
Machine Excavation Length (Rectangular)	Double	10	The length of the area of this segment that will be excavated by machine, in meters

Machine Excavation Width (Rectangular)	Double	5.5	The width of the area of this segment that will be excavated by machine, in meters
Waler Installation Duration (Rectangular)	Distribution	Constant(5)	The time it takes to install the waler of the rectangular segment, in days

Output Parameters:

Attribute Name	Type	Default value	Description
Excavated Dirt Volume	Double	0	The volume of dirt excavated from this shaft segment, measured in cubic meters after excavation

Statistics:

Attribute Name	Type	Default value	Description
Duration	Statistic		The time between the start of excavation and the end of any post-segment excavation activities, in hours

Slab Sump

Description:

This element simulates the process of excavating the sump and pouring the slab.

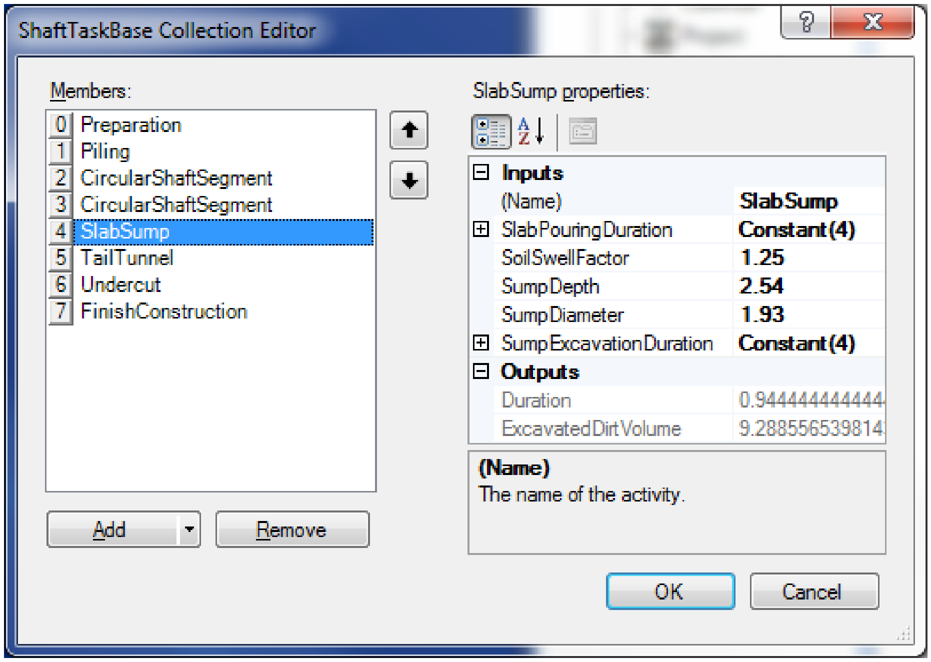


Figure 17– Slab and Sump Construction Tasks and Properties in the Shaft Element

Input Parameters:

Attribute Name	Type	Default value	Description
(Name)	String	SlabSump	The Name of the Element
Slab Pouring Duration	Distribution	Constant(4)	The time that it takes to pour the slab, in hours
Soil Swell Factor	Double	1.25	The swell factor of the soil to be excavated for the sump
Sump Depth	Double	2.54	The depth of the sump, in meters
Sump Diameter	Double	1.93	The diameter of the sump, in meters
Sump Excavation Duration	Distribution	Constant(4)	The time that it takes to excavate the sump, in hours

Output Parameters:

Attribute Name	Type	Default value	Description
Duration	Double	0	The time taken to excavate the sump and pour the slab, in days
Excavated Dirt Volume	Double	0	The volume of dirt excavated for the sump, measured in cubic meters after excavation

Statistics: None

Tail Tunnel/Undercut

Description:

This element represents the construction processes of a tail tunnel or undercut. The process is comprised by sections defined in a data table.

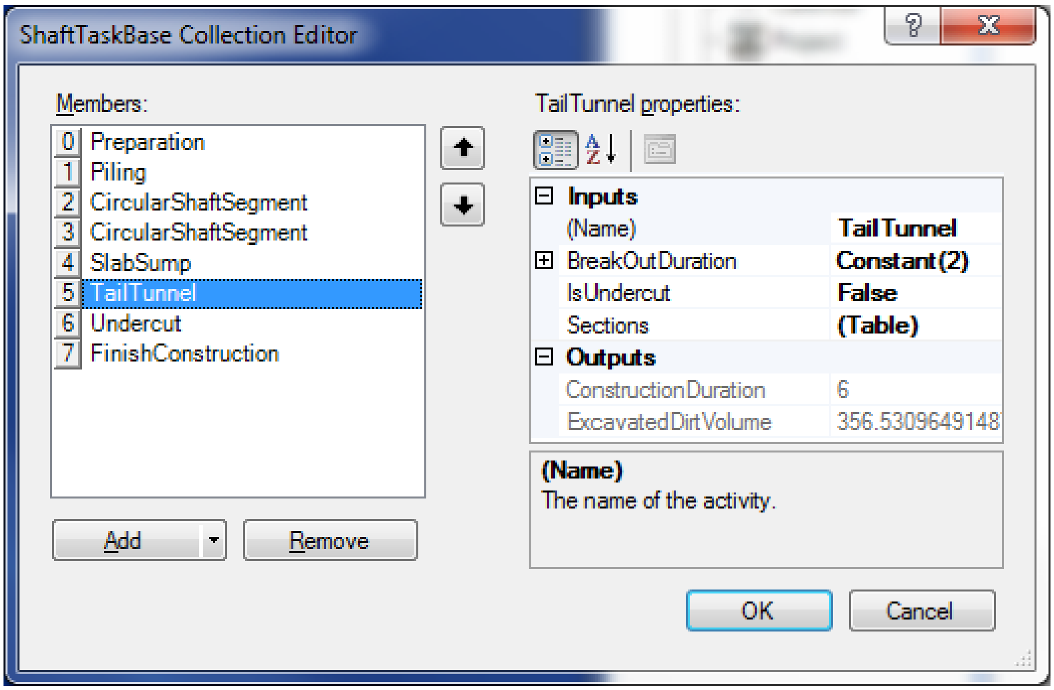


Figure 18– Tail Tunnel/Undercut Construction Tasks and Properties in the Shaft Element

Input Parameters:

Attribute Name	Type	Default value	Description
(Name)	String	TailTunnel	The Name of the Element
Break Out Duration	Distribution	Constant(2)	The duration for break out, in days
IsUndercut	Boolean		Define whether it is tail tunnel or undercut.
Sections	Table		Entries in this table determines the number of tail tunnel sections and their properties

Tail tunnel and undercut can consist of more than one section. A user can open a table and specify the properties of each section.

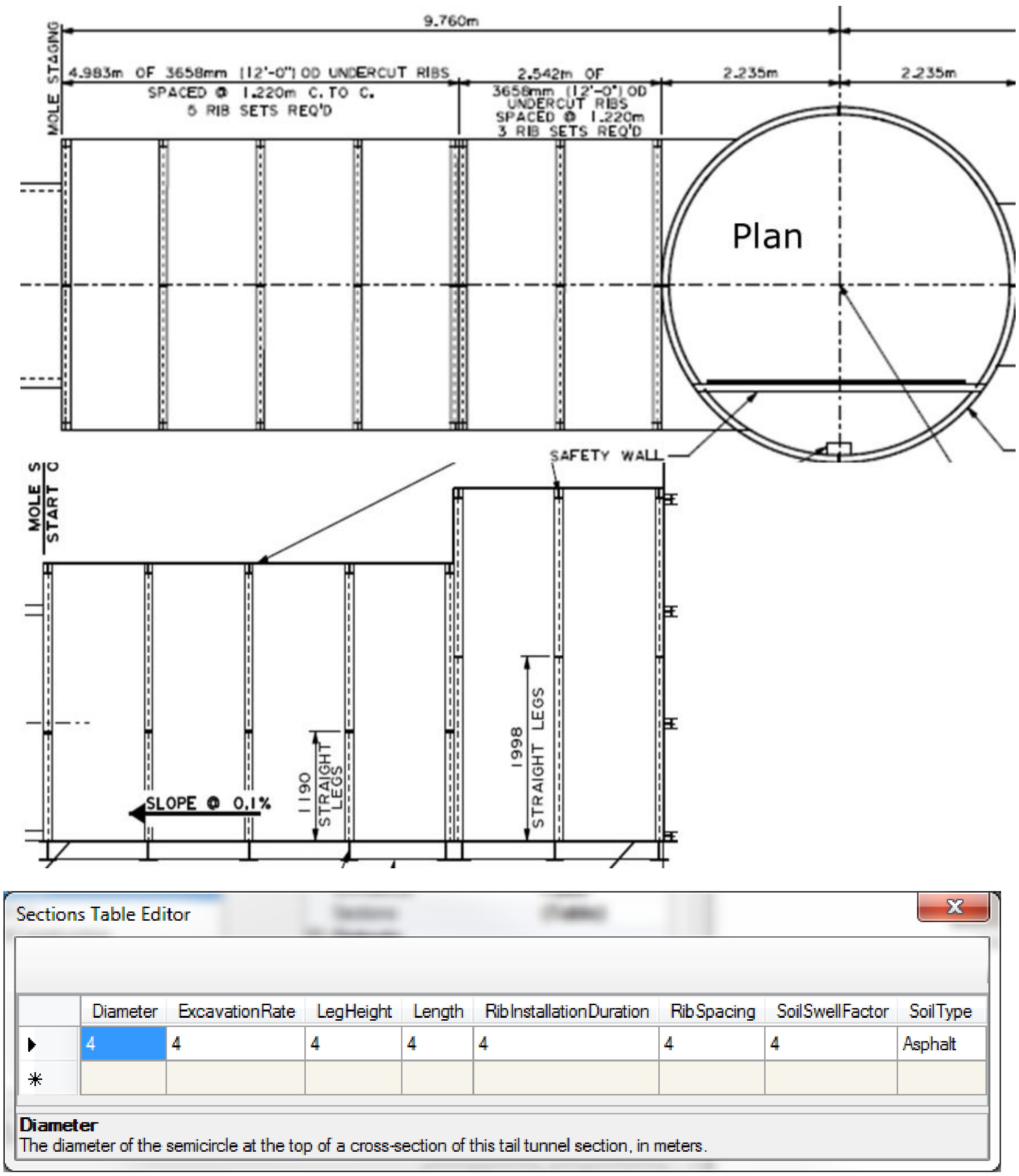


Figure 19– Tail Tunnel and Undercut Sections Specifications

Output Parameters:

Attribute Name	Type	Default value	Description
Construction Duration	Double	0	The duration of construction of the trail tunnel / undercut, in days
Excavated Dirt Volume	Double	0	The volume of dirt excavated from this tail tunnel, measured in cubic meters after excavation

Statistics: None

Finish Construction

Description:

This element marks the end of construction for this shaft; if its parent shaft is the Working Shaft, then this element also triggers the start of tunnel construction.

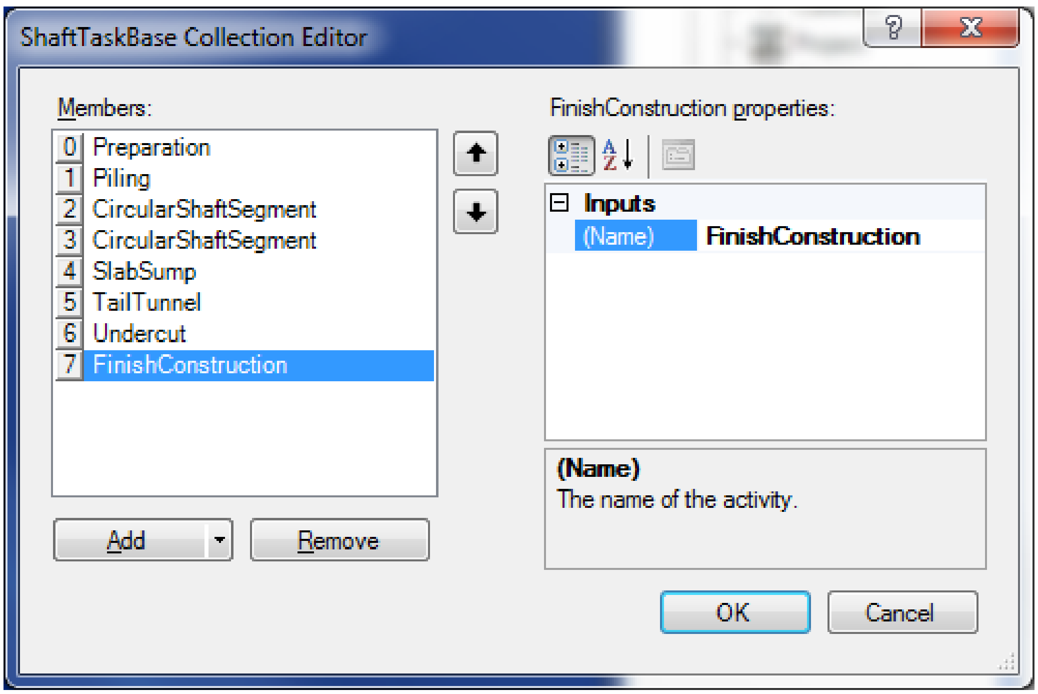


Figure 20– Finish Construction Task in the Shaft Element

Input Parameters: Attribute Name	Type	Default value	Description
(Name)	String	FinishConstruction	The Name of the Element

Output Parameters: None

Statistics: None

TUNNEL CONSTRUCTION ELEMENTS

In this category, five different elements were created to model tunnelling activities. These elements are responsible to model required tasks for tunnel excavation, as well as the dirt removal process from the tunnel face to the undercut and to the ground level, lowering down liners and materials, and many other activities inside a tunnel. A detailed description of each element is provided in the following section.

Loading/Unloading

Description:

Loading/Unloading element represents surface area. It is responsible to simulate crane activities for lowering down materials and liners and removing dirt from the tunnel to the surface.

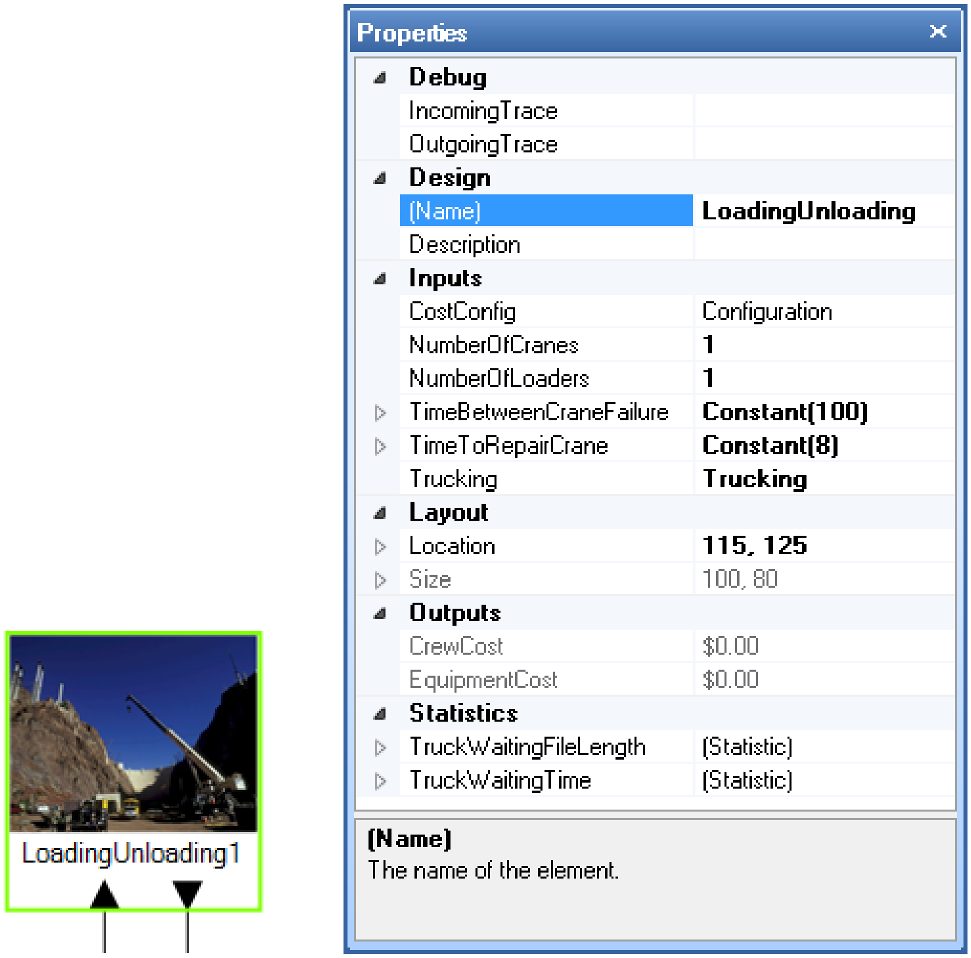


Figure 21– Shape and Properties of the Loading/Unloading Element

<u>Parameters:</u>

Design:

Attribute Name	Type	Default value	Description
(Name)	String	LoadingUnloading1	The Name of the Element
Description	String		The Description of the element

Inputs:

Attribute Name	Type	Default value	Description
Cost Config.	Custom data structure	Double	Configures cost-related properties for this element including crew and equipment hourly cost

Number of Cranes	Integer	1	The number of cranes on site for loading and unloading trains.
Number of Loaders	Integer	1	The number of loaders on site for loading dirt into trucks
Time Between Crane failure	Distribution	Constant (100)	The time form the repair of one crane to its next breakdown, in hours
Time To Repair Crane	Distribution	Constant (8)	The time it takes to repair a crane, in hours
Trucking	Enumeration	-	The corresponding trucking element used to remove dirt from this site

Output:

Attribute Name	Type	Default value	Description
Crew Cost	Double	0	The part of the cost of construction activities attributable to wages, especially for direct labour
Equipment Cost	Double	0	The part of the cost of construction activities attributable to equipment (including crane) usage

Statistics:

Attribute Name	Type	Default value	Description
Truck waiting file length	Statistic	0	Statistics for the length of the line of trucks waiting to be loaded with dirt
Truck waiting Time	Statistic	0	Statistics for the waiting time of trucks waiting to be loaded with dirt

WORKING AREA

Description:

"Working Area" element represents the area where trains travel to and from the tunnel face; this area includes undercut, tail tunnel, and shaft lower part. This area is used for TBM installation activities in TBM tunnelling and switch installation for train movement when muck carts are used for dirt removal purposes. This area can be shared between two tunnels; once the user increases the number of tunnels to two, the shape of the element will be updated automatically to show two-way tunnelling. After tunnel completion, the tunnel will be extended in the undercut area to the shaft location using hand tunnelling method. This will connect the tunnel to the manhole which will be constructed inside the shaft in the future.

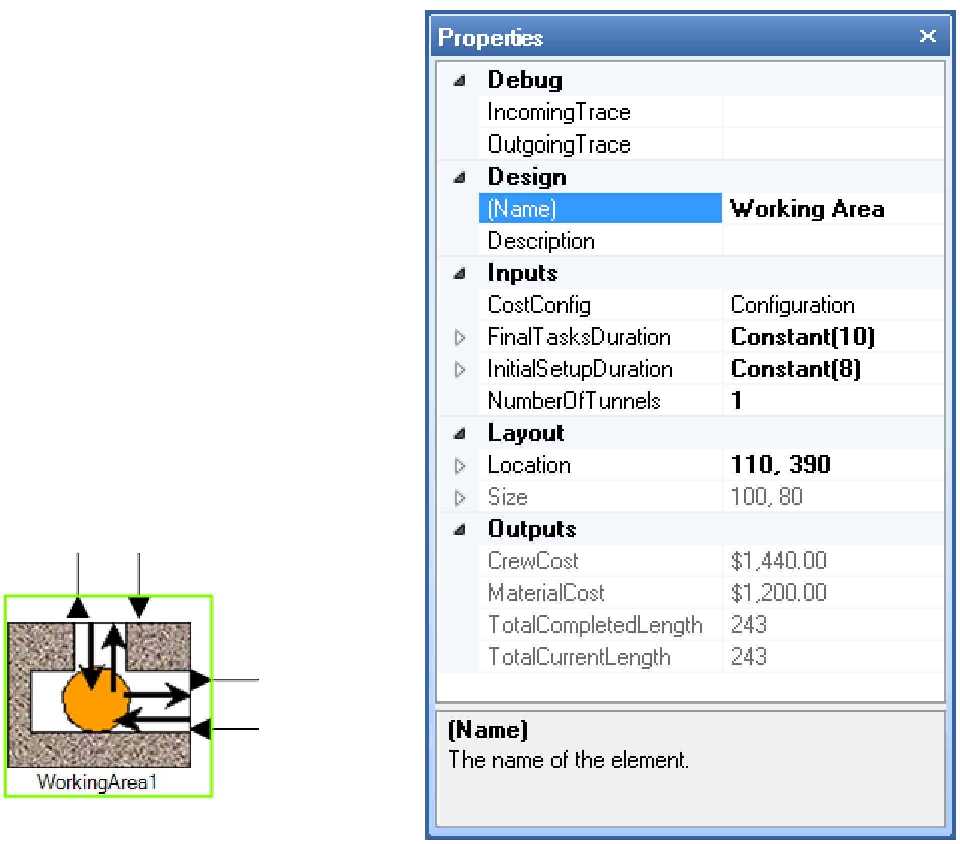


Figure 22– Shape and Properties of the Working Area Element

Properties:

Design:

Attribute Name	Type	Default value	Description
(Name)	String	WorkingArea1	The Name of the Element
Description	String		The Description of the element

Inputs:

Attribute Name	Type	Default value	Description
Cost Configuration	Custom data structure	0	The Unit cost of crew, equipment and material consumed during initial setup and final construction process.
Final Tasks Duration	Distribution	Constant (5)	Include different activities such as Manhole construction, backfilling, and removing utilities if required, in hours
Initial Setup Duration	Distribution	Constant (5)	The time that it takes to setup the working area, in hours
Number of Tunnels	Integer	1	The number of Tunnels connected to this working area

Output Parameters:

Attribute Name	Type	Default value	Description
Crew Cost	Double	0	The cost of crew during initial setup and final construction activities.
Material Cost	Double	0	The cost of material used during initial setup and final construction activities.
Total Completed Length	Double	0	Gets the length that the entire tunnel will be when completed, in meters
Total Current Length	Double	0	Gets the current chainage of the tunnel, in meters.

Statistics: None

TBM TUNNEL

Description:

TBM Tunnel element encapsulates the major excavation processes using TBM, including excavating soil section, installing liners, installing track, and performing surveying. TBM and train properties are specified in this element. Also, the user can specify multiple tunnel sections based on different soil types in the area. By adding more tunnel sections or changing the direction of the tunnel, the shape of the element will be updated accordingly.

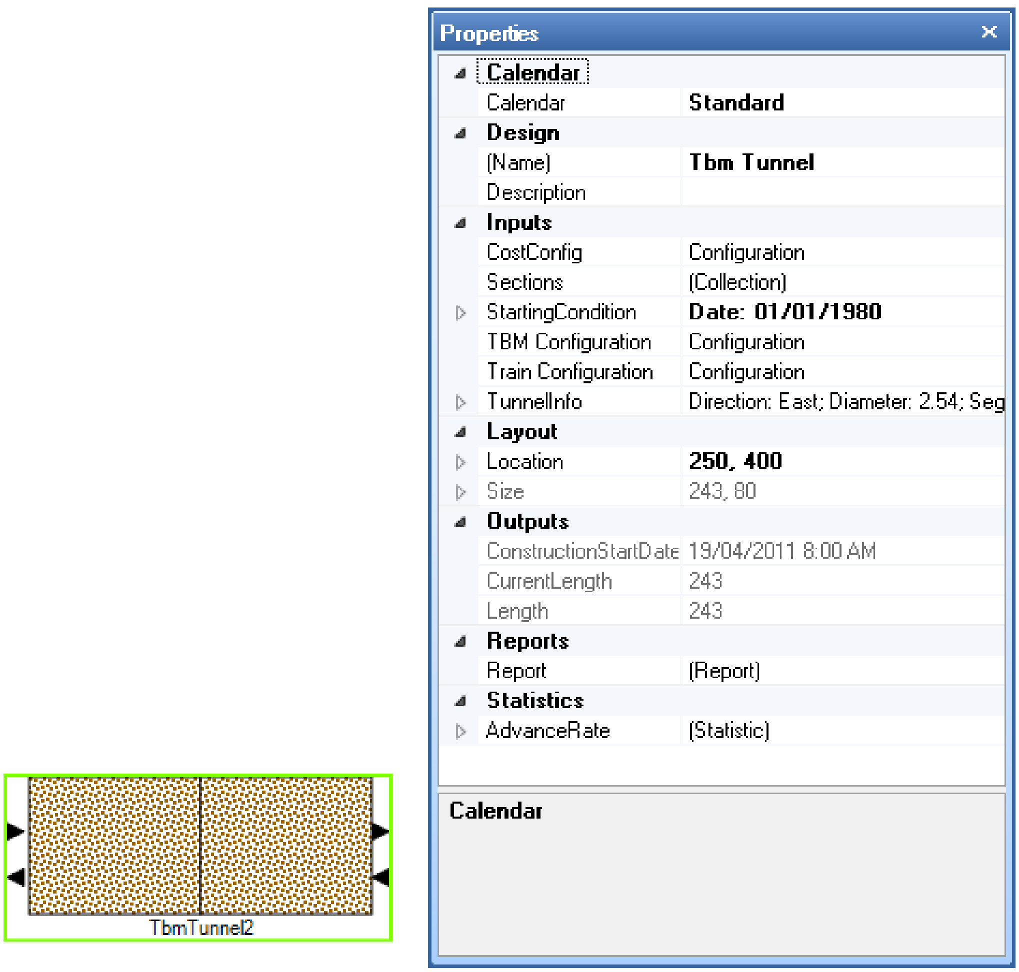


Figure 23– Shape and Properties of the TBM Tunnel Element

Properties:

Calendar:

Attribute Name	Type	Default value	Description
Calendar*	Enumeration	Standard	A calendar associated with this tunnel

Design:

Attribute Name	Type	Default value	Description
(Name)	String	TbmTunnel1	The Name of the Element
Description	String		The Description of the element

Inputs:

Attribute Name	Type	Default value	Description
Cost Config	Custom data structure	Double	Configures cost-related properties for this element including hourly cost of crew and Equipment except TBM as well as liner cost and other materials cost per meter (TBM and train cost are entered separately)
Sections	List <Tunnel Section>	-	A collection of soil layer for this tunnel. (Tunnels are divided in sections based on number of soil types along the tunnel.
Starting Condition	Custom data structure	Date	The conditions for this tunnel to start. A tunnel can start on specific date or after completing the other tunnel or after the other tunnel reaches a certain chainage.
TBM Configuration	Custom data structure	-	Configures TBM-related properties for this tunnel
Train Configuration	Custom data structure	-	Configures Train-related properties for this tunnel
Tunnel Info	Custom data structure	-	Configures a set of attributes for this tunnel such as diameter, direction, and segment length

In the following section, some input parameters will be discussed in more detail along with the screen shots.

- Cost configuration: a user can enter the cost of crew, material (anything except concrete liner), liner, and equipment (except train and TBM) in the Cost Configuration window by clicking on a small button on the right hand side of the CostConfig property (it appears when the user clicks on the property).

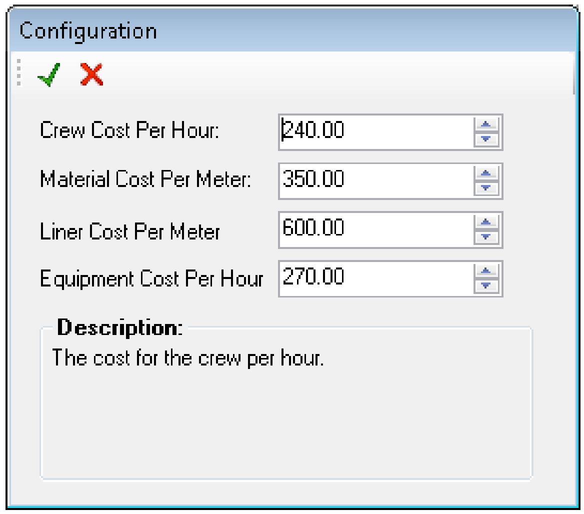


Figure 24– Cost Configuration in the TBM Tunnel Element

- Sections: the tunnel is divided into different sections based on geotechnical and geometrical information; for each section, the user can define a soil type that the section is being constructed in and enter different inputs related to that section. More than one tunnel section can be defined for one tunnel. By opening the “TBM Tunnel Section Collection Editor” window, a user can add more sections to the tunnel and update its attributes in the property grid of that section. Based on the geometry of the section, the speed of the dirt removal equipment (train or truck) can be different (in the curved section the speed of the vehicle will be lower than in the straight sections). Also, for the curve sections, the surveying intervals will be shorter, and the surveying duration can be longer. The excavation rate and lining duration will also be different based on geotechnical conditions of the section.

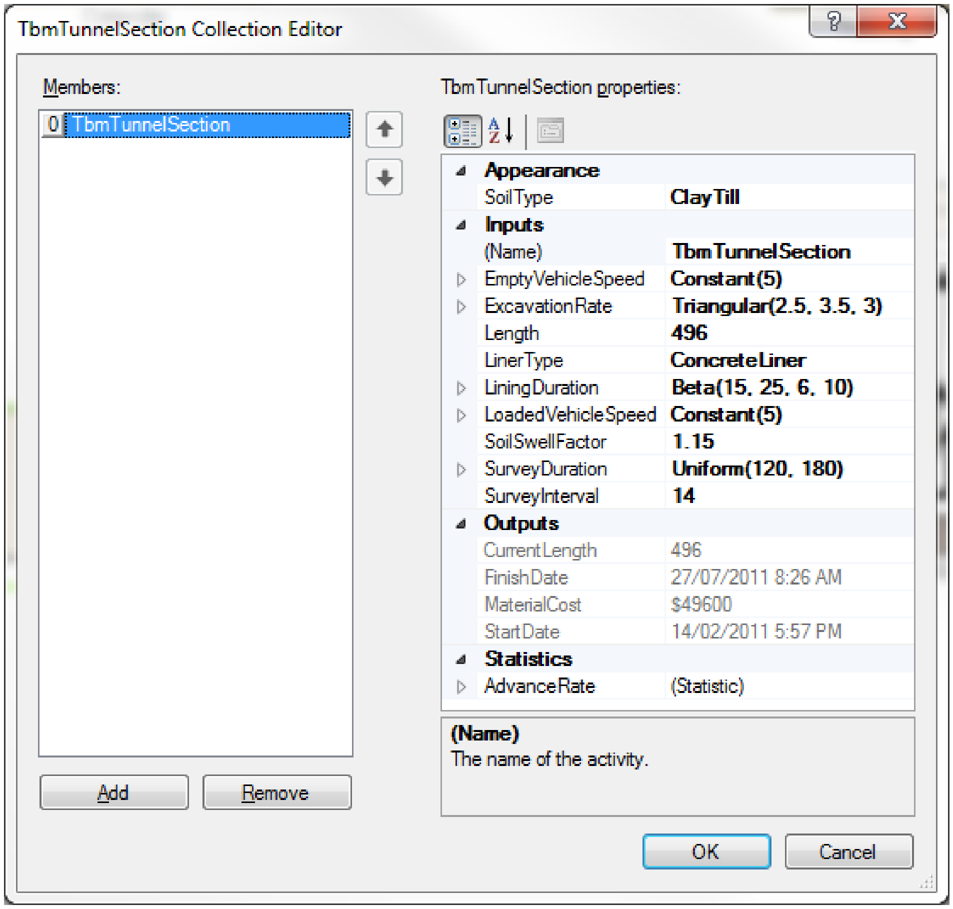


Figure 25– Tunnel Section Properties in the TBM Tunnel Element

- TBM Configuration: the following figure shows the TBM Configuration. In the Dimensions part, a user enters the general information about the TBM such as TBM length, and the conveyer and gantry length, and their section lengths (a conveyer and gantry are installed in different sections). In the Performance part, a user enters the durations for installing conveyer and gantry segments and resetting duration of TBM (in the resetting phase the TBM is fixed to the end of last installed liners to start new cycle of excavation). The Reliability part specifies the breakdown intervals, the time required to fix the TBM, and the likelihood of the need to order TBM parts (a value between 0 and 1). In the Deployment section, The TBM installation Duration and Cradle Installation

Duration are entered (The TBM is assembled on a steel cradle positioned on the slab). In the bottom section, a user enters a TBM cost per meter and the duration of driving TBM into the tunnel at the beginning of the tunnel excavation. No lining process happens until the entire TBM enters the tunnel. Then TBM starts excavating again and when the excavated length reaches the liner width, the first concrete lining sets are installed.

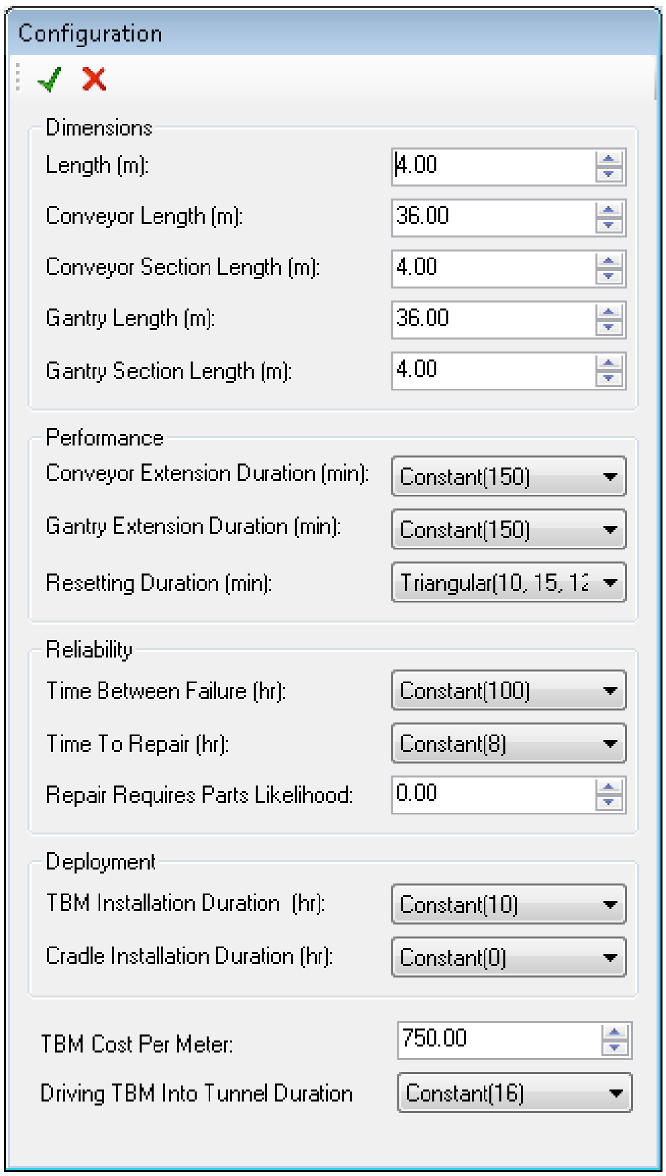


Figure 26– TBM Properties in the TBM Tunnel Element

- Train Configuration: in the train Configuration window, as shown in the following figure, the number of muck carts and their capacity, and the number of material carts are entered. Material carts are used to move liners from the undercut to the tunnel face. In the Performance section, the duration for loading and unloading dirt from carts is entered. Also, to be able to move the train closer to the conveyer, the track should be extended in specific intervals. The track extension intervals and the duration for this extension are also entered. The time between train failure and the repair time are also entered in this section. The cost per hour for using the train, and the available number of trains can be entered as well. Dispatch Interval is the time between dispatches of trains.

To use more than one train, a switch installation is needed at the undercut area to facilitate the manoeuvre of the trains in that area. The switch installation duration and the associated cost should be entered by the user. If there is more than one train, the installation length indicates that if the TBM reaches to that length, the switch installation should take place, and the second train gets dispatched. There is no switch installation if the user enters only one train.

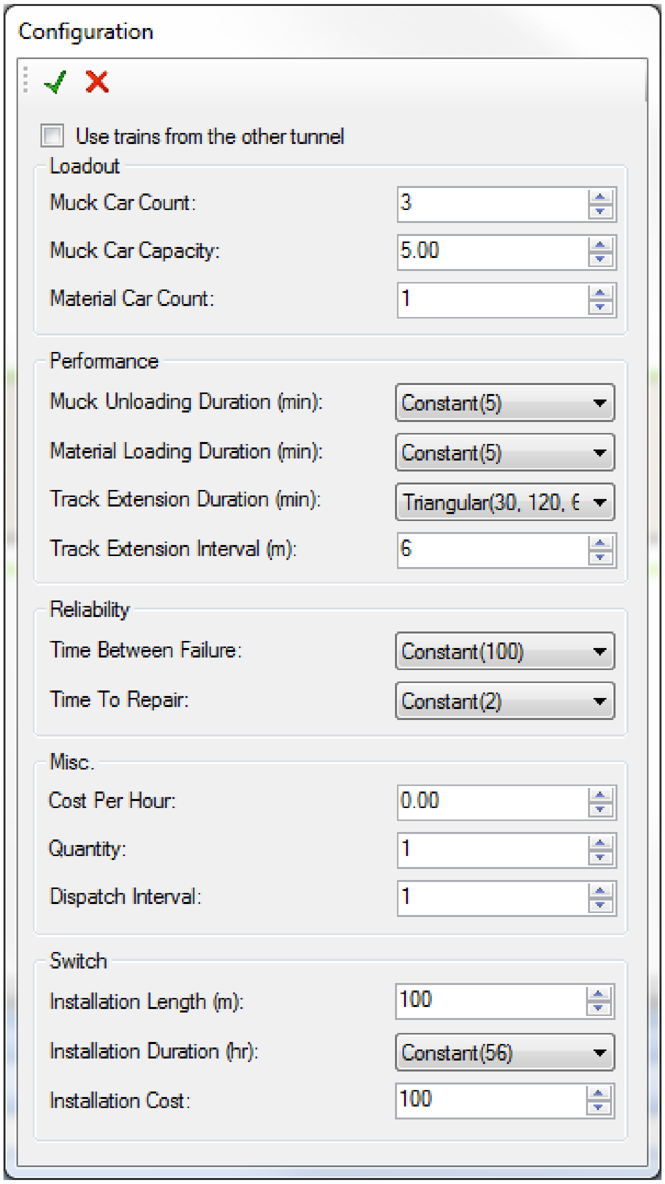
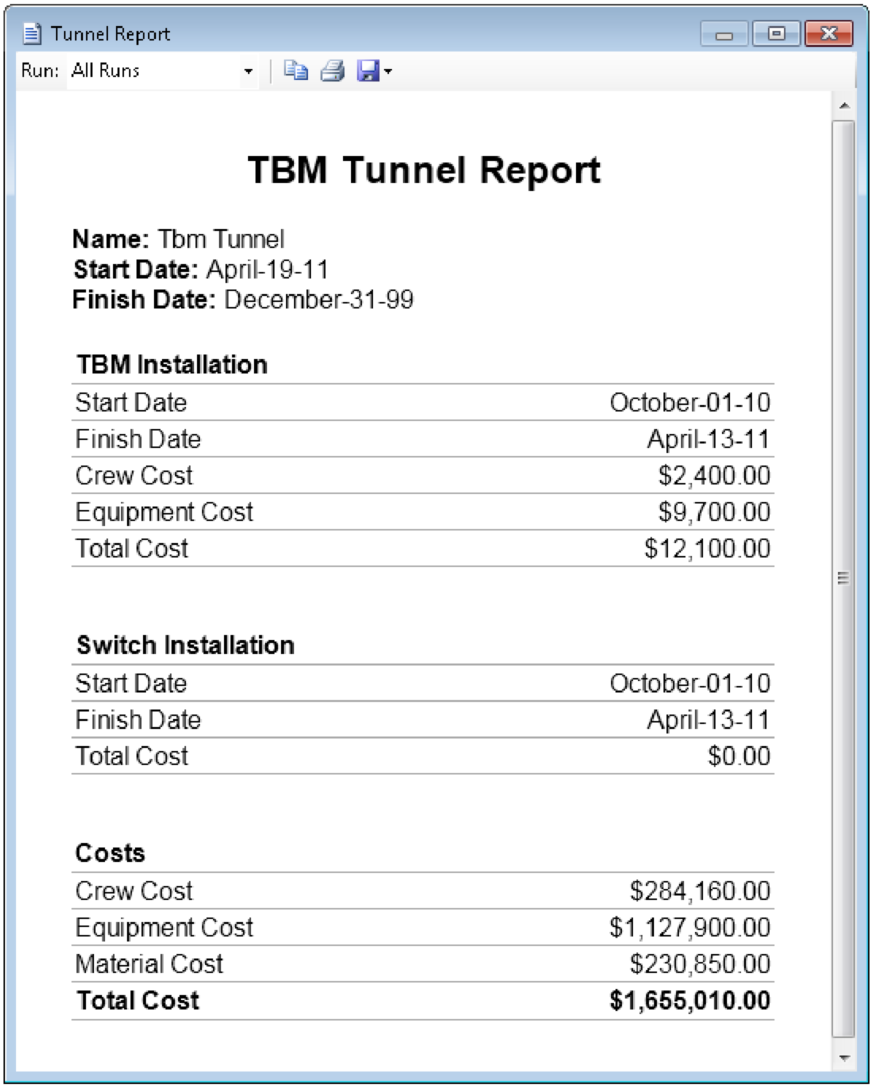


Figure 27– Dirt Removal Train Properties in the TBM Tunnel Element

Output:

Attribute Name	Type	Default value	Description
Construction Start Date	Date	0	The day on which tunnel construction begins
Current Length	Double	0	The current length of the tunnel (including unlined portion), in meters
Length	Double	200	The total length of this tunnel, in meters

Reports:



Tunnel Report

Run: All Runs

TBM Tunnel Report

Name: Tbm Tunnel
Start Date: April-19-11
Finish Date: December-31-99

TBM Installation	
Start Date	October-01-10
Finish Date	April-13-11
Crew Cost	$2,400.00
Equipment Cost	$9,700.00
Total Cost	$12,100.00

Switch Installation	
Start Date	October-01-10
Finish Date	April-13-11
Total Cost	$0.00

Costs	
Crew Cost	$284,160.00
Equipment Cost	$1,127,900.00
Material Cost	$230,850.00
Total Cost	**$1,655,010.00**

Figure 28– Tunnel Report Example in the TBM Tunnel Element

Statistics:

Attribute Name	Type	Default value	Description
Advance Rate	Statistic	0	The penetration rate of the tunnel, in meters per hour

SEQUENTIAL EXCAVATION TUNNELING ELEMENT WITH RIBS & LAGGING LINING

Description:

Two tunnel elements represent SEM tunneling method, with two different lining systems: shotcrete and ribs and lagging. These elements encapsulate the major excavation processes in SEM, including excavating soil section, installing liners, installing track (if a train is used for dirt removal), and performing surveying. A user can specify multiple tunnel sections based on different soil types and the excavation sequence in the tunnel. In this method, based on a tunnel cross section size, a train or truck can be used as dirt removal equipment.

By adding more tunnel sections or changing the direction of the tunnel, the element shape will be updated accordingly.

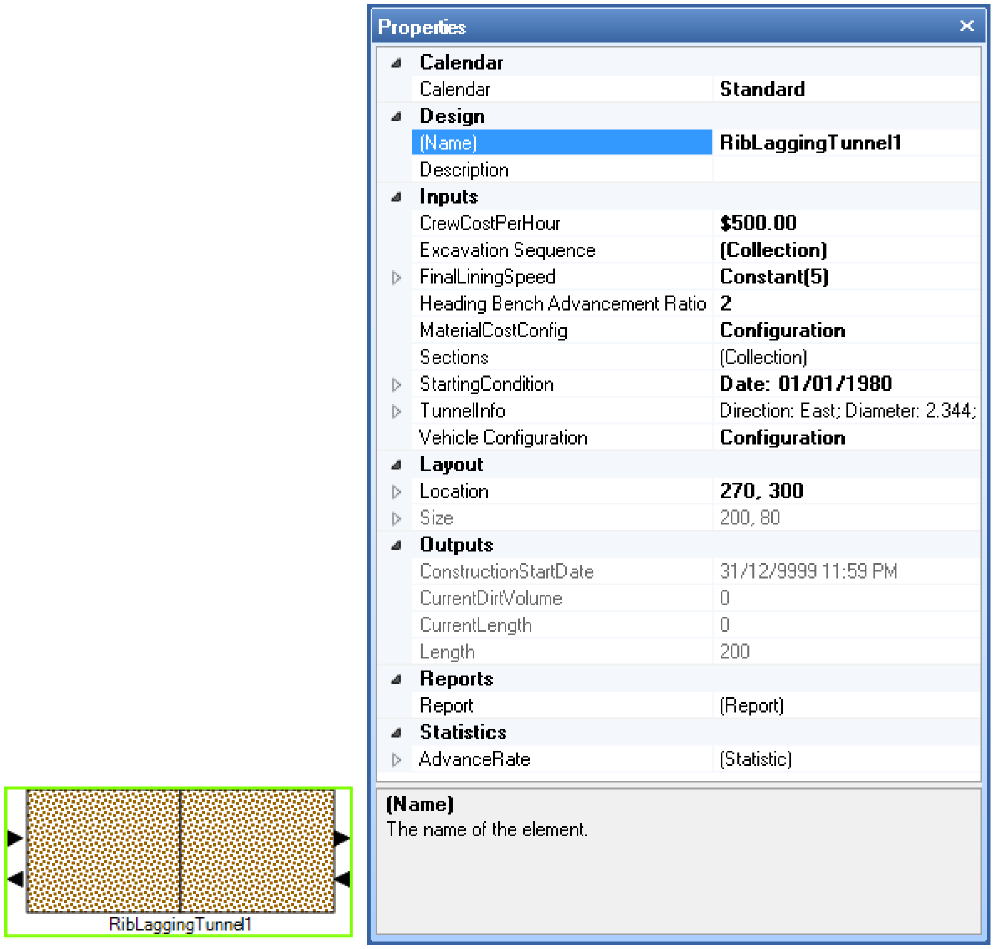


Figure 29– Shape and Properties of the Ribs and Lagging Tunnel Element

Properties:

Calendar:

Attribute Name	Type	Default value	Description
Calendar*	Enumeration	Standard	A calendar associated with this tunnel

Design:

Attribute Name	Type	Default value	Description
(Name)	String	Sequential ExcavationTunnel1	The Name of the Element
Description	String		The Description of the element

Inputs:

Attribute Name	Type	Default value	Description
Crew cost per hour	double	500	The cost of crew
Excavation Sequence	List <Tunnel cross section>	-	A collection of cross section segments defining the sequence of excavation in one section
Final lining speed	Distribution	Constant (5)	The speed of the final lining in m/hr
Heading bench advancement	2	0	The number of heading section that is equal to one bench section
Material cost config	Custom data structure	Double	Configures material cost-related properties for this element including materials for additional supports
Sections	List <Tunnel Section>	-	A collection of soil layer for this tunnel
Starting Condition	Custom data structure	Date	The conditions for this tunnel to start. A tunnel can start on specific date or after completing the other tunnel or after the other tunnel reaches a certain chainage
Tunnel Info	Custom data structure		Configures a set of attributes for this tunnel such as diameter, direction, and segment length
Vehicle Configuration	Custom data structure	-	Configures dirt removal vehicle-related properties for this tunnel

In this section, some of the inputs will be discussed in detail:

Excavation Sequence

In this input parameter, a user should specify the sequence of tunnel section excavation and its related tasks. For each small part of a section, a user enters an area, and a list of the excavation and lining tasks.

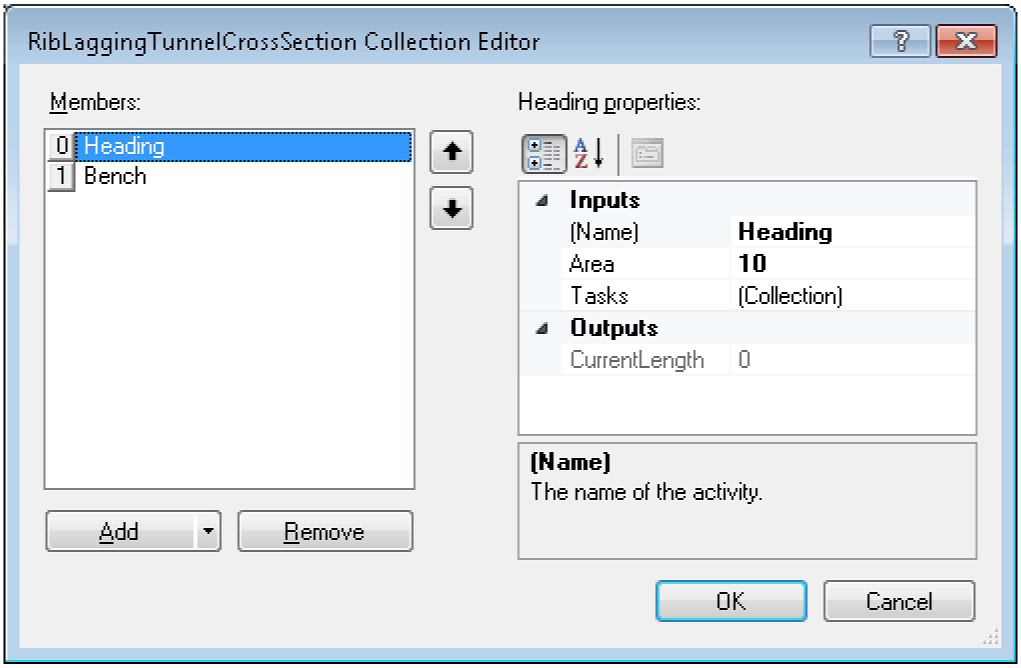


Figure 30– Ribs and Lagging Tunnel Cross Section Properties

In the tasks collection, a user can add tasks from a predefined list. For instance, for ribs and lagging support system, the following tasks are defined:

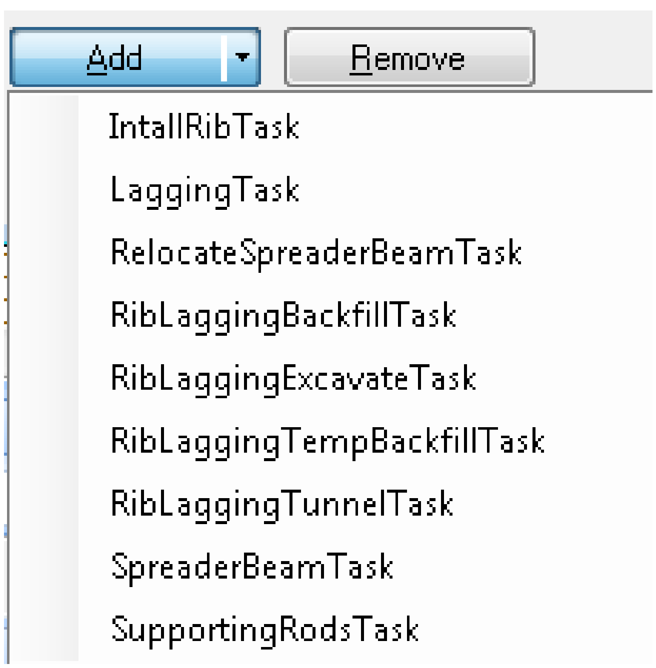


Figure 31– List of Tasks in the Ribs and Lagging Tunnel Element

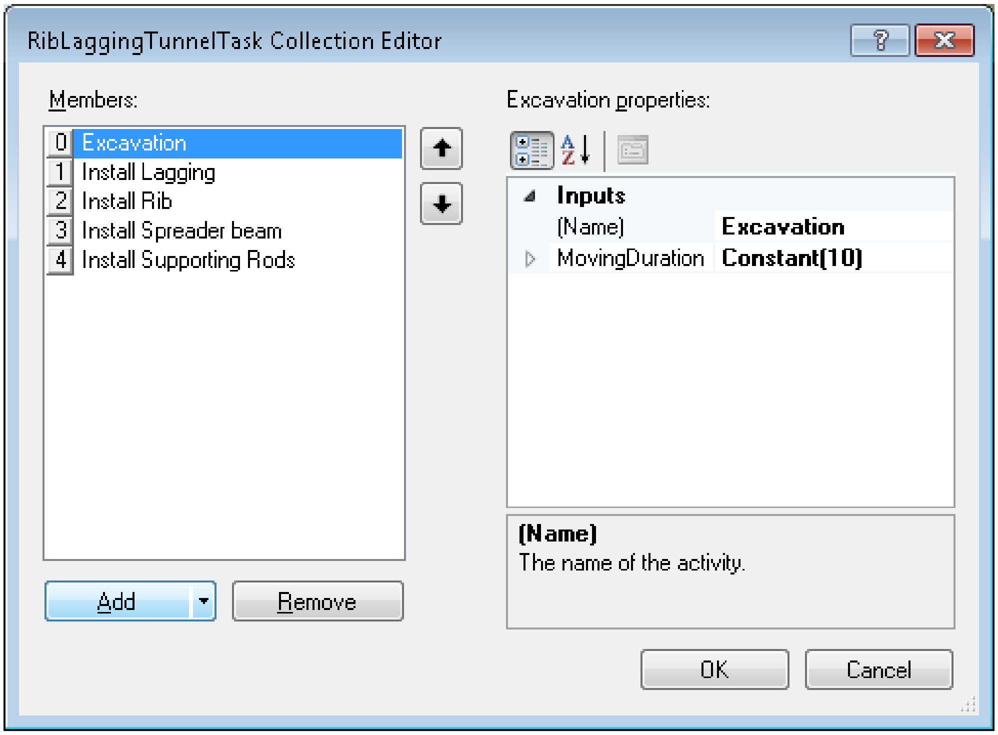


Figure 32– Task Properties in the Ribs and Lagging Tunnel Element

For each task, there is a "Moving Duration" input that shows duration to move equipment and materials and make sections ready for the next task. The user can enter zero if there is no gap between each task.

Material Cost Configuration

In this section, a user enters a unit cost for the different materials that are usually used during the excavation and lining of this tunneling system.

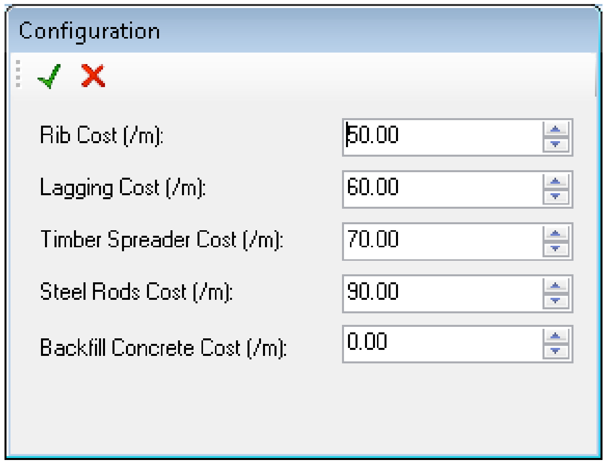


Figure 33– Cost Configuration in the Ribs and Lagging Tunnel Element

Sections

This attribute defines different sections along the tunnel length.

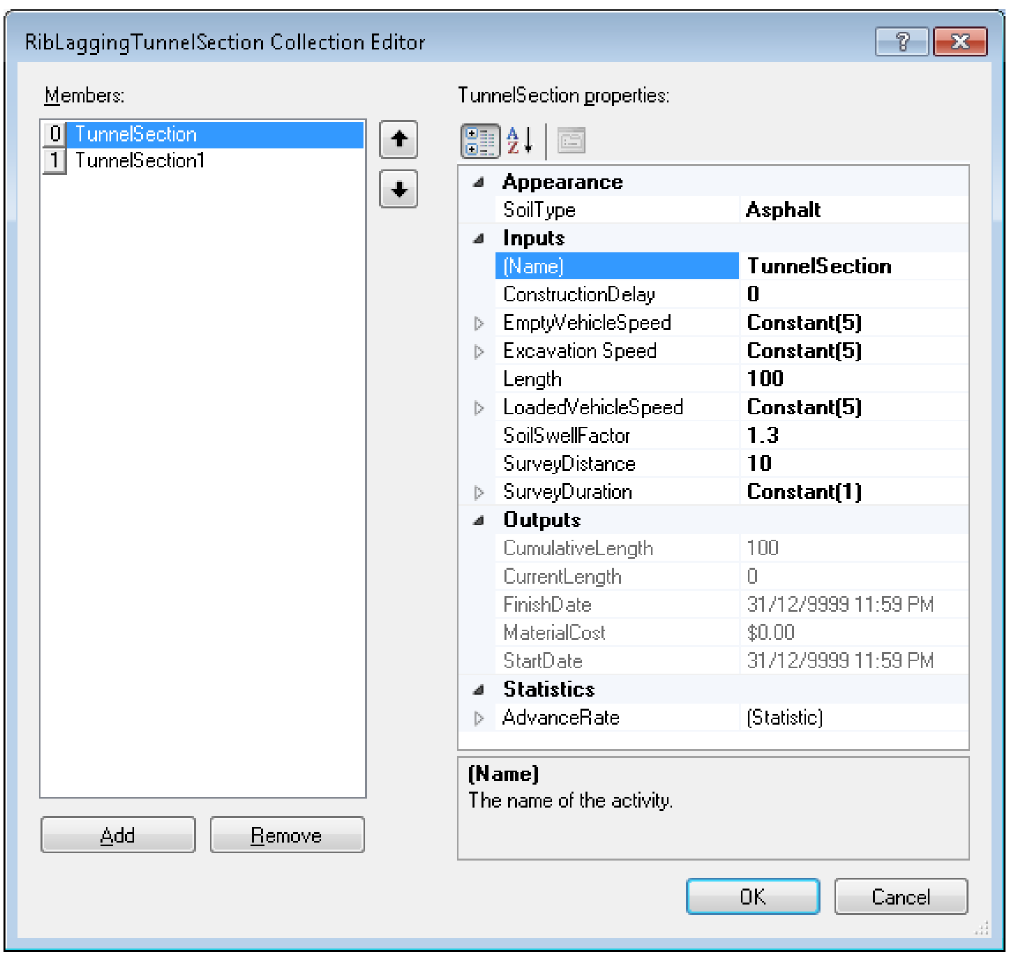


Figure 34– Tunnel Section Properties in the Ribs and Lagging Tunnel Element

Vehicle Configuration

For excavation dirt removal, either a train or a truck can be used, based on the tunnel size. As it is shown in the following figure, a user can select what kind of vehicle is used for dirt removal inside the tunnel.

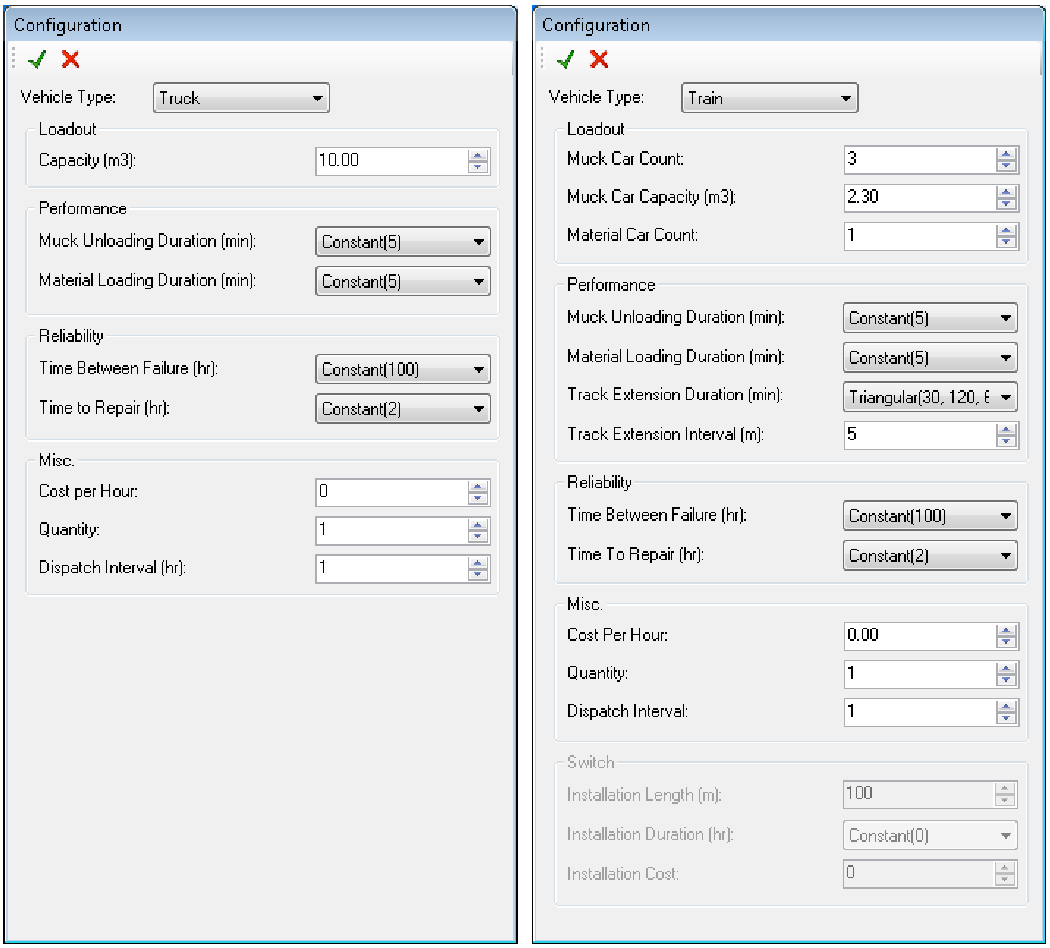


Figure 35–Properties for the Dirt Removal Vehicles in the Ribs and Lagging Tunnel Element

Outputs:

Attribute Name	Type	Default value	Description
Construction Start Date	Date		The day on which tunnel construction begins
Current Length	Double	0	The current length of the tunnel (including unlined portion), in meters

Current Dirt Volume	Double	0	The amount of dirt currently accumulated at the tunnel face, in m3
Length	Double	250	The length that the entire tunnel will be when completed, in meters

Report:

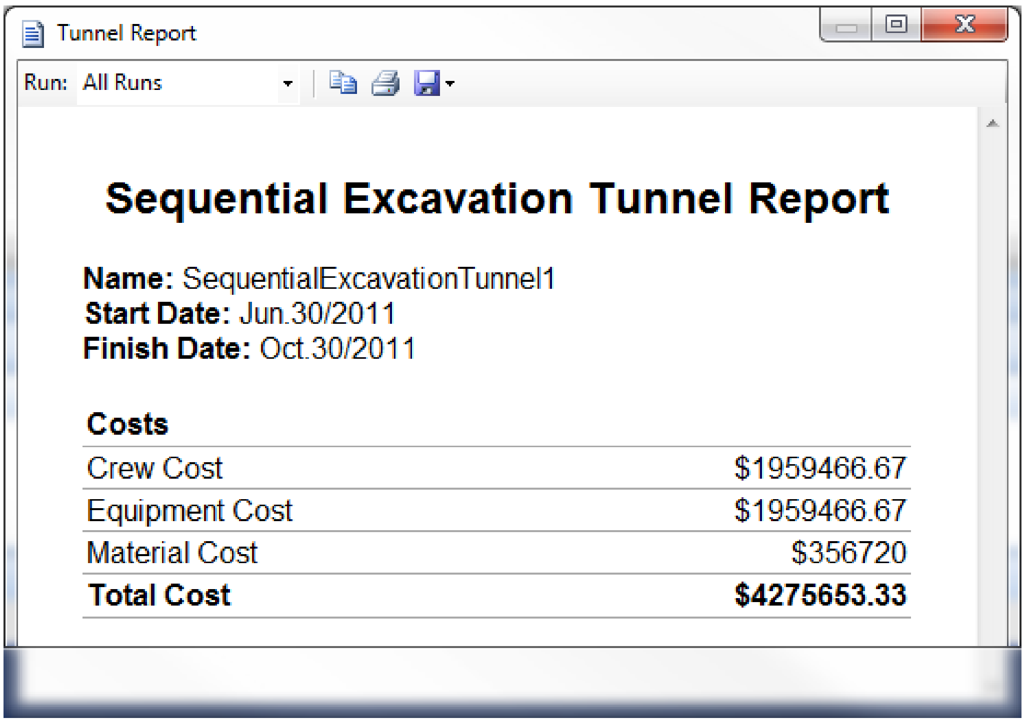


Figure 36–Tunnel Report in the Ribs and Lagging Tunnel Element

Statistics:

Attribute Name	Type	Default value	Description
Advance Rate	Statistic	0	The penetration rate of the tunnel, in meters per hour

SEQUENTIAL EXCAVATION TUNNELING ELEMENT WITH SHOTCRETE LINING

Description:

Two tunnel elements represent SEM tunneling method with two different lining systems: shotcrete and ribs and lagging. These elements encapsulate the major excavation processes in SEM, including excavating soil section, installing liners, installing track (if a train is used for dirt removal), and performing surveying. A user can specify multiple tunnel sections based on different soil types and the excavation sequence in the tunnel. In this method, based on a tunnel cross section size, a train or a truck can be used as dirt removal equipment.

By adding more tunnel sections or changing the direction of the tunnel, the element shape will be updated accordingly.

Tunnel with Shotcrete Lining

This element is similar to ribs and lagging tunnel element described in the previous section. The only difference is the list of excavation and lining tasks in the “Excavation Sequence” attribute, and material cost in the “Material Cost Config.” There are also additional inputs in the “Sections” attribute.

Excavation Sequence

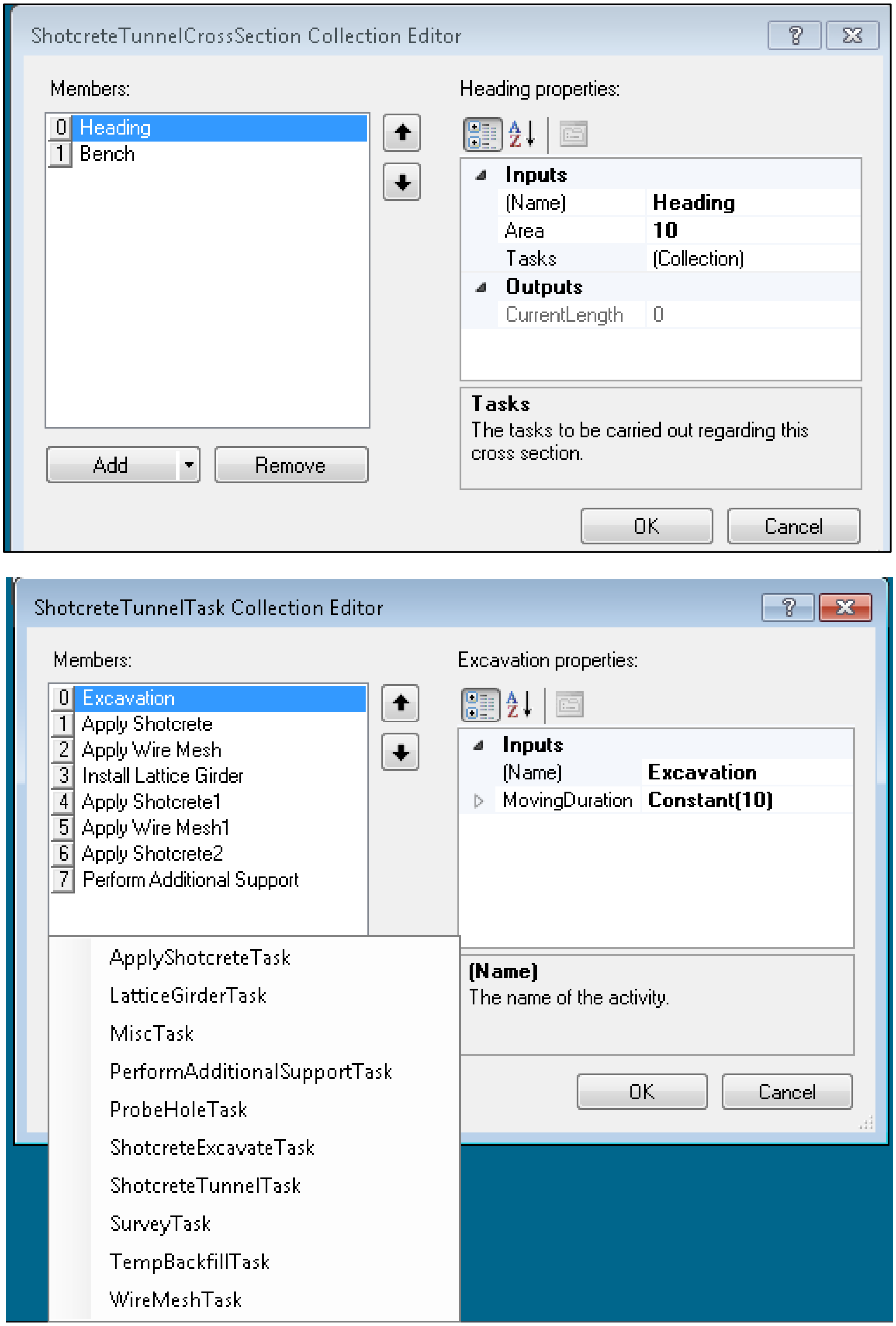


Figure 37–List of Tasks for the Tunnel Cross Sections in the Shotcrete Tunnel Element

For each task, there is a “Moving Duration” input that shows duration to move equipment and materials and make sections ready for the next task. A user can enter zero if there is no gap between each task.

Material Cost Configuration

Since different materials are used in the shotcrete system, a list of materials in the cost configuration attribute is different than that for the ribs and lagging method.

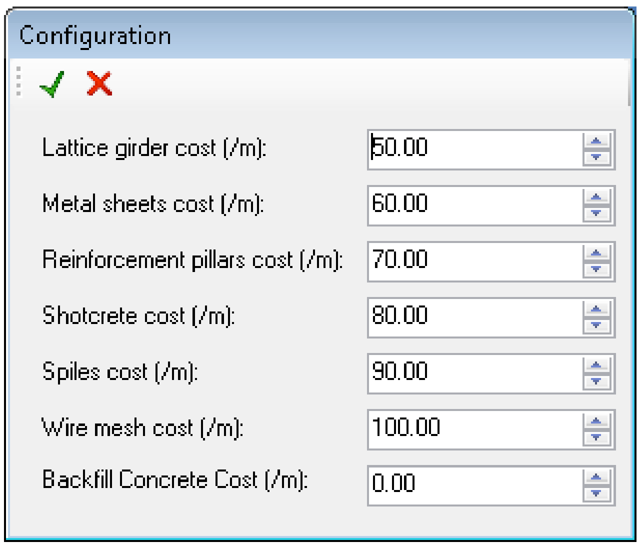


Figure 38–Cost Configuration in the Shotcrete Tunnel Element

Sections

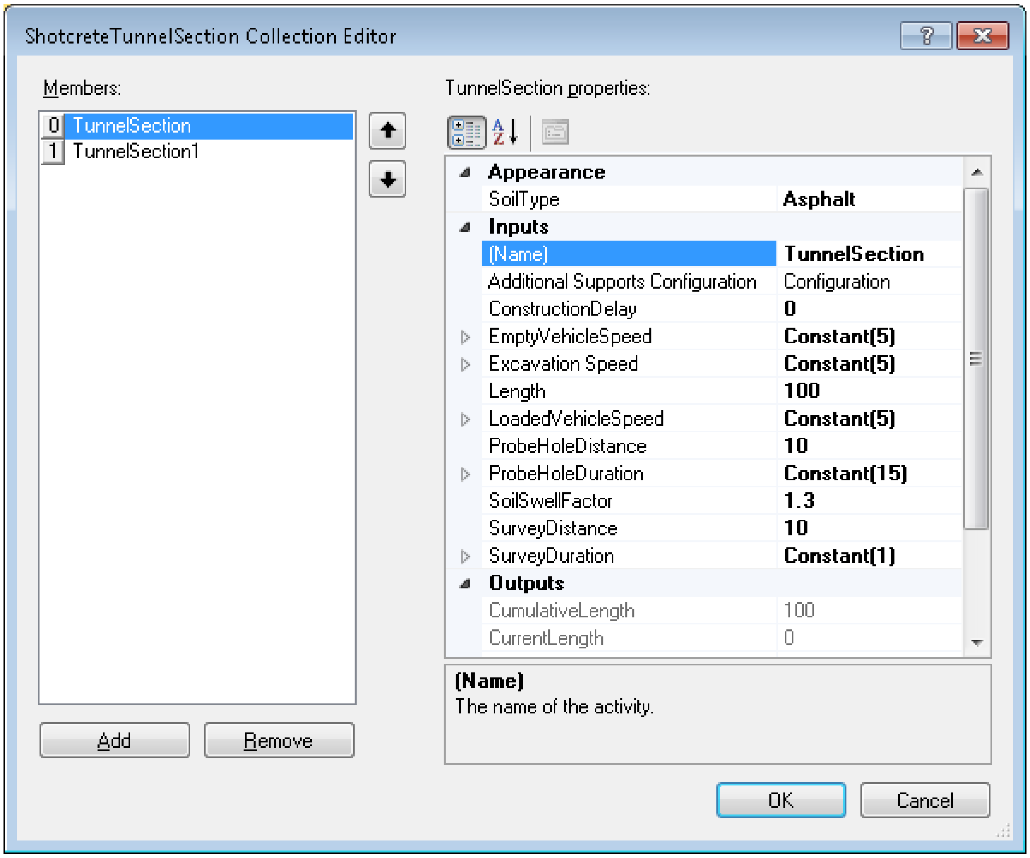


Figure 39–Tunnel Section Properties in the Shotcrete Tunnel Element

In shotcrete lining method, based on geotechnical conditions, some sections require additional supports. So for each section, a user can enter the additional support information from common materials.

Also, in the site, to investigate soil condition, the contractor digs probe holes a certain distance along a tunnel. Here, a user can also specify that distance and the excavation duration for each set of these probe holes.

The other attributes are the same as for the ribs and lagging method.

FINISH CONSTRUCTION

Description:

This element is responsible for the final activities in the construction process. In TBM tunneling, it is responsible for modeling TBM removal and hand tunneling in the undercut area, and in the SEM method, it is used to enter the duration of final tasks in the tunnel construction.

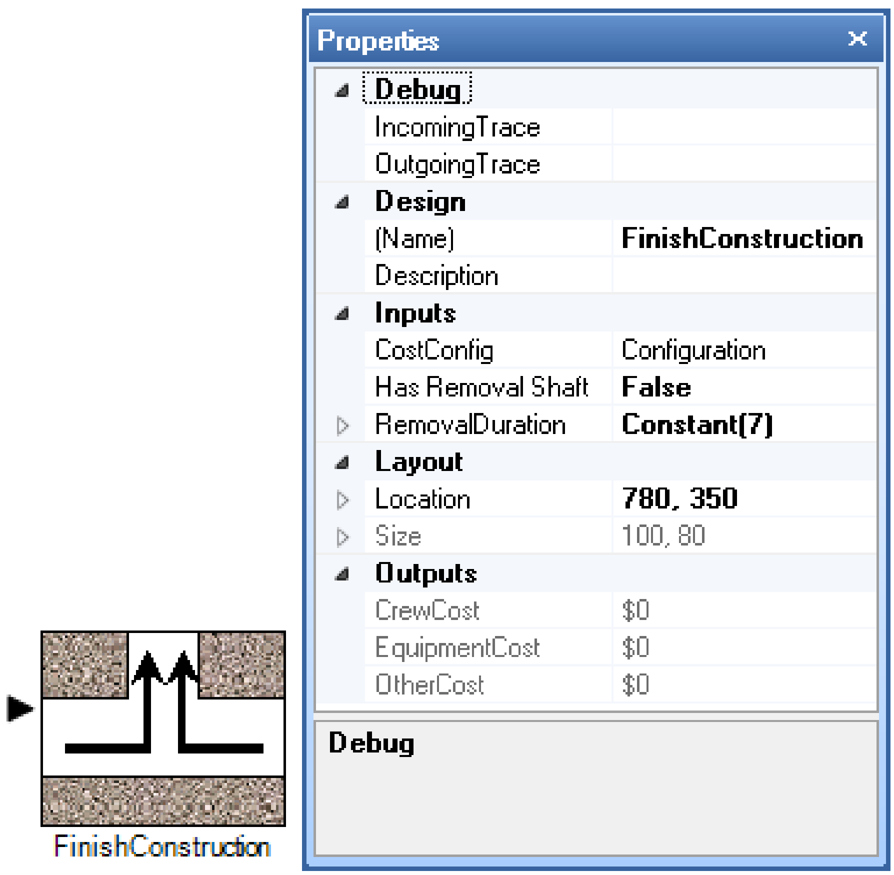


Figure 40–Shape and Properties of the Finish Construction Element

Properties:

Design:

Attribute Name	Type	Default value	Description
(Name)	String	Finish Construction	The Name of the Element
Description	String		The Description of the element

Inputs:

Attribute Name	Type	Default value	Description
Cost Config	Custom data structure	Double	Configures cost-related properties for this element including hourly cost of crew, material and equipment.
Final Tasks Duration	Distribution	Constant (14)	The time that takes to remove TBM in TBM tunneling and do final tasks in other tunnelling method.
Has removal shaft	Boolean	False	Indicates whether the process has removal shaft, if enter yes, it will create arrows to connect to removal shaft element.

Outputs:

Attribute Name	Type	Default value	Description
Crew Cost	Double	0	The part of total cost of construction activities attributable to wages for workers inside this the tunnel
Equipment Cost	Double	0	The cost of Equipment associated with the process
Material Cost	Double	0	The cost of material used to build manhole and fill shaft in TBM tunneling or do final works at the end of the tunnel for other methods.
Other Cost	Double	0	The total miscellaneous costs associated with the process
TBM Removal Start Date	Date		The date on which removal of the TBM began.

SUPPORT ELEMENTS

PROJECT

Description:

This is the parent element that will hold all other elements related to the tunnel template. The user can have more tunnel construction projects in one simulation model. The supply element is the only element that can be placed outside of the project element.

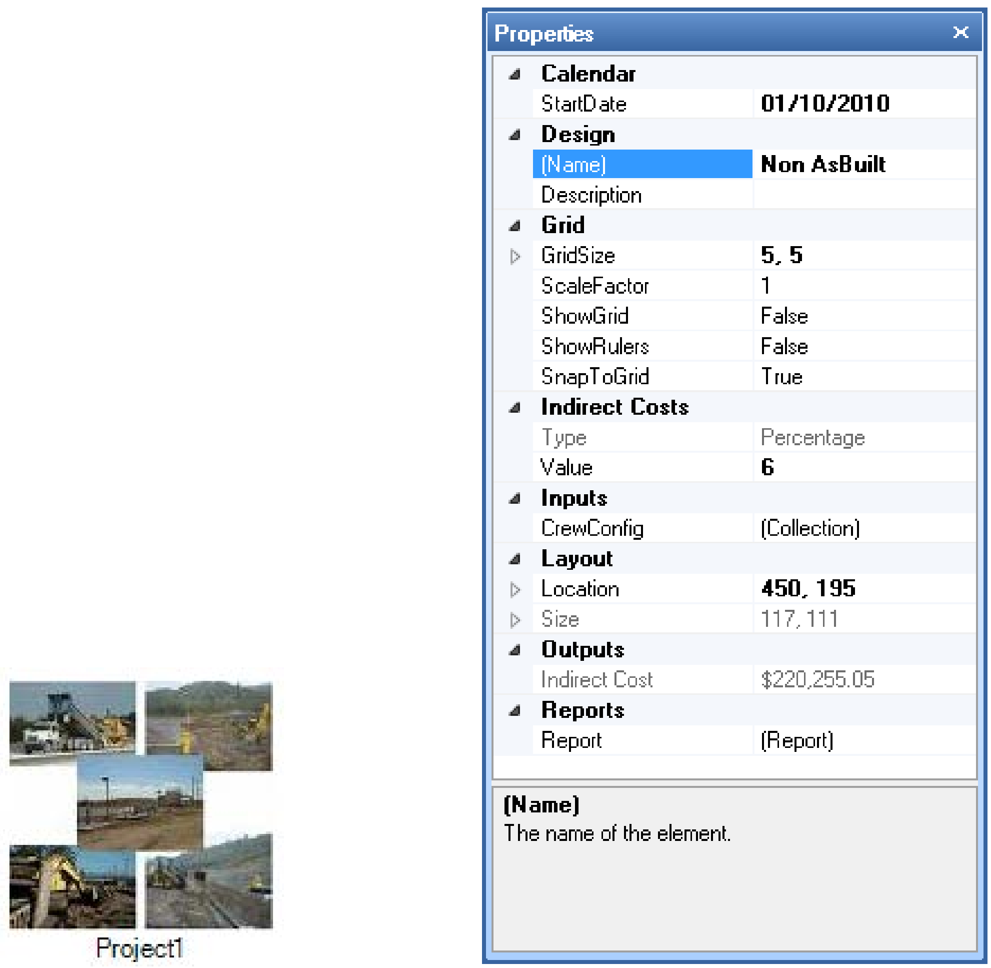


Figure 41–Shape and Properties of the Project Element

Properties:

Calendar:

Attribute Name	Type	Default value	Description
Start Date	DateTime		The start time of the project.

Design:

Attribute Name	Type	Default value	Description
(Name)	String	Project1	The Name of the Element
Description	String		The Description of the element

Indirect Costs

Attribute Name	Type	Default value	Description
Type	String	Percentage	Indicates how indirect cost is calculated
Value	Double		The percent that user enters to calculate indirect cost. Indirect cost will be calculated by multiplying the direct cost by this value.

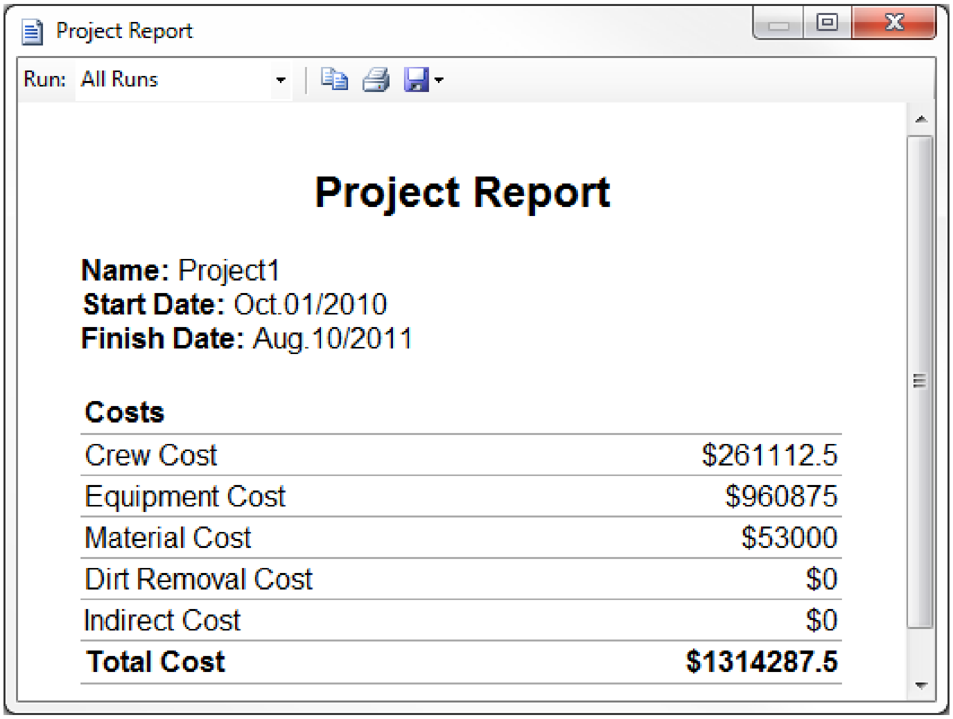
Inputs:

Attribute Name	Type	Default value	Description
Crew Config	Collection		A collection of crew in this project

Outputs:

Attribute Name	Type	Default value	Description
Indirect Cost	Double		Indicates the indirect cost of the project

Report:


Project Report

Run: All Runs

Project Report

Name: Project1
Start Date: Oct.01/2010
Finish Date: Aug.10/2011

Costs	
Crew Cost	$261112.5
Equipment Cost	$960875
Material Cost	$53000
Dirt Removal Cost	$0
Indirect Cost	$0
Total Cost	**$1314287.5**

Figure 42– Project Element Report

Attribute Name	Description
Crew Cost	The part of total cost of construction activities attributable to wages for workers in the project
Equipment Cost	The total cost of Equipment in the project
Material Cost	The total material cost for the project
Dirt Removal Cost	The cost associated to the dirt removal activities
Indirect Cost	The total indirect cost calculated based on the project duration and the hourly rate of indirect cost
Total Cost	The sum of direct and indirect cost of the project

SUPPLY

Description:

This element is used as a supplier of tunnelling materials, such as liner and rib and lagging; it is also used to supply the TBM parts, if needed. Supply element can supply material for more than one project in one simulation model. A user should add it to the project level.

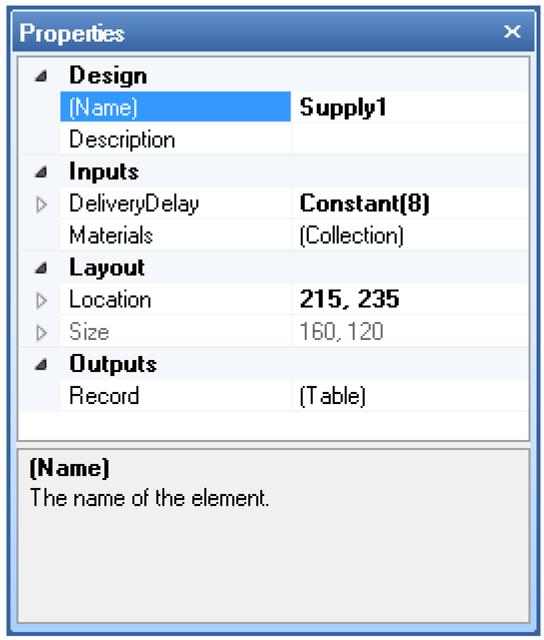


Figure 43– Shape and Properties of the Supply Element

Properties:

Design:

Attribute Name	Type	Default value	Description
(Name)	String	Supply1	The Name of the Element
Description	String		The Description of the element

Inputs:

Attribute Name	Type	Default value	Description
Delivery Delay	Distribution	Constant(8)	The time between the material order and the corresponding material delivery, in hours
Materials	Enumeration		A collection of materials that needs to be ordered and supplied during the project..

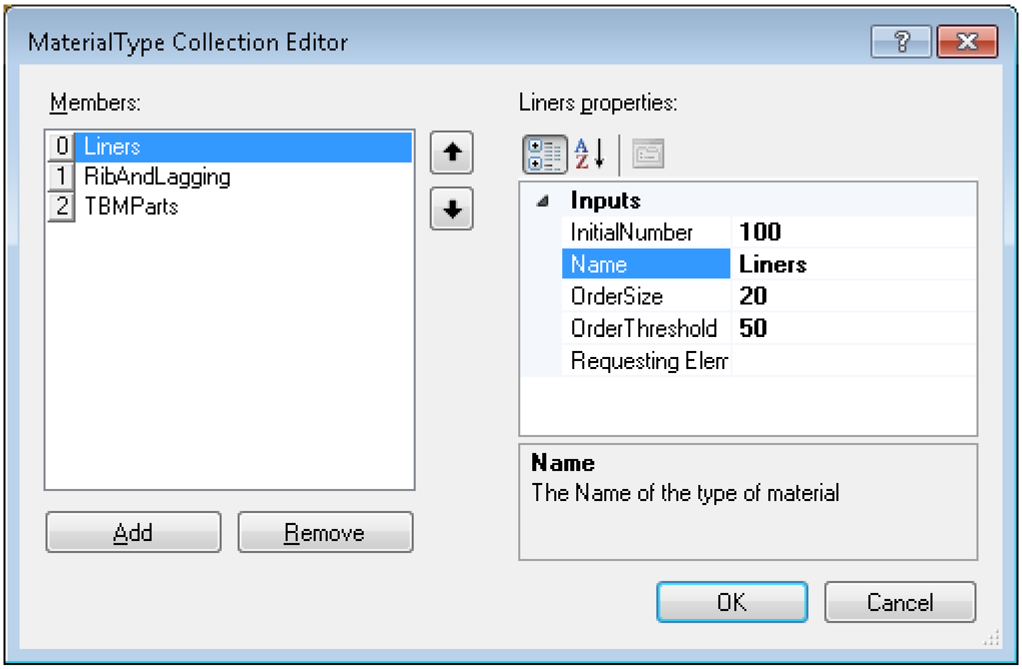


Figure 44– Material Supply Properties in the Supply Element

For each material, a user should enter the initial number of that material in the site, order size, and the threshold for ordering a new batch for the project (if available number of material gets less than this number, a new request will be created for that material).

Output parameters:

Attribute Name	Type	Default value	Description
Record	Table		A record of all material requests for this supplier

As an output value, a record of material requests and their delivery time are listed in a table as below:

Supply Table Editor

	Quantity Requested	Request Date	Delivery Date	Comment
▸	100	07/05/2012	07/05/2012	Liner order placed by LoadingUnloading1
	100	29/05/2012	29/05/2012	Liner order placed by LoadingUnloading1
	100	18/06/2012	18/06/2012	Liner order placed by LoadingUnloading1
	100	27/06/2012	27/06/2012	Liner order placed by LoadingUnloading1

Quantity
The quantity of supplies delivered.

Figure 45– Supplied Material Report in the Supply Element

Statistics: None

TRUCKING

Description:

This element represents the earth moving operation of the tunnel, which takes excavated dirt from the tunnel site to a dumping area. This element is a stand-alone element which should be added as a child element of the project element. It communicates with the corresponding Loading/Unloading element to move the excavated dirt from the shaft location to the dumping site to allow the tunnelling operation to continue. If the amount of dirt reaches the maximum volume of the dirt that can be sitting in the site, the operation will stop until the dirt is transferred to a dump site.

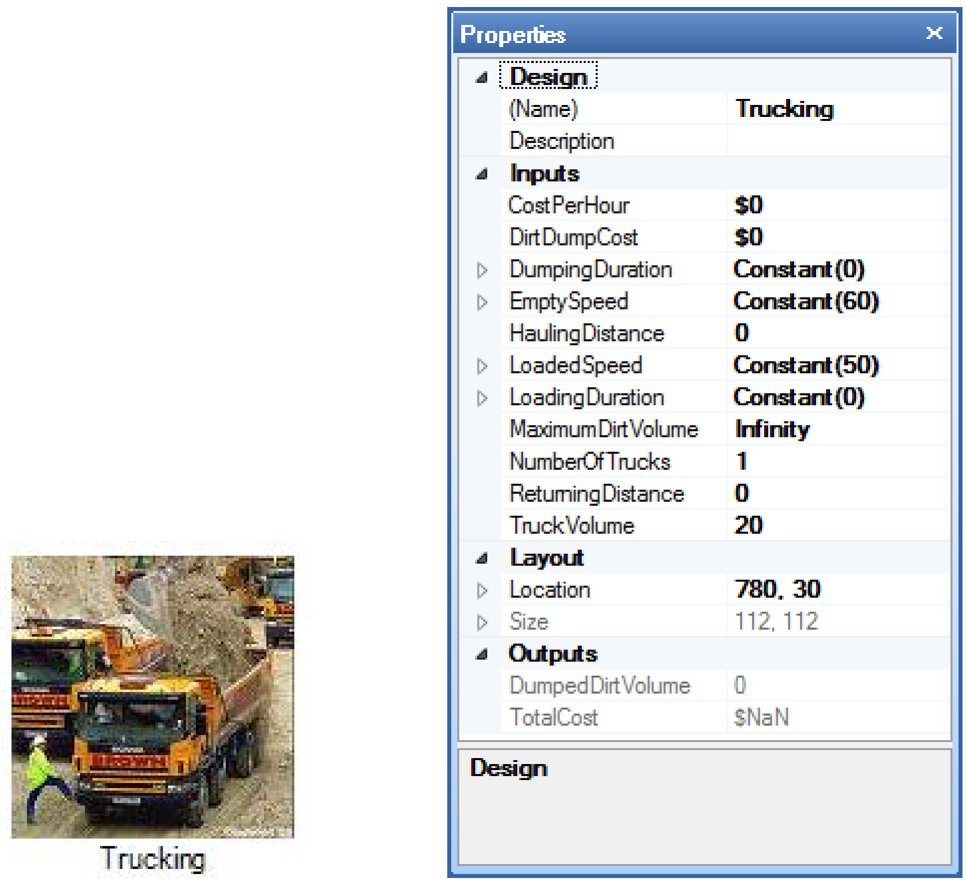


Figure 46– Shape and Properties of the Trucking Element

Properties:

Design:

Attribute Name	Type	Default value	Description
(Name)	String	Trucking1	The Name of the Element
Description	String		The Description of the element

Inputs:

Attribute Name	Type	Default value	Description
Cost per hour	Double	0	The cost per hour of removing dirt
Dirt Dump Cost	Double	0	The cost of dumping one cubic meter of dirt
Dumping Duration	Distribution	Constant(0)	The time that it takes one truck to dump its dirt in minutes
Empty Speed	Distribution	Constant(60)	The average speed of an empty truck in km/h
Hauling Distance	Double	0	The distance from the loading site to the dumping site, in km
Loaded Speed	Distribution	Constant(50)	The average speed of a loaded truck in km/h
Loading Duration	Distribution	Constant(0)	The time that it takes to load one truck with dirt using one loader, in minutes
Maximum Dirt Volume		Infinity	The maximum volume of the dirt that can be sitting in as a pile in the sites utilizing this trucking element, in cubic meters
Number of Trucks	Integer	1	The number of trucks used to carry dirt from the project site.
Returning Distance	Double	0	The distance from the dumping site to the loading site. In Km
Truck volume	Double	20	The volume of single truck in cubic meter

Output parameters:

Attribute Name	Type	Default value	Description
Dumped Dirt Volume	Double	0	The volume of dirt dumped at the dump site, in cubic meter
Total Cost	Double	0	The total cost of dirt removal activity

Statistics: None

Details of Tunnel template Programming

There are couple of details in the programming of the tunnel Simphony template and federation as listed below. Also see the Tunnel Construction User Manual Document for more details that not covered here.

1. Lining of the last meter of the tunnel:

The last meter of the any prerequisite tunnel element (ie. the last section of a tunnel element that is not connected to the Finish Construction element) will be lined before leaving the tunnel element.

During the construction of the last Tunnel (the one connected to the Finish Construction element) the TBM will be pushed out into the Finish Construction (ie. the dirt entity will transfer out of the Tunnel element towards the Finish Construction element) and the last meter will be declared as lined after FinalTasks are complete. The dirt entity is sent back through the Tunnels where the last section of the last tunnel will get updated.

```
// If this is the last section...
// If there remains unlined tunnel at the end due to length of the tbm,
// hand-line the tunnel as the tbm is pushed into the retrieval shaft.
if   (this.Tunnel.LastSection   ==   this   &&   this.CurrentLinedLength   +
this.Tunnel.SegmentLength >= this.Length)
{
// If it's the first and only tunnel, the tbm is pushed into the retrieval
shaft. Or if the tunnel is completely lined
```

```
if (this.Tunnel == this.Tunnel.LastTunnel || this.CurrentLinedLength >=
this.Length)
{
// Schedule the train to transfer through this soil layer.
this.Engine.ScheduleEvent(dirt, this.transferToTunnelFaceEvent, duration);
}
else if (!this.FinishTime.IsFinite())
{
// Move to the tunnel face and continue to line the last meter.
this.Engine.ScheduleEvent(dirt,this.Tunnel.ArriveAtTunnelFaceEvent,
duration);
}
}
else
{
// Move to the tunnel to continue excavation / lining.
this.Engine.ScheduleEvent(dirt,this.Tunnel.ArriveAtTunnelFaceEvent,
duration);
}
```

(From the TbmTunnelSection class)

Multiple tunnel elements now act as if it were a single (and the first) Tunnel. For cosye purposes especially, Simphony sends a cumulative lined length (the sum of all linedlength of each tunnel element) for reporting. To make the Tunnel elements behave like tunnel sections when a dirt entity arrives at subsequent tunnels, it maintains the number of trains, train instances, capacity etc. from the first tunnel element. (ie. we do not create additional or new set of trains). A property called *CurrentTunnel* was created and as the name suggests is a reference to the tunnel that being excavated at that moment. Only current tunnel is allowed to send updates to other federate. The *LastTunnel* is the tunnel element that is connected to a finish construction element. It is made so that the AsBuilt and the Reporting Federate are able to send and receive updates to the correct tunnel object instance, but from a user point of view it will seem like a single tunnel as being worked on.

Appendix E- User Manual for the Tunnel Construction Federation in COSYE

This research utilizes an HLA-based distributed simulation to record as-built documents in different construction projects. In this research, a tunnel construction project is used as a case application to document as-built project information in an integrated manner. The developed model is used to record the entire tunnel construction process and product information, including adequate details of project progress, resource interaction, daily site information and product data.

The tunnel federation is a distributed simulation model developed in the COSYE environment to simulate the tunneling operation and record as-built information. The federation consists of a number of components including the Simphony federate, visualization federate, report federate, and support federates.

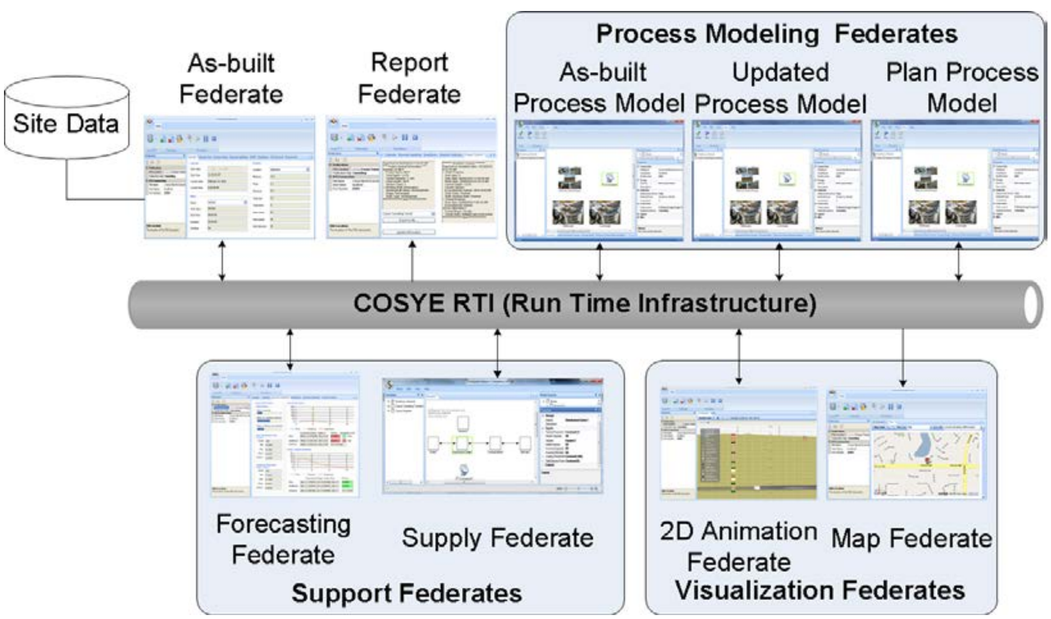


Figure 1– Tunnel Construction Federation

But for as-built documentation, three main components should join the federation: the Simphony federate to model the as-built construction operation, the report federate, and the as-built federate. The Simphony federate, which is responsible to model tunnel construction operation, is the main component of the federation. Based on the actual project status, two, three, or more Simphony federates may join the federation, which will be discussed later in this document. The as-built federate is responsible to update as-built information by connecting to the Microsoft Access file that stores plan and as-built data in multiple tables. Finally, the report federate generates a summary and detailed reports for the comparison of plan and actual project results for various project management studies. This manual explains the process of executing the federation and provides useful details of running the federation.

Tunnel Federation Execution

To run the tunnel federation for the purpose of as-built documentation, four federates should join the federation when no changes happen to the process model: plan Simphony model, as-built Simphony model, a report federate, and an as-built federate. If any changes happen to the construction process and therefore the model in Simphony requires updating, another Simphony federate with the actual process model should be created. This federate, as a new federate, can join the federation before its execution. To run the federation, the following steps should be followed:

- A user should add the RTI connection element to the project level in each Simphony model. The RTI connection has two main attributes that should be set to make the Simphony federate join the tunnel federation and communicate with other federates using the same FOM document. The "FddLocation" attribute presents the location of FOM for the Tunnel federation, and the "Federation

Name" defines the federation that a federate is joining. The FOM document is an XML file that is stored in the Cosye.Tunneling folder.

If the Simphony model is running on a different machine than the one that RTI is running on, "Host Name" and "Host Number" attributes should be changed to the name and IP address of the computer that the RTI is running on.

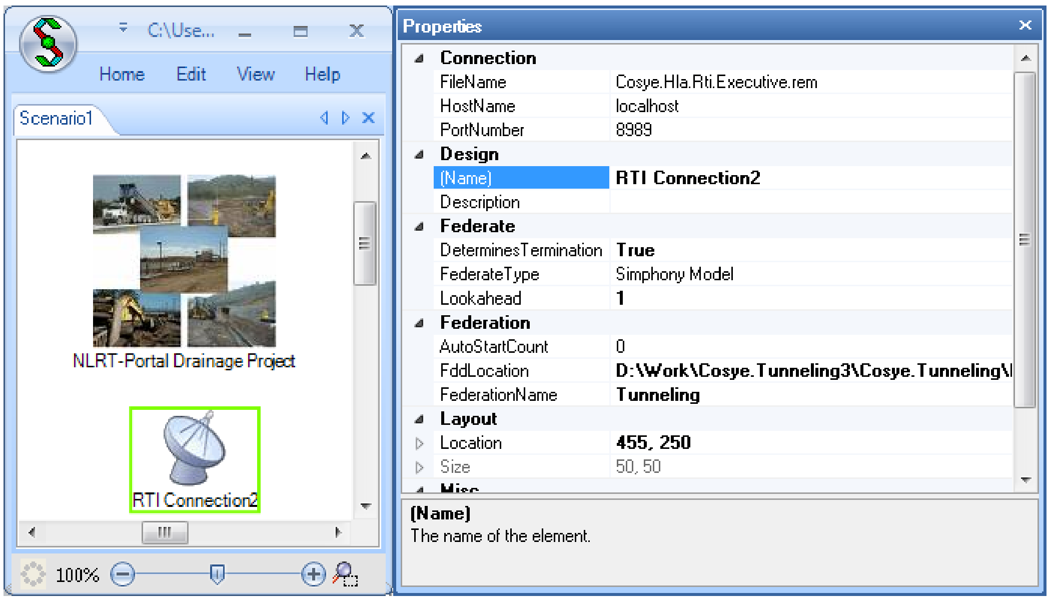


Figure 2– RTI Connection Properties in the Simphony Federate of the Tunnel Federation

- If any changes happen to the process model and a new Simphony federate joins the federation, a user should modify tunnel project names to facilitate the message sending and receiving process in the federation.
  - In the as-built Simphony model, project element should be named after the project name in the database.
  - The as-built project with a change order needs to contain project name but cannot be exactly the same as the previous one. For example, if the as-built project name is "NLRT-Portal Drainage Project," this one should be named "NLRT-Portal Drainage Project Revised."

- The non as-built project that runs just on user input can have any name but, **CANNOT** contain as-built project name (here: "NLRT-Portal Drainage Project").

The next step is to open the as-built federate. This federate should be open inside the Federate Host (shown below) which is in the COSYE folder in the program file (C:\Program Files (x86)\Cosye).

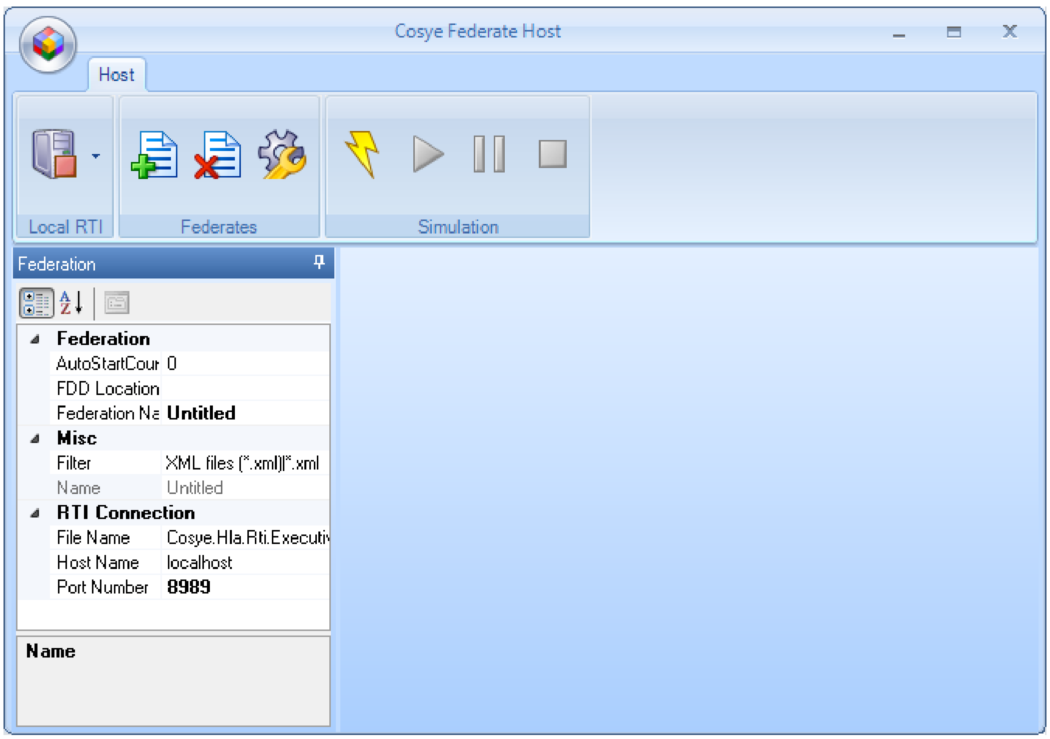


Figure 3– COSYE Federate Host

Then, a user can open the as-built federate inside the Federate Host. A user can select the XML file called “AsBuilt” from the following address:

C:\..\Cosye.Tunneling.Startup.AsBuilt\bin\Debug.

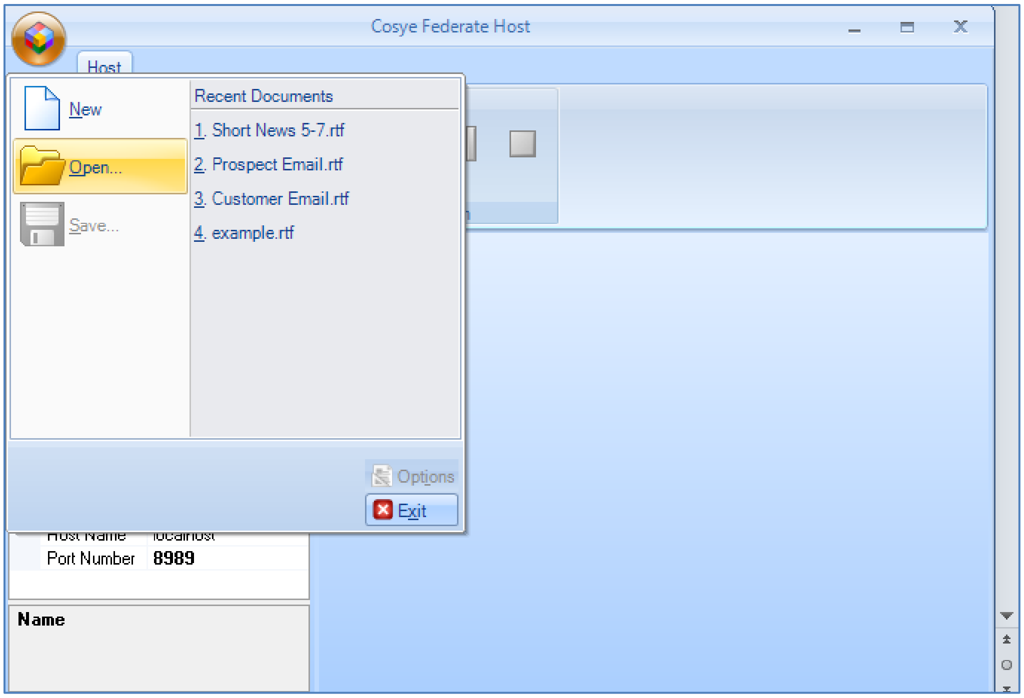


Figure 4– Opening As-built Federate inside the Federate Host

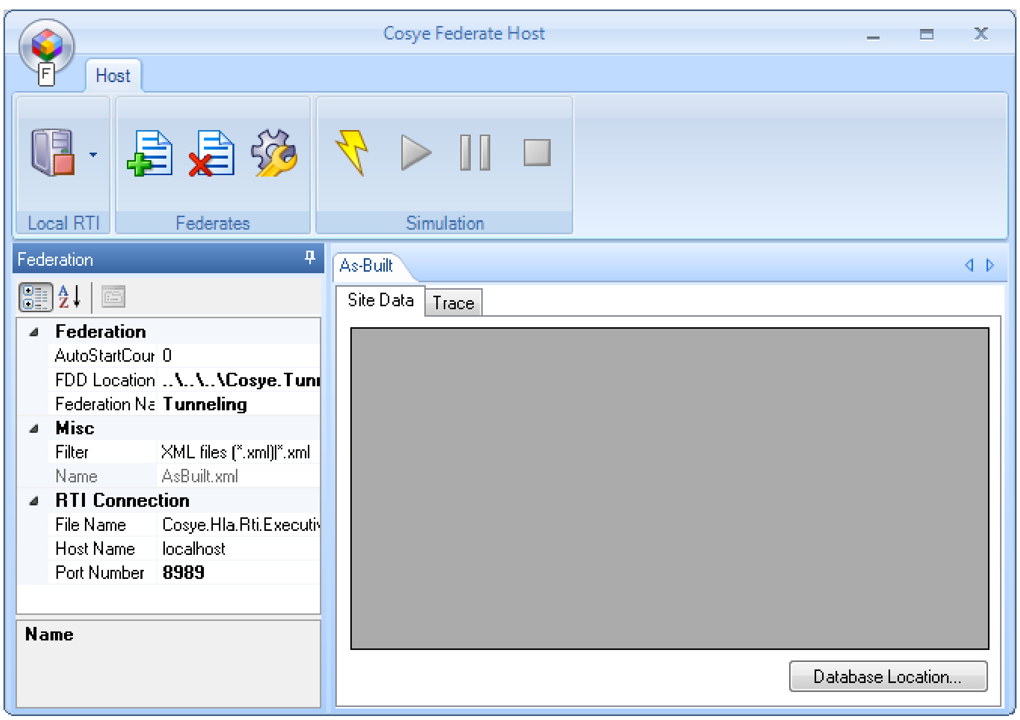


Figure 5–As-built Federate in the Tunnel Federation

In this phase, a user can specify an as-built database by clicking the “Database Location” button. A user can also select the as-built database after the federation execution starts. In this case, the as-built federate will automatically open a window and ask for the database file.

- Report federate can be opened by double-clicking on the file named “Cosye.Tunneling.ReportFederate” at the following address: C:\..\Cosye.Tunneling.ReportFederate\bin\Debug.

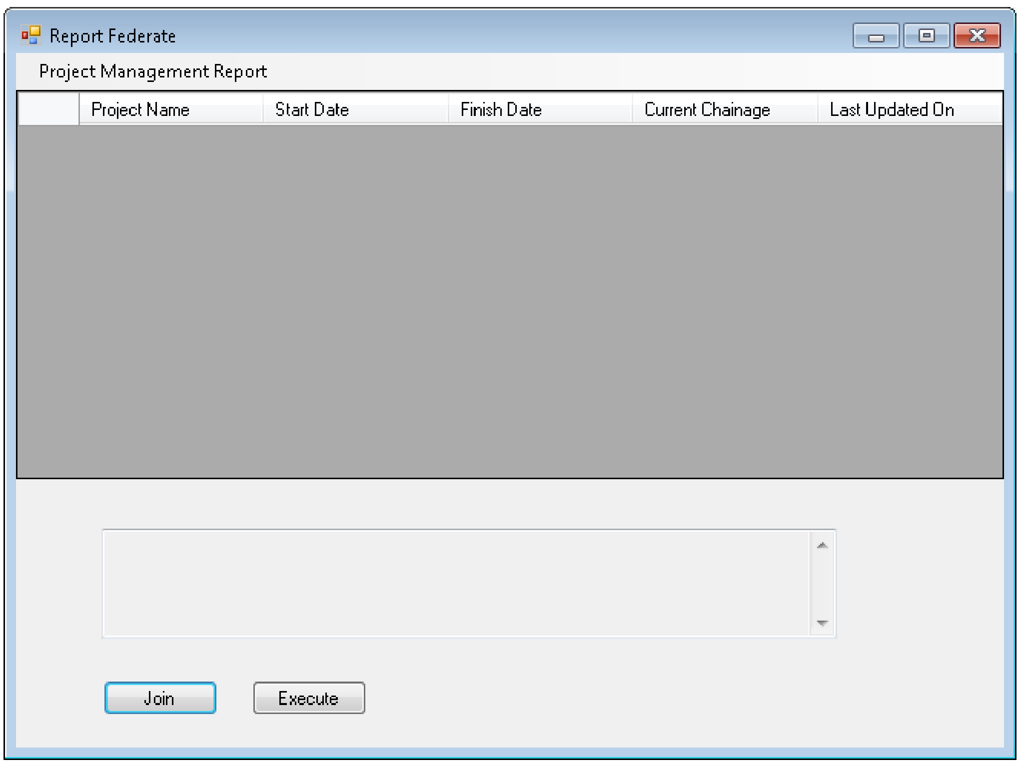


Figure 6–Report Federate in the Tunnel Federation

After all federates get ready to join the tunnel federation, COSYE should start through a federate host or by opening COSYE from the start menu.

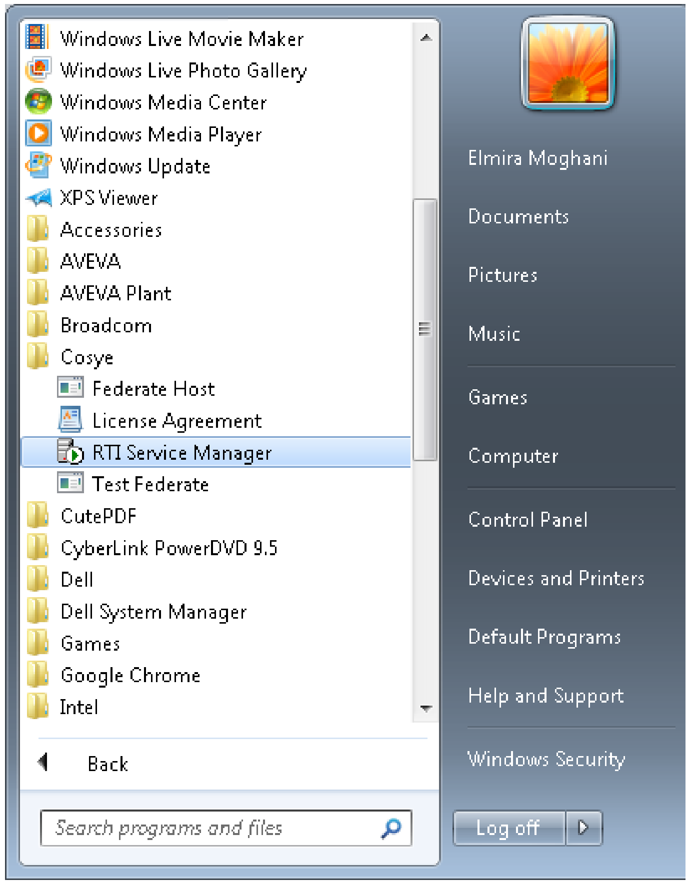


Figure 7–Running RTI from the Start Menu

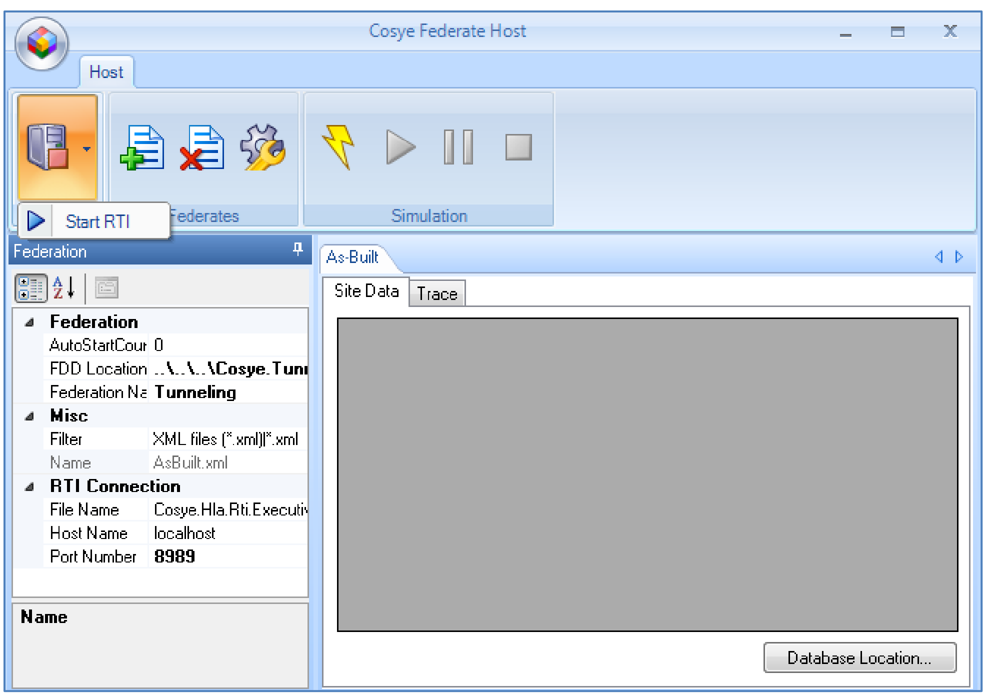


Figure 8–Start/Restart RTI Using the Federate Host

The next step is to join all federates to the federation. In the as-built federate and report federate, a user should click on the “Join” button, and in Simphony federates, a user should run the model. When all federates join the federation, a user can click on “Run Federation” or “execute” button to run the federation and wait until the simulation of Simphony models is complete.

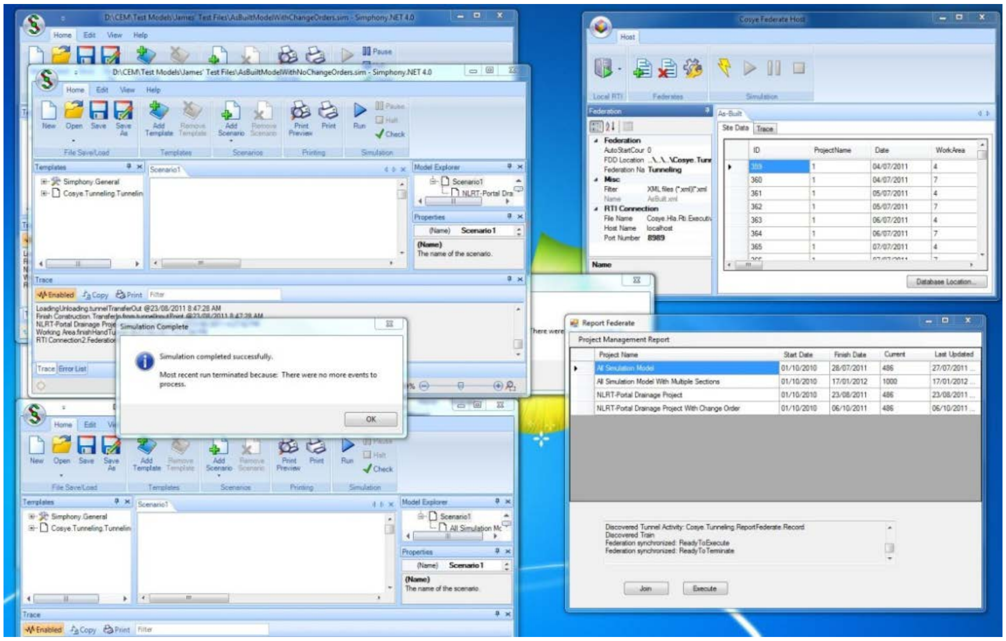


Figure 9–Running Tunnel Construction Federation to Document As-built Information

Once the simulation ends, a user can get multiple reports about the project, both for the planning and actual construction processes.

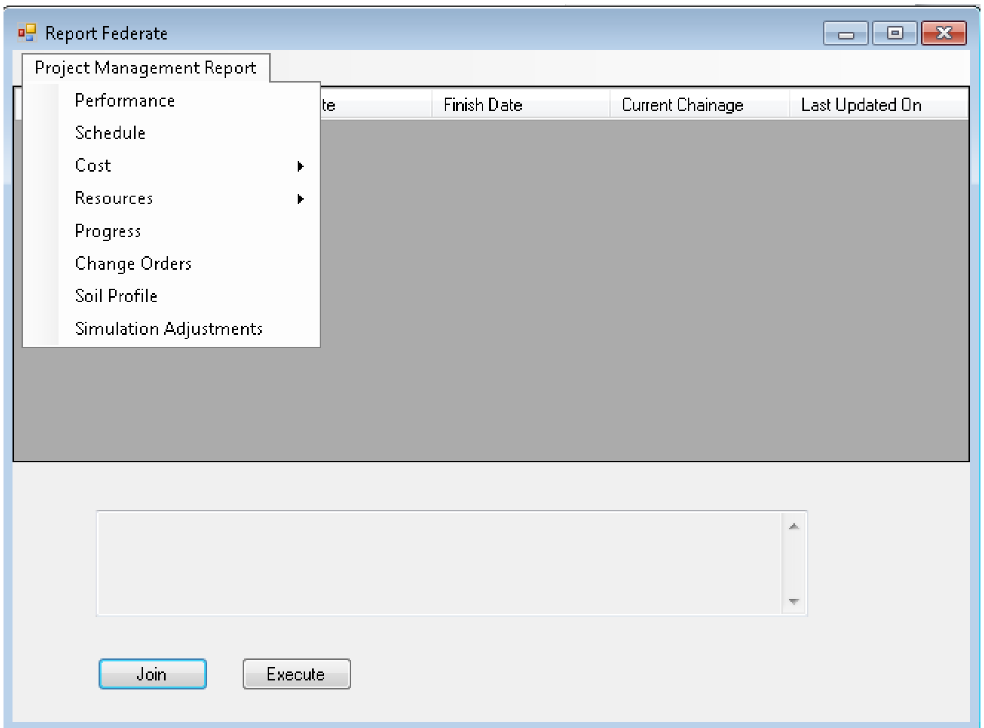


Figure 10–List of Management Reports in the Report Federate

In the middle of the simulation run, a user is also able to select a report and open it and get the latest status of a project.