University of Alberta

Developing a Generic Resource Allocation Framework for Construction Simulation

By

Hosein Taghaddos

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

Fall 2010
Edmonton, Alberta

Examining Committee

Dr. Simaan AbouRizk, Civil and Environmental Engineering

Dr. Yasser Mohamed, Civil and Environmental Engineering

Dr. SangHyun Lee, Civil and Environmental Engineering

Dr. Vivek Bindiganavile, Civil and Environmental Engineering

Dr. Michael Lipsett, Mechanical Engineering

Dr. Feniosky Peña-Mora, Civil and Environmental Engineering, Columbia University

Dedication

This thesis is dedicated with love, admiration, and respect

to my kind mother, my dear father, and my beloved sisters;

to my lovely wife;

and particularly to my most intimate and dearest friend, Imam Mahdi.

Abstract

The allocation of resources over time, referred to as resource scheduling, in large-scale construction environments is a challenging problem. Although traditional network scheduling techniques are the most popular scheduling techniques in the construction industry, they are ineffective in modeling the dynamic nature and resource interactions of large projects. Simulation based modeling or optimization techniques are also time-consuming, complicated and costly to be implemented in large-scale projects. This research is focused on developing a new framework to insert artificial intelligence inside construction simulations for facilitating the resource allocation process.

The first stage in this study was developing a framework to solve resource scheduling problems in large scale construction projects. This framework, called the Simulation Based Auction Protocol (SBAP), integrates Multi-Agent Resource Allocation (MARA) in a simulation environment. This hybrid framework deploys centralized MARA (i.e., auction protocols) whereby agents bid on different combinations of resources at the start of a simulation cycle. Agents attempt to improve their individual welfare by acquiring a combination of resources. An auctioneer is designed to allocate resources to the agents by maximizing the overall welfare of the society. Simulation is also employed to track the availability of resources, and manage resource oriented activities. This framework is implemented in two large construction applications of scheduling module

assembly yard and multiple heavy lift planning in modular construction.

The second objective of this project is to develop a generic resource allocation component for addressing optimized resource allocation in various construction projects. This component is developed in a large scale model using High Level Architecture (HLA), instead of traditional simulation environments. HLA allows splitting a large scale model, known as a federation, into a number of manageable components (i.e., federates), while maintaining interoperability between them. A generic Resource Allocation (RA) federate is designed to act as an auctioneer for federates developed based on the SBAP. Another generic federate is also built to automate the communication with the RA federate. These two generic federates can be reused in various construction federations. This framework is successfully implemented in an industrial construction process that involves different supply chains including spool fabrication, module assembly and heavy crane lifts in site construction.

Acknowledgement

"Whoever does not thank others, has not indeed thanked the God."

Apostle of the God, Mohammad (peace upon him and his family)

This thesis would not have been possible without the generous support and encouragement of my supervisors, Professor Simaan AbouRizk and Dr. Yasser Mohamed. I have been truly lucky to have both of them as my mentors. Dr. AbouRizk treated me as his son, and guided me with his keen advice, critical insight, and confidence throughout the whole dissertation process. Dr. Yasser Mohamed also inspired me with his wisdom, and his patience. I thank both of you from the bottom of my heart for your assistance in carrying out this research.

I would like to acknowledge Professor Feniosky Peña-Mora for serving as my external examiner and providing supportive advice. I am also grateful to my candidacy and doctoral committee: Dr. SangHyun Lee, Dr. Vivek Bindiganavile, and Professor Michael Lipsett, for their thoughtful review and invaluable suggestions.

I would like to take this opportunity to thank all the individuals, colleagues and friends who assisted me during this research project. I extend my sincere thanks to Rick Hermann, the Manager of Construction Engineering at PCL Industrial Management Inc., for his collaboration, support, and enthusiasm during this research project. I also appreciate the assistance of Stephen Hague, and other faculty and staff in the Construction Engineering and Management Program who assisted me in various aspects of this research. Particular gratitude is extended to Holly Parkis, the technical writer of the construction group, for reviewing this thesis vigilantly. Financial assistance from the University of Alberta and the Natural Sciences and Engineering Research Council of Canada during my PhD program is also appreciated.

Finally, I would like to express my deepest gratitude to my beloved family who taught me how to live. My sincere thanks to my lovely parents for their never ending support. I am but a product of your dreams and sacrifices. Although I have been away from you, your prayers have always paved my road to success. I also express my deep appreciation to my kind sisters, Soreh and Maedeh, and my dear brother-in-law, Amin, for their sincere support and motivation. I am also grateful to my other in-laws for their continual support and help. My special heartfelt acknowledgement must go to my loving wife Naimeh, whose warm companionship and kind support can never be appreciated enough. Thank you for your continuous love, inspiration, and patience during this journey. Words cannot express my gratitude and everlasting love toward you.

Contents

List of Tables

List of Figures

CHAPTER 1. Introduction [1]

1.1 Problem Statement

Effective allocation of resources is crucial in managing construction projects. Many large-scale construction projects are composed of several correlated projects, each of which includes several resource-constrained activities. There are also a number of scarce and expensive resources involved in construction projects. For instance, the rental cost of a heavy lift mobile crane may reach $1,500 per hour. Moreover, some of the resources (e.g., mobile crane, a skilled crew) are only available for a limited period of time.

These large-scale projects must therefore be carefully planned. Graphical representations of the project on a time scale, bar charts, and velocity diagrams were the main practical planning tools for several decades. In the late 1950s, scheduling networks such as Critical Path Method (CPM) and Program Evaluation and Review Technique (PERT) were developed by some large corporations (e.g., General Dynamics) for better management of large complex construction projects (Sawhney 1994). Since then, these scheduling networks have become popular in construction firms; most of these companies carry out some form of CPM scheduling using commercial scheduling software (e.g., MS.

[1] A part of this chapter is published at (Taghaddos 2009).

Project, Primavera) (AbouRizk and Mohamed 2002). Nevertheless, these network scheduling techniques are ineffective in modeling the dynamic nature of large-scale construction projects. CPM scheduling techniques are also weak in dealing with resource interactions (e.g., resource leveling) in multi-project construction environments. They also fail to model the stochastic nature of construction activities (Sawhney 1994; Tommelein 1998).

Simulation modeling is a powerful technique for overcoming the deficiencies of traditional scheduling networks (Sawhney 1994). It has become a reputable tool for capturing uncertainty in construction projects and for managing resources in the planning and execution phases. An elegant simulation model can answer many hypothetical scenarios. Simulation can be combined with optimization techniques such as genetic or heuristic algorithms to find an optimum resource schedule (Leu et al. 2000; Zhou 2006).

Simulation, however, has not been embraced by the majority of construction companies to allocate or schedule resources (McCabe 1997). This problem can be attributed to different causes including complexity of construction systems particularly for large-scale or multi-project construction environments, limitation of time and budget for plan development, lack of construction practitioner knowledge in simulation modeling, and inefficiency of current simulation environments for construction practitioners (Shi and AbouRizk 1994). In other words, many construction firms do not implement simulation-based or optimization techniques simply because they cannot afford or choose not to employ dedicated professionals who are expert in understanding the domain and simulation modeling and optimization algorithms. The other problem is lack of reusability of simulation-based or optimization models. Most often, it is more convenient to start from scratch rather than reuse previously developed models (El Ghandour 2007).

A number of studies indicate that inserting intelligent agents into simulation models makes the simulation process more powerful (Mohamed and AbouRizk 2005; Mukherjee 2005; Sawhney et al. 2003). Combining simulation environments with external artificial intelligence tools assists the modelers in automating the performance improvement of simulation modeling (Van Tol 2005). Nonetheless, some studies on building intelligent simulation models have revealed that intelligent simulation models in the current framework quickly become overwhelmed in terms of the size, detail, and the nature of their interactions (Van Tol 2005).

The research undertaken here is focused on developing a new framework to employ artificial intelligence inside construction simulations for facilitating the resource allocation process. The concept of framework can be defined as “a basic conceptual structure used to solve or address complex issues” (Razdan et al. 2010).

This framework can be easily employed in various large-scale construction projects to make effective allocation of resources practical. This framework also allows reusing the developed artificial intelligence in different types of construction simulations.

1.2 Scope of Research

Although construction projects have many differences and each seems unique, they share some common issues. For example, most projects are multi-unit projects, they are affected by weather conditions, and they usually have resource-driven scheduling. In the traditional approach, we try to deal with these issues (e.g., resource-driven scheduling) in each single project. However, the main scope of this research is to develop a generic framework to reuse the efforts of solving any of these issues, particularly resource scheduling, in future construction projects.

This study has employed a distributed simulation technology called High Level Architecture (HLA) to promote reusability of different modeling components. HLA allows splitting a large scale model, known as a federation, into a number of manageable components, known as federates, while maintaining interoperability between them (Taghaddos et al. 2008b). Therefore, HLA is a suitable simulation environment to reach our goal, which is enhancement of the reusability of modeling efforts such as resource-driven scheduling in future construction projects.

In the HLA environment, various federates may handle different issues of the federation. For example, Shahin (2007) has developed a federate to incorporate the impact of weather on the productivity in a construction project. The focus of this project is developing a generic Resource Allocation (RA) federate to act as an artificial intelligence inside a simulation model. This RA federate can assist different federates with optimum allocation of resources.

The journey to start the RA federate began with developing a framework to solve resource scheduling problems in large-scale construction projects. This framework, referred to as Simulation Based Auction Protocol (SBAP), is quite flexible and can be customized for different construction projects for efficient utilization of resources. SBAP can be either embedded in traditional simulation environment, as is explained in chapter 3, or employed in simulation federates in the HLA environment, as is explained in chapter 6. This framework is developed based on the centralized version of Multi-Agent Resource Allocation (MARA).

MARA is an excellent structure for allocation resources among different agents and finding a feasible or optimum resource allocation procedure. MARA is not any specific algorithm; however, it allows employing various combinatorial optimizations algorithms. Although the allocation procedure in MARA can be either centralized or distributed, the centralized allocation is suitable for this study for developing an independent federate. This centralized allocation, also referred to as auction protocol, is derived from auction theory, used primarily in

Microeconomics. In this protocol, a single element (i.e., auctioneer) decides on the allocation of resources among agents, once the agents submit their preferences over alternative allocations, referred to as bidding stage (Chevaleyre et al. 2006). Although MARA works well for resource allocation problems, it is not practical to use for resource scheduling problems in large-scale construction projects due to numerous reasons explained in chapter 2. Hence, this MARA framework is modifies to the SBAP framework to suite for solving large-scale construction problems. This framework is explained in detail in chapter 3 of this thesis.

1.3 Research Objectives

The primary objective of the presented research is to employ artificial intelligence inside construction simulations for facilitating the resource allocation process in various construction projects. To reach this goal, three auxiliary objectives have been identified:

I. To develop a resource scheduling framework, referred to as a Simulation Based Auction Protocol (SBAP) framework, based on auction protocols and discrete event simulation modeling.

II. To model a large-scale industrial construction application containing various supply chains and to spilt the model into manageable components (i.e., federates) by employing High Level Architecture (HLA).

III. To develop a generic resource allocation federate that is reusable in various construction simulations by employing HLA and the SBAP framework.

1.4 Research Methodology

The above-mentioned objectives are achieved by following the methodology below:

1. Modeling two large-scale construction project (the module assembly yard, and site construction with multiple heavy lift cranes)

   Developing a generic framework, referred to as the Simulation Based Auction Protocol (SBAP), to solve resource scheduling problem in various large-scale construction projects
2. Validating the SBAP in a fabrication problem
3. Implementing the SBAP in the above-mentioned large-scale construction case studies
4. Developing a comprehensive federation of industrial construction involving several federates (e.g., the module assembly yard and site construction) based on the HLA
5. Developing a generic Resource Allocation (RA) federate to serve various construction federates, built based on the SBAP, to allocate the resources effectively

1.5 *Implementation Environment*

The proposed SBAP framework and the developed generic resource allocation federate have been implemented in large-scale construction applications. The software systems in this development are Visual Studio 2008, Simphony, COSYE, Microsoft Access 2007 and Microsoft Visio 2007. These systems are explained below:

- Simphony is a discrete event simulation environment for construction projects based on unified modeling technology (Hajjar 1999). It allows the user to develop Special Purpose Simulation (SPS) templates as a means to facilitate the adoption of simulation by the industry (Hajjar and AbouRizk 1999). Simphony also provides highly flexible simulation tools that support graphical, hierarchical and modular modeling (AbouRizk and Mohamed 2000). The first generation of Simphony, called the legacy version of Simphony (Simphony 1.05), was developed in 1998 for the COM or ActiveX environment using object oriented programming in Visual Basic 6 (VB6). Shortly after the first release of the .NET Framework, legacy Simphony was updated to Simphony.NET in 2003. Simphony.NET was initially targeted at version 1.0 of the .NET framework. Several versions of Simphony have been released since 2003, concurrent with upgrades of the Microsoft .NET framework. The most recent version of Simphony, called Simphony.NET 3.5, is targeted at version 3.5 of the .NET framework. Simphony.NET 3.5 is a complete redevelopment of the Simphony simulation environment to be more extensible. It is also intended to be used as the simulation engine by models developed within the COSYE environment. Therefore, the programming environment in Simphony.NET 3.5 is Visual Studio, while the programming environment of the previous versions of Simphony was internal (i.e., inside Simphony).

- The Construction Simulation Environment (COSYE) is an HLA-based simulation environment developed at the University of Alberta (AbouRizk et al. 2006; AbouRizk and Robinson 2006). COSYE runs on Windows.NET, and is composed of Run-Time Infrastructure (RTI) software as the backbone of the federation, a XML editor to provide the Federation Object Model (FOM), a discrete event simulation engine (Simphony 3.5), and time-stepped federates. During run time, COSYE provides the necessary communication, information exchange, and data-sharing protocols using the RTI. The developed RTI assures simulation synchronization, coordination, and consistency between the different federates.
- Visual Studio 2008 is the main programming environment for the COSYE and Simphony.NET 3.5. It also provides the necessary connections to communicate with databases in Microsoft Access 2007 or with Microsoft Visio 2007 as the third party visualization system

Although the proposed framework is generic, some specific case studies are used to demonstrate and to validate the framework. This does not imply that this framework performs better for these case studies. The proposed framework can be employed for solving resource scheduling problems in various construction applications.

1.6 Thesis Organization

The remainder of this thesis is organized into the following chapters: Chapter 2 contains a brief overview of resource allocation, construction simulation, HLA, and MARA. It also reviews the current state-of-the-art in embedding agents inside simulation models. Chapter 3 presents the proposed SBAP framework for solving resource scheduling in large-scale construction projects. Chapters 4 and 5 discuss the implementation of the SBAP in two large-scale construction case studies of scheduling module assembly yard and industrial crane operations. Chapter 6 discusses developing generic resource allocation federates based on HLA and the SBAP framework. These federates have been successfully implemented in the industrial construction federation which involves different supply chains including drafting, material procurement, fabrication shop, module assembly (i.e., the first case study) and site construction (i.e., the second case study). Chapter 7 presents a summary of the research, its contributions as well as its limitations. This chapter also offers some suggestions for future work. Finally, two papers, titled “Simulation-based resource leveling in multi-project construction” and “Simulation-based schedule enhancement of tower cranes”, are presented in the appendixes (Taghaddos et al. 2008b; Moghani et al. 2009). Although these papers are not directly aligned with the objective of this thesis, they have been preliminary steps for the development of simulation models for the module assembly yard and mobile crane operations case studies. These appendixes are explained briefly in chapters 4 and 5 of the thesis.

CHAPTER 2. Literature Review[2]

2.1 Introduction

The main objective of this thesis is development of a generic framework for effective allocation of resources in construction simulation. This framework is founded based on the Multi Agent Resource Allocation (MARA) and High Level Architecture (HLA). The review on prior work touches four different areas: resource allocation, construction simulation, HLA, and MARA, as well as a brief review of the state of the art of embedding agents inside simulation models.

2.2 Resource Allocation

Various types of resources are involved in construction projects including manpower, equipment, materials, money, and space (Tharachai 2004; Willis 1986). Some of these resources are scarce and expensive. For example, rental cost of a heavy lift mobile crane in the province of Alberta may reach $1,500 per hour. Skilled crew is also a valuable and expensive resource in construction projects, and space can be scarce in congested construction sites.

---

[2] Parts of this chapter are published at (Taghaddos et al. 2008b; Taghaddos et al. Submitted).

Effective allocation of resources is crucial for the success of construction projects. This success implies accomplishing the project on time, in budget and with an acceptable quality. Therefore, the concept of resource allocation is introduced to the construction industry as the process of assigning available resource(s) among the various activities that are competing for the same resource(s) (Lasry et al. 2008). Each activity demands specific resources, or a range of resources, at different times. The sum total of the demands for resources from multiple projects becomes the demand for resources at the company level (Tharachai 2004).

Practically, the total amount of resources as well as the total budget of the construction companies is limited. Allocation of limited resources among various activities is sometimes referred to as constrained resource scheduling (Leu and Yang 1999; Ahuja et al. 1994). To put it another way, resource constrained scheduling deals with providing an optimum or feasible schedule for the problem by considering the limited amount of resources (Feng 1998). Providing such an optimum schedule is a complicated process, but has a key impact on the total cost and schedule of construction projects.

The efforts to solve resource allocation problems began in the 1960s and are still ongoing (Schwindt 2005). The early work in resource allocation was concerned with three types of resource allocation problems: the time-cost trade off, the project duration problem, and the resource leveling problem:

The time-cost trade off arises when durations of activities can shrink with increasing the amount of resources at the expense of extra direct cost of the project. However, reducing the duration of an activity may reduce the project cost eventually. The time-cost trade tries to find the best balance between the duration, cost and resources of activities.

1) The project duration approach attempts to schedule the project's activities to limited renewable resources such that all activities are completed within a minimum amount of time.
2) The objective of the resource leveling approach is to smooth the utilization of renewable resources over time without exceeding the Late Finish (LF) date of the project (Leu and Yang 1999; Schwindt 2005).

In general, resource allocation seeks the best schedule in terms of minimizing the total cost of the project, minimizing the project's duration or delivering by the due date, and maximizing the resource utilization (Feng 1998). All three aspects should be considered simultaneously. In other words, considering only one of these objectives, such as minimizing the total cost, is not a proper approach.

Since 50 years ago, a great deal of effort has been devoted to solving resource allocation problems using either heuristic procedures or optimum-yielding techniques. Heuristic procedures try to produce close-to-optimal solutions, while optimum-yielding (e.g., mathematical) techniques look for the best schedule (i.e., an accurate solution) (Feng 1998; Schwindt 2005). A resource-constrained scheduling problem is an NP-hard optimization problem, meaning that no algorithm can be guaranteed to find the optimum solution in a polynomial time (Pinedo 2008). Thus, it is natural to consider solving a resource scheduling problem using approximate (e.g., heuristic) methods in a reasonable amount of time. In other words, pure mathematical optimizations are not ideal for resource planners, particularly in large-scale construction projects, for several reasons. First, modeling the entire problem mathematically and imposing an integrated objective function that considers all the involved constraints in the problem is very complicated, particularly for large-scale construction projects. Second, pure mathematical techniques usually require many compromises of the problem's assumptions, which make the results of the solution unrealistic for large-scale practical problems. Moreover, most mathematical optimizations lead to one optimum solution and do not touch the other, feasible but less optimal, solutions. Finally, modeling the stochastic nature of construction problems using pure mathematical optimizations is very difficult, if not impossible (McCabe 1997; Feng 1998; Gopalakrishnan 1998).

The handbook of Demeulemeester and Herroelen (2002) provides a comprehensive state of the art of solving resource scheduling problem. This handbook overviews the previous research in exact and heuristic solutions with an emphasis on the problems that are still in need of considerable research effort (Demeulemeester and Herroelen 2002).

2.1 Construction Simulation

Simulation modeling is a powerful approach for enhancing the planning and performance of a project. Computer simulation allows the planner to model the real-world system using a computer and to experiment with a mathematical-logical model (Law 2006; Pritsker et al. 1989; Pritsker 1986). Simulation addresses the random nature, resource-driven characteristics and dynamic interactions of a construction process during operation. It is also an effective tool to allocate limited resources to different activities effectively (Shi 1999). Simulation can be employed as a standalone application or combined with other optimization models (e.g., belief networks) to find an optimum (or close-to-optimum) solution (McCabe 1997).

Early computer simulations were developed by writing code from scratch using programming languages such as FORTRAN (Mohamed and AbouRizk 2005). Developing simulation-specific programming environments has facilitated modeling generic systems by providing a number of simulation-specific libraries (Mohamed and AbouRizk 2006). These General Purpose Simulation (GPS) languages such as SLAM (Pritsker 1986) are capable of supporting simulation modeling in any domain including manufacturing, industrial engineering, and construction. Some other languages have been developed specifically for construction, such as CYCLONE (Halpin 1977). CYCLONE provided a foundation for many other simulation systems such as CIPROS (Tommelein et al.

1994), STROBOSCOPE (Martinez and Ioannou 1994), Simphony (Hajjar and AbouRizk 1996), and SDESA (Lu 2003). Simphony provides Special Purpose Simulation (SPS) templates, which facilitate modeling projects within one particular domain of construction operations. A SPS template is beneficial for a practitioner who is not knowledgeable in simulation (and particularly in its programming) to easily simulate a construction domain (AbouRizk and Mohamed 2000). Examples of these SPS templates are CRUISER (Hajjar and AbouRizk 1996), EARTHMOVER (Martinez 1998), and module assembly yard templates (Taghaddos et al. 2009).

Simulation models can be classified according to various criteria. They can be categorized as deterministic or stochastic, continuous or discrete, and local or distributed. A deterministic simulation model deals with events occurring in a predictable manner, while a stochastic simulation model deals with data with an irregular pattern (i.e., probabilistic data) (Chung 2004). A continuous simulation model represents a system of equations or mathematical models using differentiation, integration, or approximation, while a discrete simulation model manages events by logical relationships between processes (Yasantha 2001). A local simulation represents one application in one computer, while a distributed simulation consists of several autonomous simulation models that communicate through a computer network.

Previous research has identified several limitations of the traditional local simulation environments for large scale construction projects (El Ghandour 2007; AbouRizk and Robinson 2006; Wang 2006). Some of these limitations are because of computing limitations. For example, Wang (2006) showed that modeling the industrial construction involving pipe spool fabrication quickly overpowered the traditional simulation in terms of model topology and computing processing. The other limitation relates to the ability of the simulator to conceptualize, document and model a large-scale construction project. In this case, simulation analysts must be knowledgeable enough to reflect the reality of the entire system, even if it is a comprehensive project (El Ghandour 2007). Moreover, the capacity of current simulation models, even the ones with hierarchical systems, is limited to an input-output port. These models are not effective when the hierarchies are not driven by traversing entities (AbouRizk and Robinson 2006). A search for simulation technologies used by other industries that can deal with the complexity of large-scale projects concluded that High Level Architecture (HLA) is one of the most promising approaches to address these issues.

2.2 High Level Architecture (HLA)

Distributed simulation is as emerging technology to distribute "execution of a single run of a simulation program across multiple processors" (Fujimoto 2003). Distributed simulation has been developed for collaborative simulation independently since the 1970s in at least three large industries: the defense industry, the Internet/gaming industry, and the high performance computing industry (Fujimoto 2003; Gan et al. 2003). Numerous motivations for developing distributed simulation in these communities were involved, including reducing the length of execution time, integrating different simulation models together, and geographical extension of the simulation execution. In the last 30 years, several standards have been developed for distributed simulation including Distributed Interactive Simulation (DIS) and High Level Architecture (HLA).

HLA is defined as a high level (i.e., general purpose) architecture for distributed simulations. It was developed in the mid 1990s in the context of defense applications. Nevertheless, HLA was intended to be applied across a broad range of simulation application areas including engineering. HLA does not prescribe a specific implementation or software application; in other words, it was envisioned that different implementations and software would be used as computer technology advances (Fujimoto 2003). HLA is designed to integrate separate components of simulation models, referred to as federates, into a single distributed simulation model, referred to as federation (Fujimoto 2003; Kuhl et al. 1999). The main motivations for using HLA-compliant simulation model are:

a. Reusability and Interoperability

   The main intent of HLA is to promote interoperability between simulations and to aid the reusability of developed models in different contexts, ultimately reducing both the time and the cost required to create a new environment (Shahin 2007; Fujimoto 2003; Kuhl et al. 1999). In many cases, parts of the simulation model may have already been developed for other applications. However, linking those simulation models to simulate the area of interest is such a complicated task that modelers usually prefer to start from scratch. This is due to the lack of reusability and interoperability in traditional simulation models (Wang 2006). Reusability enables us to reuse a component of a simulation model (i.e., a federate) in different scenarios. Interoperability enables us to make a component of a simulation model (a federate) cooperate with other components without recoding (Shahin 2007; Kuhl et al. 1999). These features significantly facilitate developing construction simulations, because simulation models of different construction applications share a number of common components. For instance, a scheduling or breakdown federate can be used in different applications with some minor modification.

b. Integration

Another major feature of HLA is the ability to integrate several different simulations into a single simulation environment (Fujimoto 2003). Currently, simulation analysts must be knowledgeable enough to reflect the reality of the entire system in the developed model, but this is difficult, as a simulation model may be very comprehensive (El Ghandour 2007).

c. Geographical extension

The geographical extent over which the simulation is executed with HLA enables the model to be run at geographically disperse sites. This is particularly beneficial when personnel or resources (e.g., databases or special facilities) are involved in the simulation model (Gan et al. 2003). This is also useful in supply chain modeling, where building a single, centralized model sharing sensitive data is not an option (Gan et al. 2000). Distributed simulation can also greatly reduce costs by eliminating the need for the resources to be physically relocated (Fujimoto 2003).

d. Reducing the length of time to execute the simulation

One of the principal reasons for developing distributed simulation is to reduce the length of time needed to execute the simulation (Fujimoto 2003). HLA-compliant simulation allows each federate to run on a different computer, thereby speeding up the processing time of the system. However, reducing computational time greatly depends on the federation architecture.

e. Standardization of simulation components

The standardization of simulation components is another advantage of HLA. Developing a standardized approach assists modelers by saving a significant amount of time in regulating the way simulation systems are built, allowing modelers to concentrate on their knowledge of construction processes (El Ghandour 2007).

Once HLA was conceived as general purpose architecture applicable beyond defense applications, it went through the process of industry standardization. The first set of standards was published by the Simulation Interoperability Standards Committee (SISC) of the IEEE Computer Society in 2000. These IEEE standards characterize HLA by three main components:

1. *HLA rules*: a set of 10 fundamental simulation rules describing the general principles of HLA. These HLA rules must be obeyed if a federate or federation is to be regarded as HLA-compliant (Simulation Interoperability Standards Committee (SISC), IEEE. 2000).

2. *HLA interface specification*: The interface specification defines the functional interfaces between federates and underlying software, called the Run-Time Infrastructure (RTI). It also addresses the interoperability of federates and allows them to be coordinated with RTI. RTI is the software that acts as the simulation backbone, and it must conform to the HLA specifications (Figure 2-1). RTI provides services such as synchronization, communication, and data exchange between federates for supporting an HLA-compliant simulation. These services fall under six main areas: time management, object management, declaration management, federation management, ownership management, and data distribution management (Simulation Interoperability Standards Committee (SISC), IEEE. 2001a).

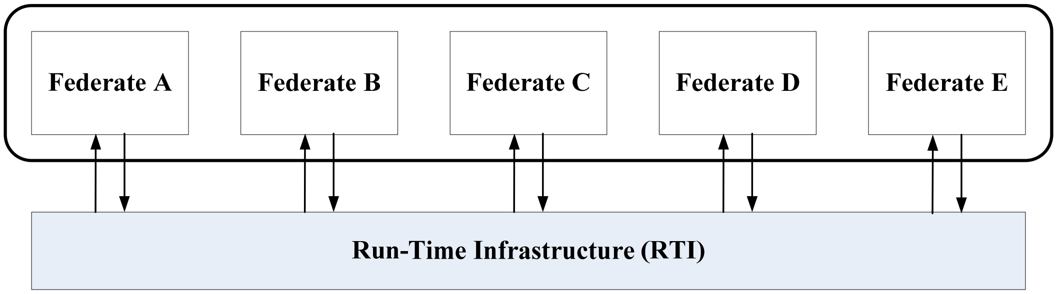


Figure 2-1 A schematic view of HLA structure

3. *Object Model Template (OMT)*: The OMT provides a common framework for data exchange between different federates. It consists of the Federation Object Model (FOM), and the Simulation Object Model (SOM). FOM describes the shared information regarding objects and interactions for the entire federation, while SOM specifies the shared data for a single federate. The main role of OMT is creating a common format in detail to promote collaborative modeling, reusability, and interoperability. In other words, OMT allows individual models to combine into a coherent simulation environment (Simulation Interoperability Standards Committee (SISC), IEEE. 2001b).

HLA is an excellent platform for embedding artificial intelligence into simulation models. Through this platform, an independent Resource Allocation (RA) federate can be developed that is reusable in various construction projects to assist different federates in allocating resources effectively. This RA federate is built in the presented research based on the Multi-Agent Resource Allocation (MARA) structure. A brief background of intelligent agents and MARA is presented as follows.

2.3 Multi-Agent Resource Allocation (MARA)

The term “agent” is widely used in a number of technologies, such as databases, artificial intelligence, operating systems, and computer networks (Bellifemine et al. 2007). There is a general consensus that autonomy is essential to the notion of agency, although no universal definition is accepted for this term (Weiss 1999). An agent tries to balance and satisfy its own objective function in the best possible way. Wooldridge and Jennings (1995) describes an agent as “a self-contained program capable of controlling its own decision-making and acting based on its perception of its environment, in pursuit of one or more objectives.” (Nwana 1996) emphasize the possession of autonomy, co-operation, and learning for the agents (Figure 2-2). These behavioral attributes assist the agent in operating without human guidance, in cooperating with other agents, and in learning (Wooldridge and Jennings 1995). The possession of all three attributes is what indicates an intelligent agent. Weiss (1999) define intelligent agents as being capable of flexible autonomous action.

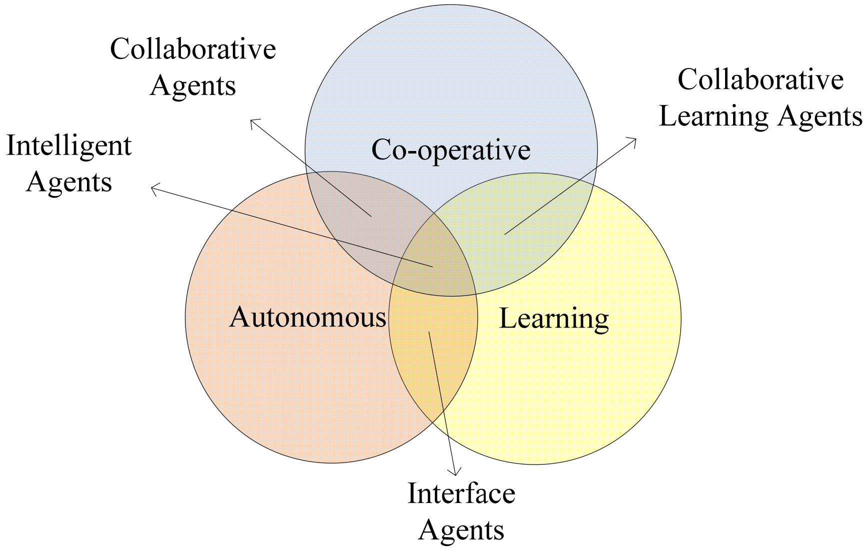


Figure 2-2 Nwana's (1996) requirements for agenthood

A number of agents can act collectively as a society and generate a multi-agent system (MAS). The main advantage of MAS is in expanding the functions of individual agents beyond their interconnected capabilities (Ren and Anumba 2004). The MAS paradigm offers an excellent structure for resource allocation problem that was discussed earlier in this chapter. This structure is called Multi-Agent Resource Allocation (MARA).

Multi-Agent Resource Allocation (MARA) is a structure for negotiating over resources and allocating them among individual agents (Chevaleyre et al. 2006; Chevaleyre et al. 2005; Moore et al. 1994). Although MARA was only introduced at the beginning of this decade, it has developed very rapidly. MARA addresses not only resource distribution issues, but also the interaction between the agents who share resources. MARA is an organized, well formulated structure that makes it suitable for applications in the industry (Liu 2009). A brief description of components of MARA is presented below.

2.3.1 Resources

Resources refer to items that are being distributed. Resources in general can be categorized into different classes: renewable resources such as manpower, machines, tools, and space are available on a period-by-period basis; non-renewable resources such as money, raw materials, and energy are available on a total project basis; doubly-constrained resources such as man-hours per day are incorporated as a combination of both renewable and non-renewable resources. In other words, these doubly-constrained resources are constrained per period as well as for the overall project. Some resources such as those with availability for a specific time interval might be partially renewable or nonrenewable. Resources can also be categorized from a divisibility viewpoint as continuous and discrete resources. Finally they can be categorized from a preemption viewpoint into preemptive and non-preemptive resources. Preemptive resources can be withdrawn from the current process tasks, and returned later, while non-preemptive resources cannot (Demeulemeester and Herroelen 2002).

2.3.2 Agents

The term “agent” was discussed in the previous section. In the MARA context, agents refer to entities which are competing over resources (Liu 2009). These agents have autonomy to declare their preferences over resources and accept or deny them. In the bidding language, agents are considered as bidders who are bidding for different combinations of resources. The agent's preferences for different resource alternatives are represented as bids.

2.3.3 Agent Preferences

The notion of individual welfare (i.e., the agent's preference) represents an agent's degree of satisfaction with a certain resource allocation. Although individual welfare may be measured quantitatively or qualitatively, here only the quantitative approach is considered. In the quantitative method, each agent is equipped with a utility function mapping an allocation to a real non-negative number called the agent's preference (Endriss et al. 2006; Chevaleyre et al. 2008). As mentioned previously, each agent preference corresponds to a bid for the agent. These bids can be expressed using different bidding languages including OR and XOR.

2.3.4 Bidding Languages

Bids can be built up with a bidding language which is concise and natural for humans to create and understand. The most basic bid request is an atomic bid, which indicates requesting one particular subset of the resources. An OR bid is a disjunction of atomic bids and can refer to any item or combination of items. However, the power of an OR bid is quite limited. For example, one may want to bid on either resource A for $100, or resource B for $200, but not both. Expressing such as example is not possible using OR bids. XOR is an exclusive OR for atomic bids, meaning that the agent is willing to accept one but not more than one of atomic bid. XOR limits the choice of the agent to only one resource and makes it easier to execute in a computer environment. Moreover, the XOR language has unlimited representational power and can represent all possible valuation functions. For example, the previous example can be easily expressed with the XOR language. However, XOR may not express some valuation functions efficiently. In general, a combination of OR and XOR provides a powerful bidding language (Liu 2009; Shoham and Leyton-Brown 2009).

2.3.5 Social Welfare

To analyze the behavior of a society of agents, in which agents co-operate or compete with each other, the concept of "social welfare" from microeconomics can be employed to evaluate the performance from a global point of view. Social welfare characterizes the well-being of overall society in relation to the welfare enjoyed by its individuals. In some cases, social welfare is best represented by utilitarian social welfare, in which the social welfare is interpreted as the sum of individual utilities. In some other cases, where a fair treatment of all agents is required, egalitarian social welfare best represents the entire welfare of the society. Egalitarian social welfare corresponds to the poorest individual welfare in the society (Chevaleyre et al. 2004; Arrow et al. 2002).

2.3.6 Resource Allocation

Allocation is defined as a particular distribution of resources among agents. MARA provides a flexible framework for incorporating different allocation procedures. Various optimization techniques can be employed in this framework to find a feasible or optimum resource allocation procedure. The allocation procedure aims to find a suitable allocation of resources. If the computation burden of the allocation procedure is shared by all agents in the society, it is called distributed resource allocation. Conversely, in a centralized resource allocation procedure, a single component manages the allocation of resources among agents (Chevaleyre et al. 2006; Buisman et al. 2007). This approach has emerged from combinatorial auctions in microeconomics and is the focus of the allocation procedure in this study.

2.3.7 Auctions

Auctions are structured ways of allocating scarce resources among agents. Auction protocols can be categorized as single good, multi-unit, or combinatorial auctions (Shoham and Leyton-Brown 2009; Cramton et al. 2006). Auction algorithms include an auctioneer, an explicit centralized component. The auction algorithm mechanism consists of two main steps: first, agents submit their bids for different combinations of resources so that their individual welfare is maximized or their revenue is minimized. Second, the auctioneer makes the final assignment by solving the Winner Determination Problem (WDP). Solving WDP corresponds to maximizing the social welfare of society or minimizing the total revenue of society. WDP can be solved with various WDP optimization algorithms, including the greedy algorithm, linear programming, and competitive equilibrium.

2.4 Embedding Agents inside Simulation Models

Improving and optimizing the simulation model is exhausting for the user, whether modeler or scheduler. He or she must build and observe the model, find areas for improvement, and implement the required changes to be able to make decisions at different phases of the simulation experiment. A number of studies indicate that inserting intelligent agents into the simulation model significantly empowers the simulation process (Mohamed and AbouRizk 2005; Mukherjee 2005; Sawhney et al. 2003).

Some researchers in the construction management have attempted to embed agents inside a simulation model and to produce automated evaluation tools (McCabe 1997). McCabe (1997) integrated belief networks into simulation to provide diagnostics for evaluating their performance, considering some performance measurement indices such as queue length, queue wait, server quantity, server utilization, and customer delay indexes. At the end of each simulation run, these indexes were calculated to evaluate remedial actions for the next simulation run by means of a belief network. Van Tol (2005) tried to make this approach dynamic to allow the simulation to react to the changes during the simulation run. However, he did not create an autonomous agent that was readily applicable in different construction projects. One of the main problems was that the process interactions of construction simulations such as Simphony (Hajjar and AbouRizk 1999) do not provide an efficient environment for handling autonomous generic agents. (Van Tol 2005) indicated that employing artificial intelligence outside of a simulation program heavily taxes the performance of the computer. Mohamed and AbouRizk (2005) also pointed out that embedding an agent inside the simulation model as an event scheduling simulation approach results in an inefficient simulation processing time which renders the simulation model impractical for realistic large-scale models. Reusing the agent in other simulation models also requires extensive recoding; sometimes it is easier to simply start from scratch instead.

This project proposes a framework to embed intelligent agents inside a simulation model based on HLA and MARA structures. This framework provides a generic resource allocation federate for facilitating and enhancing the resource allocation process in construction simulation. This federate can collaborate with different federates, or can be utilized in different federations. More detail of this framework is provided in the next chapters.

CHAPTER 3. Simulation Based Auction Protocol for Resource Scheduling Problems [3]

3.1 Introduction

Multi-Agent Resource Allocation (MARA) was explained in the previous chapter as a generic structure to allocate resources among different agents and to find a feasible or optimum resource allocation procedure. In the centralized resource allocation approach of MARA, known as the auction protocol, the agents submit their bids (i.e., preference over different combinations of resources) so that their individual welfare is maximized. Then an auctioneer decides on the allocation of resources among agents to maximize the social welfare of the entire society of agents (Chevaleyre et al. 2006). This protocol works well for resource allocation problems, but is not yet standardized for resource scheduling (i.e., resource allocation over time) problems. However, resource scheduling issues frequently arise in the allocation of resources in large-scale construction projects.

This chapter is focused on developing a Simulation Based Auction Protocol (SBAP) to solve resource scheduling in large-scale construction projects. Discussion in this chapter starts with a brief review to the auction protocol. Then the SBAP framework is introduced for solving resource scheduling problems in

---

[3] A version of this chapter is published at (Taghaddos et al. Submitted).

construction. The chapter concludes with implementing the SBAP in a typical fabrication scheduling example. In the next two chapters, the SBAP is implemented in two large-case construction applications.

3.2 Review of Auction Protocols

Auctions are important mechanisms for buying and selling goods or services. Some commonly used types of auctions are: English, Japanese, Dutch, first price sealed-bid, and second-price sealed-bid (Vickrey) auctions. Auctions can also be categorized as single-good, multi-unit, position, and combinatorial auctions (Shoham and Leyton-Brown 2009; Klemperer 1999). The literature on auction theory flourished when Vickrey introduced the concept of "second price" in 1961 (Vickrey 1961; Kutanoglu and Wu David 1999). Consequently, several hundred papers were written to study auction theory and competitive bidding in various applications (Stark and Rothkopf 1979; Rothkopf and Harstad 1994; Engelbrecht-Wiggans 1980; Smith 1991). An auction protocol is not any particular algorithm, but is a protocol to negotiate over resources and to allocate them among self-interested agents (Chevaleyre et al. 2006; Chevaleyre et al. 2005; Moore et al. 1994).

Auctions include an auctioneer, an explicit centralized component which collects the bids for resources from a wide range of bidders and awards the winning agents to maximize the auctioneer's payoff (Shoham and Leyton-Brown 2009). The auctioneer makes this final assignment by solving a Winner

Determination Problem (WDP). The WDP in a combinatorial auction, given the agents' declared utility function for different combinations of resources, can be solved by finding the allocation that maximizes social welfare. This problem can be expressed by (Shoham and Leyton-Brown 2009):

$$\begin{cases} \text{Maximize} \quad \sum_{i \in N} \sum_{S \subseteq R} \mathrm{u_i}(S) \mathrm{x_{s,i}} \\ \text{Subject to:} \\ \sum_{j \in S} \sum_{i \in N} x_{S,i} \leq 1 \quad \forall j \in R \\ \sum_{S \subseteq R} x_{S,i} \leq 1 \quad \forall i \in N \\ x_{S,i} = \{0,1\} \quad \forall S \subseteq X, i \in N \end{cases} \tag{3-1}$$

where N is the set of agents, R is the set of resources, S is the bundle of allocated resources to agent i, $u_i(S)$ is utility function of agent i for the bundle S, and $x_{S,i}$ is a binary variable in which $x_{S,i} = 1$ indicates that the bundle S is allocated to agent i. The first constraint in Equation (3-1) ensures that no overlapping bundle of resources is allocated. The second constraint ensures that no agent receives more than one bundle of resources. The last constraint makes the WDP an Integer Programming (IP) problem, which is a Linear Program (LP) with an added constraint of having integer variables (Shoham and Leyton-Brown 2009)(Wilken et al. 2000). It is well known that the IP problem is NP-complete, meaning that no polynomial-time algorithm (i.e., fast solution) to solve the problem is known. There are two approaches to get around the computational issue in solving IP problems.

First, heuristic (i.e., approximate) methods can always be employed to solve WDP problems. Heuristic algorithms can be categorized as complete and incomplete methods. Complete methods (e.g., tree search algorithm) are guaranteed to find an optimal solution, if it exists. However, it is may take a very long period of time to reach the optimum solution. Incomplete methods (e.g., the greedy algorithm, local search algorithm) are not guaranteed to find an optimal solution, but they can reach an approximate solution within a limited amount of time. Both complete and incomplete algorithms often perform well and can solve many practical problems, despite their theoretical caveats (Shoham and Leyton-Brown 2009).

Second, a polynomial-time solution exists to solve an IP problem in some circumstances. For example, Dynamic Programming (DP) is an efficient algorithm to solve WDP in cubic time, if bundles of resources do not contain more than two resources. Another example is when the constraint matrix is Total Unimodularity (TU), meaning that the determinant of every square sub-matrix is 0, 1, or -1. In these cases, the WDP can be solved with an LP. Single-dimensional problems are a subclass of TU matrices. At the following, the assignment problem is reviewed briefly as a single-dimensional WDP problem (Shoham and Leyton-Brown 2009).

3.3 Assignment Problems

The assignment problem is a single-dimensional combinatorial allocation problem, where a set of n agents bid for a set of m resources incurring some cost or profit (v). Each agent may bid for one or maximum m resources. This problem consists of a set N of n agents, a set R of m resources, a set M of all possible assignment pairs ($M \subseteq N \times R$), and a value function v mapping each assignment pair to a real positive value ($v: M \rightarrow \Re^{+}$). In a symmetric problem, n agents and m resources (m = n) have to be matched on a one-to-one basis (Shoham and Leyton-Brown 2009; Bertsekas 1992).

The role of the auctioneer is assigning each resource to a maximum of one agent so that the social welfare (i.e., total profit) is maximized or the total cost is minimized. Equation (3-2) expresses the WDP as a LP (Shoham and Leyton-Brown 2009; Bertsekas 1988):

$$\begin{cases} \text{Maximize} \quad \sum_{(i,j)\in M} v(i,j)x_{i,j} \\ \text{Subject to:} \\ \quad \sum_{j|(i,j)\in M} x_{i,j} \leq 1 \quad \forall i \in N \\ \quad \sum_{i|(i,j)\in M} x_{i,j} \leq 1 \quad \forall j \in R \end{cases} \tag{3-2}$$

where x is a binary indicator matrix in which $x_{i,j} = 1$ indicates that agent i and resource j, or the pair (i,j), is selected, and $x_{i,j} = 0$ indicates otherwise.

Table 3-1 represent a simple assignment problem that consists of 3 agents (a_1, a_2, a_3) and 3 resources (r_1, r_2, r_3). In this example, agent 1 is willing to bid \$2 on resource r_1 and \$4 on resource r_2, but is not willing to bid on resource r_3 at all. The highlighted values in Table 3-1, [(agent 1, r_1), (agent 2, r_2), (agent 3, r_3)], represent the optimum solution with maximum profit for the system. In general, the optimum solution of the entire system differs with the optimum solution for each individual agent. For instance, the optimum selection for agent 1 is receiving resource r_2, while the optimum selection for the entire system is assigning r_1 to agent 1.

Table 3-1 A simple assignment problem (Wellman and others 2001)

Resources:	r_1	r_2	r_3
agent 1	**$ 2**	$ 4	$ -
agent 2	$ 1	**$ 5**	$ -
agent 3	$ 1	$ 3	**$ 2**

The assignment problem can be solved with various optimization techniques. It can be solved with an LP (e.g., simplex algorithm) in polynomial time with complexity order of $O(n^3)$. However, the solving time for a large-scale WDP in practice might be too high. The LP solution is also not robust to changes in the problem specifications. Therefore, more robust solutions, called auction algorithms, are proposed for solving symmetric assignment problems based on the economic nature of competitive equilibrium. These algorithms are not the focus of this thesis; however, they are briefly presented here to be employed in future improvements (Shoham and Leyton-Brown 2009).

Auction algorithms are fundamentally different approaches for solving linear networks in an assignment problem using parallelism (Bertsekas 1992). One of the robust auction algorithms for the symmetric assignment problems is an “ascending-auction algorithm” (ε-competitive equilibrium), which resembles the

English auction in auction theory. The main concept in this algorithm is competitive equilibrium, a balanced economic situation in which resources are priced such that no buyer or seller can improve its bargaining position. As a result, the marginal benefit of the agents is equivalent to the marginal cost of the resources (Callan and Thomas 2004). Similarly, in the ascending-auction algorithm, each resource is associated with a price (p_j) and a utility, which is defined for each pair of possible assignment as u(i, j) = $v(i,j)$- p_j. An assignment and a set of prices are in "competitive equilibrium" condition, if no agent can gain any profit by bidding for another resource at the current prices ($u(i,j) \geq u(i,k), \forall k$). If a feasible assignment and a set of price for the agents (shadow price) are in competitive equilibrium condition, the assignment is an optimal one. Furthermore, there is always a set of certain prices for resources satisfying the competitive equilibrium condition for any optimal solution (Shoham and Leyton-Brown 2009).

The ascending-auction algorithm consists of several rounds, where in each round an agent bids for different resources and selects the one with the minimum value. In this algorithm, the initial price of resources is set at zero. At each round, the current price of non-selected resources stays the same and the current price of selected resources is added by the bid increment of difference between the price of the first and second-best agents ($b_i = u(i,j) - max_{k:(i,k)\epsilon M, k \neq j)} u(i,k) + \epsilon$). Table 3-2 demonstrates solving the previous stated example with an ascending-auction algorithm. Details of this algorithm are elaborated in (Shoham and

Leyton-Brown 2009; Bertsekas 1992); it is proved that the ascending-auction algorithm converges to the optimum solution with a better performance than Linear Program (LP) methods.

Table 3-2 Solving the previous assignment problem with ascending-auction algorithm

round	x1	x2	x3	bidder	x1	x2	x3	preferred	bid increment	current assignment		
	p1	p2	p3		v (i,x1) - p_1	v (i,x2) - p_2	v (i,x3) - p_3	object	(e=.1)	x1	x2	x3
0	0	0	0	1	2	4	0	x2	2.1		1	
1	0	2.1	0	2	7	2.9	0	x1	4.2	2	1	
2	4.2	2.1	0	3	-3.2	27.9	2	x2	26	2	3	
3	4.2	28.1	0	1	-2.2	-24.1	0	x3	2.3	2	3	1

3.4 Scheduling Problems

The scheduling problem is a resource allocation problem, where agents bid for the resources over time (Shoham and Leyton-Brown 2009; Wellman et al. 2001). In this case, each agent is willing to pay for a set of resources to accomplish a time-constrained task in certain duration. The time constraint of a task might be release time, deadline, or precedence constraints (Wang et al. 2007a).

An auction based scheduling mechanism allows resource scheduling to be both locally autonomous and aligned with the global interest of the society (Kutanoglu and Wu David 1999; Parkes and Ungar 2001). The role of the auctioneer is to set a feasible schedule, in which a set of time slots is assigned to each agent so that the total cost is minimized or the total revenue is maximized. Despite the importance of auction based scheduling, limited attention has been devoted to this area.

Some researchers have dealt with this problem by imposing a discretization of the time line into finite time slots and mapping the time slots into indivisible distinct resources (Kutanoglu and Wu David 1999; Wellman et al. 2001). This method converts a scheduling problem to a WDP in auction protocols. Similar to the WDP, a scheduling problem can be encoded as an Integer Program (IP) as the following:

$$
\begin{cases}
\text{Maximize} & \sum_{S\subseteq X, i\in N} v_i(S)x_{i,S} \\
\text{Subject to:} & \\
\quad \sum_{S\subseteq X} x_{i,S} \leq 1 & \forall i \in N \\
\quad \sum_{S\subseteq X, j\in S, i\in N} x_{i,S} \leq 1 & \forall j \in X \\
\quad x_{i,S} \in \{0,1\} & \forall S \subseteq X, i \in N
\end{cases}
\tag{3-3}
$$

where N is a set of n agents, X is a set of m discrete and consecutive time slots, S is a subset of A, $x_{i,S}$ is a binary variable with 1 value, if agent i is assigned to bundle S, and $v_i(S)$ is valuation function of agent i for bundle S (Shoham and Leyton-Brown 2009).

It is clear that scheduling problems have higher complexity than assignment problems. One of the ramifications of this higher complexity is that generally, scheduling problems are known as NP-hard problems and cannot be solved in polynomial time (Pinedo 2008), while assignment problems can be solved with an LP in polynomial time.

Similar to assignment problems, a general form of ascending-auction algorithm is proposed to solve the above IP problem. In this algorithm, the time slots are considered as the resources, each with a price, and each agent bids for the time slots that maximize their surplus. The major difference between this method and the ascending-auction algorithm is that the bid increment here is a constant value, while previously the bid price was calculated by the bid increment. The convergence of algorithm depends on the amount of the bid increment. Moreover, it does not always converge to an optimum solution (Shoham and Leyton-Brown

2009; Bertsekas 1992; Wellman et al. 2001). The following factory scheduling example is a simple typical example to illustrate the algorithm.

3.4.1 Illustrative Example: Fabrication Scheduling Problem

Consider a busy day in a factory with only one work station that can work in one shift from 09:00 until 17:00. There are eight time slots which can be allocated to the production of a customer order. There are four customers who will pay a certain amount if their job is completed by the deadline as shown in Table 3-3. Assume that the minimum price that the factory can accept for each time slot is $3 per hour, which is called the reserve price (Shoham and Leyton-Brown 2009).

Table 3-3 Fabrication scheduling example

Job	**Duration (hr)**	**Deadline**	**Worth ($)**
1	2	13:00	10
2	2	12:00	16
3	1	12:00	6
4	4	17:00	14.5

The late start and late finish, the float, the worth, and the duration of the tasks are summarized in Figure 3-1. The total floats of the tasks, which are the amount of time a task can delay the project completion date, are shown by the dotted line in this figure.

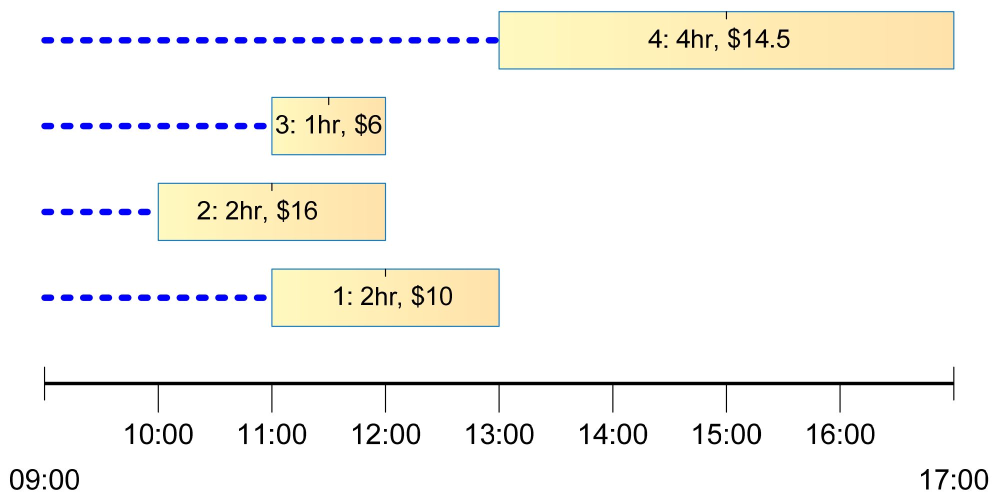


Figure 3-1 Floats and deadlines in fabrication scheduling example

Implementing the generalized ascending-auction algorithm with an increment of $0.25 will lead to the optimum solution in this problem (Table 3-4). However, the algorithm does not converge to the competitive equilibrium if the increment is changed to $1. It is also possible that this algorithm does not converge to the optimum solution regardless of the amount of the increment (Shoham and Leyton-Brown 2009). The optimum solution of this problem, solved by a generalized ascending-auction algorithm, is shown in Figure 3-2.

Table 3-4 Solving factory scheduling with generalize ascending-auction algorithm

round	bidder	Slots bid on	F_1	F_2	F_3	F_4	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8
0	1	9,10	9,10	Nothing	Nothing	Nothing	3.25	3.25	3	3	3	3	3	3
1	2	10,11	9	10,11	Nothing	Nothing	3.25	3.5	3.25	3	3	3	3	3
2	3	9	Nothing	10,11	9	Nothing	3.5	3.5	3.25	3	3	3	3	3
...	...	...	...	...	...	...								
24	1	Nothing	11,12	9,10	Nothing	13,14,15,16	6.25	6.25	6.25	3.25	3.25	3.25	3.25	3.25

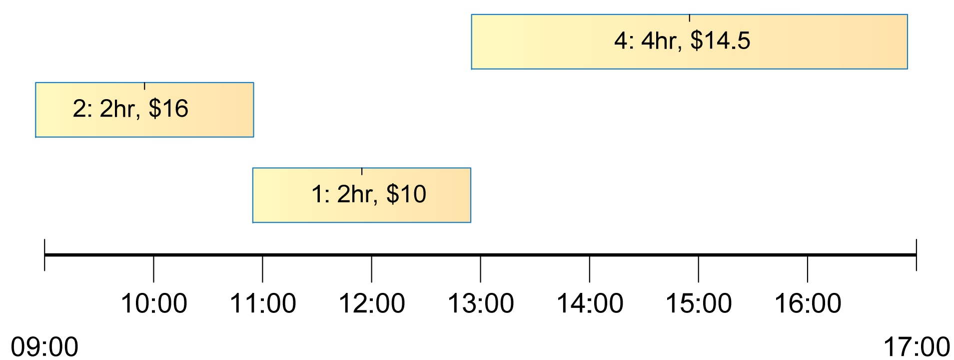


Figure 3-2 Optimum schedule for factory example

3.4.2 Deficiency of the Discretization Approach for Scheduling Problems

Imposing a discretization of the time line into finite time slots and mapping the time slots into indivisible distinct resources works well for small scale problems such as the above-mentioned example. However, this approach does not perform efficiently when facing a large time window in actual industrial or construction problems; for example, a project with 10 resources in a "week" time window generates more than one million time slots. In fact, the number of bids in a combinatorial auction is exponentially dependent on number of resources, which worsens the situation (Wang et al. 2007b). To mitigate this issue, Wang et al. (2007) proposed an expressive bidding language that allows bidding for a set of tasks under release time and deadline constraints. However, this approach brings up new challenges of validating the feasibility of solutions and handling multi-attribute WDP. In the next section, a novel approach is proposed to solve scheduling problems in large-scale construction problems.

3.5 Proposed Simulation Based Auction Protocol (SBAP) for Resource Scheduling Problems

In the traditional approach of simulation-based resource allocation, resources are distributed between different entities on a First-In-First-Out (FIFO) basis. When two entities request resources simultaneously without any priority, available resources are allocated to one of them either arbitrarily or based on the first request in the simulation language. However, capturing shared resources can be prioritized in most simulation-based approaches, meaning that once a resource becomes available, it can be allocated to an activity (i.e., an agent or entity) with respect to the priority dispatching rule (Mohamed et al. 2007). For example, Figure 3-3 shows that the first activity with higher priority (pr = 10) has captured the available resource, while both activities require the resource simultaneously. Priority plays the same role as the agent's preference in MARA terminology. In MARA, agents declare their priority (i.e., bidding price) for different combinations of resources. The auctioneer allocates the sources to maximize the total welfare in the society.

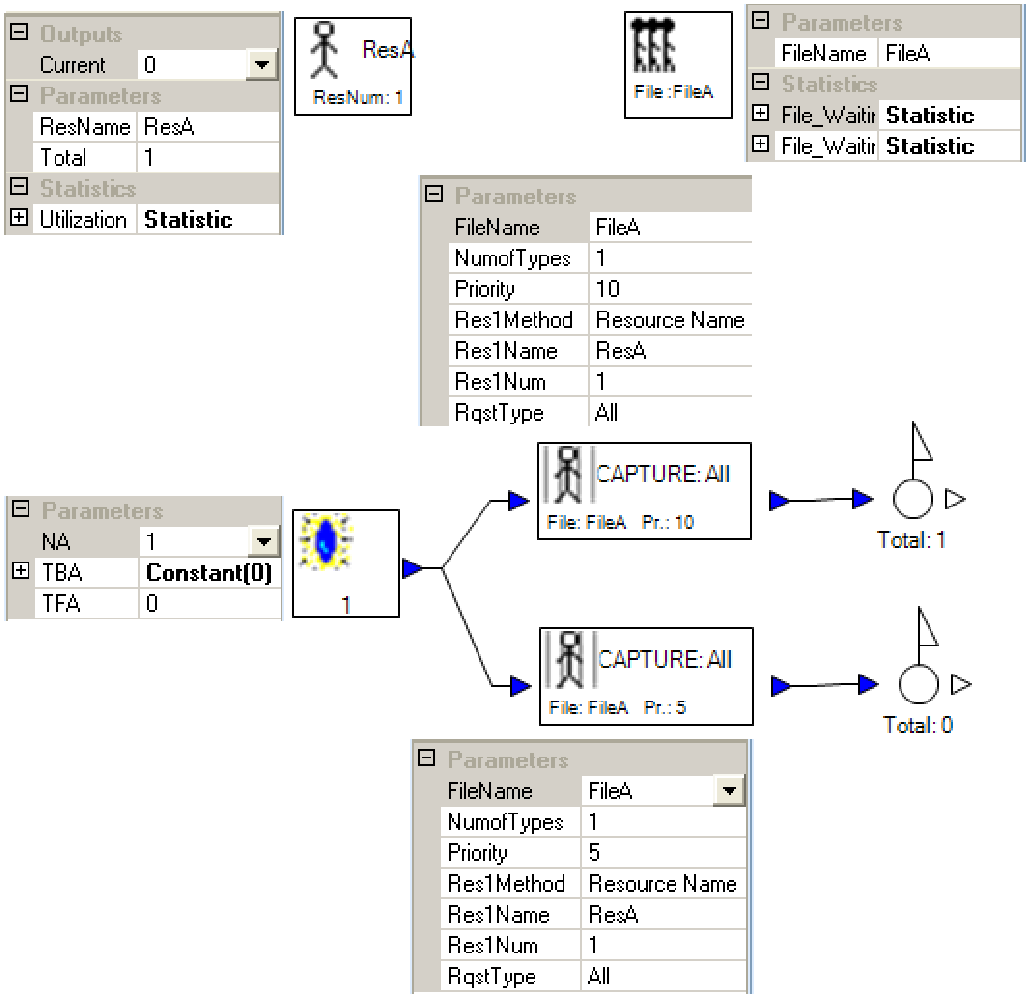


Figure 3-3 Capturing shared resources with priority

Although auction protocols provide a flexible framework to solve resource assignment problems, there is no such current framework to solve resource scheduling problems. This chapter builds on the previous research and proposes the Simulation Based Auction Protocol (SBAP) to solve resource scheduling problems. SBAP's underlying idea is based on "persistence of vision," the phenomena where a series of slightly different images shown in rapid succession produce the perception of motion (Thompson and Bordwell 2010). A motion picture, which is composed of individual frames shown in rapid succession, appears as a moving image because of persistence of vision (Thompson and Bordwell 2010). Similarly, a scheduling problem is a collection of assignment problems, if the time element is considered in the bid price of the agents. An overview of SBAP is shown in Figure 3-4.

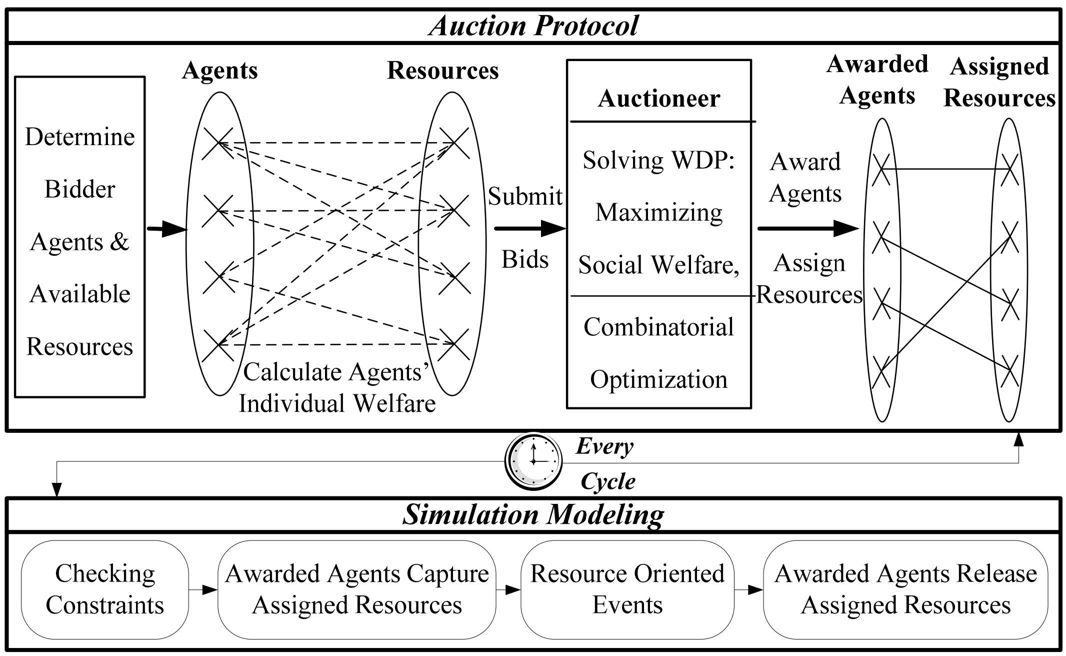


Figure 3-4 Components of the SBAP

The SBAP framework solves scheduling problems by holding a number of auctions on a regular basis with d_t time steps. In each auction, agents or activities (a_1, a_2, a_3, …, a_k), which are located at a time window before the time of the current auction plus ΔT, bid for available resources with a competitive cost or profit as demonstrated in Figure 3-5. The values of d_t and ΔT depend on the application.

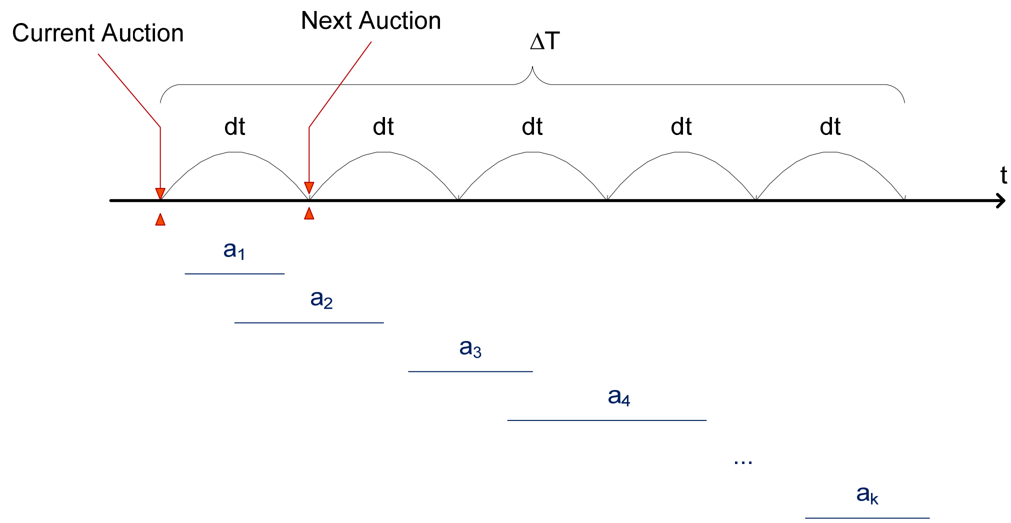


Figure 3-5 The proposed SBAP approach to solve resource scheduling problems

The bidding price of the agents for different combinations of resources depends on many factors including the availability period of the resources, the early start and the late finish (deadline) for the activities, worth (or cost) of the activities, and the float (or slack) of the activities. For example, if the agent passes its deadline, it may either not bid for resources or still bid for the resources but with a higher cost due to penalty factors. In practical large scale problems, a discrete event simulation model is employed to track the availability of resources, capture and release the resources, schedule the upcoming activities and advance the time. The flowchart of this methodology is summarized below. The UML activity diagram of this methodology is shown in Figure 3-6.

1. *Period of time (i.e., week) =1*

2. *Consider agents arriving in the time window and satisfy the constraints:*

   A=$\{a_1,a_2,a_3,...,a_n\}$

3. *Consider available resources in the next period of time:* *R*=$\{r_1,r_2,r_3,...,r_m\}$
4. *Find bidding cost for different combinations of resources*
5. *Solve Winner Determination Problem (WDP)*
6. *For each winning agent:*
   - 6.1. *If the winning agent captures the assigned resource by the next period of time*
     - 6.1.1. *Allocate the resource to the winning agent*
     - 6.1.2. *Remove the agent from list of arrived agents*
     - 6.1.3. *Use the discrete event simulation to capture and release the assigned resources and perform the support activities*
7. *If the time has not reached the end of project's duration*
   - 7.1. *Bidding Cycle = Bidding Cycle + 1*
   - 7.2. *Go to Step 2*

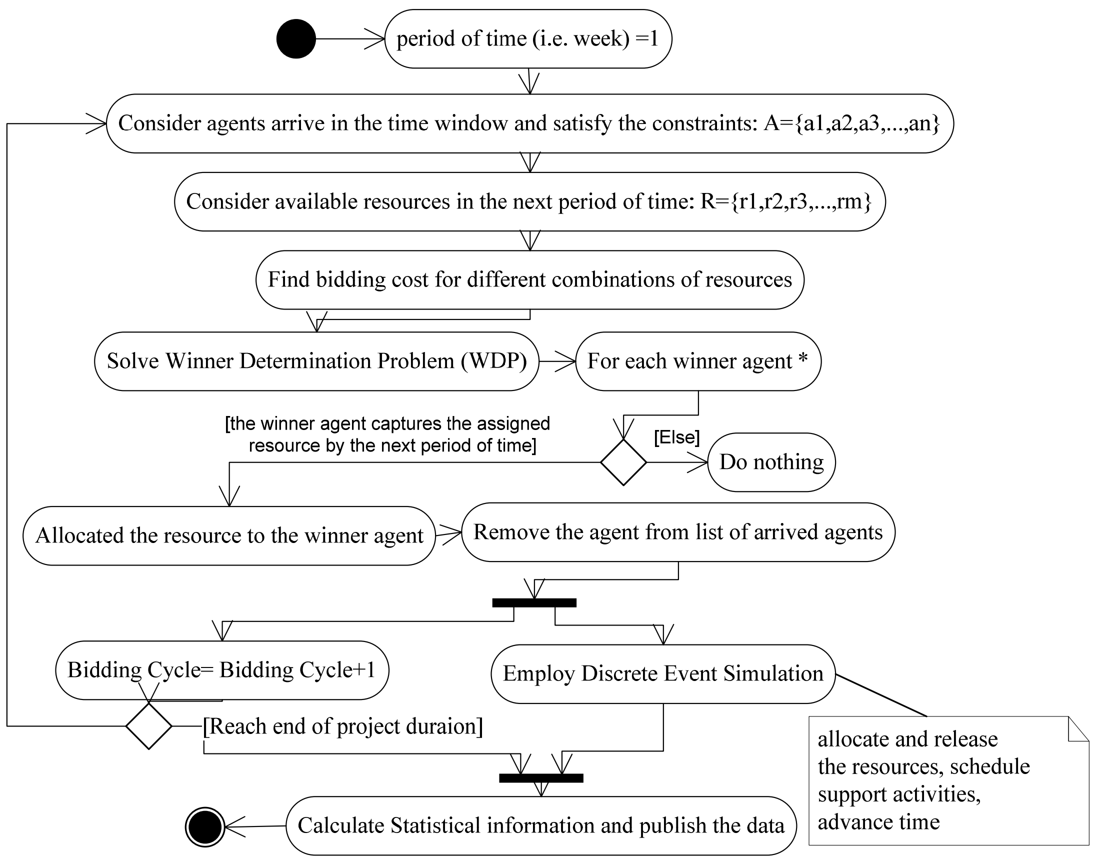


Figure 3-6 UML activity diagram of the SBAP

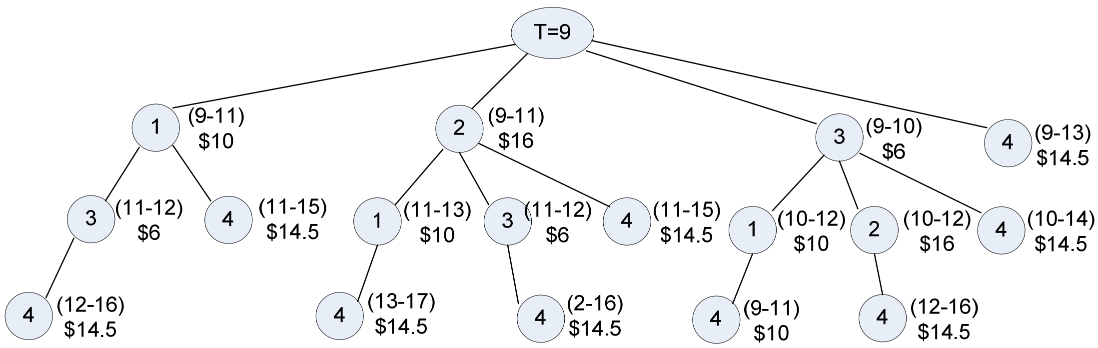
The SBAP is a very flexible approach for large scale practical problems. Each practical problem usually has its own constraints, which can be embedded into the framework. The critical issue in this methodology is to find bidding prices so that they represent the problem constraints (e.g., the early start of activities), worth and float of the activities, etc. Other constraints can be handled through the simulation model before or after the bidding stage. This approach is illustrated by the previous fabrication scheduling problem.

Because this problem takes place over an 8 hour period, ΔT and d_t are assumed to be 8 hours and 1 hour, respectively. It means that the auctions are held every hour, starting from 09:00. The four bidding agents, corresponding to the four tasks, compete over the only available resource (i.e., the work station). This problem is solved using the proposed framework based on two different approaches for calculation of the bid price as follows.

3.5.1 Bid Price Calculation using Search Algorithm

In this approach, the bidding price of agents is calculated by searching through various schedules in the time window preceded by the agent's task. Then, the schedule maximizing the revenue is selected to calculate the bidding price. For example, the action tree shown in Figure 3-7(a) demonstrates different possible tasks that may be scheduled after selection of the first agent's task. The bid price for each agent can be calculated by adding the first task's worth with the maximum revenue of its proceeding tasks.

a. At time = 09:00



b. At time = 11:00

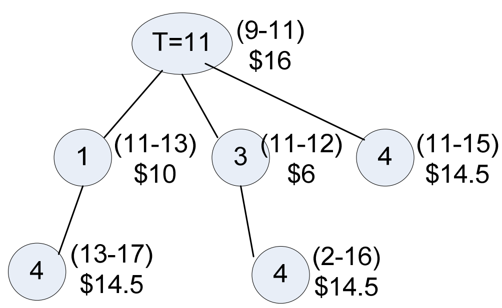


Figure 3-7 Action tree for the fabrication scheduling problem

Table 3-5 demonstrates calculation of the bidding price and allocation of the resource in the auctions. In the first auction at time = 09:00, four agents participate in the auction and the second agent (i.e., task 2) wins the auction with $40.5 bid value. In the second auction at time = 10:00, no agent bids at this auction, because there is no available resource. Therefore, the time is advanced to 11:00 without any specific action. The action tree in the third auction, Figure 3-7(b), is part of the previous tree shown in Figure 3.7(a), because no agent has been added to the time window. Therefore, agent 1 wins the auction with a $24.5 bid value. Similar to time = 10:00, no auction will be held at time = 12:00. In the last auction at time = 13:00, only agent 4 can participate in the auction and receive the resource.

Table 3-5 Calculation of bidding prices using the search algorithm

Auction's Time	Available Resource	P_1	P_2	P_3	P_4	Winning Agent	Duration (hr)
9:00 AM	1	10 + max (6+14.5, 14.5) = $30.5	16 + max (10+14.5, 6+14.5, 14.5) = $40.5	6 + max (10+14.5, 16+14.5, 14.5) = $36.5	$14.5	2	2
10:00 AM	0	---	---	---	---	---	---
11:00 AM	1	10 + 14.5 = $24.5	---	6 + 14.5 = $20.5	$14.5	1	2
12:00 AM	0	---	---	---	---	---	---
13:00 AM	1	---	---	---	$14.5	4	4

This approach results in the same optimum schedule found from the generalized ascending-auction algorithm (Figure 3-2). This algorithm is expected to lead to the exact or near-exact solution in the most cases. However, it grows exponentially when the number of agents and resources in the time window are increased. Therefore, it is not practical in large scale problems, and is simply used to demonstrate how the best bidding price can be calculated using different techniques. Another heuristic algorithm for this fabrication scheduling problem is proposed in the next section.

3.5.2 Bid Price Calculation using an Heuristic Algorithm

In this method, the bidding cost is estimated by considering different parameters that affect the decision making process in the real world. For example, in the fabrication scheduling problem, the first important parameter is worth (or cost) of the activities, which represent the unit profit (or unit cost) and duration of activities. The other important parameter that should affect the bidding price is the total float of activities. Activities with less float have less flexibility and are more urgent. Therefore, the bidding price of agents in each auction can be calculated by Equation (3-4):

$$P_i = \frac{Worth_i}{1+Float_i} \qquad (3\text{-}4)$$

In this example, if a job is completed after its deadline, it is worth nothing. It is assumed that no agent will participate in the auction if it passes its deadline (float ≥ 0). In Equation (3-4), a value of one is added to the float to avoid division by zero. Implementing this equation leads to the allocation procedure demonstrated in Table 3-6.

Table 3-6 Calculation of bidding prices using the heuristic algorithm

Auction's Time	Available Resource	P_1	P_2	P_3	P_4	Winning Agent	Duration (hr)
9:00 AM	1	10/(1+2) =\$3.3	16/(1+1) =\$8	6/(1+2) =\$2	14.5/(1+4) =\$2.9	2	2
10:00 AM	0	---	---	---	---	---	---
11:00 AM	1	10/(1+0) = \$10	---	6/(1+0) = \$6	14.5/(1+2) = \$4.8	1	2
12:00 AM	0	---	---	---	---	---	---
13:00 AM	1	---	---	---	14.5/(1+0) = \$14.5	4	4

In the first auction at time = 09:00, the second agent wins the first auction with a bidding price of \$8, while there are four agents competing for the resource. As in the previous approach, no auction will be held at time 10:00, because no resource is available. At the third auction at time = 11:00, the first agent with a bidding price of \$10, wins the auction and receives the resource. No auction will be held at time = 12:00. The only bidding agent at time = 13:00 is agent 4, which wins the auction. This approach leads to the same optimum approach shown in Figure 3.6.

However, this approach is a heuristic approach and does not converge to an optimum solution in all cases.

3.6 Summary

This chapter presents a Simulation Based Auction Protocol (SBAP) to solve resource scheduling problems in large-scale construction industry projects. This hybrid framework integrates auction protocols and discrete event simulation modeling. The auction protocol can employ different combinatorial optimization techniques to solve WDP and to allocate resources among the bidding agents.

In this framework, auctions are held on a regular basis, whereby some of the agents bid for different resources or combinations of resources so that their individual welfare is maximized. Agents can represent jobs in a factory scheduling problem or a construction unit (i.e., a module) in modular construction of an industrial plant. The bidding price should be customized according to the nature of the problem by considering the project's influencing factors such as the actual cost, availability of resources, early start and late finish of the activities, worth (or cost), or float of the activities. Then an auctioneer awards the winning agents and assigns the available resources to the agents to maximize the social welfare or to minimize the total cost based on a combinatorial optimization. As it is explained in the next two chapters, a simulation model would be very useful in large scale practical problems to track the availability of resources, capture and release the resources, and schedule the upcoming activities.

The SBAP is described in a typical example of fabrication scheduling in this chapter. Two different approaches are presented to calculate the bidding price and the results are validated by having the optimum solution of the problem. The SBAP will be implemented in two large-scale construction case studies in the next two chapters.

CHAPTER 4. Case Study 1: Application of the SBAP for Modular Construction at the Assembly Yard[4]

4.1 Introduction

In the previous chapter, a Simulation Based Auction Protocol (SBAP) was introduced to solve resource scheduling problems in large-scale construction projects. This framework allows the integration of the auction protocol and discrete event simulation modeling. The auction protocol can employ different combinatorial optimization algorithms to allocate resources among bidding agents. The SBAP in the previous chapter was described in a typical example of fabrication scheduling. In this chapter, the SBAP is implemented in a large-scale construction problem of module assembly yard. This problem involves developing a realistic schedule for a module assembly yard that satisfies the project constraints and uncertainties. Scheduling the module assembly yard involves a number of uncertain factors and resource constraints in the yard as well as the fabrication shop. These factors pose a challenge for the scheduler to

---

[4] Parts of this chapter are published at (Taghaddos et al. 2008a; Taghaddos et al. 2008b; Taghaddos et al. 2009)

optimize the use of available resources (e.g., space, crew) and to meet the project's delivery deadlines (Mohamed et al. 2007). This chapter demonstrates that the proposed SBAP framework is very flexible and practical for such large-scale or multi-project construction environments. The SBAP provides a suitable framework to experiment with different affecting parameters and to enhance the decision making process.

4.2 Scheduling a Module Assembly Yard

Modular construction is common practice for building industrial plants in the Alberta oil sands region. Modularization not only minimizes time and the cost to construct on site in northern Alberta's harsh weather conditions, but also improves the safety and quality of the project (Schimmoller 1998; Burke and Miller 1998; Maru and Kawahata 2002). In construction, a "module" refers to a pre-constructed unit that can easily be inter-connected to make a structure (e.g., refineries and oil-processing plants, buildings). A module is usually made of pre-assembled components such as structural steel frames, racks of pipes, cables, equipment, or a combination of miscellaneous components. Its dimension is designed to fit in a transporter for transfer to the construction site (Mohamed et al. 2007; Borrego 2004). In fact, every module represents a unique construction project with its own internal design and components. Pipe spool modules are usually assembled off-site in a yard near the spool fabrication shop. This yard is known as the module assembly yard.

Figure 4-1 displays the layout of the module assembly yard of PCL Industrial Management Inc., located in Nisku, Alberta. A typical module assembly yard is divided into a number of areas, which are called "lots." Each lot is composed of a number of rows, which are called "bays." Generally, 2 to 5 modules can be placed in a bay depending on the size of the modules and the bay.

Figure 4-1 PCL module assembly yard in Nisku, Alberta

The assembly process (e.g., structural steel erection, equipment installation, electrical work, heat-tracing, insulation, fireproofing, and instrumentation) can begin once the required components are prepared by the spool fabrication shop and other supply changes (Taghaddos et al. 2008a; Mohamed et al. 2007). To assemble a module, first it should be placed in a suitable space in the yard depending on its type, length, early start date (representing the time that all the components are available), estimated duration of the assembly process, and shipping date (Mohamed et al. 2007). The assembly process consists of a number of activities with a range of durations. These activities precede each other with certain lags (Start-to-Start (SS) relationships) based on historical information. For instance, the piping activity cannot be started before 20 days after the start of structural steel activity. In other words, the minimum lag between the piping and structural steel activities is 20 days. Once all the required activities are completed, the space in front of the module in the bay is empty, and a transporter is available in the yard, the module can be shipped out to the construction site (Taghaddos et al. 2009).

Scheduling the module assembly yard is a multi-project resource-constrained scheduling problem. A number of different types of resources are involved in the module assembly yard, including the space, skilled crews (e.g., structural steel crew, piping crew), and transporters. These resources are constrained in several ways: 1) Limited space is available in the assembly yard. This space is very valuable is a busy season of the yard. 2) The maximum number of crews for different tasks as well as the rate of hiring skilled crews (ramp-up) is limited. For example, the ramp-up (i.e., the slope of the manpower loading curve) cannot exceed 10 crews in a week. 3) The maximum number of transporters (number of shipments per day) is also limited.

Aside from the resource limitations, some other constraints should be considered in the schedule. Each module (i.e., project) should be delivered to the site by a certain date depending on the client request, and a module cannot be started earlier than a certain date based on the capacity of spool fabrication shop and other supply changes. The clients may request to ship the modules to the site in a certain order. A realistic schedule should also meet a number of physical constraints based on the yard layout, and several logical constraints, as determined by the superintendent, including the following:

1) The superintendent may decide to assemble a module in a specific bay, or across a specific set of bays, e.g., {Bay A1, Bay A2, Bay A3, Bay A4, Bay B1, Bay B2, Bay B3, Bay B4}, depending on the module type and the availability of equipment.
2) The superintendent may prefer to place a module in a certain lot(s) based on availability of equipment or crews. For example, he may prefer to place a module preferably in lot A, otherwise in lot B, and so on.
3) Modules can only be shipped out by a transporter once there is empty space in the front of the module in the bay. Otherwise, a crane is required to lift the module and place it in the transporter, which is very costly (Mohamed et al. 2007).

A typical module assembly yard may contain a few hundred modules when it faces a high workload. Scheduling such a multi-project environment with so many constraints involved is very complicated when using network-based scheduling tools (e.g., Primavera). Modeling modules with traditional CPM techniques requires relationships between modules in a bay. Hence, scheduling such a dynamic system with CPM-based techniques is a tedious exercise. For instance, if there are not enough workers to perform an activity on a module that is in front of the bay, the finish time of that activity is delayed. As a result, the modules in the back of the bay are stocked until the module in the front is finished. More importantly, adjusting the schedule because of any changes in the progress of the work or resource availability is very difficult. Finally, the optimum allocation of resources and the resource leveling are also serious challenges in CPM-based approaches (Mohamed et al. 2007).

This chapter presents the development of a simulation-based scheduling system for the module assembly yard. This system is intended to serve not only as a scheduling system, but also as a tracking system. In the other word, the model reads both as-planned information and as-built (i.e., actual) information while the project is progressing. Those modules, which are in the middle of the assembly process at the time of running the model, have some as built and some as planned information. The input data of this system are entered through a database interface, developed in Microsoft Access 2007. This data is updated by the yard's scheduler on a regular basis (e.g., weekly). The discrete event simulation model reads the input data from the database, simulates the overall process in the module assembly yard, and generates various graphical output data including module location and resource utilization. These graphical reports enable the analysis of pertinent decisions.

In the following, first the developed database interface for the module assembly yard is elaborated. Then, a discrete event simulation model of the module assembly yard, which was not built based on the Simulation Based Auction Protocol (SBAP), is presented. However, it is also explained how some components of the SBAP framework had been considered in this simulation model unintentionally. The previous attempts to develop an agent based simulation model of the module assembly yard based on a MARA structure are explained, and finally a discrete event simulation model developed using the SBAP framework is presented. This case study demonstrates how the SBAP enhances the resource allocation procedure in large-scale construction problems.

4. 3 Database Interface

From the user's perspective, the database interface is the entry point for the scheduling process. This user interface has been implemented in Microsoft Access using Visual Basic for Applications (VBA). The Access database includes general information about the simulation, the space allocation parameters, and the assembly parameters of modules (Figure 4-2). The "Set Up" menu allows the user to define some general information about the simulation parameters, projects, scenarios, layout (e.g., number of bays, their size and location), and classification of modules.

Figure 4-2 shows the input data regarding the "project parameters". These parameters include the list of the assembly activities, the range of crews in the yards, and some general information for the simulation model and space allocation. In this menu, "ramp up" determines the maximum amount of people that the company can hire to reach the maximum number of resources.

Figure 4-3 illustrates the input data to define different scenarios for the assembly process. Each scenario can be either retrieved from past historical data and modified for the current project, or defined from scratch. A scenario defines applicable activities for the module assembly, range (i.e., minimum, maximum, average) of duration and manpower to perform activities, the overlaps and the type of calendar for each activity. Each scenario serves as a template for specific types of the modules. However, the information about duration or manpower of the required activities can be overridden in the database. Usually at the early stage of the project, this information is estimated based on the superintendent's experience and historical data. Once the drawings arrive at the site, the estimate from duration and man-hours is revised based on more accurate data. The recent version of the simulation-based system has also the capability to level the manpower curve by checking the "resource level" box in Figure 4-2. This feature is crucial in the schedule to reduce the fluctuation of manpower curve and to make it smooth.

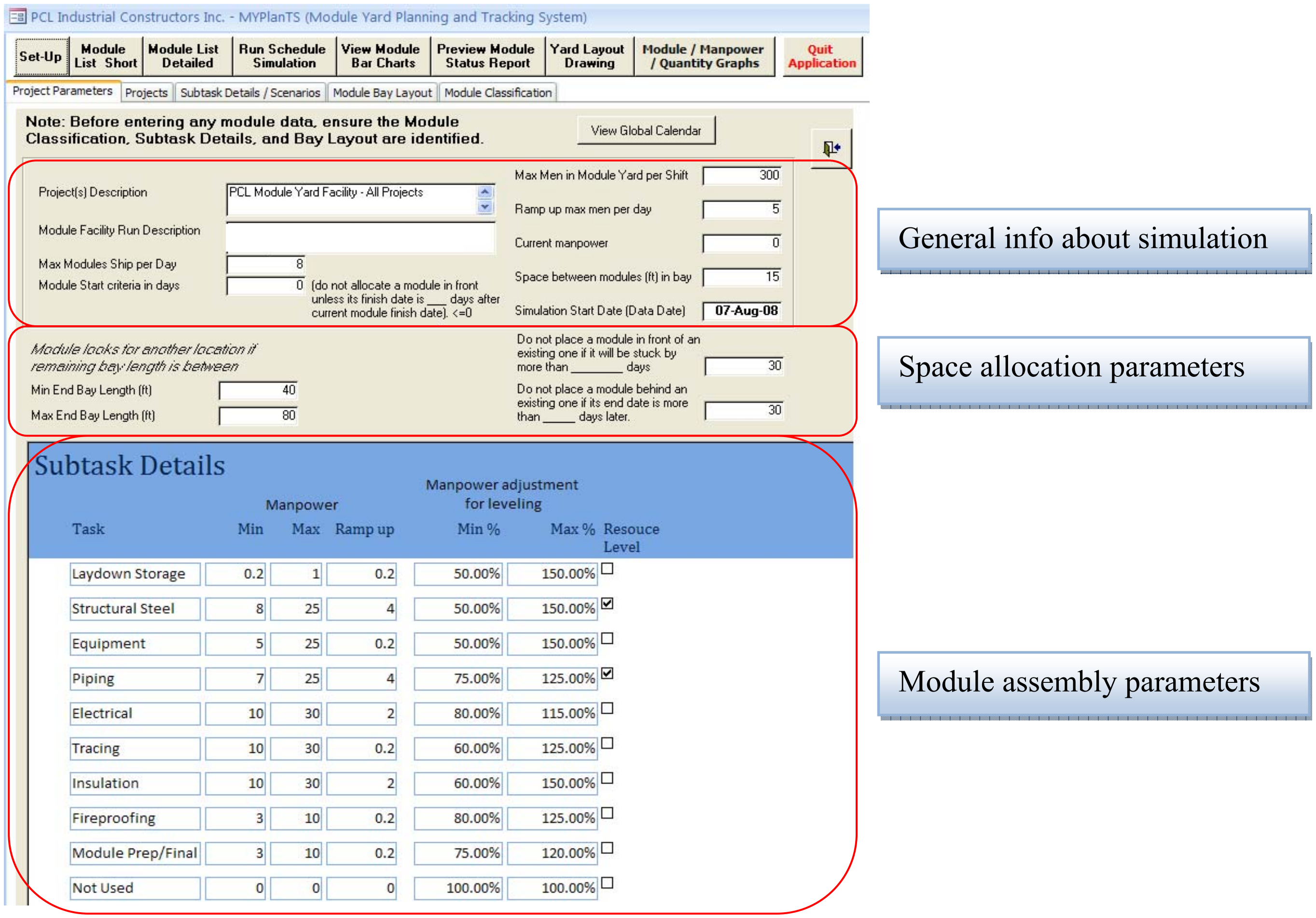
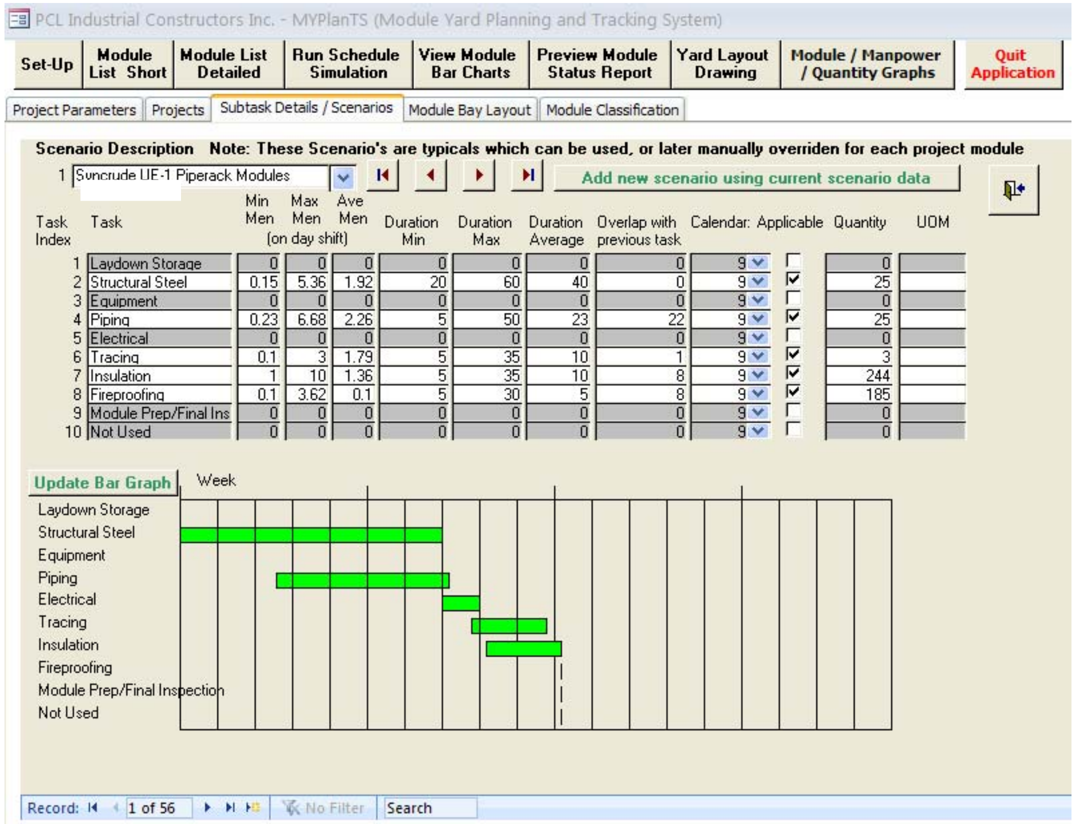


Figure 4-2 Input data regarding the project parameters in the developed planning and tracking system



Task Index	Task	Min Men (on day shift)	Max Men (on day shift)	Ave Men (on day shift)	Duration Min	Duration Max	Duration Average	Overlap with previous task	Calendar	Applicable	Quantity	UOM
1	Laydown Storage	0	0	0	0	0	0	0	9	☐	0	
2	Structural Steel	0.15	5.36	1.92	20	60	40	0	9	☑	25	
3	Equipment	0	0	0	0	0	0	0	9	☐	0	
4	Piping	0.23	6.68	2.26	5	50	23	22	9	☑	25	
5	Electrical	0	0	0	0	0	0	0	9	☐	0	
6	Tracing	0.1	3	1.79	5	35	10	1	9	☑	3	
7	Insulation	1	10	1.36	5	35	10	8	9	☑	244	
8	Fireproofing	0.1	3.62	0.1	5	30	5	8	9	☑	185	
9	Module Prep/Final Ins	0	0	0	0	0	0	0	9	☐	0	
10	Not Used	0	0	0	0	0	0	0	9	☐	0	

Figure 4-3 Input data regarding the scenarios

Figure 4-4 shows some general information for the modules including the type, dimensions, scenario, erection order, planned start date, and planned ship date of modules. This system links each module to one of predefined scenarios, defined in the Set Up menu. The user may wish to erect the modules at the site based on a specific order. In this case, the shipment of the modules to the site should follow the same order.

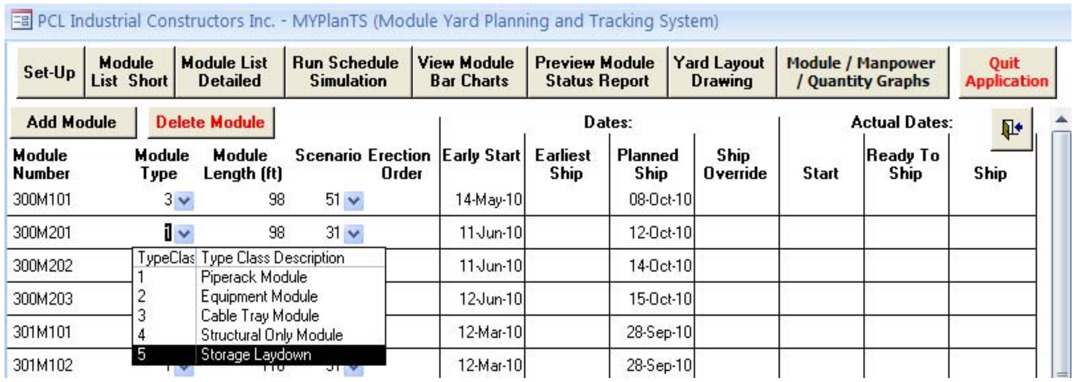


Figure 4-4 Input data the modules

The system has the flexibility to define a number of bays in a group and force the system to place the module in that group. This system also allows for activating the lots only for a specific period of time (Figure 4-5). This feature is beneficial for taking into account the future expansion of the yard or unavailability of a lot in a certain period of time due to various reasons.

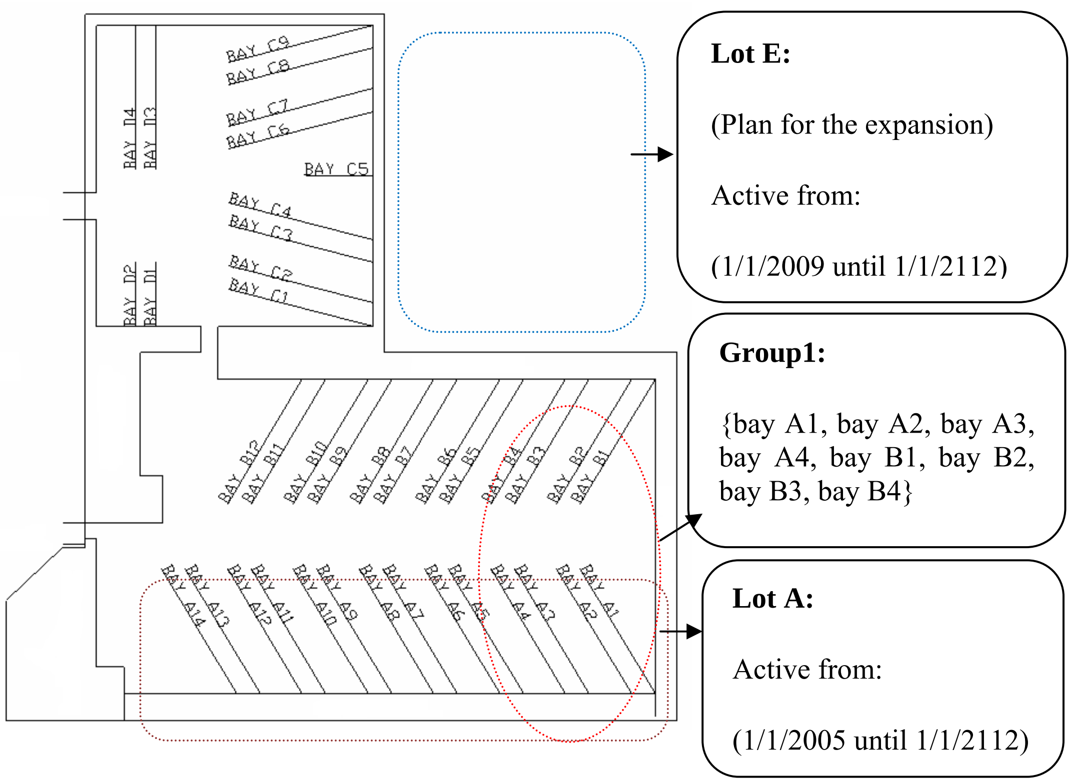


Figure 4-5 Layout of the module yard and an example of bay grouping

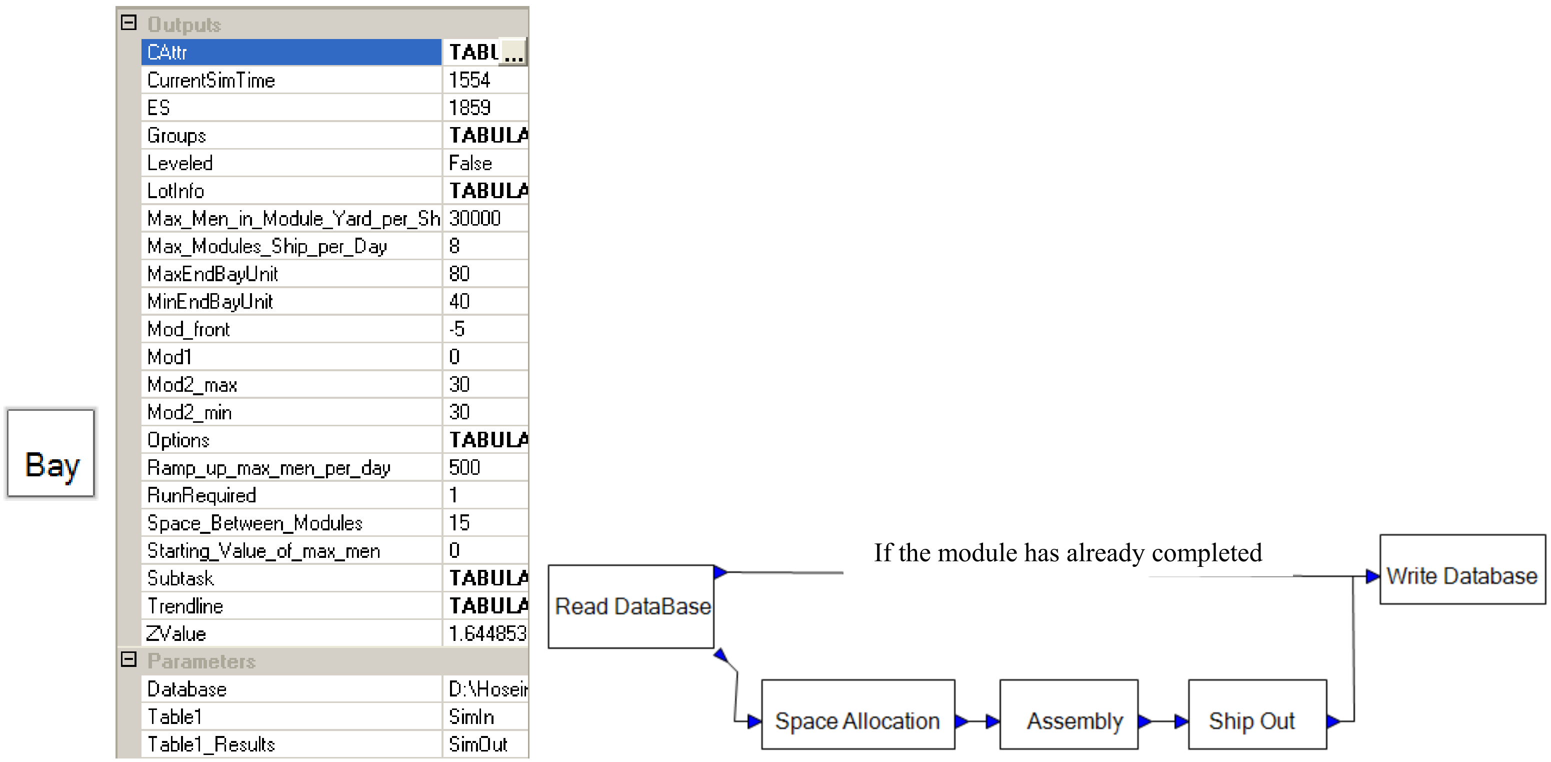
4.4 Scheduling Module Assembly Yard without the SBAP

The first simulation model for the module assembly yard was developed by Davila Borrego (2004) at the University of Alberta. This model employs process interaction elements (a common template) of the discrete event simulation model to allocate space in the bays and different skilled crews to the modules (Borrego 2004). This simulation model is then translated from the legacy version of Simphony to the .NET version of Simphony in 2005 to speed up the simulation model. Although the previous simulation models were useful for planning the module assembly yard of PCL Industrial Management Inc., the company required some more features and enhancement in scheduling the resources to make use of the model in practice. The simulation model was further developed and expanded over a period of about four years, and this new simulation model is compared with Borrego's simulation model.

4.4.1 Structure of the simulation model

This simulation model represents modules as entities that flow into the system. Attributes of these entities (e.g., early start, size, priority, duration and manpower of activities) are retrieved from the database. The model also represents space in the yard, different types of crews, and transporters as resources that should be allocated to different entities. This simulation model employs a priority dispatching rule so that a module with less float captures the required resource with higher priority. Therefore, if two modules request a resource (e.g., space, crew, transporter) simultaneously, the module with a higher priority will capture the resource first. Priorities of modules which are in-progress (i.e., as-built modules) are also set higher than the priorities of as-planned modules. Similar to Borrego's model, the initial priority of the modules is calculated according to their total float, defined as the planned ship date minus early start date minus duration (Borrego 2004). However, this priority in the new simulation model is updated during the simulation model if there is any delay in the assembly process.

As depicted in Figure 4-6, the developed model in Simphony.NET has two hierarchical levels. The parent element (Figure 4-6-a) reads the required data from the Access database and makes the data accessible for all the child elements. In the child level (Figure 4-6-b), the simulation model assigns resources (i.e., space, skilled crew, transporters) to different modules, while satisfying the available constraints. In the child level, first the model generates an entity with an initial priority and other attributes corresponding to every module at the early start time (or actual start time) of each module. The initial priority of the modules is calculated according to their total float, which is planned ship date minus early start date minus duration. If the module is already completed, the corresponding entity is sent directly to the last element to write the results into the database. Otherwise, the entity is sent to the ”Space Allocation,” “Assembly,” and “Release” elements, consecutively. In space allocation elements, some heuristic rules are employed to place a module in a suitable space in the yard. In assembly element, the required number of crew has to be captured to perform the activities in a predefined sequence (e.g., with enough overlaps) with a duration. After all the required activities to assemble a module are completed, the module waits until the space in front of the module becomes empty and a transporter becomes available to ship out the module. The UML activity diagram of the simulation model is elaborated in Figures 4-7. Some important components of the simulation model are discussed in the following sections.



a) Parent level   b) Child level

Figure 4-6 Simulation model of the module assembly yard

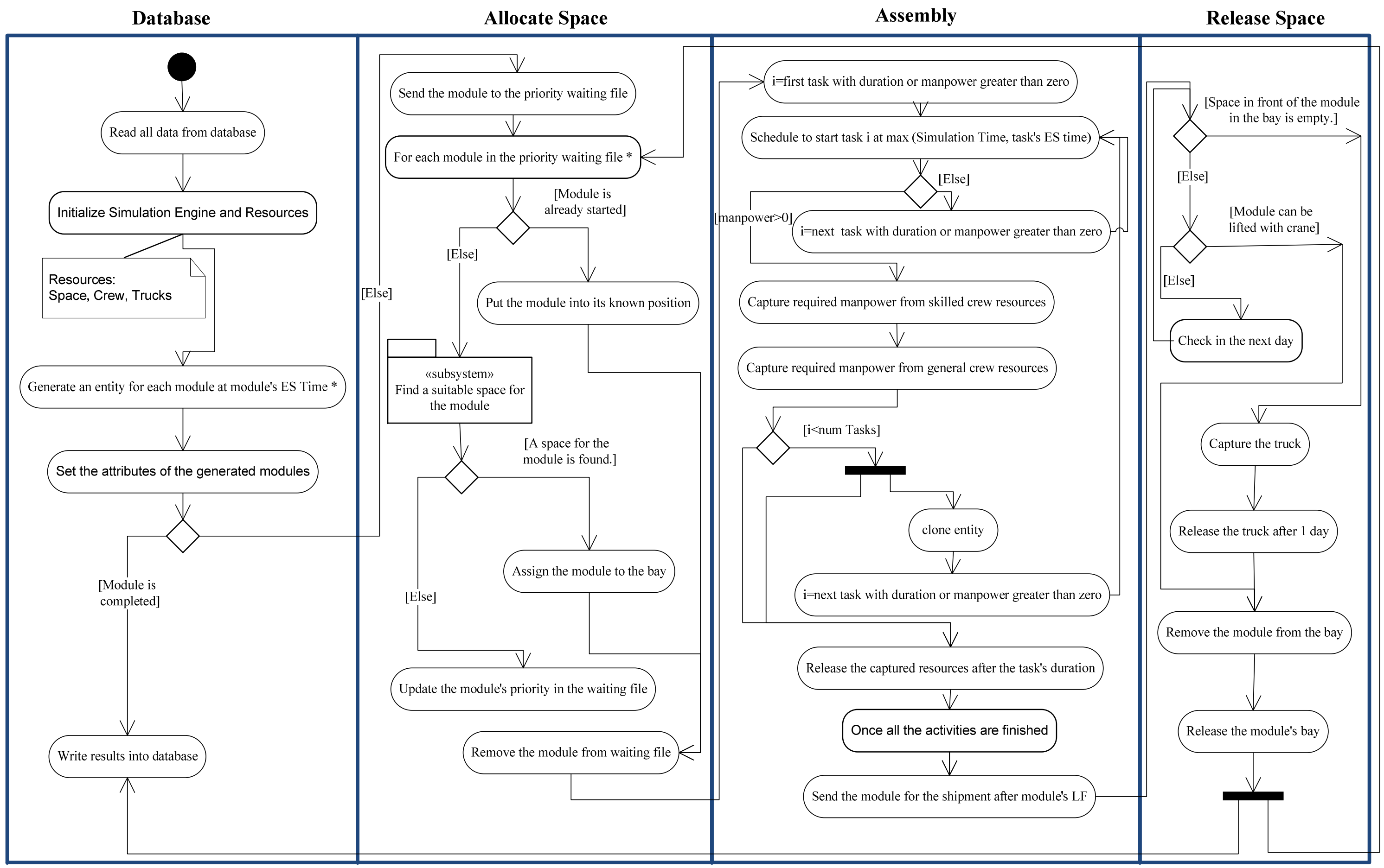


Figure 4-7 The UML activity diagram of the module yard's simulation model

4.4.2 Communicating with the database

The developed simulation model reads the data from a database, produces a schedule for the assembly yard which satisfies the available constraints, and generates some graphical reports. The simulation model runs behind the scenes, and ordinary users only work with the database interface. Once the simulation modeling is also performing resource leveling on manpower curves, this communication is a two pass communication as shown in Figure 4-8.

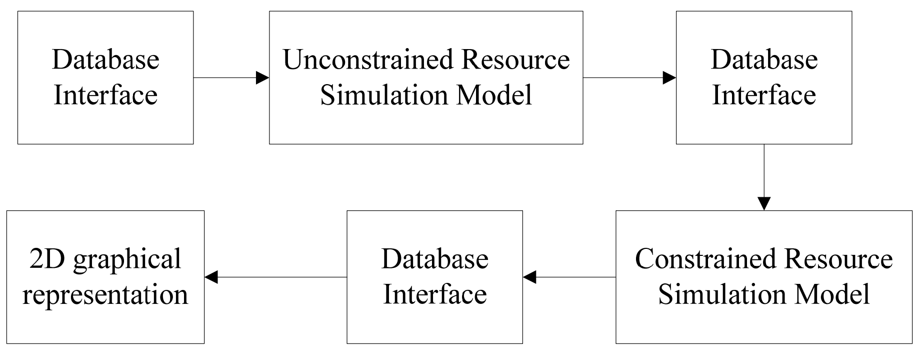


Figure 4-8 Two pass simulation-based approach for resource leveling

4.4.3 Module's preferences

In a space allocation procedure, it is assumed that a module selects a bay in a lot from the front of a lot to the back (i.e., starting from bay #1 to bay #n). Borrego's model assumes that each type of module prefers to be located in a certain lot in the yard. In an industrial project, modules can be categorized in into cable tray, equipment, miscellaneous, pipe rack, and structural modules (Figure 4-9). In this case, pipe rack modules may be preferably routed to lot A, or equipment modules may be preferably routed to lot B (Borrego 2004).

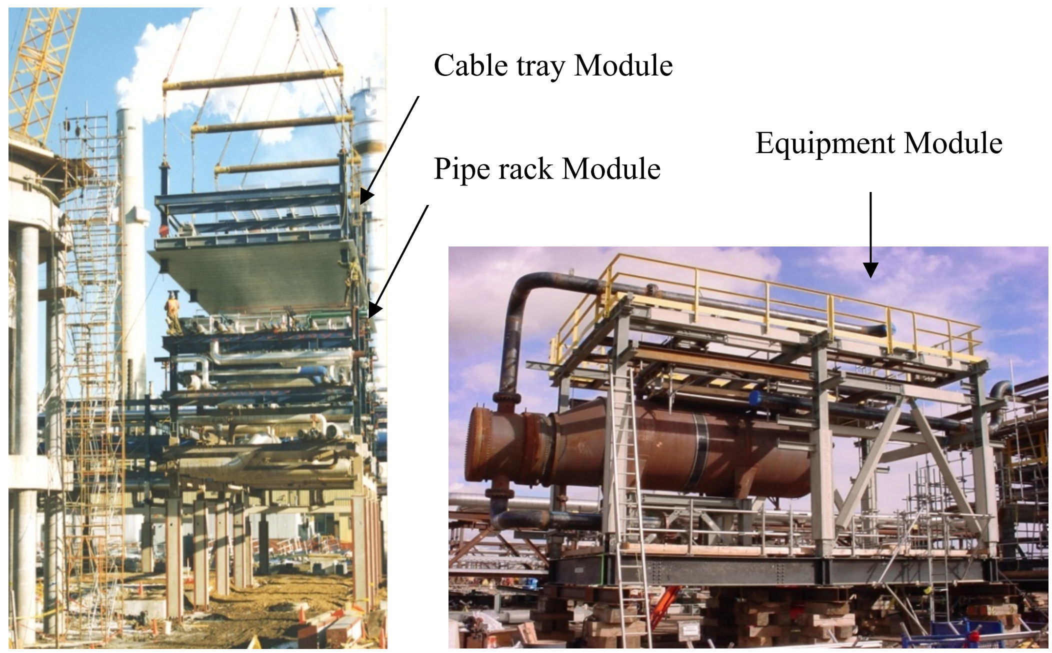


Figure 4-9 Different types of modules

Borrego's simulation model associates each type of module with a certain lot in the yard. If no space in that lot was available, the module had to wait until a free space in the preferred lot became available. The company asked us to substitute the lot preference with an array (Table 4-1); now, by associating an option ID with each module, the simulation model checks different lots in a certain order to find a suitable space for the module. For example, {Lot A, Lot C, Lot B, Lot D, Lot E} indicates the first priority is to find a space in the bays of Lot A, the second priority is to find a space in the bays of Lot B, and so on. If the simulation model finds no space in lot A, the bays in the second lot (B) are searched.

Table 4-1 Option table in the database

Option	Lot 1	Lot 2	Lot 3	Lot 4	Lot 5
1	A	B	C	D	E
2	B	A	C	D	E
3	C	A	B	D	E
4	D	A	B	C	E
5	E	A	B	C	D
6	B	C	D	A	E

We also advanced Borrego's simulation model by allowing the superintendant to place specific modules in a certain bays (in one lot or different lots). This decision might be made for several reasons including availability of certain equipment in specific areas. Therefore, a "group table" was designed in the new model to specify the bays in a group (Table 4-2). This group of bays (group 1) is shown in Figure 4-5. This feature allows to the scheduler to associate a module with a certain group ID in order to force the simulation model to allocate the module only to the bays of the specified group.

Table 4-2 Group table in the database

Bay	Size	Group
A01	290	1
A02	290	1
A03	290	1
A04	290	1
A05	290	
…		
B01	290	1
B02	290	1
B03	290	1
B04	290	1
B05	290	
…		

4.4.4 Space allocation

To allocate the space in the bay to different modules, Borrego considered each bay as a resource with the capacity of a certain virtual space unit. For example, he assumed that longer bays (e.g., bays in lot A) contain 30 virtual units and shorter bays (e.g., bays in lot D) contain 15 virtual units. In this case, a module that requires half of a long bay request 15 virtual space units of that bay (Borrego 2004). This approach does not allow a module to acquire half a short bay or a quarter of a long bay, because 15 divided by 2 does not have an integer value.

In the new simulation model, we considered each bay as a resource by two attributes of space-in-front and space-in-back. The space-in-front is the amount of empty space at the front of a bay and the space-in-back is the amount of empty space at the back of a bay. When the bay is empty, the space-in-front is zero and the space-in-back equals the length of the module. The values of space-in-front and space-in-back attributes change (increase, or decrease) gradually during the simulation run by placing a module in the front or back of a bay. This new approach offers us two benefits: first, the integer virtual unit is replaced with double values that represent the real space in the bay. Secondly, the new simulation model differentiates between space at the front and the back of the bay, while Borrego's simulation model only looks at the space in the back of the bay and places modules in the bays from the back to the front.

A great deal of effort is spent in this simulation model to ensure effective utilization of the available space in the assembly yard. This model employs two heuristic rules, the time and space criteria, to optimize the use of space in the yard. These criteria not only enhance the utilization of space in the yard, but also affect the utilization of other resources and on-time delivery of the modules.

4.4.4.1 Time criterion

It was mentioned that the model of the module assembly yard has a spatial constraint on the unloading module: transporters have access to only one end of the bays. Therefore, a module cannot be moved until all the modules before it are already completed and removed. This makes the problem more complicated. If a module in the front has a delay, the module in the back has to wait until the completion date of the front module. Delay of a module is acceptable for a certain buffer period (e.g., 2 days' delay), whose length is subject to optimization. Borrego considered this constraint in the simulation model by enforcing a time criterion inside the simulation model. This time criterion helps to avoid a module waiting idle at the back of another module more than the buffer period. This time criterion requires tracking the finish time of the module at the back of the bay (End-Back-Module), as well as the finish time of the last modules scheduled in the bay (Finish-Current-Module). The time criterion is simply checking the "estimated" finish time of the modules to be less than the smaller of these two values plus the buffer period as shown in Equation 4-1 (Borrego 2004).

$$Module's\ Finish\ Time \leq$$

$$Min\left(EndBack\text{-}Module,\ Finish\text{-}Current\text{-}Module\right) + Buffer \qquad (4\text{-}1)$$

This criterion avoids placing a module in front of other modules, if it is estimated to finish after the modules in the back of the bay, taking into consideration the buffer period. To take a case in point, consider the example shown in Figure 4-10 with 5 days' acceptable buffer. The estimated finish time of a new module should be less than Min (50, 40) +5.

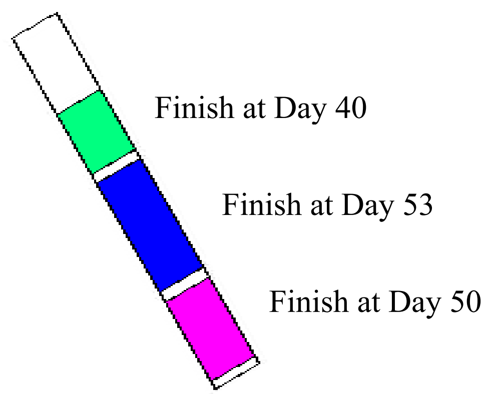


Figure 4-10 Example for space allocation in the yard

However, this time criterion does not always work properly. Figure 4-11 shows a simple example with 5 days' acceptable buffer. The estimated finish time of a new module should be less than Min (50, 44) +5, which is day 49. In this case, if this module finishes on day 49, the module which is estimated to finish on day 40 has to wait 9 days after its completion, which is more than the acceptable buffer.

Therefore, we modified the time criterion for capturing the space in front of the bay. The new time criterion for placing a module in front of a bay dictates the estimated finish time of a module to be less than the minimum finish time of all modules in the bay (Min-Finish-Module) plus the buffer period as shown in Equation 4-2:

$$Module's\ Finish\ Time \ \leq \ Min\text{-}Finish\text{-}Time \ + Buffer \quad (4\text{-}2)$$

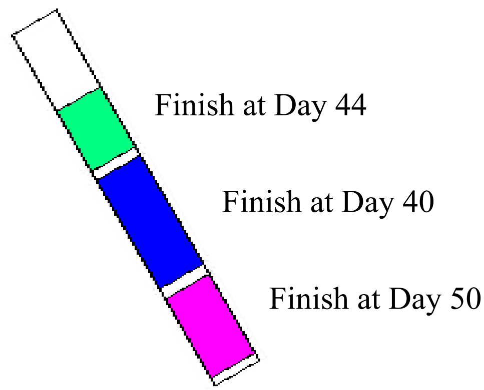


Figure 4-11 The second example for space allocation in the module assembly yard

Similarly for placing a module in back of a bay, the estimated finish time of a module must be more than the maximum finish Time of all modules in the bay (Max-Finish-Module) plus the buffer period as shown in Equation 4-3:

$$Max\text{-}Finish\text{-}Time \ + Buffer \ \leq \ Module's\ Finish\ Time \quad (4\text{-}3)$$

Please note that the estimated finish time of a module is only a rough estimate at the space-allocating phase, because it may change based on resource (e.g., space, manpower, transporter) limitations.

4.4.4.2 Space criterion

Beside the time criterion shown in Equations 4-2 and 4-3, a space criterion is also considered in the new model to utilize the space in the yard further. The space criterion recommends placement of a module in a bay so that remaining space in the bay will be either small enough (e.g., less than 20 feet) or will be large enough to be able to place another module in the remaining space (e.g., more than 80 feet). Therefore, the minimum and maximum limits of module length in a project (Mod_min, Mod_max) have to be defined to optimize space allocation as in the following:

$$\textit{The remaining space in a bay } \boldsymbol{should} \textit{ not be between } (Mod\_min, Mod\_max). \tag{4-4}$$

However, the space criterion is a preferred (not mandatory) criterion. In other words, if no space is found for a module according to the space criterion, this criterion will be ignored. The space and time criteria are two heuristic rules to improve the space utilization and on time delivery in the module assembly yard. The UML activity diagram of space allocation procedure which is a composite element of the simulation UML diagram is shown in Figure 4-12.

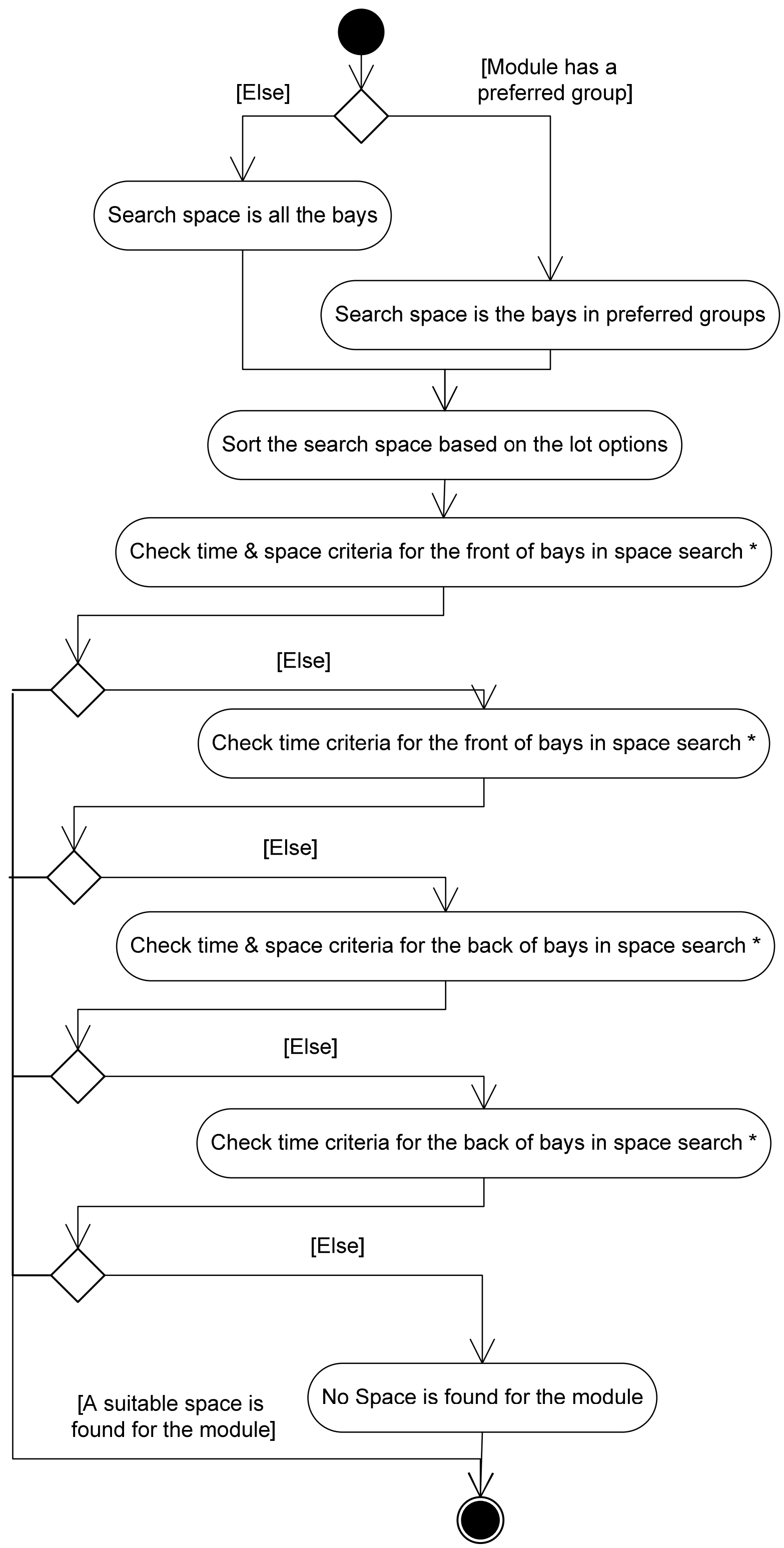


Figure 4-12 Finding a suitable space for a module in the assembly yard

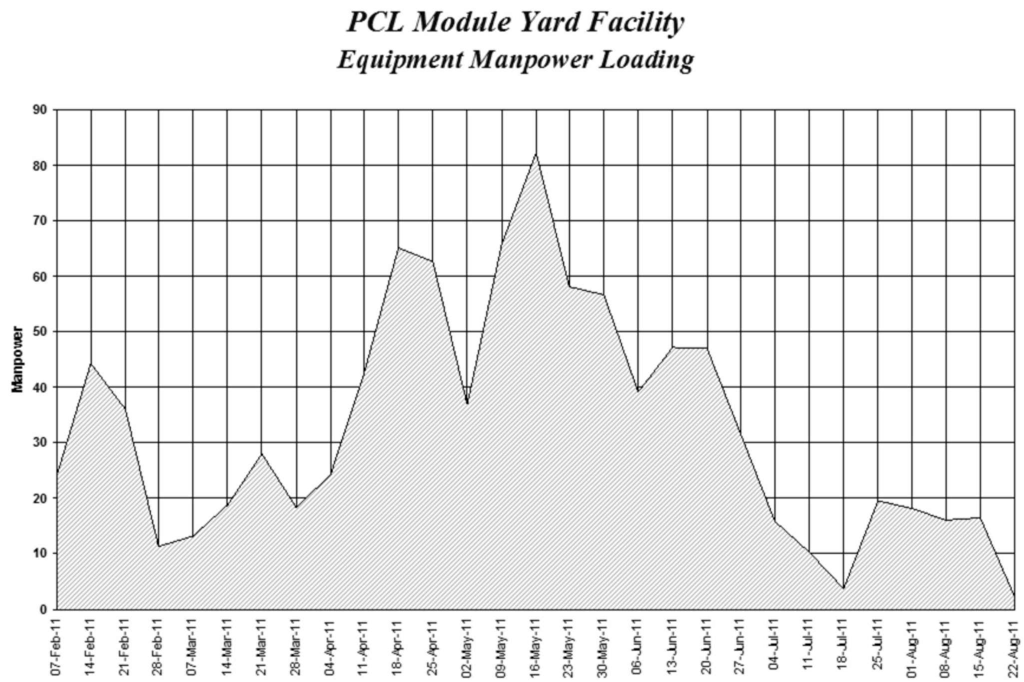
4.4.4.3 Erection criterion

The last important criterion that is considered in some projects is the client's request to ship out the modules in a specific order. In this case, the modules of a project with specific erection order have to be placed in a bay based on ascending erection order. In the other words, a module with a lower erection order must be in front of a module with a higher erection order.

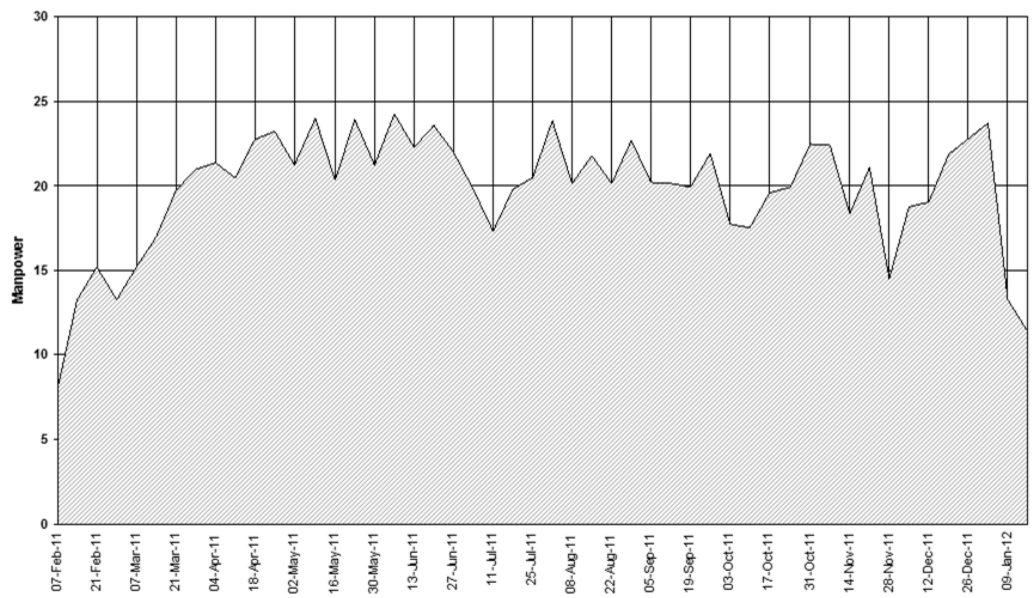
4.4.5 Crew allocation

Allocation of crew in this kind of multi-project environment is a very complicated task. Manpower limitation results in a delay in the start time of some activities and those delayed modules may impact the delivery date of the other modules in the same bay. Similar to the space allocation, distribution of available manpower among different activities is based on the priority dispatching rule. This priority is calculated based on the available float of a module, and is updated during the simulation model once there is a delay in start time of an activity. This approach equates the minimum slack algorithm in the job shop scheduling with a resource constraint. This method is a heuristic planning method that gives priority to the task with the least remaining slack and then updates the slack of all other tasks (Russell and Norvig 2003).

This simulation-based approach results in a logical schedule. However, the fluctuation of resources is not favorable in practice. For example, Figure 4-13 shows the results of equipment manpower loading using the unleveled simulation model. This figure shows that the number of piping crew in the yard varies from 10 to 80 crews in 3 months and then is reduced to 5 crews 3 months later. These variations are not favorable for the industry. We employed a two-pass simulation-based resource leveling approach to level (i.e., smooth out) the manpower curves in the project. The details of the approach are beyond the scope of this thesis; however, the two-pass simulation-based approach is well explained in the paper in Appendix 1 (Taghaddos et al. 2008a).



a) Unleveled manpower loading curve



b) Leveled manpower loading curve

Figure 4-13 Leveling the manpower curve of equipment crew

4.5 Scheduling Module Assembly Yard using the MARA

After developing the simulation model in Simphony.NET, Liu (2009) at the University of Alberta implemented a MARA approach in modeling module assembly yard. He developed the model in an agent based simulation platform called Repast.NET, which provides a library of classes for developing agent based simulation in a Java-based language. Agent based simulation advances the time on a regular basis (i.e., tick) regardless of the events schedule, while discrete event simulation only advances the time when an event is scheduled (Liu 2009).

In this model, Liu (2009) considered modules as agents, and bays (i.e., space) in the yard as resources. Modules can bid for the available space in the bays after their early start date. The preference of a module for a bay, also known as the utility function, is expressed as a function of time factors and type factors. The time factors include updated total float (UFF_i) of the module, and the amount of time that the module blocks the shipment of the last module in the bay ($BlockShipPenalty_{ij}$). The type factors include the module's preference over a certain bay (BP_j). All these factors are taken into account in Equation (4-5) to calculate a module's preference to a bay (Liu 2009; Liu and Mohamed 2008):

$$u_i(B_j) = C + BP_j - 10 \times UFF_i - BlockShipPenalty_{ij} \qquad (4\text{-}5)$$

Where, (4-6)

$u_i(B_j)$ is the utility function (i.e., preference) of the module M_i to the bay B_j

C is a large constant value to ensure the utility function is positive (e.g., 200)

BP_j is a binary value, which is 1 when the preferred bay of the module M_i is the bay B_j

UFF_i is the updated free float of the module M_i, calculated by LF_i- (Current time + Duration $_i$)

$BlockShipPenalty_{ij} = Max[(LF_i – LF_k)\times Penalty_{block}, 0]$

LF_i= Late finish time of the module (M_i)

LF_k= Late finish time of the last module (M_k) in the bay B_j

In Equation (4-5), some factors appear with a positive sign, because increasing those factors increases the value of the utility function. For example, if a module is bidding for space in its preferred bay, the utility function is expected to be more. Some other factors appear with a negative sign, because increasing those factors reduces the value of the utility function. For instance, the longer the amount of time a module blocks the bay, the less the utility function should be. Similarly, a module with less updated total float is more urgent; thus, its utility function should be more. In Equation (4-5), a multiplier of 10 is used for UFF to amplify the effect of this factor. (Liu 2009; Liu and Mohamed 2008)

In this model, the eligible modules, which have passed their ES date, calculate their utility function for suitable bays in every time tick. Once all the eligible modules submit their bids for the applicable bays, an auctioneer distributes the bays among the modules to maximize the social welfare. The utilitarian social welfare is used in this model because all agents are identical from the project point of view (Liu 2009; Liu and Mohamed 2008).

4.6 Scheduling a Module Assembly Yard using the SBAP

The previous MARA model was developed successfully in the agent based simulation platform, where time advances every tick regardless of the events' schedule. The discrete event simulation in the proposed SBAP resembles the agent based simulation in the sense that some events are scheduled on a regular basis to hold auctions during the simulation model. However, the SBAP is more advantageous because the regular events in the SBAP are scheduled only for holding auctions and are independent of other simulation events. In other words, the simulation events can be scheduled in durations with scale of minutes, while the regular auction events are held every day or every week (every dt in general). The discrete event simulation platform also enables embedding of the SBAP within the HLA environment as will be explained in chapter 6. The other advantage of the SBAP framework is that agents in the SBAP can bid for and reserve the resources from ΔT unit of time before their ES date, while approach agents in the MARA can bid for the resources only after the ES date (Figure 3-3). Finally the SBAP approach empowers the modeler to adjust the utility function and the parameters of ΔT and dt to reach the most efficient resource allocation, with the least cost and with on-time delivery.

Aside from the fundamental differences between the MARA and the SBAP frameworks, the modeling approach of the previous simulation model developed by Liu (2009) is different from the presented simulation model in this section. This SBAP-based simulation model is based on the updated simulation model and has the following differences from the previous MARA-based model:

- The amount of time that a module blocks the shipment of the last module in the bay (i.e., $BlockShipPenalty_{ij}$) in the previous MARA-based model is only dependent on the late finish of the last module in the bay, while in this SBAP-based model is a function of estimated finish time (not late finish time) of all the modules in the bay, not only the last module. This estimated finish time of the modules is updated as the simulation model progresses.
- A block penalty factor, called BlockBayPenalty, is also added into the SBAP-based model, when a module sits in the back of a bay and keeps all the space in the front of the bay idle. This situation happens when the difference between the estimated finish time of the module and the maximum estimated finish time of existing modules has a high value.
- Modules in the MARA-based model bid only for the back of bays (similar to Borrego's model), while the modules in the SBAP-based model bids for

both back and front of bays. However, it is preferable to place a module in the front of a bay.

- The preference of a module for a bay in the MARA-based model is a binary preference (i.e., 0 or 1), while in this SBAP-based model it is an integer variable with different degrees. In other words, modules in the MARA-based model only have a preference for the bays in one lot and no preference for the other lots, while modules in the SBAP-based model have preferences for all lots but at different levels.
- In the previous MARA-based model, no attempt was made to minimize the wasted space in the bays, while there is an attempt in this SBAP-based model to minimize the wasted space in the bays.
- In the new SBAP-based model, a Block Priority Penalty (BPP) is added to the utility function, similar to the Block Ship Penalty (BSP). This factor encourages placing the modules with higher priorities in front of the modules with lower priorities. This results in delivery of modules at the front of a bay earlier than the modules at the back of a bay.

Before applying the SBAP to the simulation model explained in section 4.4.2, the developed simulation model in Simphony.NET (version 1.1.3.13) was translated to Simphony.NET 3.5. This new simulation environment has been redesigned completely to be more extensible than the previous version of Simphony.NET Simphony.NET 3.5 provides the convenience of programming in the Visual Studio environment, while the programming environment in previous versions of

Simphony.NET (e.g., version 1.1.3.13) was the internal programming environment inside Simphony. Developing simulation models in Simphony.NET 3.5 is much more convenient and flexible than before.

After translating the simulation model of the module assembly yard from Simphony.NET to Simphony.NET 3.5, the model was modified to follow the SBAP framework. This SBAP framework has simplified the structure and has improved the performance of the simulation model. In the SBAP framework, agents represent modules in the module yard and compete over the resources similar to the model developed by Liu (2009). Regular auctions are held to distribute available resources to the bidding agents and the auctioneer allocates the resources to maximize the social welfare. A discrete event simulation model is also employed to allocate and release the assigned resources and schedule different activities. The main components of this framework, the auction protocol and the discrete event simulation model, are explained below:

4.6.1 Auction Protocol

In this project, the SBAP is employed to facilitate the space allocation procedure, the most challenging resource allocation in the simulation model. In the current simulation model, skilled crews are allocated to modules simply based on the priority, which is calculated as updated total float. It is mentioned in section 3.5 that the priority is a simplified utility function in a resource allocation procedure using the MARA structure. However, the allocation of space requires a more comprehensive utility function to reflect reality.

Auctions are held in the SBAP framework on a regular basis. In each auction, the modules whose bidding time (i.e., early start time) is prior to the current auction time plus the length of Time Window (TW) will bid for the space in front or back of the bays. If a module has a group preference, it only bids for space in the bays in the module's group. Each module can bid for space either in front or in back of the bays with a proposed utility function. However, modules do not bid for space in the bays that do not contain enough space available. The utility function (i.e., individual welfare or bidding price) for each bidding module is calculated as follows:

$$U_i(B_j) = -C_1 \times UTF_i + C_2 \times LP_j - C_3 \times BlockShipPenalty_{ij} - C_4 \times BlockBayPenalty_{ij} - C_5 \times BlockPriorityPenalty_{ij} - C_6 \times WasteUnits_{ij} - C_7 \times CrewCongestion_j + C_8 \times FB_j \quad (4\text{-}7)$$

where

$u_i(B_j)$ is the utility function (i.e., preference) of the module M_i to the bay B_j

UTF_i is updated total float of the module M_i, calculated by LF_i- (Current time + Duration $_i$).

LP_j is an integer value between (1-4) based on the priority of the lot in the option array.

$BlockShipPenalty_{ij}$ is the maximum duration blocking in the bay

$$= \begin{cases} Max[FT_i - Min\ (FT\ of\ the\ modules\ in\ the\ bay), 0), \\ \qquad if\ space\ in\ the\ front\ of\ the\ bay \\ \\ Max[Max\ (FT\ of\ the\ modules\ in\ the\ bay) - FT_i, 0), \\ \qquad if\ space\ in\ the\ back\ of\ the\ bay \end{cases}$$

$BlockPriorityPenalty_{ij}$ is the maximum priority blocking in the bay

$$= \begin{cases} Max[Max\ (pririty\ of\ the\ modules\ in\ the\ bay), 0), \\ \qquad if\ space\ in\ the\ front\ of\ the\ bay \\ \\ Max[Pr_i - Min\ (pririty\ of\ the\ modules\ in\ the\ bay), 0), \\ \qquad if\ space\ in\ the\ back\ of\ the\ bay \end{cases}$$

FT is an estimate of finish time of a module when it is placed in the bay.

$WasteUnits_{ij}$ is the amount of waste unit left in the bay after placing the module.

$CrewCongestion_j$ is the congestion of crew in a bay and its adjacent bays.

FB_j is a “front bay” binary variable, whose value is 1 if the space is in front of the bay j, and zero otherwise.

C_i are multipliers which normalize the effect of different parameters based on their range and priority in the decision making process. For example, the impact of BlockShipPenalty should be more than the impact of BlockBayPenalty. These multipliers also may change according to other factors. For example, if WasteUnits is less than the minimum length of the modules, the multiplier should have a high penalty factor. Otherwise, this multiplier is a lower penalty factor. The amounts of these values are not presented here because of confidentiality concerns for PCL Industrial Management Inc.

Once all the bidding modules submit their bids, the auctioneer allocates available resources (i.e., the front and back of bays) to some of the bidding modules based on a combinatorial optimization. In this example, the greedy algorithm is chosen to award the winning modules so that the utilitarian social welfare is maximized. Although the greedy algorithm is an incomplete heuristic method, it often performs well in practice (Shoham and Leyton-Brown 2009; Klemperer 1999). It allocates resources by adding one bid at a time and never reconsidering a bid after its allocation (Shoham and Leyton-Brown 2009; Klemperer 1999). In this allocation procedure, modules first have to be sorted according to their priority (i.e., UFF in this problem). The allocation procedure then proceeds as explained below:

- Loop over modules:
    - Price = Minimum of acceptable price & IndexRes1 = -1

- Loop over submitted bids of the module
    - If the requested resource is still available

      and bidding price > Price then

        - Price = bidding price
        - IndexRes1 = ID of the bid's resource
- End Loop
- If IndexRes1 > -1 then:
    - Assign the resource with ID = IndexRes1 to the bidding agent.
    - Remove the agent and resource from the list of agents and resources.

- End Loop

The ascending-auction algorithm is also employed to solve the WDP in this assignment problem, as was explained in chapter 3. However, the assignment problem in this WDP is not necessarily symmetric, meaning that the number of agents and resources are not equal. This problem has been solved by adding some dummy resources to the problem and then ignoring the winner agents for the dummy resources.

Once the winner modules are awarded, only those modules that can capture the assigned space prior to the next auction will capture the space. This approach allows some of the bays to remain empty for the important modules that are coming in the near future (after ΔT days). The amount of ΔT can be adjusted to reach the best performance.

4.6.2 Discrete Event Simulation Model

The discrete-event simulation model in this study is similar to the updated simulation model explained in section 4.4.2. The UML activity diagram of the new simulation model based on the SBAP framework is shown in Figure 4-14. The main difference between this simulation model and the previous simulation model without the SBAP, depicted in Figure 4-7, is in the first and second swim lane (i.e., partition) of the UML activity diagram.

In the previous simulation model developed without the SBAP framework, generated modules are sent to “Allocate Space” to find a suitable space. The algorithm for finding a suitable space is a composite element, shown in Figure 4-10. This element includes a number of loops to look at the front and back of available bays by considering the time and space criteria. Once the first suitable space is found for a module, it is sent to the assembly part.

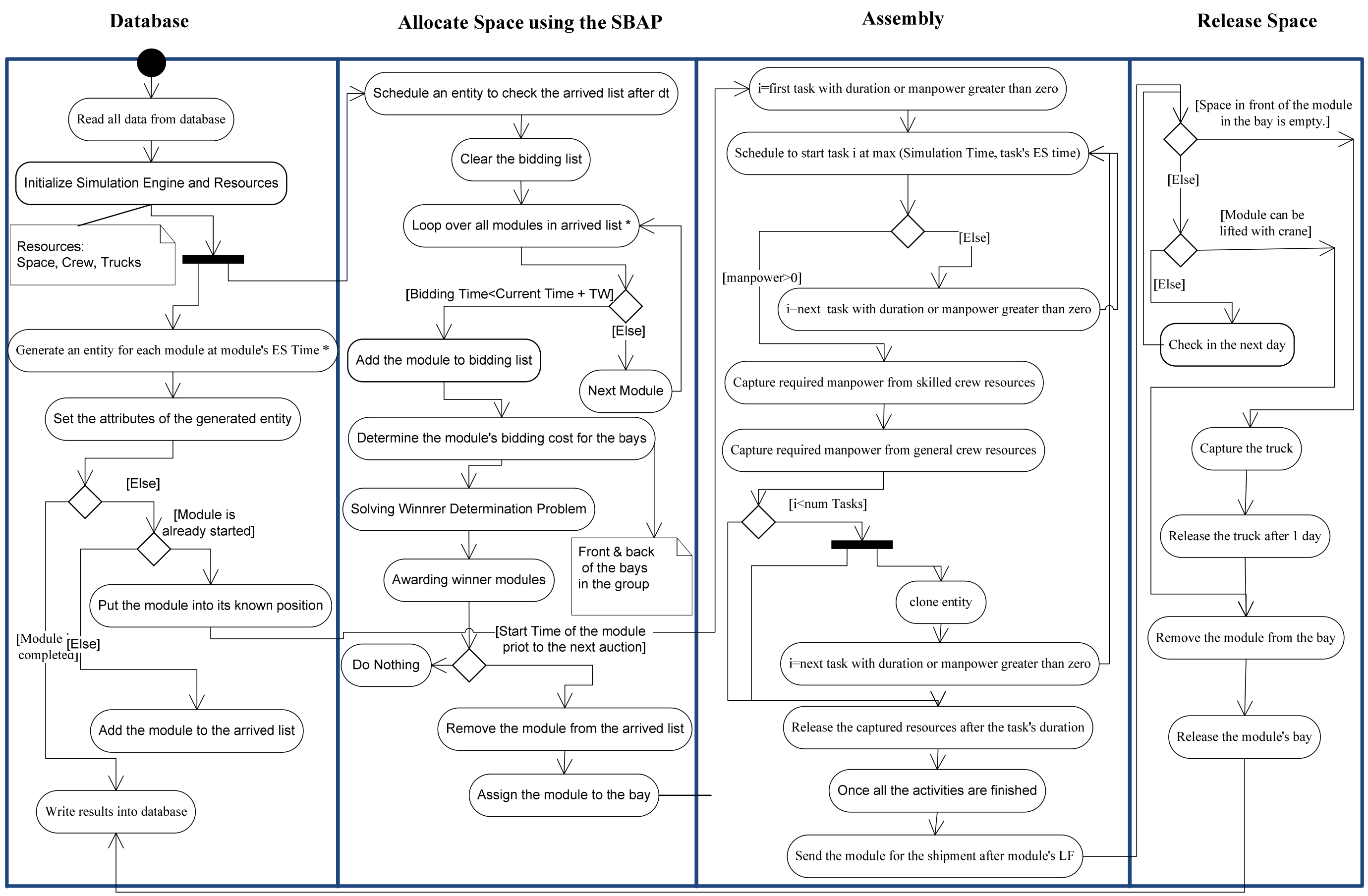


Figure 4-14 UML activity diagram of the module yard's simulation model developed based on the SBAP

In the new simulation model developed based on the SBAP framework, generated modules are sent to an “arrived list.” Modules in this list are checked regularly (every dt) to bid for the resources. Once the bidding time (i.e., early start time) of a module is less than the current simulation time plus the length of the Time Window (TW), it can bid for the front and back space available in the bays in its group. Once the modules in the “bidding list” submit their bids for different resources, the auctioneer solves the Winner Determination Problem (WDP) using a combinatorial optimization algorithm. Those awarded agents whose start date (i.e., early start date) is prior to the time of the next auction (current time + dt), capture the assigned resource. Then the simulation model assigns skilled crews and transporters and releases them, similar to the previous simulation model.

The SBAP not only simplifies and structures space allocation procedures in the simulation model, it also enhances the performance of the entire schedule. The developed system is also efficient in terms of the computational speed. For a large-scale practical problem with 200 modules, each module having 10 activities, the computational process takes approximately 5 to 10 minutes depending on the problem constraints and resource leveling.

In this SBAP-based model, all affecting factors including on-time delivery and space utilization are considered together. The rates of different factors can be adjusted to reach the best performance, and different modules are compared in solving the WDP and maximizing the social welfare. However, the greedy algorithm only looks at one entity at a time. Therefore, it is worth trying other combinatorial optimizations (e.g., dynamic programming) that look at the entire system in solving the WDP.

The developed simulation model is validated by comparing the simulation results with Primavera in a pilot project with a limited number of modules and an unlimited number of bays. The results of the simulation model are nearly identical with the results of the Primavera schedule in these special cases.

4.7 Limitations of the Scheduling Module Assembly Yard using the SBAP

One of the main challenges of the module yard's simulation model is linking the simulation model of the module yard to the simulation model of the fabrication shop. In practice, a module assembly process in the module yard starts once all required spools are fabricated in the fabrication shop. Currently, these two models work independently, although the shop and yard belong to the same company. Additionally, the simulation model of module yard and site construction should also be connected. Otherwise, the scheduler has to link them manually and re-enter the changes whenever there is a delay in the fabrication shop. In chapter 6, we introduce High Level Architecture to develop a comprehensive model of the entire industrial construction. This comprehensive simulation model allows us to link these dependent and separately developed simulation models together.

CHAPTER 5. Case Study 2: Application of the SBAP for Multiple Heavy Lift Planning [5]

5.1 Introduction

The Simulation Based Auction Protocol (SBAP) was introduced in chapter 3 for solving resource scheduling in large-scale construction problems and implemented successfully in a large-scale problem of a module assembly yard in chapter 4. This chapter presents implementing the SBAP in another case study of heavy lift planning. This case study demonstrates the capabilities and flexibilities of the proposed approach in a comprehensive multiple heavy lift planning system. Implementing the SBAP on this real application reduces the total cost and enhances the schedule and safety of the project.

5.2 Heavy Lift Planning

Heavy lifting in industrial construction involves the installation or replacement of the preassembled modules, which arrive from the module assembly yard of the fabrication shop, or other pieces of plant equipment (e.g., reactors), coming from

[5] Parts of this chapter are published at (Taghaddos et al. Submitted; Taghaddos et al. Forthcoming 2010).

different supply chains. These prefabricated objects may weigh up to 1000 tons (Hornaday et al. 1993; Lin and Haas 1996). In North American industrial construction, such heavy lifts are usually performed with mobile cranes (Shapira and Glascock 1996). Mobile cranes are one of the most critical resources in construction sites and are involved in various activities across the site. Unlike the majority of tower cranes, mobile cranes have to move to where installation activities utilize them (Tantisevi and Akinci 2008). To lift a module, mobile cranes can be located in different places with appropriate configurations based on their lifting radii and boom lengths. The configurations and locations of mobile cranes can be selected based on obstructions in the site, construction sequence, the congestion of the site and many other factors.

Heavy lift planning is usually performed on an iterative and trial-and-error basis to reach a reliable plan that satisfies all the constraints (which are discussed further below) with an acceptable confidence level. Prior to a lifting activity, several supporting tasks have to be planned, including changing the location, configuration, or rigging of the crane, or preparing the ground under the crane foundation and the pick point of the object (Varghese et al. 1997). It may take weeks of manual planning to derive an optimum system with minimum risk involved. However, these efforts are worthwhile and have significant impacts on the cost, schedule and safety of heavy industrial projects (Lin and Haas 1996). For example, in Alberta industrial heartlands, the rental cost of mobile cranes ranges from \$100/hr to \$1,700/hr. In addition, the cost of assistance equipment and laborers (e.g., $300/hr) should be added to the lifting cost. Supporting tasks may take a day or several days depending on the size of the crane and the required location, configuration, and rigging of the crane. Consequently, optimizing a heavy lift plan in a mega project can save millions of dollars.

Developing lift planning faces two main challenges in the current practice of the construction industry: first, lift planning is mostly done intuitively and is highly dependent on the lift planner's experience and skills (Hornaday et al. 1993; Shapira and Glascock 1996; Tantisevi and Akinci 2008). Although some computer-aided planning tools have been developed in the last two decades, none of them produces an integrated planning tool that considers all the lifts throughout the project. Second, lift planners often have to reschedule the lift plan in the construction stage because of delays in execution and unforeseen scenarios arising during the project (Hornaday et al. 1993; Mahalingam et al. 2000). Although some heavy lift planning tools have been developed for updating the lift plan in unforeseen scenarios, they only validate the feasibility of single lifts (Mahalingam et al. 2000).

5.3 State of the Art: Heavy Lift Planning

Heavy lift planning methods continue to undergo major changes because of both the increasing lifting capacity of cranes and advancing computer technologies, including simulation modeling, database information systems, computer-aided design (CAD), artificial intelligence (AI) tools, and geographical information systems (GIS) (Hornaday et al. 1993). These tools have enabled industry practitioners and academics to advance heavy lift planning and make the system more efficient.

In the construction industry, two companies have published their progress toward developing computer-aided heavy lift planning systems. Brown & Root (today known as KBR) developed a Computer-Aided Rigging (CAR) system to automate calculations of rigging analyses and documentation of rigging plans in a CAD platform (Brown 1991). Bechtel developed an Automated Lift Planning System (ALPS) to facilitate crane selection, rigging analysis, and 3D lift simulation (Bennett and Ditlinger 1996; William and Bennet 1996). Other commercial systems exist to assist the lift planner in investigating the feasibility of a single heavy lift. These systems include Compu Crane, Crane Lift Planner, 3D liftPlan, Cranimax, Kranxpert.

Efforts in computerizing heavy lift planning were initiated by the University of Texas in 1989 (Reddy and Varghese 2002). Varghese et al. (1997) interviewed several lift experts and identified various criteria in the planning process: 1) Availability of the crane, 2) Access to site, 3) Access to lift area, 4) Crane location, 5) Lift path clearances and factor of safety, 6) Ground support during lift, and 7) Removal from lift area. These lift criteria evaluate the feasibility, not the optimality, of the lift. A number of influencing factors such as crane type, crane configuration, site layout, and construction sequence or schedule influence these feasibility criteria; the lift planner can reach a feasible solution by adjusting the influencing parameters in different scenarios. A Heavy Lift Planning System (HeLPS) was also developed at the University of Texas to assist the lift planners with the 4^{th}, 5^{th} and 6^{th} lift evaluation criteria, and to animate the crane motions on a structure jobsite by utilizing CAD models of the site, vessels, and cranes (Varghese et al. 1997). Some other researchers have focused on analyzing the site accessibility (Hornaday et al. 1993; Lin and Haas 1996; Varghese and O'Connor 1995), identifying the possible crane locations (Lin and Haas 1996; Tantisevi and Akinci 2008) and automating the path-planning task (Reddy and Varghese 2002; Sivakumar et al. 2003; Deen Ali et al. 2005).

Aside from the construction industry, crane scheduling problems arise frequently in manufacturing systems and port container terminals under different settings. However, these problems are solved with different assumptions and the reported results are not directly applicable to mobile crane scheduling. For instance, some research assumes that the cranes at a berth or a shop are identical (Tamaki et al. 2004), or does not address many support tasks (e.g., mobilization, ground preparation, storage) common in heavy lifts with mobile cranes.

While they are useful, all the heavy lift planning systems previously mentioned are limited to modeling only some of the lift evaluation criteria. There is no comprehensive system that determines the required types of cranes in the preliminary planning stage and the lifting schedule for all the lifts during the detailed planning stage of the project, while concurrently considering all the planning criteria over the entire construction period. The number of variables in the entire system makes developing an integrated system very difficult. This research builds on previous studies and proposes a novel approach to determine the required types of cranes and to produce the lifting schedule for the entire project. This system enables the lift planner to consider all the lift evaluation criteria identified by Wolfhope (1991) as well as some other logical constraints.

5.4 Case Study

The case study focuses on multiple heavy lift planning in the modular construction of an industrial plant in Alberta, Canada. Multiple heavy lift planning for mobile crane operations in industrial construction is a complicated problem which is not yet solved completely. An automated multiple lift planning system has been identified by many researchers and industry practitioners as a cost efficient tool (Hornaday et al. 1993; Lin and Haas 1996; Varghese et al. 1997). Such a system is expected to select types of mobile cranes, and to provide a lift schedule once the information regarding the modules and the construction schedule is available with reasonable accuracy. This system is also expected to work at either the final planning stage with a detailed lift plan or at the execution stage when the project has evolved and some cranes have already lifted some modules in the site.

A module is a construction unit made of pre-assembled components such as structural steel frames, racks of pipes, cables, equipment, or a combination of miscellaneous components. Modules are built within or beside a large indoor fabrication facility and transported to the site by transporters (Taghaddos et al. 2009; Borrego 2004). Modular construction has become popular in the last century in the construction of remote facilities particularly in areas with harsh weather conditions (e.g., the province of Alberta). This popularity is due to numerous benefits for both constructor and client, including shorter project duration, reduced number of workers onsite, reduced project costs, improved safety and quality, more flexible construction processes, and eased site congestion (Taghaddos et al. 2008a). Figure 5-1 shows the module installation in a typical congested industrial plant. Modular construction of such a large industrial plant includes lifting several hundred prefabricated modules, which may each weigh up to 1000 tons (Hornaday et al. 1993; Lin and Haas 1996).

Planning multiple heavy lifts in congested industrial plants, particularly if the plant is built by modular construction, is a complicated and time consuming process. It includes determining the appropriate type, configuration, and rigging of the crane for each lift; preparing the pre-lift location of the module and the crane; verifying the lift capacity rating of the crane; and ensuring sufficient clearance during the lifting process (Lin and Haas 1996). Several activities may take place to prepare a crane to lift a module. These activities include mobilizing the crane to the site, changing the location, configuration and rigging of the crane, preparing the ground and putting mats underneath the crane's foundation. Each of these activities requires a different amount of time depending on the type of the crane, which may range from 1 day to 2 weeks.

Figure 5-1 A typical congested industrial plant

Figure 5-2 shows the locations of cranes in the construction site. Modules can also be lifted from different cranes' locations and pick-points. Mobile cranes can also have different configurations based on their lifting radii and boom lengths. The configurations and locations of mobile cranes can be selected based on obstructions in the site, construction sequence, the congestion of the site and many other factors.

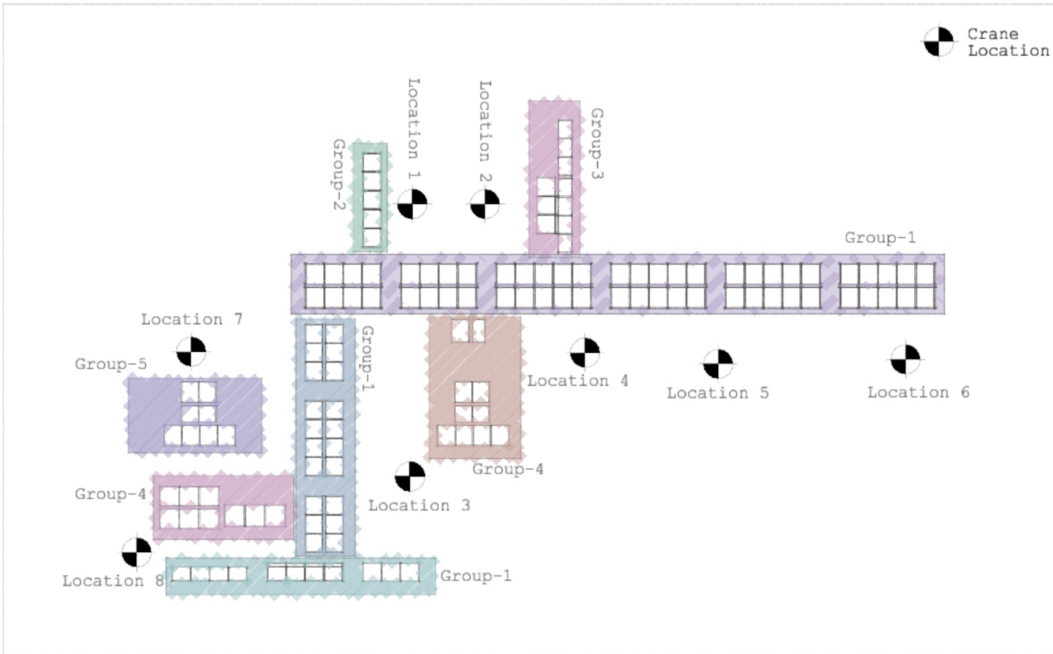


Figure 5-2 Modular installation and the plot plan with crane locations

Such a modular construction involves two main concerns. First, assembling prefabricated modules and interconnecting or connecting them to the other structures requires following certain sequences. The other concern in heavy lift planning is avoiding physical obstructions. In the industry practice, satisfying some of the constraints is mandatory, while satisfying others is a preference. For instance, a top module *must* be placed after the bottom module, but it is *preferable* not to place a module between two adjacent modules.

Lifting a module with a mobile crane requires some supporting tasks, both preceding and succeeding the lift (Figure 5-3). For example, if the crane is not already on the site, it has to be mobilized into the site. A mobilization task requires different amounts of time depending on the type of crane. While the crane is mobilized on a location in the site, the ground underneath the crane has to be prepared and supporting mats must be put down. If the crane moves to a new location that does not have the required ground preparation or mats, the same activities have to be repeated for the new location. The lifting duration also depends on the type of mobile crane.

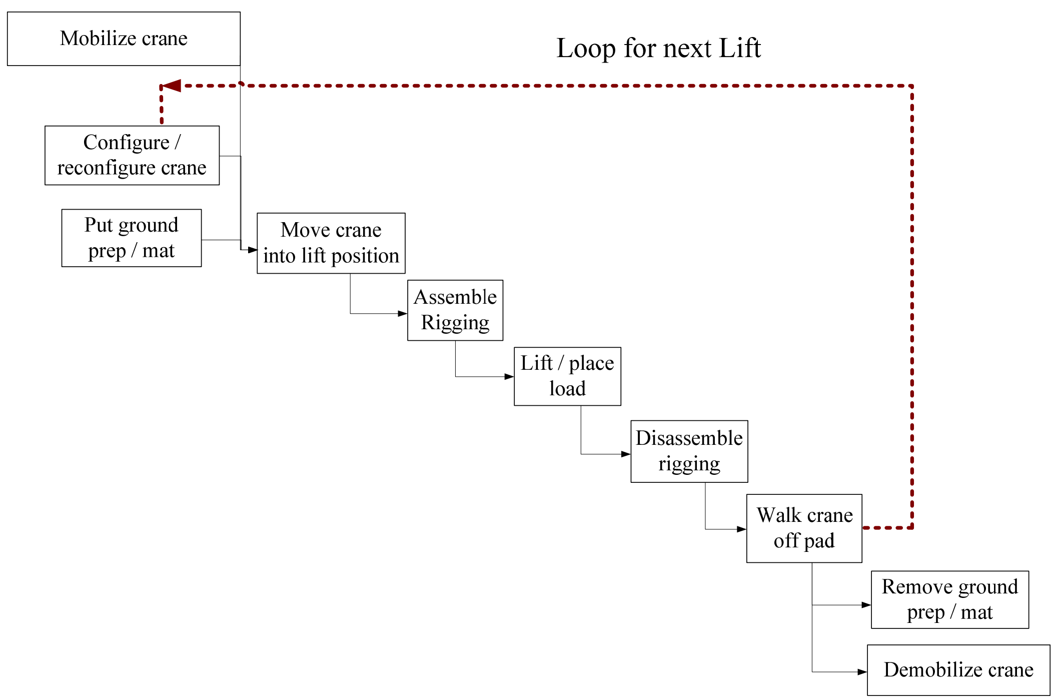


Figure 5-3 Crane's supporting tasks to lift a module

A crane is usually mobilized in the first required configuration to lift a module. However, it does not come with any rigging and the required rigging (i.e., 1 point, 12 points) has to be assembled after its mobilization. If the required configuration, location, or rigging of a lift differ from the current ones, a certain amount of time is required to reconfigure the crane, relocate it, or change its rigging. However, prior to changing the location or configuration of a crane, its rigging has to be disassembled and then assembled in the new position or configuration. Once a crane is not required in the site anymore, it has to be walked off the pad and demobilized. Simultaneously the ground preparation materials and the mats have to be removed.

The other important concept in the installation process is the logical constraint of the project. For example, a bottom module in a structure has to precede its top modules; modules have to be lifted in sequence because it is very difficult to place a module between two adjacent modules; lifting some modules may obstruct the lifting path of some other modules; some activities cannot be interrupted and should be continuously finished in a working day (e.g., lifting). There are also other constraints including availability of cranes, or possible locations.

The objective of this project is determining the best types of cranes to be used during the construction stage as well as finding a proper lift schedule that finishes the project on time and in budget, while satisfying the project's constraints. The lift schedule is expected to include the following information:

1) Mobilization and demobilization time of the cranes
2) Location of the cranes for each lift and the time to change their locations
3) Configuration of cranes and the time to change their configurations
4) Rigging of cranes and the time to assemble and disassemble their riggings

5) Locations underneath the cranes that have to be prepared and the time to prepare or to remove the ground preparation.
6) Pick-points and lifting time of the modules
7) Storage time of the modules
8) Sequence of module installation
9) The total cost and duration to accomplish the project

Solving such a large-scale problem with traditional optimization techniques is very difficult and time consuming. Some researchers have developed heavy lift planning systems to improve or automate some aspects of heavy lift planning (Tantisevi and Akinci 2008; Varghese et al. 1997; Reddy and Varghese 2002). However, no comprehensive research which considers all the above motioned objectives under one umbrella has yet been performed. The objective of this research is to develop a comprehensive lift planning system by employing the proposed SBAP framework.

Building upon the previous research (Hornaday et al. 1993; Varghese et al. 1997), the proposed heavy lift planning system is illustrated in Figure 5-4. This system reads the input data from a database, performs its analysis to select the best cranes with the proper locations and configurations, and produces a lifting schedule. After writing the output data into the database, a post simulation animation reads the required schedule and the spatial information and demonstrates the lifting process during the entire construction life cycle. The designed database system and the developed simulation model based on the SBAP are elaborated as follows.

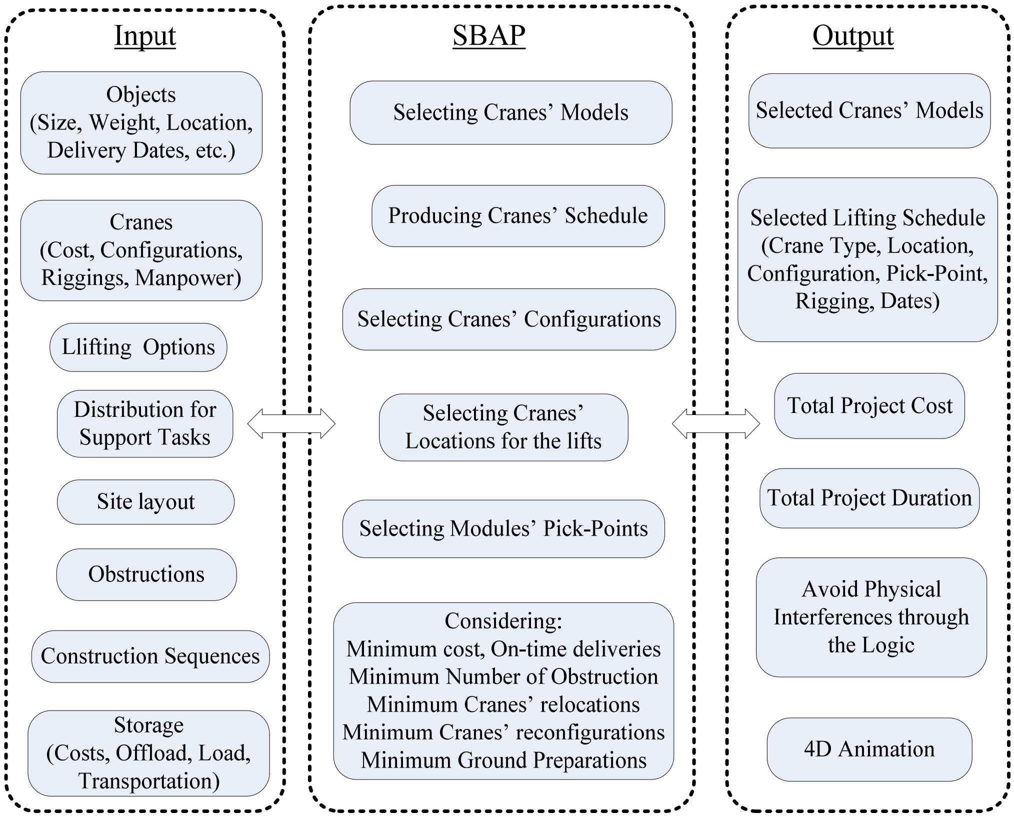


Figure 5-4 Main elements of the heavy lift planning system

5.5 Database System

The database is the entry point of the heavy lift planning system. It includes the following information regarding objects, cranes, site layout, etc.:

a) Object data:

Object data include spatial and physical information of the objects (i.e., modules, vessels) such as their size, weight, and destination. Moreover, the lift planner team has to estimate the delivery date of the objects to the site as well as their assembly Early Start (ES) dates, which depend on other constraints (e.g., structures, foundations). Lifting of the object can be performed on the later of the delivery date and the ES date of the object. Moreover, they have to be preliminarily designed to determine the required type of the rigging for each lift.

b) Resource data:

The lift planners have to also determine a vast majority of the data related to the scarce resources in the project.

b.1) Crane

The first type of scarce resource in heavy lift planning is mobile cranes. The required information for all the potential cranes that might be utilized in the lifting process should be defined. This information includes the type and size of the cranes, their dimensions, their availability periods, the estimated durations of their support tasks (e.g., changing the configuration, ground preparation), and the costs (e.g., rental cost, mobilization cost). The user also can restrict the maximum number of cranes to be mobilized in the site to a certain value. If the model starts from the middle of a project, the initial status (location, configuration and rigging) of the mobilized cranes should be determined.

b.2) Locations of cranes

The other type of scarce resource is the potential locations of the cranes in the construction site. These spots are scarce in a congested site with limited space for the heavy lift cranes. The lift planner should consider all the potential spots with their geographical coordinates in the construction layout as well as their availability periods. Access to the site and lift area can be reflected in these availability periods. Moreover, these location spots are defined so that enough space will be available to assemble and disassemble the mobile cranes.

b.3) Pick points of lifts

The potential pick-points of the lifts are the third type of resource in this system. Similar to the locations of the cranes, the availability periods of potential pick points have to be determined. As a result, this system models the access not only to the site and the cranes, but also to the objects. The major difference between locations of the cranes and pick-points of the lifts is that two cranes cannot sit in the same location area simultaneously, but two objects can sit in the same lift point at the same time. However, two objects cannot be lifted from the same pick point simultaneously.

c) Preliminary lift analysis

An important component of the SBAP is a preliminary analysis to define some lift options for each module. Each option includes a crane ID with a certain configuration ID, location ID, and pick-point ID. This analysis considers many lift feasibility factors including the capacities of cranes during lifts, the radii of the cranes, the configurations of the cranes, the locations of cranes and objects, the lengths of booms and jibs of cranes, and the lift path clearances. Some of the lift options may become invalid because of obstructions. In this case, if a certain module is placed first, then another object cannot be placed anymore from a particular location and with a particular configuration. Sometimes the location of a crane is the place where a module will be, before the module is installed. This information is stored in the database to be handled with the simulation model.

d) Storage data

An object can be installed at the site on a date which is later than the early start date and the delivery date of the object. However, if the object cannot be lifted into place by a proper crane by the next day, it will be sent to the storage area. Storing a module incurs some costs to transport the object and store it in the storage. The planner has to also estimate the cost to mobilize transport equipment, the cost to load and offload the module, as well as the daily rental cost of the storage area.

e) Sequence logic

The construction process of an industrial site must follow a certain logic. The presented heavy lift planning system provides the flexibility of providing the predecessors and successors as well as various types of lag (SS, SF, FS, and FF) for the objects. For instance, the module on top should be assembled after the one on the bottom. There is another logical sequence that is preferable to consider in practice to avoid placing a module between two adjacent modules. Therefore, the modules should be grouped to some rows and each row has to be built from one side. In some places such as at the intersection of rows or at the corners, a module may be included in two row groups (Figure 5-5).

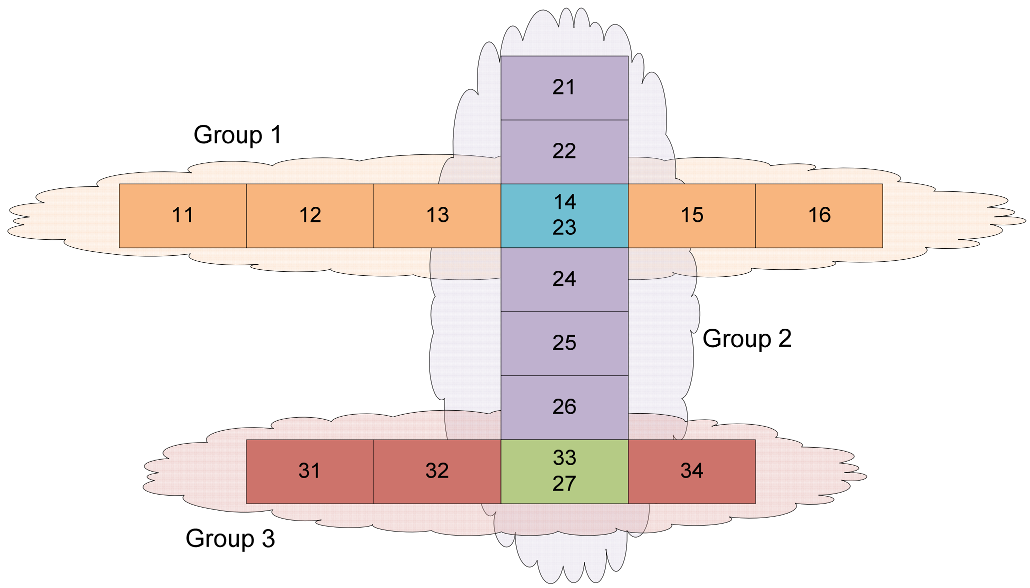


Figure 5-5 Sequence of lifting module

5.6 Semi-Manual Multiple Heavy Lift Planning

To explain the automated heavy lift planning process, first the semi-manual process of the heavy lift planning, which is the current practice of many construction firms, must be explained. Then attempts to automate this process using the SBAP are elaborated. The traditional practice of multiple heavy lift planning in the PCL Industrial Management Inc. is as follows:

1. Select the most favorable crane based on the weight of objects.
2. Choose the maximum number of cranes for the project by considering the budget and the schedule.
3. Find out the best locations for the cranes on site:
    a. The least movement of cranes.
    b. The least ground preparations and supporting mats for cranes

    c. The least overlap with the foundation and other activities.
    d. The least obstruction by high towers and permanent objects.
4. Calculate the crane percentage and boom clearance for each object using the assumed locations.
5. Choose the best crane configurations and locations based on the maximum clearances and minimum crane percentages.
6. Minimize changing the configurations of the cranes on the site.
7. Set the earliest install date for each object based on the schedule and the early arrival date for each object.
8. Create 4D simulation using the install dates and the best lift options of cranes.
9. Review the results and identify potential problems:
    a. Any clash between the crane boom and the body with the permanent and installed objects on the site.
    b. The installation sequence should satisfy the common practice and logics of construction.
    c. The crane location is available at the time of lift and is not disrupting any other activities occurring simultaneously
    d. The pick-points of cranes must be available at the time of lift.
    e. The number of cranes working at the same time is limited to maximum number of cranes.
    f. One crane cannot be used in multiple lifts simultaneously.

g. If the crane must be relocated, enough time should be available for moving the crane.

h. If the crane must be replaced with a new crane, enough time should be available for mobilizing the new crane.

i. The schedule must be adjusted to minimize the number of crane movements.

j. If the rigging must be changed, enough time should be available for assembling the new rigging arrangement.

10. This process must be repeated with different combinations of cranes to find out the best solution.

11. Total cost will be calculated based on the maximum duration of cranes on the site and daily rental cost plus the rigging crew expense for each lift. The rental cost for rigging components is also considered in the final cost. Each cost item includes multiple activities or additional costs:

    a. Rental Cost = Equipment + Operator + Insurance + Fuel

    b. Mobilization Cost = Transportation + Rigging labor + Rental cost for smaller assist crane

    c. Demobilization costs, which usually consist of the same activities as mobilization with a shorter duration.

    d. Reconfiguration = Rental cost for assist crane + Rigging laborer

Reviewing the 4d schedule helps to find the outstanding issues. Nevertheless, the solution is highly dependent on the experience and knowledge of the person or the group in charge. The manual process becomes more complicated when more factors affecting the final cost are considered. Increasing the number of cranes used on site makes comparing alternatives and finding the best option more difficult. In this traditional practice, the cost of delaying the project or the storage cost on the site has not being considered.

5.7 Automated Multiple Heavy Lift Planning using the SBAP

The SBAP is the main processing unit of the multiple heavy lift planning system. This system selects the best cranes with the best configurations and locations, arranges to mobilize the cranes to the site at the appropriate time, schedules the required time for lifting modules or storing them, or arranges to reconfigure or relocate the cranes. "Best" does not imply the global optimum of the system, which is an unrealistic expectation for such a large scheduling system with so many variables (i.e., an NP-hard problem). It only means a near-optimum solution or a solution with a significant improvement over traditional scheduling methods.

The main challenge in the heavy lift planning is distributing the limited resources (e.g., mobile cranes and space) among different modules over time so that the entire system is optimized and the available constraints are satisfied. Consequently, the problem can be translated to a large-scale resource scheduling problem with several constraints. Solving such a large-scale resource scheduling problem with the traditional optimization algorithms is impractical. This research proposes to employ the SBAP framework to distribute the resources among different modules over time based on auction protocols. This framework maps the resource scheduling problems to combinatorial optimization problems by holding regular auctions (i.e., daily, weekly).

In this system, agents represent modules in the construction site and compete over the resources. Heavy lift cranes are the first type of resource; they are limited and costly and create bottlenecks in the construction process. Space for the cranes and pick-points are another type of resource; space is critical in congested industrial plants. In this framework, auctions are held at the start of a simulation cycle (e.g., every week, or every day), at which modules bid for different combinations of the resources. Then the auctioneer allocates the resources by minimizing the total bidding cost. A discrete event simulation model is also employed to allocate and release the assigned resources, and to schedule different support activities. Figure 5-6 depicts a schematic view of the SBAP. The main components of the SBAP are the auction protocol and the simulation modeling, which are explained below.

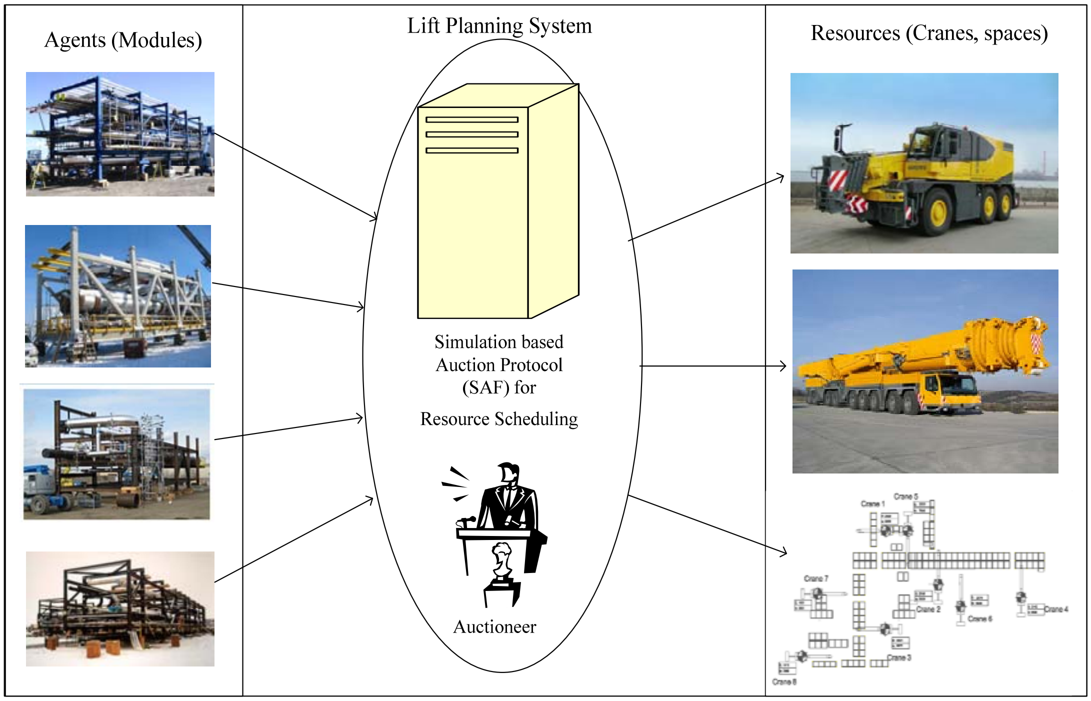


Figure 5-6 Schematic view of the heavy lift planning system

5.7.1 Auction Protocol

In the proposed SBAP framework, a number of auctions are held regularly (e.g., on a daily basis). In each auction, agents (i.e., modules in this problem) whose lifting time is prior to the current simulation time plus the length of Time Window (TW) and whose conditions satisfy the available constraints bid for different combinations of resources. These agents have to determine their bidding costs for different combinations of resources. Then the designed auctioneer employs an allocating procedure to distribute the available resources among bidding agents such that the total cost is minimized. However, after awarding the winning agents, only those winning agents that require the crane to start its support activities prior to the next auction will capture the crane.

As mentioned above, a preliminary analysis was performed to determine a number of lifting options for each module prior to this project. A lifting option corresponds to a tender with a combination of resources that the module bids on. Each lifting option associates modules with a specific crane with a particular configuration and location, as shown in Table 5-1. For example, in this table, the module with object ID 3 can be lifted with 8 different options, which involve 3 cranes (i.e., the crane IDs 3,4, and 8), 4 cranes' locations (i.e., the location detail IDs 1, 2, 3, and 7), and 3 pick-points (i.e., the pick-point IDs 1, 3, and 7). This table shows part of the option table, which also includes information related to the configuration, rigging, radius, boom clearance, and capacity of cranes in different lifting options. This preliminary analysis is performed by considering the capacity of cranes, the required radius to lift the object, the weight of objects, and the length of the boom and jib of the cranes. Some of the lifting options may become invalid during construction because of obstruction by the other modules. These circumstances are all identified in the preliminary analysis and stored in the database.

5-1 Modules' lifting options

OptionID	ObjectID	CraneID	LocationDetailID	PickPointID	ConfigID	RiggingID	RadiusMin	RadiusMax	BoomClearance	PositionRadius	CapPercAtPositionRadius
14136	3	8	1	1	18585	8	78.7	255.9	13.14	23.94	24
14137	3	8	2	1	18585	8	150.9	255.9	25.81	44.67	36
14138	3	8	3	3	18585	8	255.9	255.9	49.57	74.31	83
14139	3	8	7	7	18585	8	229.7	255.9	40.45	61.23	69
14140	3	3	1	1	19895	8	78.7	150.9	12.16	23.94	44
14141	3	3	2	1	19895	8	150.9	150.9	22.59	44.67	66
14142	3	4	1	1	20692	8	79	124	11.61	23.94	45
14143	3	4	1	1	20715	8	79	98	12.60	23.94	53
14144	4	8	1	1	18585	8	78.7	295.3	11.82	23.94	15
14145	4	8	2	1	18585	8	150.9	295.3	23.38	44.67	24
14146	4	8	3	3	18585	8	255.9	295.3	45.96	74.31	54
14147	4	8	4	3	18585	8	269	295.3	45.37	80.48	59
14148	4	8	7	7	18585	8	229.7	295.3	37.31	61.23	45
14149	4	2	1	1	18899	8	80	105	10.85	23.94	68
14150	4	3	1	1	19895	8	78.7	150.9	10.30	23.94	31
14151	4	3	2	1	19895	8	150.9	150.9	19.35	44.67	46
14152	4	4	1	1	20692	8	79	164	9.55	23.94	31
14153	4	4	2	1	20692	8	151	164	17.35	44.67	72
14154	4	4	1	1	20715	8	79	138	10.98	23.94	36
14155	9	8	1	1	18585	6	150.9	295.3	25.08	39.29	20
14156	9	8	2	1	18585	6	78.7	295.3	8.20	17.93	13
14157	9	8	3	3	18585	6	295.3	295.3	57.81	85.71	61
14158	9	8	4	3	18585	6	137.8	295.3	30.42	40.55	18

One of the main components of the proposed algorithm is the calculation of the bidding prices for the modules in an auction. Each module checks different combinations of resources in its lifting options. If the module can be lifted by the resource combinations in the lifting option before the time of the next auction, it proposes a bid for that lifting option to the auctioneer. The bid price (C_b) for each module is calculated by approximating the real cost incurred in the construction, which is the summation of the actual lifting cost (C_a) and the idle cost (C_i) of the crane as follows:

$$C_b = C_a + C_i \tag{5-1}$$

Actual cost of a crane (C_a) is calculated by considering the cost incurred for performing support activates of the crane (C_{cs}), the storage cost of the module (C_{sm}) and the associated cost for any delay in lifting the module (C_{dm}):

$$C_a = C_{sc} + C_{sm} + C_{dm} \tag{5-2}$$

The cost incurred for support activities include the rental cost of the crane (C_r), mobilization cost (C_m), demobilization cost (C_{dm}), and extra manpower cost to change rigging of the crane (C_{mp}).

$$C_{cs} = C_r + C_m + C_{dm} + C_{mp} \tag{5-3}$$

Equation 5-4 shows that the rental cost of the crane (C_r) depends on the hourly rental cost of the crane (C_{cr}) and the duration that the crane is busy with the support tasks (D_b). The required duration for performing the crane's support activities includes the required duration for mobilizing the crane (D_m), demobilizing the crane (D_{dm}), preparing the ground condition (D_{gp}), relocating the crane (D_{rl}), reconfiguring the crane (D_{rc}), rigging down (D_{rd}), rigging up (D_{ru}) and lifting the module (D_l), according to Equation 5-5:

$$C_r = C_{cr} \times D_b \tag{5-4}$$

$$D_b = D_m + D_{dm} + D_{gp} + D_{rl} + D_{rc} + D_{rd} + D_{ru} + D_l \tag{5-5}$$

A module has to be stored in the storage area if the lifting date is more than a day after the delivery date of the module into the site. incurring a storage cost (Cs) which depends on the hourly rental cost of the storage area (C_{sr}), the lifting time of the module (T_l), the delivery time of the module (T_d), the offload cost (C_o), the mobilizing cost of transporting equipment (C_{mt}) and the loading cost of the module (C_l), as shown in Equation 5-6:

$$C_{sm} = C_{sr} \times (T_l - T_d) + C_o + C_{mt} + C_l \tag{5-6}$$

A delay cost has to be considered in the actual cost, if the module is going to be lifted after its late finish date. This delay cost applies to modules with a late finish time (T_{lf})). The delay cost of a module (C_d) depends on the penalty cost (C_p), and is calculated by Equation (5-7):

$$C_{dm} = C_p \times (T_l - T_{lf}) \tag{5-7}$$

Aside from the actual cost of the crane, the idle cost of the crane should also be considered in the bidding cost, because of two reasons: first, it minimizes the idle duration of cranes, which affects the total cost of the project; second, all the incurred costs would otherwise be associated with the modules. At the end of the project, the total lifting cost would be simply the summation of the lifting costs for all modules. The idle cost of the crane directly depends on its idle duration (D_i), which is the time that the crane does not lift any heavy lift module between the last release time of the crane (T_{cr}) and the start time of the preparation activities for the next lift (T_{cs}). T_{cs} should be after the current auction time (T_{Now}), the module's arrival time (T_d) and its early start time (T_{ES}). Therefore, the idle cost of the crane can be calculated as follows in Equations 5-8, 5-9, and 5-10:

$$C_i = C_r \times D_i \tag{5-8}$$

$$D_i = T_{cr} - T_{cs} \tag{5-9}$$

$$T_{cs} = \text{Max}\ (T_{Now}, T_d, T_{ES}) \tag{5-10}$$

These formulas are customized for the case study of this problem. However, they can be modified accordingly for different heavy lift planning systems. The main intention of presenting these formulas is to clarify the different factors that can be incorporated in the bidding price. After submitting the bid price, an auctioneer allocates the resources among different agents.

The auctioneer can allocate the resources using any WDP algorithm encoded in the program. In this study, the greedy and ascending-auction algorithms are employed to solve the WDP and to award the winning agents. These algorithms can be substituted by other optimization algorithms without much difficulty.

Figure 5-7 demonstrates a simple crane scheduling example, where two modules (M101 and M102) are available at $\Delta T=10$ and two cranes (Cr 1 and Cr 2) are in place with specific configurations (Conf 100 and Conf 200), locations (Loc1 and Loc2), and riggings (Rig 1 and Rig

2). The last release dates of crane 1 and 2 are September 29 and 30, respectively. It is also assumed that auctions are held on a daily basis (d_t=1). At an auction on October 1st, it is assumed that each of two modules have two options with bidding prices shown on Table 5-2. For example, the lift cost of module M101 with crane 1 is less than the lift cost of module M102 with the same crane. The required preparation duration of this lift is much less because the last configuration of crane 1 is the same as the required configuration of lifting M102 with crane 1.

Table 5-2 Bid prices of modules

Resources:	Cr_1,Loc_2	Cr_2,Loc_2
Module 101	$6,000	+∞
Module 102	+∞	+∞
Resources:	Cr_1,Loc_3	Cr_2,Loc_3
Module 101	+∞	**$6,300**
Module 102	+∞	+∞
Resources:	Cr_1,Loc_5	Cr_2,Loc_5
Module 101	+∞	+∞
Module 102	**$4,500**	$7,500

The matrix shown in Table 5-2 is indeed a 3D matrix, but for the sake of simplicity, the dimension of location is shown in different tables. After receiving the bid submissions, the auctioneer allocates the resources (i.e., cranes and locations) by minimizing the total cost. Therefore, the auctioneers consider +∞ for the bids that do not have any bid price, to avoid selecting them.

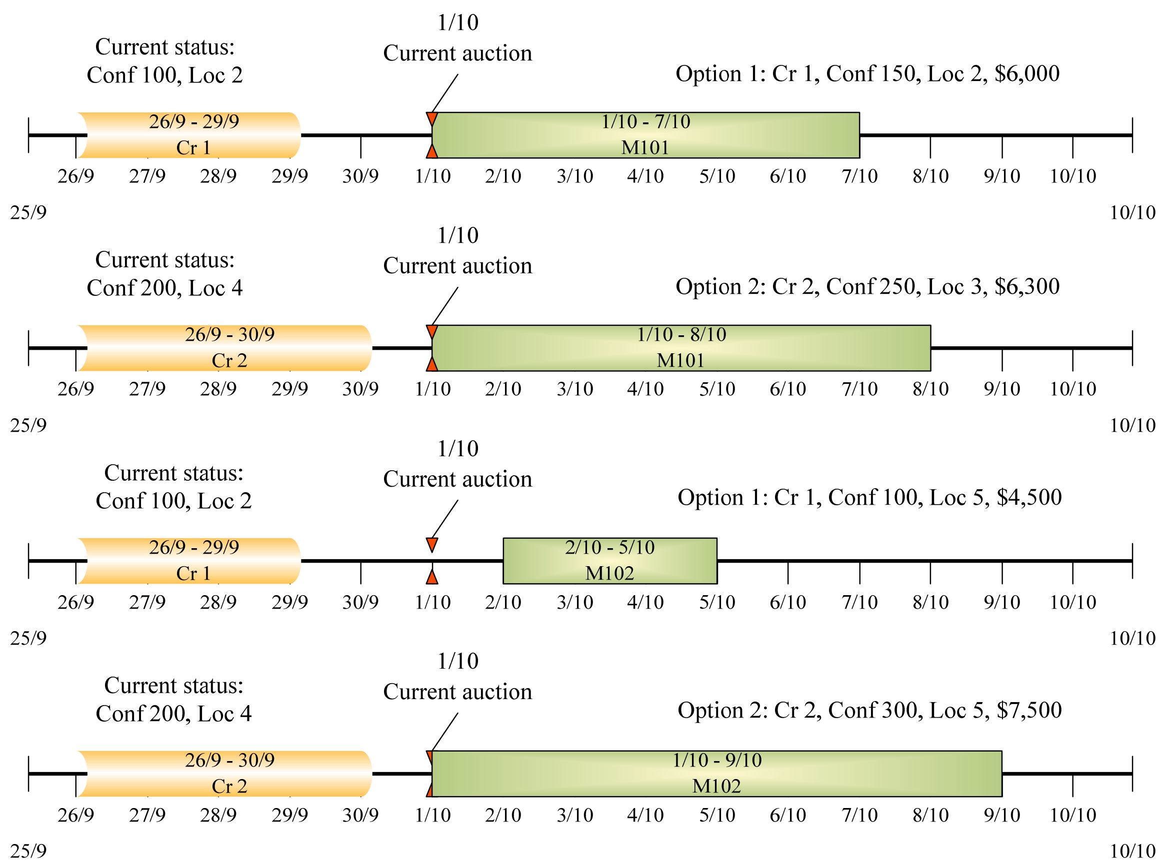
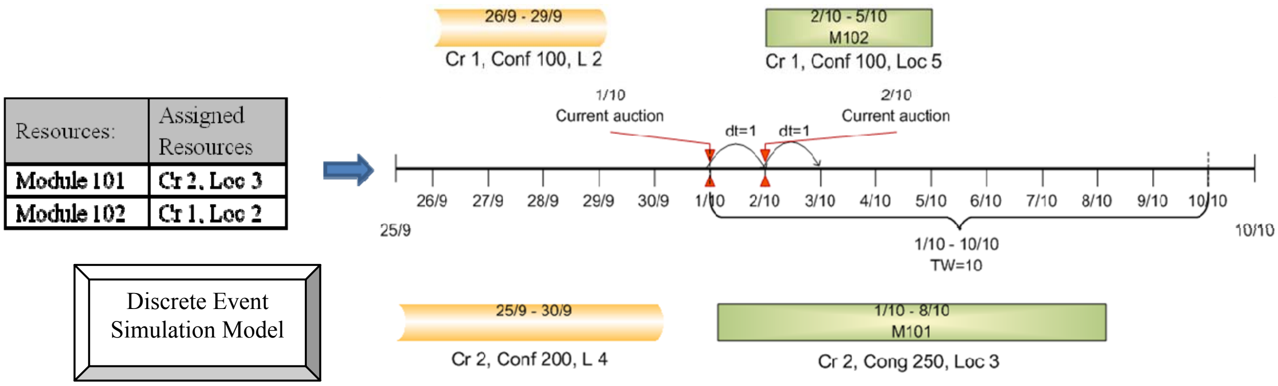


Figure 5-7 Crane scheduling example

When the auctioneer solves the WDP, it assigns the highlighted value in Table 5-2 to the module. Figure 5-8 demonstrates the allocation of resources to the modules in this example. Module 101 captures crane 2 and location 3 at the current auction, and module 102 can capture crane 1 and location 3 at the next auction. Crane 1 is not allocated to module 102 at this auction on October 1st, because the capture time of the resources for this module is not before the next auction time. But this module reserves crane 1 to capture it in the next auctions at August 2nd. This approach provides the possibility to update the decision about resource assignment by gaining new information about released resources. For instance, if a crane is not currently available but it is released before the next auction, it can be assigned to a module in the next auction.



5-8 Allocated cranes to the modules in the example

5.7.2 Discrete Event Simulation Model

Once the auctioneer assigns the resources to some of the modules, if the lifting date of the module is prior to the time of the next auction, the simulation model allocates the resources to the module at the module's capture time (T_c). In other words, the capture date of resources has to be controlled after determining the winning agents. If the capture date is prior to the next auction, the simulation model schedules capturing the resource, performing the supporting activities, and releasing the assigned resources afterward (Taghaddos et al. Submitted). Moreover, several logical constraints (e.g., assembly sequence, predecessor modules, obstruction, and availability of resources) have to be embedded inside this framework. A discrete event simulation model is developed to track the availability of resources, to capture and release the resources, to schedule the cranes' support activities, to satisfy the project's constraints, to consider the shifts and working days, and to advance the simulation time.

The simulation model is developed using the core services of the Simphony 3.5 simulation engine, which is is the new generation of Simphony.NET based on .NET version 3.5 and developed within the Visual Studio 2008 programming environment (Hajjar and AbouRizk 1999). This simulation model has the capability to run different scenarios with various combinations of resources and to pick the best one. The "best" implies a balance between the least cost and the least duration, and does not mean the global optimum.

According to the initial resources that are available, different results may be obtained. Therefore, it is best to run the program several times for different scenarios and compare the results to find the best solution. This methodology is described briefly in the following steps:

1. *Loop over different scenarios. For each scenario:*
    - *1.1. At each period of time:*
        - *1.1.1. For all the modules that arrive within or before the period of time and satisfy the constraints:*
            - *1.1.1.1. Each module bids for different combinations of resources (cranes, locations of cranes) in the lifting options.*
            - *1.1.1.2. The auctioneer assigns available resources among the bidding agents so that the total cost of the system is minimized.*
            - *1.1.1.3. Go to the next period of time.*
    - *1.2. Allocate and release the resources, schedule different support activities, and execute the rest of simulation model.*
    - *1.3. Stores the results into the database.*
2. *Continue until the simulation time reaches the end of project duration.*

This framework has four different components. First, there is a layer that produces different scenarios (step 1). Second, there is a bidding stage in which some of the agents (modules) submit their bids for different combinations of resources with a bidding price (step 1.1.1). Third, an auctioneer awards winning agents by solving WDP using any optimization technique (step 1.1.2). Finally there is a discrete event simulation engine that allocates and releases the resources and advances the simulation time. The second and the third components were explained in the previous section. The first and the fourth components are elaborated below:

Different scenarios can be produced by a recursive algorithm or a Genetic Algorithm (GA). The first approach produces all the possible scenarios considering the crane database and the value of the maximum number of cranes that can be mobilized to the site (i.e., MaxCranes). For example, if the MaxCranes is 3 and there are 4 total cranes available (c_1,c_2, c_3, c_4), the recursive algorithm produces the following 10 scenarios as available cranes for the simulation model:(c_1), (c_2), (c_3), (c_4), (c_1, c_2), (c_1, c_3), (c_1, c_4), (c_2, c_3), (c_2, c_4), (c_3, c_4). If the number of all possible scenarios is too large, a Genetic Algorithm (GA) is the alternative solution. In this case, scenarios are represented by a population of chromosomes. Each chromosome represent as array of bits, in which a bit represents availability of a crane and has a binary value (0, or 1). The fitness function can represent the total cost, duration or combinations of both. The population begins with randomly generated individuals and evolves by crossover or mutation to result in a better fitness function. The GA terminates when the number of generations has reached the set maximum number. In each auction (time t), some agents (i.e., modules) bid for different combinations of resources as follows:

1. *Define resources (cranes, cranes' locations, modules' pick-points) based on the database information, waiting-files to capture resources.*
2. *Synchronize time of the simulation engine with working hours based on the selected shift and weekdays.*
3. *Generate modules based on modules' information and schedule their arrival at the site.*
4. *At the start of a simulation cycle (every day):*
    - 4.1. *Loop over all the arrived modules. For each module:*
        - 4.1.1. *Check various constraints (e.g., if the required structures are built, the predecessor modules are installed, and the modules are built in a logical sequence).*
        - 4.1.2. *If the constraints are satisfied, loop over potential lifting options of the module. For each option (a combination of resources):*
            - 4.1.2.1. *Check the constraints regarding availability of required resources, maximum number of cranes, and obstruction.*
            - 4.1.2.2. *If all constraints are satisfied, determine the bidding cost and let the module bid for the resources in the option.*
    - 4.2. *Assign available resources (cranes, locations) among the bidding modules based on minimizing the total cost by employing a greedy optimization algorithm.*
5. *Run the simulation model based on the resource assignment results as follows:*
    - 5.1. *Capture the assigned crane and location.*
    - 5.2. *Mobilize the crane, if it is not yet mobilized.*
    - 5.3. *Prepare the ground and put the mats underneath crane, if there aren't any.*

5.4. Reconfigure the crane, if the module needs a different crane configuration.

5.5. Move the crane into new lift position, if the module needs a different crane location.

5.6. Disassemble the previous rigging and assemble the new one, if the module needs a different rigging.

5.7. Capture the pick-point.

5.8. Lift the module, if the lifting process will finish on the same date. Otherwise, lift the module at the beginning of the next business day.

5.9. Release all the module's resources.

6. Finalize the scenario:

6.1. Disassemble the rigging

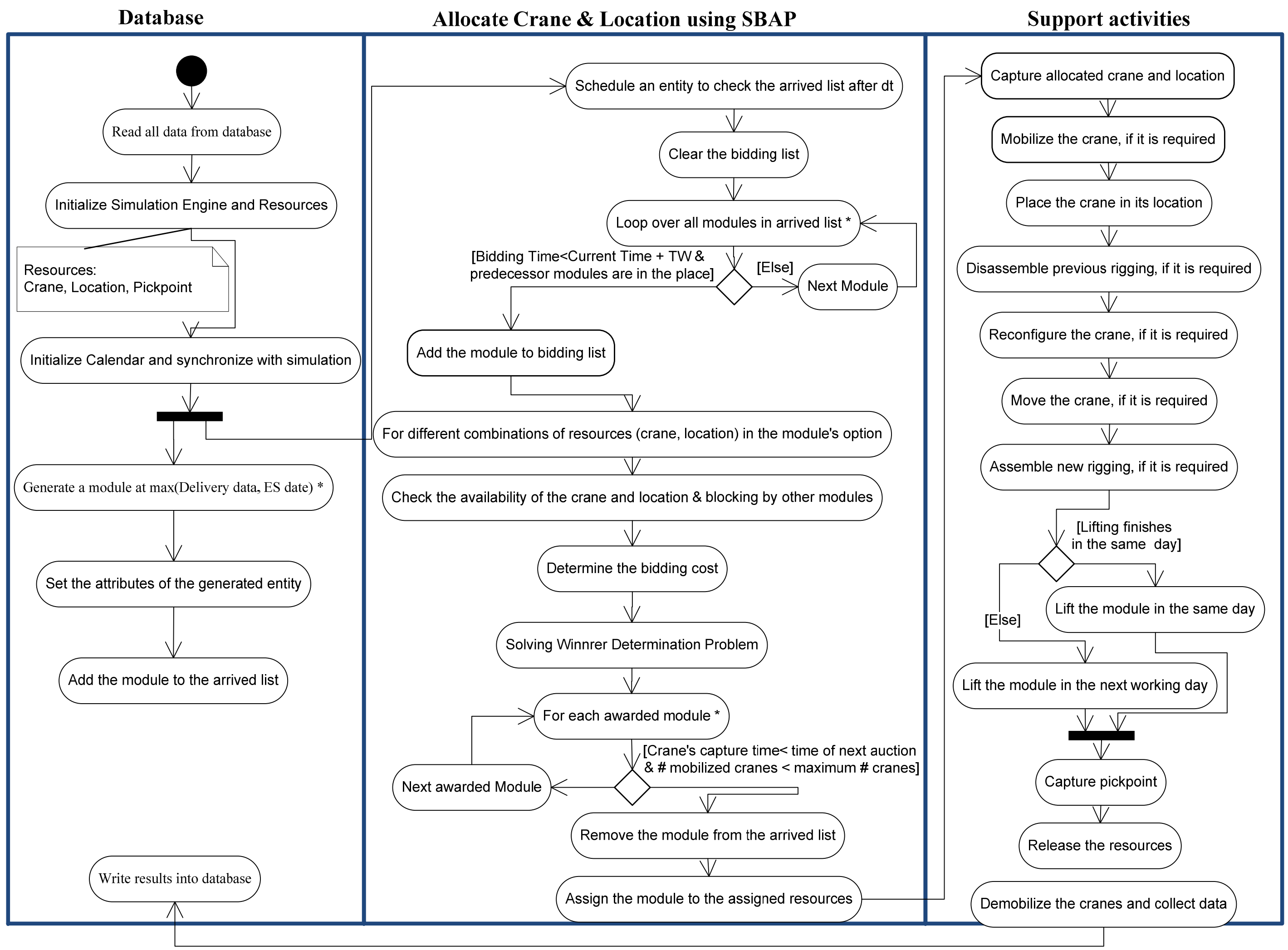
6.2. Walk the cranes off the pads

6.3. Demobilize all the used cranes

6.4. Remove the ground preparations and the mats

6.5. Calculate the total cost and duration for the scenario

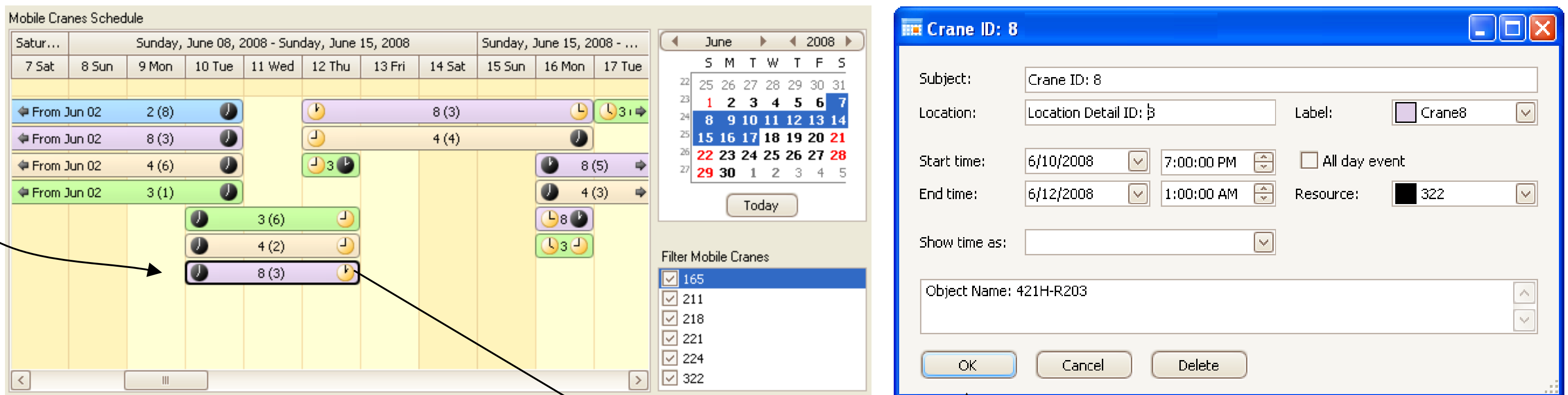
After the auctioneer assigns resources to the modules, the above-explained simulation model is employed to capture and release the resources and to advance time. The flowchart of the simulation model is elaborated in Figure 5-9. The simulation model works for different scenarios; the best schedule is demonstrated in Figure 5-10. The results include a table in the database that includes the crane type, location of the mobile crane, the pick-point, and the configuration and rigging of the crane for lifting each module. The table also includes the start time of the crane's support tasks (e.g., mobilization, relocation, and reconfiguration) and the lift cost for each lift. Figure 5-10.b shows part of this schedule using a time schedule developed by the DXperience add-on in Visual Studio 2008.



5-9 UML Activity Diagram of the simulation model of the heavy lift planning based on the SBAP

ObjectName	CraneID	LocationDetailID	ConfigID	RiggingID	StartTime	MobilizeTime	DisAssembleRigTime	ReconfigureTime	MoveTime	AssembleRigTime	LiftTime	FinishTime	LiftCost
421H-R202	8	3	18585	7	6/2/2008 4:00:00 AM	6/2/2008 4:00:00 AM				6/10/2008 4:00:00 AM	6/10/2008 2:00:00 PM	6/10/2008 7:00:00 PM	682850.0
421R-M211	4	6	20692	8	6/2/2008 4:00:00 AM	6/2/2008 4:00:00 AM				6/10/2008 4:00:00 AM	6/10/2008 2:00:00 PM	6/10/2008 7:00:00 PM	248350.0
421R-R201	3	1	19895	7	6/2/2008 4:00:00 AM	6/2/2008 4:00:00 AM				6/10/2008 4:00:00 AM	6/10/2008 2:00:00 PM	6/10/2008 7:00:00 PM	682850.0
421H-R201	2	8	18899	7	6/2/2008 4:00:00 AM	6/2/2008 4:00:00 AM				6/10/2008 4:00:00 AM	6/10/2008 2:00:00 PM	6/10/2008 7:00:00 PM	204975.0
421H-R203	8	3	18585	6	6/10/2008 7:00:00 PM		6/10/2008 7:00:00 PM			6/11/2008 9:00:00 AM	6/11/2008 4:00:00 PM	6/12/2008 1:00:00 AM	109200.0
421R-M212	3	6	19895	8	6/10/2008 7:00:00 PM		6/10/2008 7:00:00 PM		6/11/2008 9:00:00 AM	6/11/2008 2:00:00 PM	6/12/2008 4:00:00 AM	6/12/2008 9:00:00 AM	133600.0
421R-R203	4	2	20692	7	6/10/2008 7:00:00 PM		6/10/2008 7:00:00 PM		6/11/2008 9:00:00 AM	6/11/2008 2:00:00 PM	6/12/2008 4:00:00 AM	6/12/2008 9:00:00 AM	124600.0
421H-M402	8	3	18585	7	6/12/2008 1:00:00 AM		6/12/2008 1:00:00 AM			6/12/2008 8:00:00 AM	6/12/2008 6:00:00 PM	6/16/2008 3:00:00 AM	119200.0
421R-M311	3	6	19895	8	6/12/2008 9:00:00 AM						6/12/2008 9:00:00 AM	6/12/2008 2:00:00 PM	19550.0
421R-R205	4	4	20692	8	6/12/2008 9:00:00 AM		6/12/2008 9:00:00 AM		6/12/2008 7:00:00 PM	6/16/2008 4:00:00 AM	6/16/2008 2:00:00 PM	6/16/2008 7:00:00 PM	137900.0
421R-R202	8	1	18585	7	6/16/2008 3:00:00 AM				6/16/2008 3:00:00 AM		6/16/2008 8:00:00 AM	6/16/2008 1:00:00 PM	25800.0
421R-M312	3	6	19895	8	6/16/2008 4:00:00 AM						6/16/2008 4:00:00 AM	6/16/2008 9:00:00 AM	18750.0
421R-R209	8	5	18585	8	6/16/2008 1:00:00 PM		6/16/2008 1:00:00 PM		6/17/2008 3:00:00 AM	6/17/2008 8:00:00 AM	6/17/2008 6:00:00 PM	6/18/2008 3:00:00 AM	141100.0
421R-R219	4	3	20692	6	6/16/2008 7:00:00 PM		6/16/2008 7:00:00 PM		6/17/2008 9:00:00 AM	6/17/2008 2:00:00 PM	6/18/2008 1:00:00 AM	6/18/2008 6:00:00 AM	119850.0
421R-R215	3	7	19895	6	6/17/2008 4:00:00 AM		6/17/2008 4:00:00 AM		6/17/2008 2:00:00 PM	6/17/2008 7:00:00 PM	6/18/2008 6:00:00 AM	6/18/2008 11:00:00 AM	134200.0
421R-R207	8	5	18585	8	6/18/2008 3:00:00 AM						6/18/2008 3:00:00 AM	6/18/2008 8:00:00 AM	25000.0

a) Tabular schedule



5-10 Schedules of multiple heavy lift obtained from the developed simulation model

CHAPTER 6. Developing Generic Resource Allocation Federate using High Level Architecture (HLA)

6.1 Introduction

Chapter 4 and 5 reviewed simulation-based modeling of a module assembly yard, and site construction based on the proposed Simulation-Based Auction Protocol (SBAP). Module assembly and site construction are considered as two supply chains of the industrial construction, which also includes drafting, material procurement and supply, and spool fabrication. This chapter discusses building a comprehensive simulation model for the entire industrial construction using High Level Architecture (HLA).

This chapter focuses on developing a federate (or federates) to facilitate allocation of resources in construction projects. Developing the SBAP framework brings us one step closer toward this objective. It was mentioned in the previous chapters that the Winner Determination Problem (WDP) is an independent component of the SBAP framework that can be solved using various combinatorial optimization algorithms. Therefore, a Resource Allocation (RA) federate can be developed to act as an auctioneer in the SBAP framework and to solve the WDP. This RA federate can be developed for a generic domain to collaborate with different federates in construction federations. This federate can be employed either in the industrial federation or in other federations with different scopes (e.g., the tunneling federation). However, it is obvious that those federates that are supposed to employ the RA federate should be developed based on the proposed SBAP framework.

Another focus of this chapter is developing a generic federate, called the Resource Allocation Base (RAB), to automate the communication with the RA federate. This federate should be inherited by federates that are going to communicate with the RA federate (i.e., the auctioneer) to submit bids and award the winning agents. This federate is tested in the two federates in the industrial federation and works successfully.

6.2 Industrial Construction Federation

Industrial construction covers a wide range of supply chains including drafting, material procurement, spool fabrication, module assembly and site construction. Previous research has shown that the traditional simulation environments are not effective in modeling the entire process of industrial construction for several reasons (El Ghandour 2007; Wang 2006): first, developing a single simulation model that covers all the supply chains of a large scale industrial construction with one simulator is very time consuming and expensive for a corporation. The simulator has to spend several hundred hours in each supply chain (e.g., the spool fabrication shop) to gain the required knowledge and to become expert in the area. This process of gaining knowledge of entire system is very time consuming for one simulator. Second, the traditional simulation environments do not offer a suitable platform for decomposing the large-scale model and developing it using teamwork efforts. Third, reusing the developed simulation model in future projects or in the other parts of current projects is also very challenging. Generally simulators prefer to start from scratch rather than reuse previously developed simulation models. Moreover, the limitations of the processing system are a serious issue in simulating a large-scale model and running it several times to investigate different scenarios. Beside the deficiencies in interoperability, reusability and computing ability, lack of product presentation, knowledge reuse and standardization are other shortcomings of traditional simulation environments.

High Level Architecture (HLA) is a promising approach for building a comprehensive model of large-scale projects. HLA integrates separate components of a simulation model, referred to as federates, into a single distributed simulation model, referred to as a federation. The main intent of HLA is to promote interoperability between simulations and to aid in reusing models in different contexts, ultimately reducing the time and cost required to create a new simulation environment. HLA also helps to decompose a large scale simulation model and to develop each federate separately with a different simulator. These federates can be run on different computers to enhance the computational ability of the system (Shahin 2007; Fujimoto 2003; Kuhl et al. 1999).

The comprehensive simulation model of industrial construction, built in HLA, is referred to as the "industrial federation." This federation consists of several federates, shown in Figure 6-1, some of which are main simulation federates and some of which are supportive (i.e., domain independent) federates. Drafting, procurement, spool fabrication, module assembly, and site construction are currently main simulation federates, each of which simulates one of the supply chains in industrial construction. Calendar, weather, visualization, schedule, resource allocation (RA), and resource allocation base (RAB) act as supportive federates that serve one or several federates in industrial federation or even other types of federations (e.g., a tunneling federation).

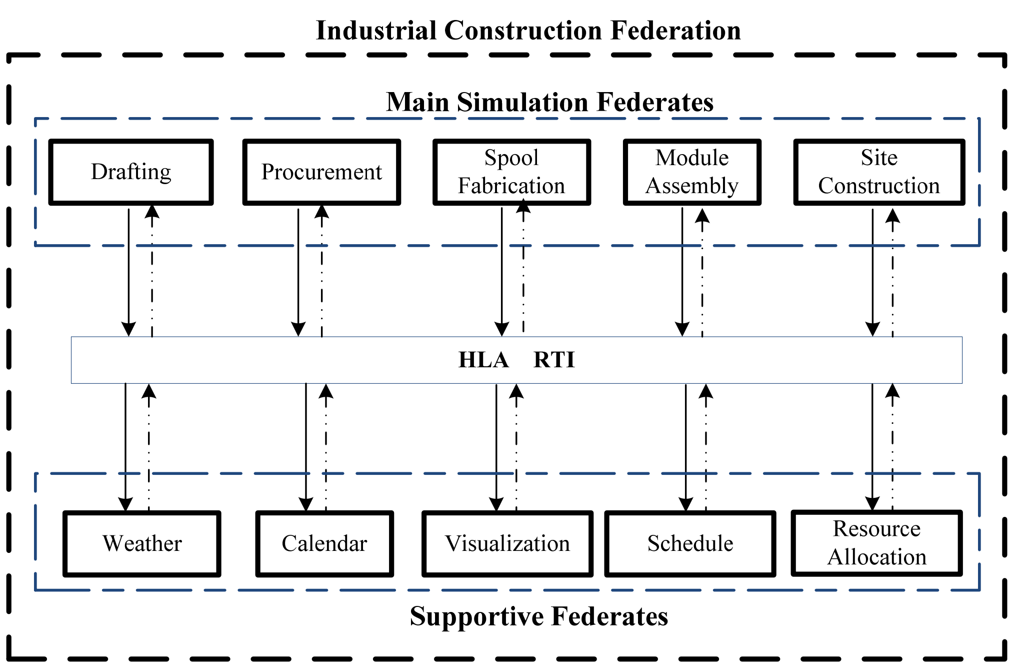


Figure 6-1 Industrial construction federation

The industrial federation is implemented in an HLA-based simulation environment, referred to as Construction Simulation Environment (COSYE) software (AbouRizk et al. 2006; AbouRizk and Robinson 2006). COSYE provides the necessary communication, information exchange, and data-sharing protocols during run-time using Run-Time Infrastructure (RTI). This RTI, developed at the University of Alberta, assures simulation synchronization, coordination, and consistency between different federates. The compatibility of the developed RTI in the current version of COSYE environment (3.5.1.10) with the IEEE HLA standard is presented in Hague (2008).

The industrial federation is one of the major federations under development at the University of Alberta funded by the Natural Science and Engineering Research Council (NSERC) of Canada/Alberta under the Construction Industry Research Chair (IRC) and Collaborative Research and Development (CRD) grants. Three graduate students including myself have been responsible for developing different federates of the industrial federation based on distributed teamwork. Different members of the team can have access to this federation or modify different federates using a source control software package called Microsoft Visual SourceSafe (VSS).

Figure 6-2 shows the solution explorer of the industrial federation. This project currently consists of 10 different projects. 'Cosye.Industrial' includes the Object Model Template (OMT) of the entire federation. 'Cosye.Industrial.FabShop', ‘Cosye.Industrial.Yard’, and ‘Cosye.Industrial.SiteConstruction’ are the main simulation federates of the spool fabrication shop, the module assembly yard, and the construction site. Other projects in the solution are service federates. ‘Cosye.Industrial.Calendar’ is a service federate that synchronizes the calendar time with RTI time by considering the weekend, shifts, and overtime hours. ‘Cosye.ResourceAllocation’ is the RA service federate that acts as the auctioneer and can be employed for optimum allocation of resources by different federates. ‘Cosye.ResourceAllocationBase’ is the RAB federate that facilitates interaction with the RA federate. Those federates that employ the RA federate should inherit from the RAB federate.

In development of the industrial federation, I have been involved in the full development of the module assembly yard, construction site, RA, RAB, schedule, and module yard’s visualization federates. Currently two other graduate students are working toward developing other federates to reach an execution level for the industry.

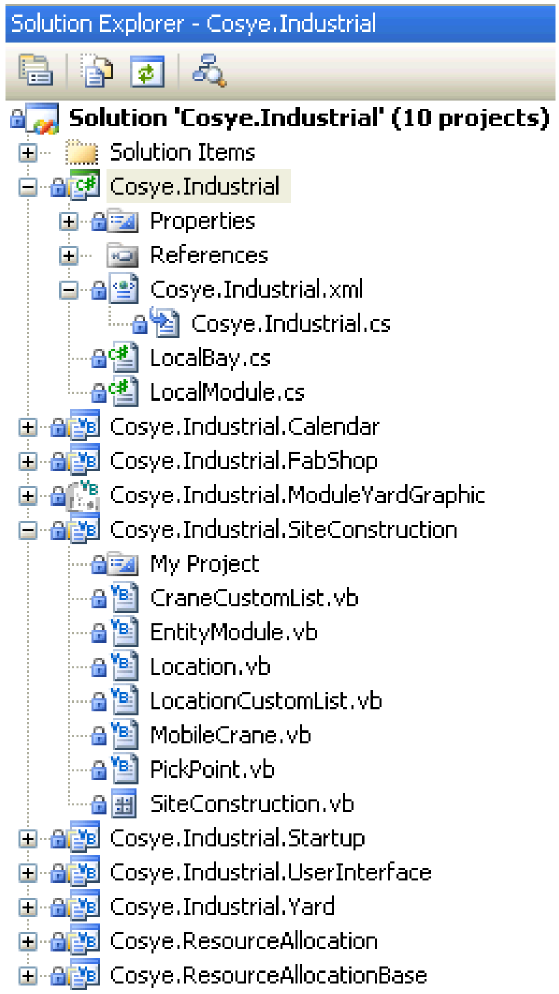


Figure 6-2 Solution explorer of the industrial federation

The schedule and module yard's visualization federates do not appear in the solution explorer in Figure 6-2 and are developed in two other projects. The schedule federate produces the schedule of the construction site, similar to the one shown in Figure 5-12 in chapter 5. This schedule federate can show the real-time schedule as the simulation model of the project is running, while the previous schedule component, in the model presented in chapter 5, only shows the schedule at the end of the simulation run. The module yard's visualization federate is also a real-time graphical federate developed using Visual Studio Tools for Office (VSTO). This federate connects to Microsoft Visio 2007 and represents the module assembly process in the module yard, while the module yard federates is running. This federate has been recently replaced by a 3D visualization federate, which is under development by another graduate student. This federate visualizes the processes happening in the module yard and construction site interactively (Figure 6-3). The weather prediction federate is also under development by the tunneling federation team members. Once that federate is completed, it can be employed in this industrial federation, too (Taghaddos et al. 2009).

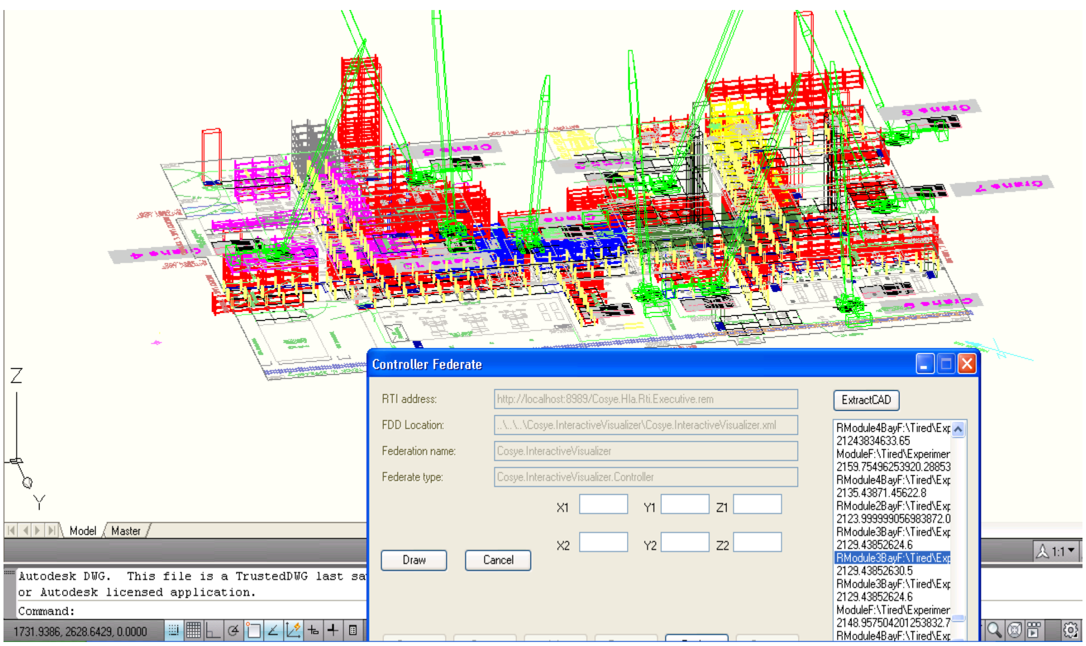


Figure 6-3 Visualization federate of the construction site

6.2.1 Time Management of Industrial Federation

Regarding the time management in HLA, a federate can be time-regulating or time-constrained, or neither, or both. Advancement of logical time in a time-regulating federate regulates the rest of the federation (specifically those federates that are time-constrained), while a time-constrained federate is constrained by the rest of the federation (specifically those federates that are time-regulating) (Kuhl et al. 1999). All the simulation federates in the industrial federation (i.e., the spool fabrication shop, module assembly yard, and site construction federates) as well as the calendar federate and the RA federate and the RAB federate are both time-regulating and time-constrained. However, the visualization federates are only time-constrained because they are dependent on the progress of other simulation federates.

For the sake of interpretability and reusability of the developed federates, a standard unit of time is chosen in all federates in the industrial federation as well as federates in other federations (e.g., tunnelling). This standard unit of time is chosen as one second, because the duration of a construction operation is rarely a fraction of a second. The look-ahead time of all federates in this industrial federation is also set to one unit (i.e., one second).

6.2.2 Object Model Template of Industrial Federation

The Object Model Template (OMT) is an essential part of the HLA framework to promote collaborative modeling, reusability, and interoperability. The OMT provides a common framework for data exchange between different federates. According to the IEEE standard (Simulation Interoperability Standards Committee (SISC), IEEE. 2001a), the OMT consists of the Federation Object Model (FOM), the Simulation Object Model (SOM), and the Management Object Model (MOM). The FOM describes the shared information regarding objects and interactions for the entire federation. The SOM specifies the shared data for a single federate. Finally, the MOM identifies objects and interactions to manage the federation.

The current version of the COSYE environment (3.5.1.10) does not yet describe SOM; however it does describe FOM using an XML format. This FOM defines the objects and interactions which are shared by the entire federation, as well as their attributes, parameters, types, and sharing methods. The FOM of the industrial federation, shown in Figure 6-4, was originally developed based on the previous research conducted by Wang (2006) in knowledge structuring of large-based industrial construction. This FOM was a comprehensive object model for the industrial federation that is beneficial for the final stages of the project development.

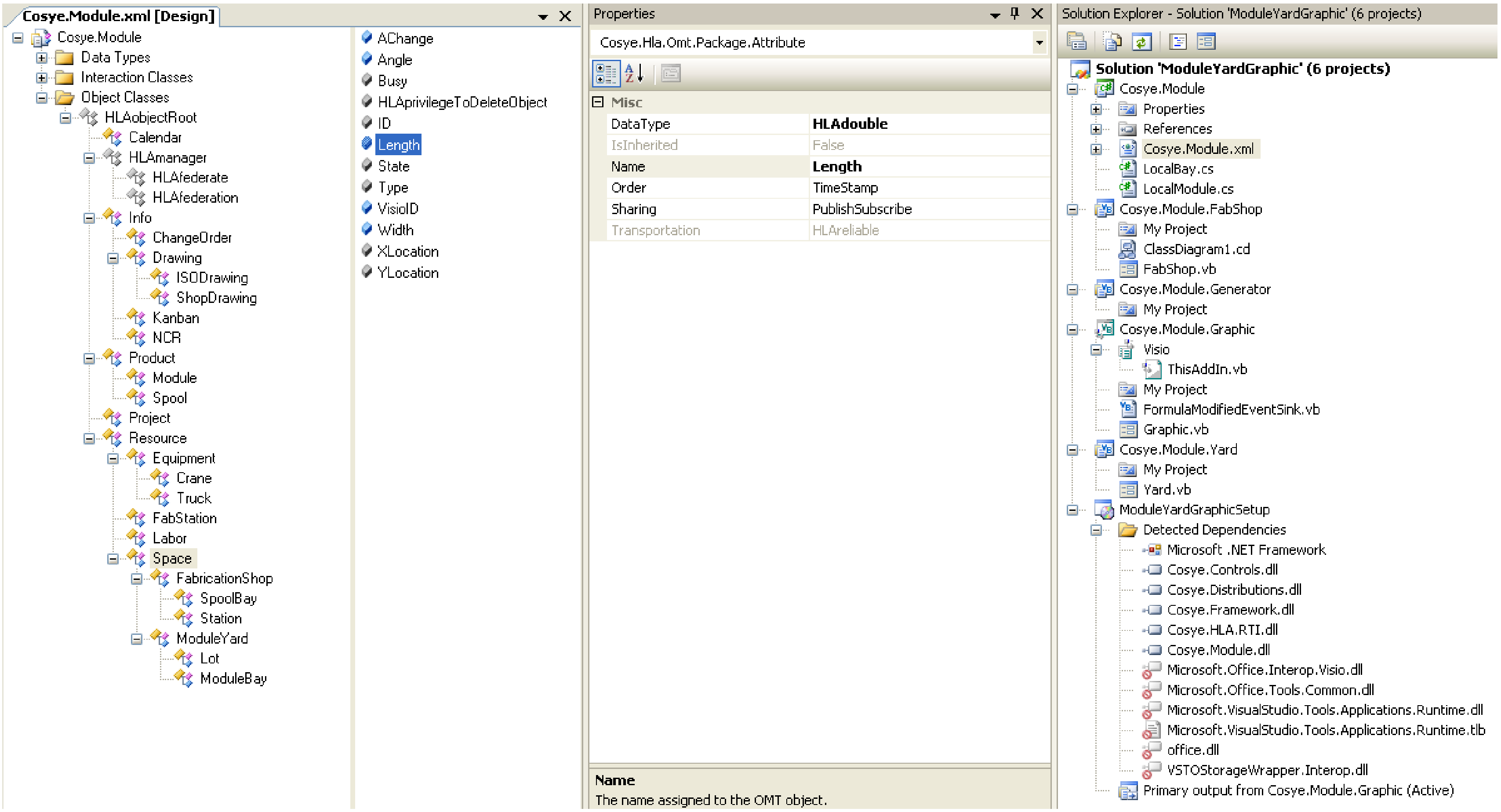


Figure 6-4 Original Federation Object Model (FOM) of the industrial federation

However, many of the objects and attributes in that FOM were redundant at the current development stage. Therefore, we decided to shrink the FOM and remove the redundant objects and attributes that were not in used in this level of development to avoid confusion between different developers (Figure 6-5). However, the removed objects can be added to the FOM as the development is progressing.

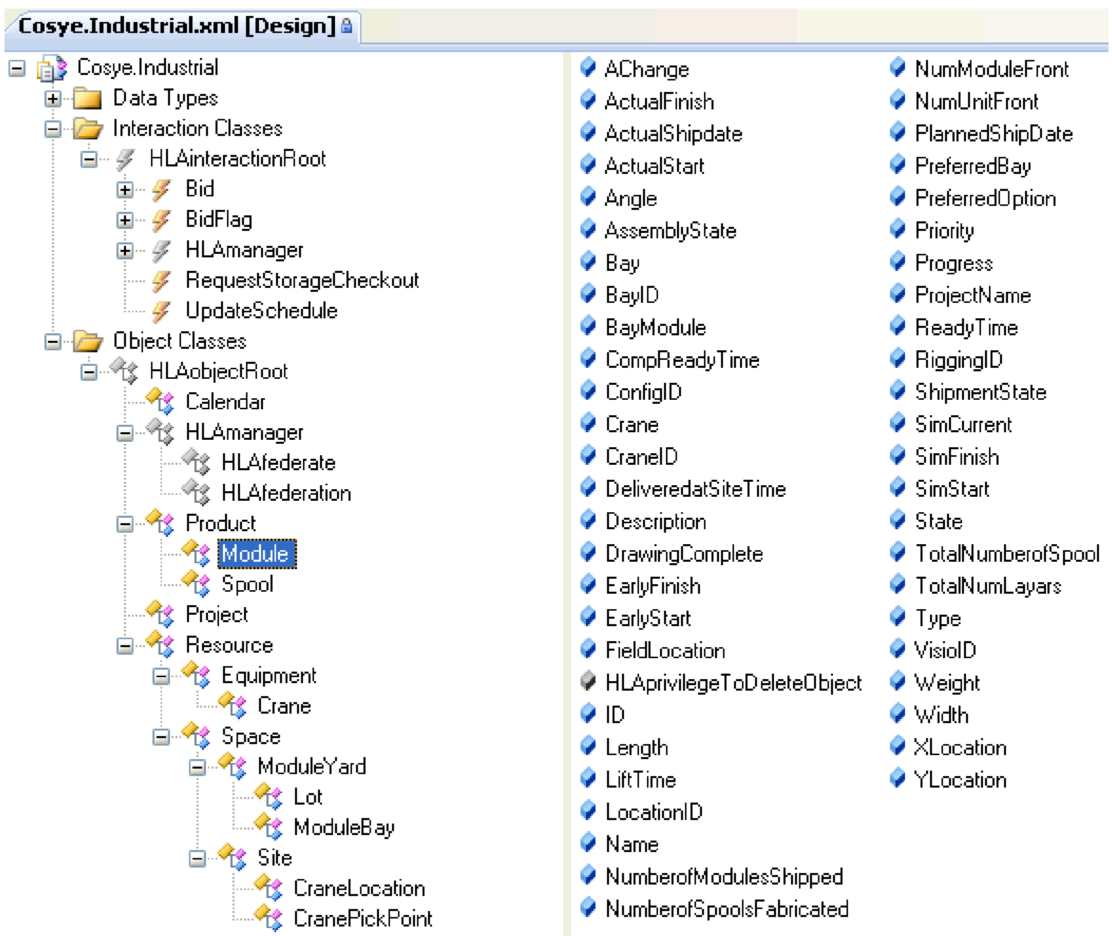


Figure 6-5 Recent Federation Object Model (FOM) of the industrial federation

It is seen in Figures 6-5 and 6-6 that the FOM has a tree structure for the objects and interactions. The first level of the new FOM contains four main entities: product, resource, calendar, and project. The calendar and the project do not have further subdivisions (i.e., children). This object possesses attributes regarding the shift, date, and general information of the project. However, the resource has a number of children including equipment (e.g., a crane) and space (e.g., space in the module yard or space at the construction site). Each child in the FOM inherits attributes of its parent. For example in Figure 6-6, 'Crane' is a child of the resource and inherits all the attributes of the resource (e.g., state attribute). Some of the resources such as the crew in the module assembly federate or the pick-point in the site construction federate are defined as internal objects inside the federate and are not defined in the FOM.

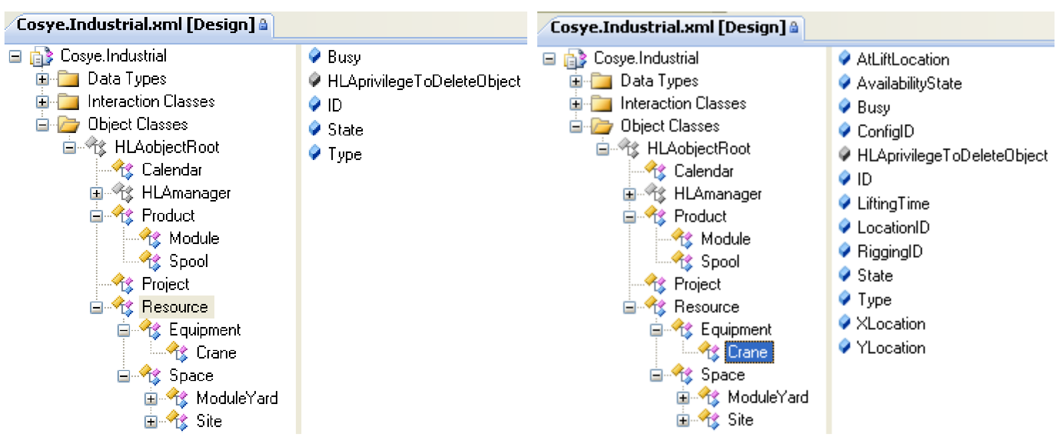


Figure 6-6 Inheritance in Federation Object Model (FOM)

The product is another main object in the parent level of the FOM. A product is defined as something that can be produced during different processes. It can be a pipe, a spool component, or a completed spool in the fabrication shop, or can be a module or division in the module assembly yard or the site construction. The relationship between these products is a 'part of' relationship, meaning that, for example, a pipe is part of a spool, or a spool is part of a module, and a module is part of a division. However, all these objects are a 'type of' product and inherit from the product in the FOM. This relationship is depicted in Figure 6-7. At this level of development, only the spool and the module are modeled in the industrial federation (Taghaddos et al. 2009).

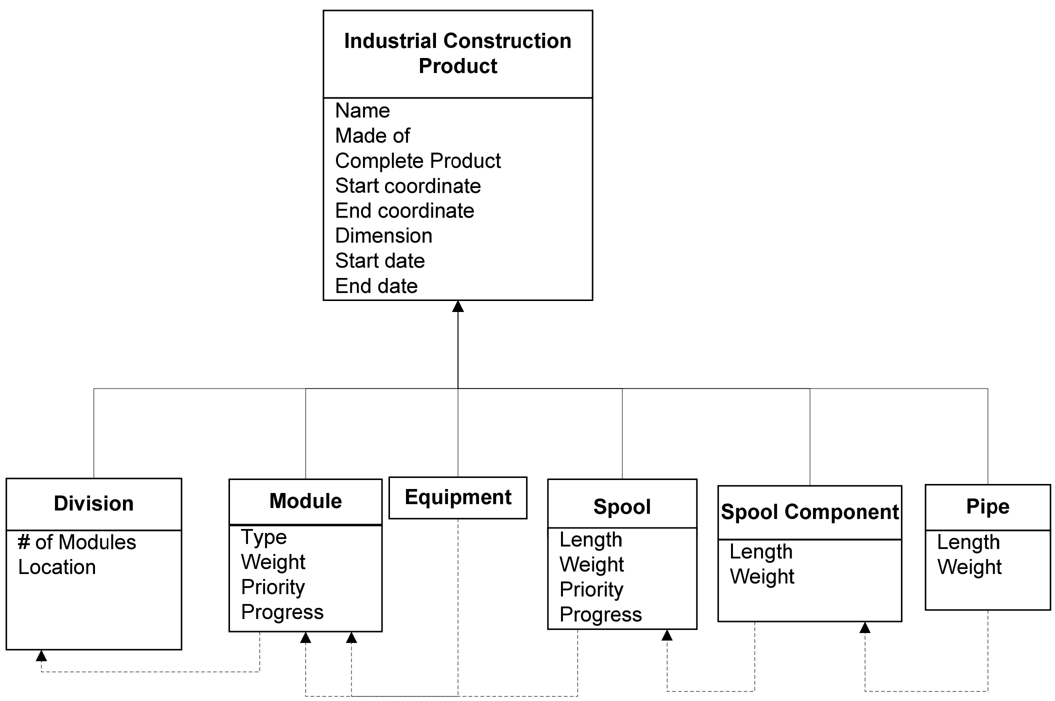


Figure 6-7 The relationship between various types of products in the industrial construction

HLA provides two means of communication for exchanging information between federates:

1) Updating an object's attribute:

Each object defined in the FOM also has several attributes. Each object may also have several instances in the federation. Each attribute of an instance of an object can be owned by a maximum of one federate and its value can be updated (i.e., modified) only by the owner federate. Once the owner federate declares an update of an attribute to the RTI, the RTI informs the new value of the attribute to all federates that have subscribed (i.e., are interested in) that attribute.

2) Sending interaction:

An interaction is a temporary message sent between different federates via RTI, while updating an attribute is a more permanent message that can be retrieved in the future. Similar to the object's attributes, an interaction has a number of parameters that are defined in the FOM. For example, the attribute class structure of the industrial federation is shown in Figure 6-8.

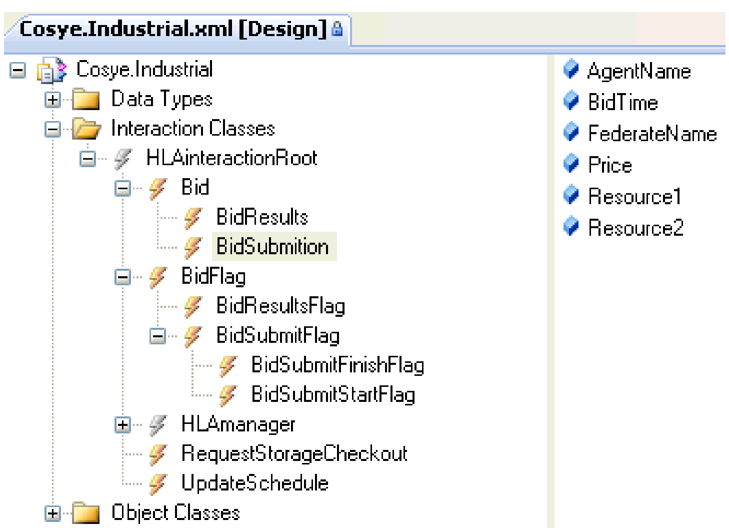


Figure 6-8 Attribute class structure of the industrial federation

6.2.3 Fabrication Shop Federate

The objective of this federate is to simulate the process of fabricating spools in the fabrication shop. Figure 6-9 shows a typical spool in the fabrication shop, which is composed of several pipes and connections that are welded together (Sadeghi and Fayek 2008). There are several stations in the spool fabrication shop including cutting, fitting, welding, Quality Control (QC), stress release, hydro test, painting, and shipping. Figure 6-10 depicts the typical processes of a spool fabrication shop (Wang 2006; Wang et al. 2009; Song et al. 2006; Song 2004).

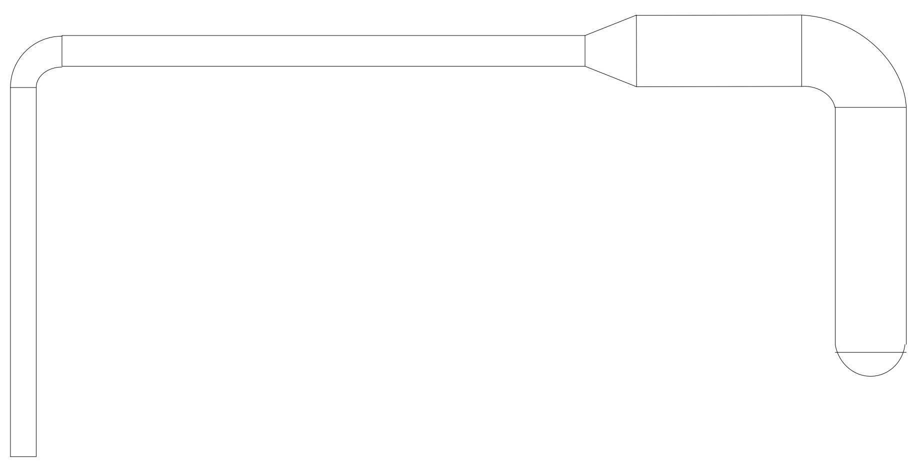


Figure 6-9 A typical spool in the fabrication shop

The fabrication shop federate reads the required information about the spools, pipes, connections, and layout of the shop from the database. Then it simulates the fabrication of a spool using various stations in the fabrication shop. Once all the spools to assemble a module are fabricated in the shop, the fabrication shop sends a message to the module yard federate to start the assembly process in the module assembly yard.

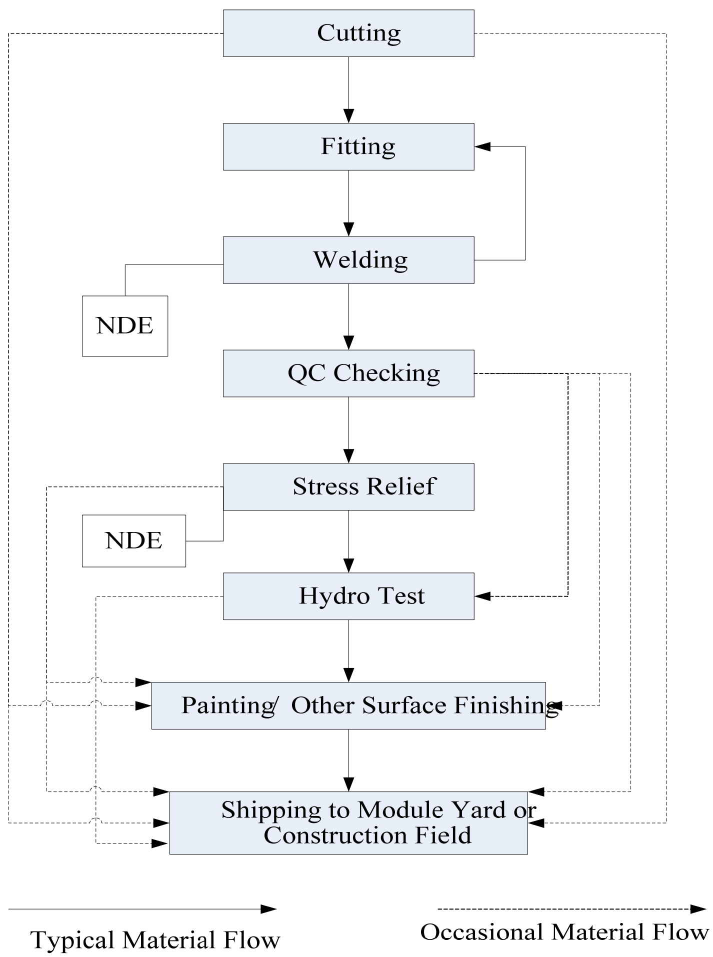


Figure 6-10 Typical fabrication process

Currently, the spool fabrication federate is preliminarily developed and assumes that a spool is composed from only one component fabricated in a station. However, one of my colleagues is involved in developing this federate to reflect the reality. Detailed discussion of this federate is beyond the scope of this thesis.

6.2.4 Module Yard Federate

The module yard federate is one of the main simulation federates designed to simulate the assembly processes of modules in the module assembly yard. The current industry practice is to start the assembly process of a module in the module yard, once all the required spools are fabricated in the spool fabrication shop. However, we have suggested the partner company change its policy in the near future by splitting the assembly process of a module into three levels, similar to building a three-story building. In this case, the assembly process of each level can be begun once the required spools of that level are fabricated in the fabrication shop. Having an integrated model of spool fabrication and module assembly enables the company to make pertinent decisions such as estimating the start time of a module or different levels of a module in the assembly yard.

The simulation model of the assembly yard federate is developed based on the standalone simulation model of the assembly yard explained in chapter 4, which was built in the Simphony 3.5 simulation environment. The Simphony 3.5 simulation engine is developed to be embedded inside the COSYE environment. Therefore, the developed simulation model of the assembly yard federate resembles the previous standalone simulation model of the assembly yard. The main differences of the simulation model in this federate and previous standalone simulation model are as follows:

1) In the previous simulation model, all the modules were generated at the beginning of the simulation model and scheduled at their Early Start (ES) time. In contrast, a module in the industrial federation is generated at the fabrication shop federate once all the required spools are fabricated in the spool fabrication shop. Then the fabrication shop federate updates the state of the module from 'fabshop' to 'ModuleYard'. Once this change is reflected to the module yard federate via RTI, this federate starts the assembly process at the earlier of either the Early Start (ES) time of the module or the current time of the module assembly federate.

2) In the previous simulation model, the entire simulation model is executed once, while the simulation model of the module yard federate is run on a step-by-step basis. At each step, the federate advances the time of simulation engine from the current time of the simulation engine until the granted time of the federate (Figure 6-11).

```
Private Sub fedAmb_TimeAdvanceGrant(ByVal sender As System.Object, ByVal e As
Cosye.Hla.Rti.TimeAdvanceGrantEventArgs) Handles fedAmb.TimeAdvanceGrant

 'Process any internal events that should occur at the current time.
    MyEngine.Simulate(fedAmb.CurrentTime)
    'Update the user interface.
        ...
    'Advance time to the time of the next internal event.
    rtiAmb.NextMessageRequest(MyEngine.TimeNext)

End Sub
```

Figure 6-11 Time advancement in the module yard federate

3) The most important enhancement in the module yard federate over the previous simulation model is separating the auctioneer's component as an

independent federate, referred to as the Resource Allocation (RA) federate. The auctioneer is a main component in the SBAP framework to solve the WDP. The RA federate is called from the module yard federate on a regular basis to maximize the social welfare of the bidding agents. Moreover, the module yard federate inherits from the Resource Allocation Base (RAB) federate to automate its communication with the RA federate. Implementation of the SBAP approach in the module yard federate and its communication with the RA federate is explained in section 6.3.

6.2.5 Site Construction Federate

The site construction federate is another main simulation federate designed to simulate the crane operations and modular construction in the site. Once a module is assembled in the module assembly yard, it is shipped to the construction site by a transporter. Then it has to be lifted to its predetermined position, once a proper mobile crane in an accessible location with suitable configuration and rigging is available and the predecessor modules (e.g., the bottom modules) are placed in advance. There are also several other constraints in this problem, elaborated in chapter 5. Similar to the module yard federate, the simulation model of the site construction federate is developed based on the standalone simulation model of the site construction in Simphony 3.5, explained in chapter 5.

Simulation modeling of the site construction based on the SBAP was discussed in detail in chapter 5. The SBAP allows the allocation of resources (e.g., mobile cranes and cranes' locations) among different agents (i.e., modules) using an auction protocol. The simulation model of the site construction federate has all three differences discussed in the previous section with the standalone simulation model developed in chapter 5: First there is a link between the module assembly yard and the construction site in the federation, while there was no link in the standalone simulation model. Moreover, simulation time advances in the site construction federate according to a step function that depends on the other federates. Lastly, the WDP in the site construction federate is solved using another RA federate. Similar to the module assembly federate, the site construction federate inherits from the RAB federate to facilitate its communication with the RA federate.

6.2.6 Calendar Federate

The main role of this generic federate is to take into consideration national holidays and long weekends, as well as the number of working hours and overtime hours during the project. This federate provides a form, shown in Figure 6-12, to input the working hours and overtime hours during the week. This form also enables the simulator to determine the holidays (e.g., long weekends) during the project. This calendar federate synchronizes the federate time with calendar time. Thus, all federates that are interested in advancing time according to the calendar can register for the updates of this federate. In practice, there is also the potential of having two different instances of this federate to consider two different calendars in the federation for different activities. For example, the working hours and shifts involved in pouring concrete may differ from the ones of piping activity.

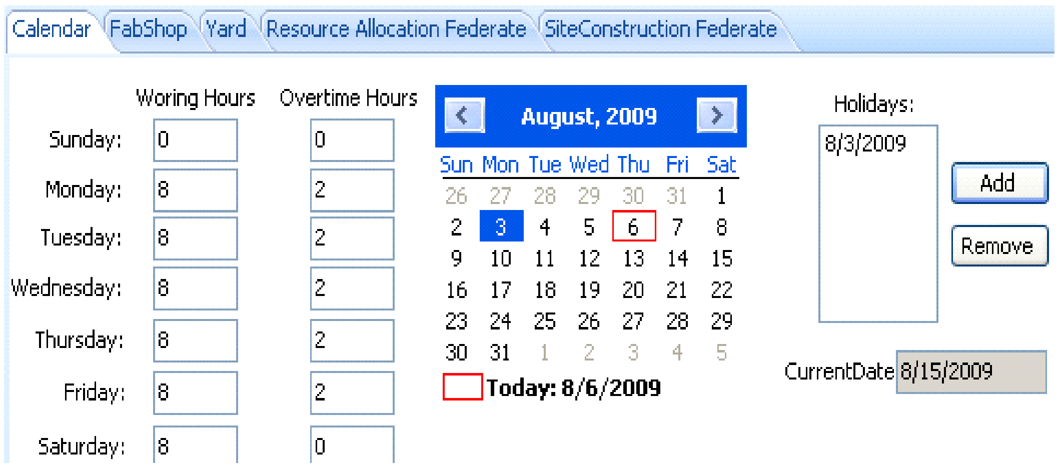


Figure 6-12 Calendar federate

This federate starts from the start date of the project, and reads the amount of working hours (regular + overtime) in each day. Then it updates attributes of an instance of calendar object, called 'MyCalendar', every working day as it is shown in Figure 6-13.

```
Private Sub fedAmb_TimeAdvanceGrant(ByVal sender As System.Object, ByVal e As
Cosye.Hla.Rti.TimeAdvanceGrantEventArgs) Handles fedAmb.TimeAdvanceGrant
        MyCalendar.UpdateAttributeValues(e.theTime + fedAmb.Lookahead)
        TxtCurDate.Text = MyCalendar.CurrentDate.ToShortDateString
        MyCalendar.CurrentDate = NextWorkingDay(MyCalendar.CurrentDate)
        MyCalendar.DayHours = DateHours(MyCalendar.CurrentDate)
        MyCalendar.DayNo = MyCalendar.DayNo + 1
        rtiAmb.NextMessageRequest(e.theTime + MyCalendar.DayHours * 3600)
End Sub
```

Figure 6-13 Time advancement in the calander federate

6.3 Developing Generic Resource Allocation Federates

It is mentioned in chapter 3 that the auction protocol contains two main stages. First, the bidding agents have to submit their bids for different feasible combinations of resources. Second, an auctioneer has to be embedded in the system to solve the Winner Determination Problem (WDP) and to award the winning agents. An independent Resource Allocation (RA) federate is designed in the federation to solve the WDP. This RA federate communicates with the main simulation federate, reads different bid alternatives of different agents, employs a combinatorial optimization algorithm, and allocates available resources to the agents so that the social welfare is maximized (or the total cost is minimized). The Resource Allocation Base (RAB) federate is also developed to automate the communication with the RA federate. These two federates are presented in the following sections.

6.3.1 RA-Base Federate

The Resource Allocation Base (RAB) federate has been designed to facilitate communication with the Resource Allocation (RA) federate. The RAB federate sends and receives several interactions to pass the information between the main simulation federate and the RA federate. The RAB federate loops through different combinations of bidding agents, sends their bid alternatives to the RA federate, and returns the assigned resource of the winning agents to the simulation federate. All federates that employ the SBAP can easily inherit this federate to automate their communication with the RA federate.

It is mentioned in chapter 3 that the SBAP framework proposes holding regular (e.g., daily) auctions to allocate available resources among bidding agents. In each auction, those bidding agents that are eligible to participate in the auction are stored in a list called 'BiddingEntityList'. Then the simulation model has to determine the utility function (or bidding price) of the agents for different combinations of resources. Different bid alternatives of the agents, including feasible resource combinations and the bid price, are stored in a list, called BidList. After determining all the agent's bids in BidList, the model loops over these bid alternatives of agents and submits the bids (i.e., resource combinations and bid price) of agents to the auctioneer. This process is done by sending three types of interactions, shown by the solid lines in Figure 6-14, including:

1) 'MyBidSubmitStartFlag' interaction is sent as a flag to indicate the start of bid submission interactions. This interaction also declares the auction time, name of the main simulation (e.g., module yard) federate, the optimization operation (i.e., min or max), number of bidding agents, number of the first type of resources, and number of the second type of resources.
2) 'MyBidSubmission' interaction is sent to declare the details of each individual bid including the auction time and name of the main simulation (e.g., module yard) federate, name of the agent, name of the resources (resource 1 and

resource 2) to bid on, and the bid price of the agent for the proposed resource combination.

3) 'MyBidSubmitFinishFlag' interaction is sent as a flag to indicate the end of bid submission interactions. This interaction also declares the auction time and name of the main simulation (e.g., module yard) federate.

In all these types of interactions, the auction time and name of the main simulation federate is sent to avoid confusion in the RA federate. This is because the RA federate may receive these types of interactions from different federates or at different times. Therefore, declaring the name or time of these interactions assists the RA federate in distinguishing the interactions sent from different federates or at different times.

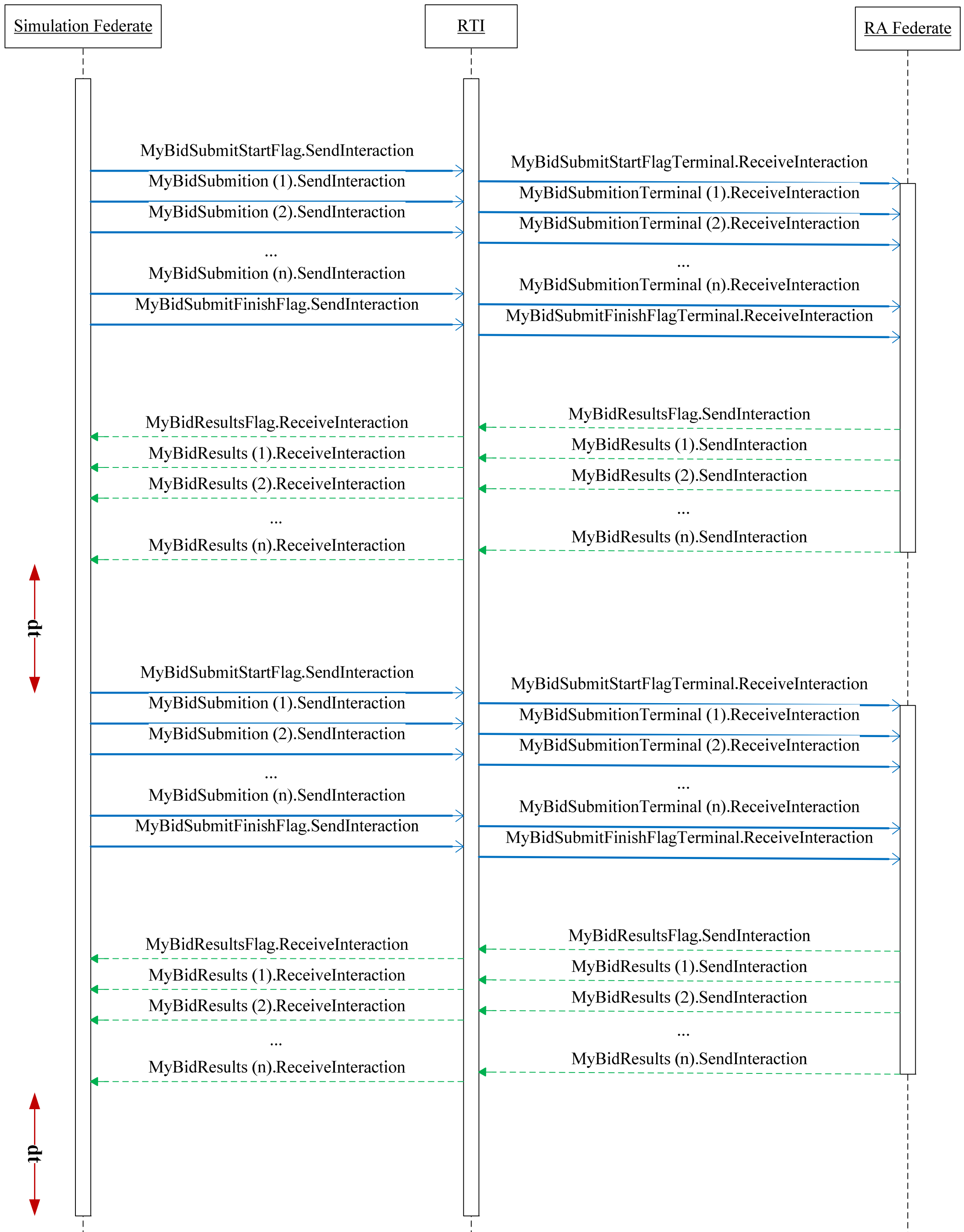


Figure 6-14 Sequence Diagram of communication between the RA federate and simulation federate

When the RA federate receives the above interactions from a main simulation federate, it solves the WDP. Then it returns some interaction to the simulation federate to inform it about the winning agents and assigned resources. These interactions, shown by the dash lines in Figure 6-14, include:

1) 'MyBidResultsFlag' interaction is sent as a flag to indicate the start of bid results interactions. This interaction also declares the auction time and name of the main simulation (e.g., module yard) federate that the results are sent for.
2) 'MyBidResults' interaction is sent to declare details of bid results including the auction time and name of the main simulation federate (e.g., module yard federate), name of the winning agent, and name of the assigned resources (resource 1 and resource 2).

Similar to the previous types of interactions, the auction time and name of the main simulation federate are sent in all results interactions. Therefore, only the simulation federate expecting for the results will make use of these bid results. It is also worth mentioning that the RA federate does not need to send a flag to declare the end of bid results. Whenever the federate receives an interaction for an awarded agent, it investigates allocation of resources to the winning agent. This process ends when the interaction for the last awarded agent is received.

Figure 6-15 depicts the class diagram of the RAB federate. This class diagram includes two major classes that should be inherited by any federate developed based on the SBAP: 1) ResourceAllocationBase class, and 2) BiddingAgent class.

1) In the COSYE environment, generally, the main class of most federates inherits from the FederateControl class. However, the main class of the simulation federates that are structured based on the SBAP framework should have some common properties to communicate with the RA federate including the federate name, bid time, mybidsubmitstartflag, myBidSubmission, mybidsubmitfinishflag, mybidresultsflag, and mybidresults. It should also have two methods to go through the bid alternatives of the agents and submit their bids, and finally allocate resources to the winning agents. To facilitate the communication with the RA federate, these properties and methods are defined in the ResourceAllocationBase class of the RAB federate. Therefore, any federate structured based on the SBAP framework should inherit the ResourceAllocationBase class of the RAB federate instead of the FederateControl class to automate its communication with the RA federate.

2) Moreover, an entity in the simulation models inherits from the entity class in Simphony simulation environment. However, those entities that are going to bid for the resources in the SBAP framework always have some generic attributes including the name, assigned resource 1, assigned resource 2, priority, different resource combinations to bid on as well as the corresponding bidding prices for each combination. The priority attribute is required to sort the agents and allocate resources to them

according to their priority. These attributes are all defined in a class of the RAB federate, called BiddingAgent class. Therefore a bidding entity in the SBAP framework should inherit from the BiddingAgent class.

In the RAB federate, a CastingList class is also designed to convert the type of the bidding entity in the simulation model to the type of BiddingAgent and vice versa. This class is developed because the generic property of BiddingAgentsList in the base class is a collection of BiddingAgent, while the bidding entity in the simulation federate is supposed to be added to this list. This CastingList class helps to convert the type of bidding entity to BiddingAgent autamitically. In other words, we can easily add the bidding entity inside the simulation model to the BiddingEntityList.

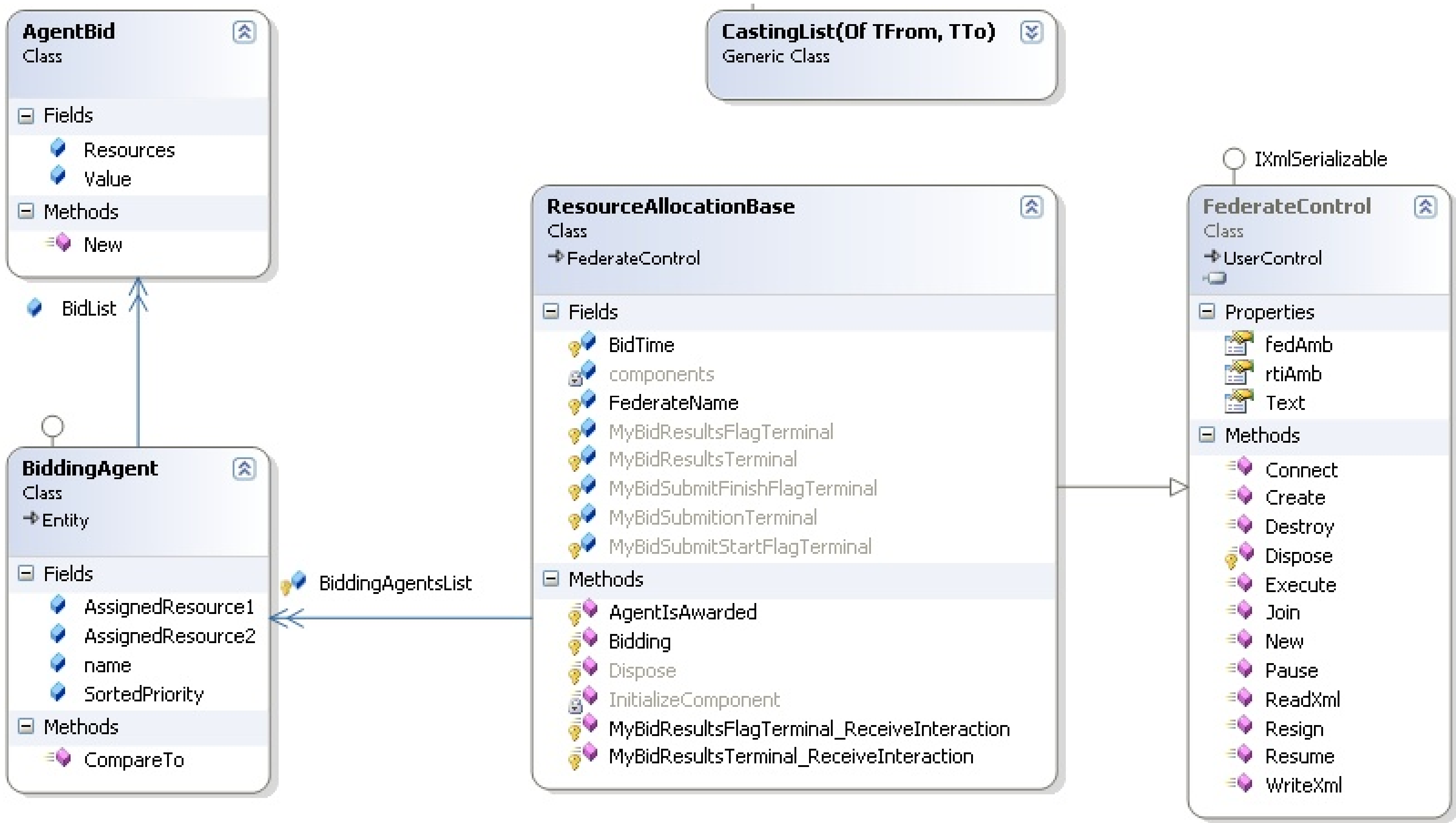


Figure 6-15 Class diagram of Resource Allocation Base (RAB) federate

Figure 6-16 shows the inheritance of the module yard and site construction simulation federates from the RAB federate. Then the simulation federate has to put all the bidding entities in the BiddingEntityList and determine their bidding price for feasible combinations of resources. The rest of communication is performed automatically by calling the Bidding function with the required parameters. The input parameters of this function are the optimization operation (minimum or maximum), the number of the first type of resources, and the number of the second type of resources. For instance in the yard federate and site construction federate we have:

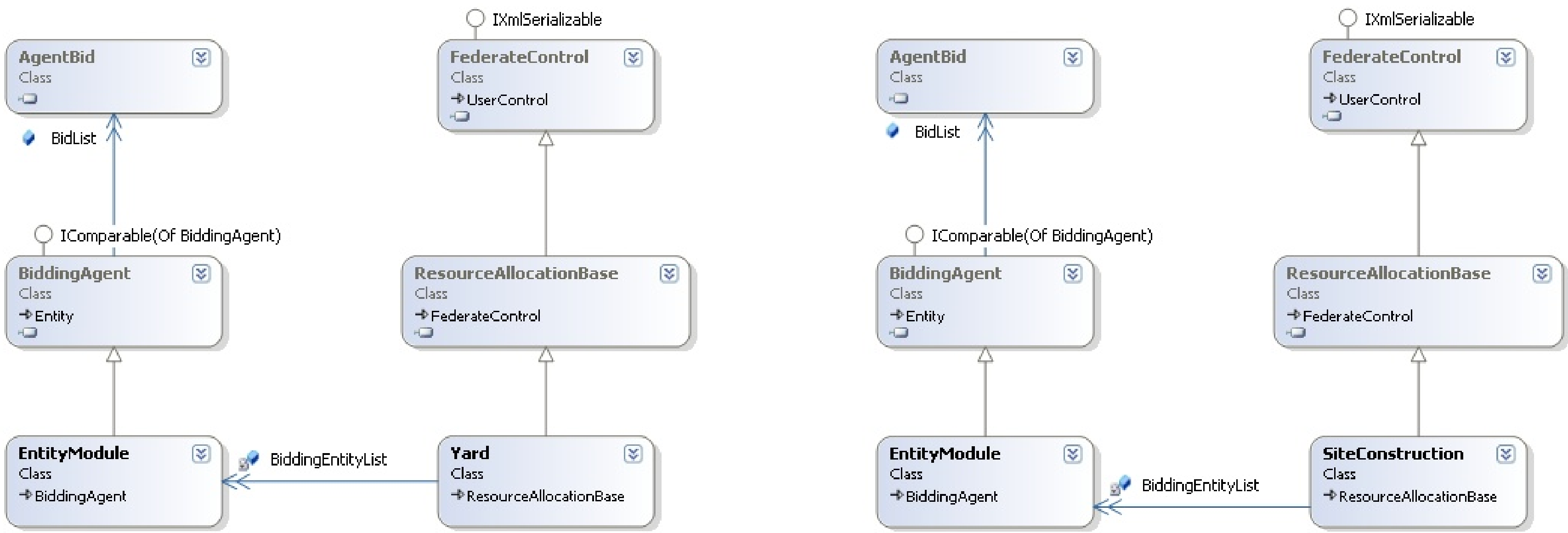
Bidding(OptimizationOperation.Max, NumBays, 1), @ Assembly yard federate (6-1)

Bidding (OptimizationOperation.Min, MyCraneResource.Count,
MyLocationResource.Count) @ Site construction federate (6-2)

Where;

In the yard federate, the optimization parameter is the minimum; the number of the first type of resource is the number of bays; and the number of the second type of resource is 1 (i.e., nothing).

In the site construction federate, the optimization parameters is the maximum; the number of the first type of resource is the number of cranes, and the number of the second type of resource is the number of cranes' locations.



a) Yard Federate b) Site Federate

Figure 6-16 Inheritance of the module yard federate and site federate from the Resource Allocation Base (RAB) federate

6.3.2 Resource Allocation Federate

This federate is designed to act as an auctioneer to allocate resources among the bidding agents. In other words, this federate is supposed to solve the Winner Determination Problem (WDP) in the combinatorial optimization. Currently this federate can allocate two types of resource (e.g., crane and location) to a number of bidding agents based on a greedy algorithm or ascending-auction algorithm. This federate can easily be expanded to allocate n type of resources to several agents using a combinatorial optimization. Figure 6-17 shows the schematic view of this RA federate in the industrial federation.

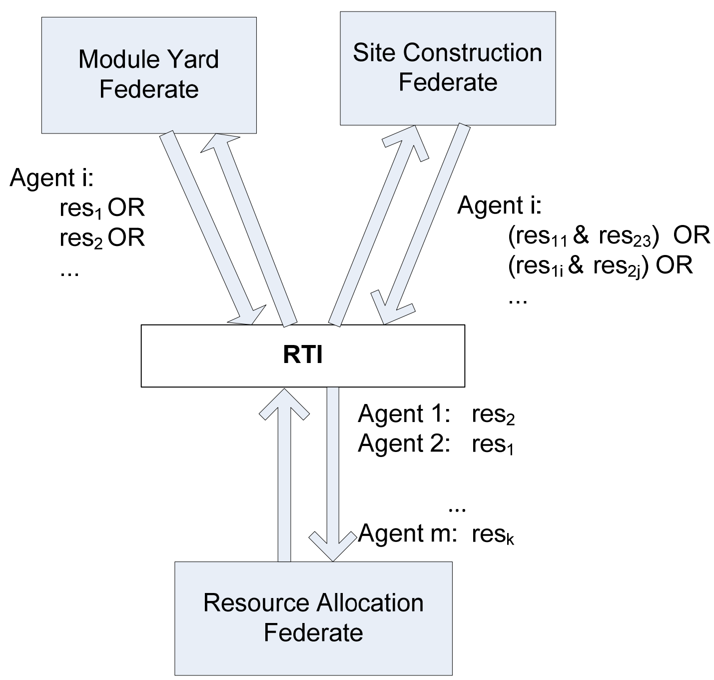


Figure 6-17 Resource allocation federate

This federate was initially designed to pass information with the simulation federates through a database. Then the database interface to pass information was replaced by the HLA interactions. The allocation algorithm in this federate is summarized below:

Initialize

1. The federate receives a ' BidSubmitStartFlag' interaction,

1.1. Create a three-dimensional array to store agents name in the first dimension, name of the first type of the resource in the second dimension, and the name of the second type of the resource in the third dimension.

1.2. Set the initial value of all the cells in this array to MaxAcceptablePrice

1.3. Set the federate name, operation type, and bid time values from the received BidSubmitStartFlag interaction

1.4. Create a two-dimension table to store the agent names and assigned resources

Receiving bids

2. The federate receives a ' BidSubmission' interaction,

2.1. Make sure the name and time of the interaction matches with federate name, and bid time values

2.2. If Operation type is minimum then set the corresponding cell in the 3D array with the price value of the received BidSubmission interaction, *ElseIf Operation type is maximum then set the corresponding cell in the 3D array with the -1*price value of the received BidSubmission interaction.*

Solving WDP:

3. *Solving the WDP, once the federate receives an 'BidSubmitFinishFlag' interaction,*

Awarding winning agents

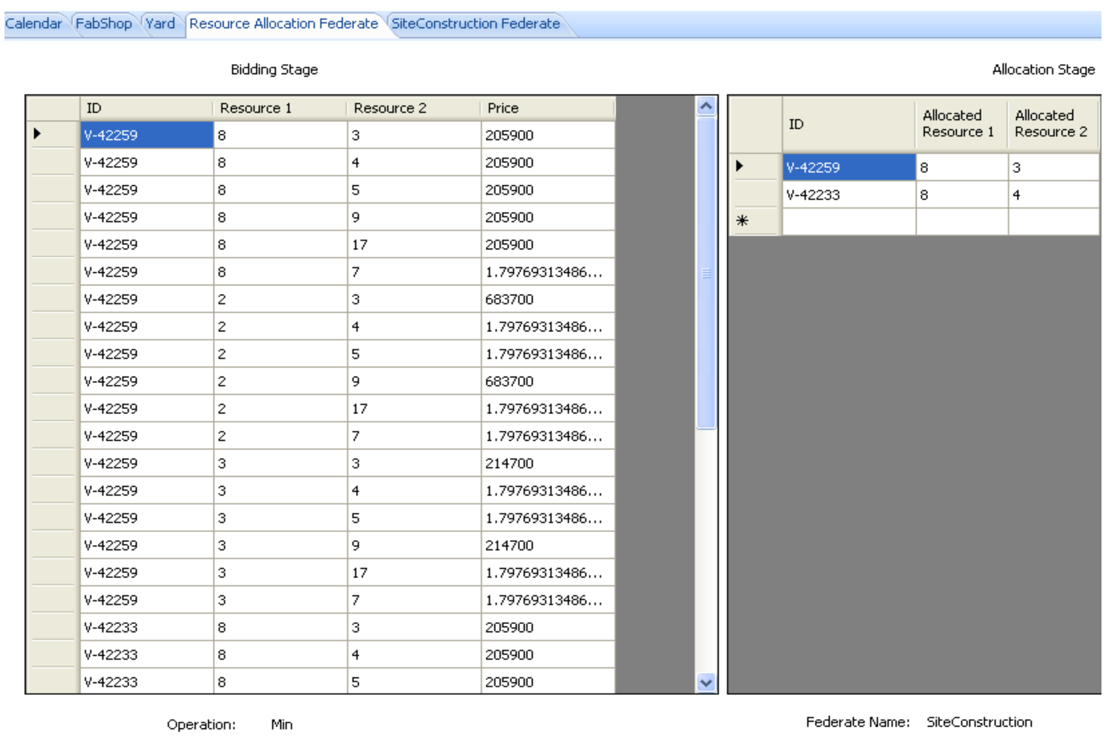
4. *Send the BidResultsFlag flag with the name of the simulation federate and bid time*

4.1. *Loop over the received agents*

4.1.1. *If the assigned resources is not null, send an BidResults interaction with the name of resources (resource type 1 and 2)*

The WDP can be solved by various combinatorial optimization algorithms. In this project, a greedy algorithm is employed to solve the WDP in general conditions. This algorithm sorts the agents according to a criterion (e.g., start time) and allocates the best combination of resources to an agent at a time. However, the greedy algorithm never reconsiders a bid after its allocation and proceeds until the available resources are assigned to the agents or all the bids of the agents are reviewed. The greedy algorithm works well in practical large WDPs, but does not result in the optimum solution with the maximum social welfare necessarily. The ascending-auction algorithm is another appealing algorithm to solve the WDP. This algorithm, explained in chapter 3, adjust the price of the resources in several iterations until the supply and demand reaches equilibrium. The WDP is solved in this study when the agents request for maximum two combinations of resources.

Figure 6-18 shows results of one of the auctions in the RA federate. Currently, the RA federate employs a greedy search algorithm to allocate the resources among agents. However, the optimization algorithms can be advanced in the next phase to incorporate various combinatorial optimization algorithms (e.g., dynamic programming).



ID	Resource 1	Resource 2	Price
V-42259	8	3	205900
V-42259	8	4	205900
V-42259	8	5	205900
V-42259	8	9	205900
V-42259	8	17	205900
V-42259	8	7	1.79769313486...
V-42259	2	3	683700
V-42259	2	4	1.79769313486...
V-42259	2	5	1.79769313486...
V-42259	2	9	683700
V-42259	2	17	1.79769313486...
V-42259	2	7	1.79769313486...
V-42259	3	3	214700
V-42259	3	4	1.79769313486...
V-42259	3	5	1.79769313486...
V-42259	3	9	214700
V-42259	3	17	1.79769313486...
V-42259	3	7	1.79769313486...
V-42233	8	3	205900
V-42233	8	4	205900
V-42233	8	5	205900

ID	Allocated Resource 1	Allocated Resource 2
V-42259	8	3
V-42233	8	4

Figure 6-18 A typical auction in the RA federate

6.4 Summary

Industrial construction is a large-scale process involving several supply chains, including drafting, material procurement, spool fabrication, module assembly, and site construction. Modeling the entire system at an appropriate level of detail cannot be achieved using traditional construction simulation modeling tools. In this study, High Level Architecture (HLA) was employed to decompose the simulation model into smaller and more manageable components, known as federates. These federates are developed independently and connected together in the Construction Synthetic Environment (COSYE), an HLA-based simulation environment developed at the University of Alberta. Currently, the industrial federation includes several main federates (e.g., procurement, fabrication shop, module yard, site construction) and some supportive federates (e.g., calendar, resource allocation, visualization).

In addition to developing the main simulation federates (e.g., module yard, site construction federates), this chapter focuses on developing two federates to facilitate an optimum allocation of resources in construction federates. Those simulation federates structured based on the SBAP can employ the Resource Allocation (RA) federate to act as an auctioneer and to solve the WDP. A Resource Allocation Base (RAB) federate is also designed to automate the communication with the RA federate. This federate should be inherited by the simulation federates communicating with the RA federate (i.e., auctioneer) to submit bids and award the winning agents. These federates are validated in the industrial federation and work successfully. However, there is room for improvement that will be mentioned in the next chapter.

CHAPTER 7. Conclusions and Recommendations

7.1 Research Summary

The research presented in this thesis was motivated by the lack of an organized structure to facilitate effective resource scheduling in construction projects. Scheduling networks (e.g., CPM, PERT) were developed about half a century ago and have become popular in the construction industry. However, these techniques have several limitations in modeling the dynamic and stochastic nature of large scale construction projects. Artificial intelligence, simulation modeling, or their combinations have also been employed to enhance allocation of resources in construction projects. However, they are not embraced in the construction industry for modeling realistic large scale projects. This unpopularity is due to various reasons including limitations of time and budget in the planning phase, complexity of such optimization or simulation models, and lack of future reusability for such complicated models. Thus, the main objective of this research was to employ artificial intelligence inside construction simulations for facilitating the resource allocation process in various construction projects. This primary goal was divided into three auxiliary objectives.

The first auxiliary objective, addressed in chapter 1, was:

I. To develop a resource scheduling framework, referred to as a Simulation Based Auction Protocol (SBAP) framework, based on auction protocols and discrete event simulation modeling.

This objective was achieved in chapter 3 by designing the SBAP framework to solve resource scheduling problems in large-scale construction projects. This hybrid framework integrates discrete event simulation modeling and centralized Multi-Agent Resource Allocation (MARA), which is referred to as an auction protocol. The SBAP can employ different combinatorial optimization techniques to allocate resources among the bidding agents. In this framework, auctions are held on a regular basis, whereby some of the agents bid for different resources or combinations of resources so that their individual welfare is maximized. Agents can represent jobs in a factory scheduling problem or construction units (i.e., modules) in the modular construction of an industrial plant. The degree of satisfaction of an agent over a certain resource allocation is represented by its individual welfare (i.e., bidding price). Once agents submitted their bid price over different combinations of resources, an auctioneer solves the Winner Determination Problem (WDP) to maximize the social welfare or minimizing the total cost based on a combinatorial optimization. Then it awards winning agents and assigns the available resources to them. This thesis first elaborates and validates the SBAP in chapter 3. Then it explains the capability and flexibility of the SBAP by implementing it in two large-scale construction case studies in chapters 4 and 5.

The second auxiliary objective, indicated in chapter 1, was:

II. To model a large-scale industrial construction application containing various supply chains and to spilt the model into manageable components (i.e., federates) by employing High Level Architecture (HLA).

This objective was achieved in chapter 6 with the development of a large scale construction case study that involves different supply chains including drafting, material procurement, spool fabrication, module assembly and site construction. This large-scale model is developed by employing the High Level Architecture (HLA). HLA allows splitting of a large scale model, known as a federation, into a number of manageable components, known as federates, while maintaining interoperability between them. In the industrial federation, each of the supply chains has been (or will be) modeled as an independent federate that serves as a main simulation federate. Each federate works independently even on different computing processes. Some supportive federates are also developed in this federation, such as the calendar federate and the visualization federate, that serve one or several federates in the industrial federation or even other types of federations (e.g., the tunneling federation)

The last and most significant auxiliary objective, stated in chapter 1, was:

III. To develop a generic resource allocation federate that is reusable in various construction simulations by employing HLA and the SBAP framework.

The Resource Allocation (RA) federate(s) employ artificial intelligence to facilitate optimum allocation of resources in different federates. These RA federates are designed using a generic approach to serve different federates in industrial construction federation or even other federations. In this project, two types of RA federate are developed.

a) First, an RA federate is developed that acts as an auctioneer. Once all the bidding agents submit their bids for different combinations of resources, this federate solves the WDP to maximize the social welfare. Then it awards some of the agents and returns the information of the simulation federate.
b) Second, an RAB federate is designed to automate the communication with the RA federate. This federate acts as a parent federate for the simulation federates that employ the SBAP. In other words, the simulation federates developed based on the SBAP have to inherit from the RAB federate. Then, both sending information about the bids and receiving information about the results of bids are done automatically.

In short, these three developmental steps correspond to three auxiliary objectives identified in chapter 1 of the thesis. This project has resulted in an excellent platform to embed artificial intelligence inside the simulation model. This framework assists the developers to reuse the developed artificial intelligence and simulation models in the future projects.

7.2 Research Contributions

The presented research has led to numerous contributions to enhance allocation of resources. The developed framework has also empowered construction simulation in modeling large-scale projects. The main contributions of this thesis can be summarized in the themes below:

1) Introducing a Simulation Based Auction Protocol (SBAP) for allocating resources in large-scale construction projects.
2) Developing a comprehensive simulation federation of industrial construction based on High Level Architecture (HLA).
3) Developing generic resource allocation federates for facilitating allocation of resources in simulation federates that employ the SBAP framework.

   The developed resource allocation is generic and ready to use for all simulation models developed based on the proposed SBAP framework, as long as they are designed as federates based on HLA standards. The use of the RA federate is not exclusive to a specific construction project.

7.3 Lessons learned

Many lessons can be drawn from this research project. The list below presents a summary of lessons learned during the development of this thesis. Considering these lessons is very useful in making the development of similar research projects both easier and faster.

1) Planning before starting the research project and at the beginning of each stage of the project is critical. This planning stage is also critical prior to simulation modeling of construction projects. Like many engineers, in the past I began developing simulation models or designing databases with implementation, neglecting any proper planning using a simple algorithm or flowchart, UML, or Entity-Relationship (ER) diagrams. During this project, I found that the planning stage is not only not a waste of time, but actually significantly reduces the time needed for the reimplementation or modification of the model or database.
2) Team development requires special consideration. This project implements HLA to develop distributed simulation models in construction. Although High Level Architecture (HLA) offers several advantages to the simulation world, team development may involve new challenges.
    a. Prior to development, the team should focus on early planning to determine a well defined strategy for the function and boundaries of the federation and various federates.

b. Responsibilities of each developer should be well defined. Providing a development schedule can help to synchronize the efforts of various team members to achieve similar levels of maturity in different federates. Significant scheduling effort in this kind of parallel development is critical for efficient modeling of a large-scale construction project.

c. Reading, understanding and following the IEEE standards is crucial for developing the HLA-compliant simulation models (Simulation Interoperability Standards Committee (SISC), IEEE. 2000), (Simulation Interoperability Standards Committee (SISC), IEEE. 2001a), (Simulation Interoperability Standards Committee (SISC), IEEE. 2001b). These standards describe the general principles of HLA, the communication methods and synchronization with RTI. They also explain the structure of the Object Model Template (OMT) which promotes reusability and interoperability of federates. These standards are particularly beneficial for the developer team to define the "required" objects and the interactions in the federation.

d. In order to reduce confusion among the developers, internal objects of federates should not be defined inside the Object Model Template (OMT) of the entire federation. The developers should pay extra attention to the structure of OMT at the beginning, because changing the structure of the OMT can be very costly (i.e., time-consuming) after development has begun.

3) Borrowing ideas from other disciplines (multi-disciplinary collaboration) can have a key impact on improving construction projects. Due to various reasons, the efficiency of construction projects is far behind other disciplines such as manufacturing, computing science, or the military industry. In this project, we borrowed several ideas and methodologies (e.g., HLA, artificial intelligence, MARA or auction theory) from other industries and used them successfully in construction projects.
4) Active interaction with the construction industry is significant in the applicability of research projects. This collaboration requires further education and creating enthusiasm about research projects in the staff of construction companies. In the presented research project, the results of the main case studies explained in chapter 4 and 5 could not have been achieved without such active collaboration. Engineers inside PCL Industrial Management Inc. were actively involved in the development of the simulation-based schedule of the module assembly yard and heavy lift planning. In hindsight, a better active collaboration could have resulted in further improvement of the tower crane allocation project presented in the appendix.

7.4 Recommendations for Future Research

This research exposes numerous areas that have the potential for further study. The areas which should be investigated in greater detail are:

1) Determine a bidding price (i.e. utility function) that represents different affecting factors in the decision making process is a critical step in the proposed framework. Further research is required to formulate the utility function for various construction projects to facilitate the employment of this framework in the construction industry. I also recommend experimentation with the SBAP parameters, and improvement of the framework by calibrating the parameters (e.g., ΔT) for different construction projects.
2) Enhancement of the RA federate by incorporating:
    a. Different combinatorial optimization algorithms (e.g., dynamic programming).
    b. Other types of auction protocols (e.g., multi-unit auctions, position auctions)
    c. Various bidding languages
    d. Other methods (e.g., egalitarian) to calculate social welfare
3) Improve the communication between federates by exchanging bidding information with XML; currently the main simulation federates communicate with the Resource Allocation (RA) federate through sending interactions.

4) Employ the Resource Allocation (RA) federate in other industrial construction federates (e.g., fabrication shop) or even other types of federations. The RA federate is successfully employed in planning the module assembly yard as well as heavy lift planning. However, this RA federate has the potential to be employed in other federates.
5) Define some resource utilization indexes (e.g., resource leveling) in the agents' utility function to consider the fluctuation of resources and to allocate the resources more efficiently.

Bibliography

AbouRizk, S. M., Al-Hussein, M., and Mohamed, Y. (2006). "Development of Synthetic Environments for Select Construction Applications." *Collaborative Research and Development Grant*.

AbouRizk, S. M., and Robinson, F. A. (2006). "Synthetic Environments for Construction Planning and Control." *NSERC Industrial Research Chair Research Program*.

AbouRizk, S., and Mohamed, Y. (2002). "Optimal construction project planning." *Proceedings of the 34th conference on Winter simulation,* IEEE, 1704-1708.

AbouRizk, S. M., and Mohamed, Y. (2000). "Simphony - an integrated environment for construction simulation." *Proceedings of the 32nd conference on Winter simulation,* IEEE, 1907-1914.

Ahuja, H. N., Dozzi, S. P., and AbouRizk, S. M. (1994). *Project management : techniques in planning and controlling construction projects.* J. Wiley, New York.

Anonymous (2010). "StateMaster - Encyclopedia: Framework." http://www.statemaster.com/encyclopedia/Framework (5/23/2010, 2010).

Arrow, K. J., Sen, A. K., and Suzumura, K. (2002). *Handbook of Social Choice and Welfare.* North Holland.

Bellifemine, F. L., Caire, G., and Greenwood, D. (2007). *Developing Multi-Agent Systems with JADE.* John Wiley, University of Michigan.

Bennett, C., and Ditlinger, S. (1996). "Bechtel automated lift planning system." *Proceeding of Robotics for challenging environments,* 401-409.

Bertsekas, D. (1988). "The auction algorithm: A distributed relaxation method for the assignment problem." *Annals of Operations Research,* 14(1), 105-123.

Bertsekas, D. P. (1992). "Auction algorithms for network flow problems: A tutorial introduction." *Computational Optimization and Applications,* 1(1), 7-66.

Borrego, D. (2004). "Simulation-based scheduling of module assembly yards with logical and physical constraints." PhD thesis, University of Alberta, Canada.

Brown and Root (1991). "C.A.R.: Computer Aided Rigging Manual." Brown and Root, Houston.

Buisman, H., Kruitbosch, G., and Peek, N. (2007). "Simulation in multiagent resource allocation." *Proceedings of the 1st NSVKI Student Conference,* Utrecht University, Utrecht, The Netherlands.

Burke, G. P., and Miller, R. C. (1998). "Modularization Speeds Construction." *Power Engineering Magazine,* 102(1), 20-22.

Callan, S., and Thomas, J. M. (2004). *Environmental Economics & Management : Theory, Policy, and Applications (Chapter 2: Modeling the Market Process).* Thomson/South-Western, Mason, Ohio.

Chevaleyre, Y., Dunne, P. E., Endriss, U., Lang, J., Lemare, M., Maudet, N., Padget, J., Phelps, S., Rodruez-Aguilar, J. A., and Sousa, P. (2006). "Issues in multi-agent resource allocation." *Informatica,* 30(1), 3-31.

Chevaleyre, Y., Dunne, P. E., Endriss, U., Lang, J., Maudet, N., and Rodríguez-Aguilar, J. (2005). "Multiagent resource allocation." *Knowl.Eng.Rev.,* 20(2), 143-149.

Chevaleyre, Y., Endriss, U., Estivie, S., and Maudet, N. (2008). "Multiagent resource allocation in k-additive domains: Preference representation and complexity." *Annals of Operations Research,* 163(1), 49-62.

Chevaleyre, Y., Endriss, U., Estivie, S., and Maudet, N. (2004). "Multiagent resource allocation with k-additive utility functions." *Proc. DIMACS-LAMSADE workshop on computer science and decision theory,* 83-100.

Chung, C. A. (2004). *Simulation modeling handbook : a practical approach.* CRC Press, Boca Raton.

Cramton, P., Shoham, Y., and Steinberg, R. (2006). *Combinatorial Auctions.* The MIT Press, Cambridge, Massachusetts.

Deen Ali, M. S. A., Babu, N. R., and Varghese, K. (2005). "Collision free path planning of cooperative crane manipulators using genetic algorithm." *J.Comput.Civ.Eng.,* 19(2), 182-193.

Demeulemeester, E. L., and Herroelen, W. (2002). *Project scheduling: a research handbook.* Kluwer Academic Publishers, Boston.

El Ghandour, W. (2007). "Integrated information-based construction simulation system." PhD thesis, University of Alberta, Canada.

Endriss, U., Maudet, N., Sadri, F., and Toni, F. (2006). "Negotiating socially optimal allocations of resources." *Journal of Artificial Intelligence Research,* 25 315-348.

Engelbrecht-Wiggans, R. (1980). "Auctions and Bidding Models: a Survey." *Management Science,* 26(2), 119-142.

Feng, C. (1998). "Applications of genetic algorithms for multi-objective resource allocation of construction projects under certainty." PhD thesis, University of Illinois at Urbana-Champaign, Illinois, USA.

Fujimoto, R. M. (2003). "Distributed simulation systems." *Proceedings of the 35th conference on Winter simulation,* 124-134.

Gan, B. P., Liu, L., Jain, S., Turner, S. J., Cai, W., and Hsu, W. J. (2000). "Distributed supply chain simulation across enterprise boundaries." *Proceedings of the 32nd conference on Winter simulation,* IEEE, 1245-1251.

Gan, B. P., Low, M. Y. H., Wei, J., Wang, X., Turner, S. J., and Cai, W. (2003). "Synchronization and management of shared state in HLA-based distributed simulation." *Proceedings of the 35th conference on Winter simulation,* IEEE, 847-854.

Gopalakrishnan, K. N. (1998). "Design and development of a decision support system for scheduling steel fabrication projects." PhD thesis, University of Alberta, Canada.

Hague, S. (2008). "RTI Compatibility with the IEEE 1516 HLA Standard." (18/01/2010, 2010).

Hajjar, D. (1999). "A unified modeling methodology for simulation-based planning of construction projects." PhD thesis, University of Alberta, Canada.

Hajjar, D., and AbouRizk, S. M. (1999). "Simphony: an environment for building special purpose construction simulation tools." *Proceedings of the 31st conference on Winter simulation,* ACM, 998-1006.

Hajjar, D., and AbouRizk, S. M. (1996). "Building a special purposes simulation tool for earth moving operations." *Proceedings of the 28th conference on Winter simulation,* IEEE, 1313-1320.

Halpin, D. W. (1977). "CYCLONE - Method for modeling job site processes." *ASCE J Constr Div,* 103(3), 489-499.

Hornaday, W. C., Haas, C. T., and O'Connor, J. T. (1993). "Computer-aided planning for heavy lifts." *J.Constr.Eng.Manage.,* 119(3), 498-515.

Klemperer, P. (1999). "Auction theory: A guide to the literature." *Journal of Economic Surveys,* 13(3), 277-286.

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.

Kutanoglu, E., and Wu David, S. (1999). "On combinatorial auction and Lagrangean relaxation for distributed resource scheduling Next." *Institute of Industrial Engineers (IIE) Transactions,* 31(9), 813-826.

Lasry, A., Carter, M. W., and Zaric, G. S. (2008). "S4HARA: System for HIV/AIDS resource allocation." *Cost Effectiveness & Resource Allocation,* 6 1-19.

Law, A. M. (2006). *Simulation Modeling and Analysis.* McGraw-Hill, New York, USA.

Leu, S. S., Yang, C. H., and Huang, J. C. (2000). "Resource leveling in construction by GA-based optimization and its DSS application." *Journal of Automation in Construction,* 10 27-41.

Leu, S., and Yang, C. (1999). "GA-Based Multicriteria Optimal Model for Construction Scheduling." *J.Constr.Eng.Manage.,* 125(6), 420-427.

Lin, K. -., and Haas, C. T. (1996). "Multiple heavy lifts optimization." *J.Constr.Eng.Manage.,* 122(4), 354-362.

Liu, Y. (2009). "Modeling of Industrial Construction Processes Using Multi Agent Resource Allocation Framework." M.Sc. thesis, University of Alberta, Canada.

Liu, Y., and Mohamed, Y. (2008). "Multi-agent resource allocation (MARA) for modeling construction processes." *Proceedings of the 40th Conference on Winter Simulation,* 2361-2369.

Lu, M. (2003). "Simplified discrete-event simulation approach for construction simulation." *J.Constr.Eng.Manage.,* 129(5), 537-546.

Mahalingam, A., Nair, H. S., and Varghese, K. (2000). "A computer-aided heavy lift planning model." *Computing in Civil and Building Engineering,* 2 996-1003.

Martinez, J. C., and Ioannou, P. G. (1994). "General purpose simulation with stroboscope." *Proceedings of the 26th conference on Winter simulation,* IEEE, 1159-1166.

Martinez, J. C. (1998). "EarthMover - Simulation tool for earthwork planning." *Proceedings of the 30th conference on Winter simulation,* IEEE, 1263-1271.

Maru, A., and Kawahata, J. (2002). "Hitachi Modularization Technology." *Nucl.Plant J.,* 20(5), 39.

McCabe, B. Y. (1997). "An automated modeling approach for construction performance improvement using simulation and belief networks." PhD thesis, University of Alberta, Canada.

Mohamed, Y., and AbouRizk, S. M. (2005). "Framework for building intelligent simulation models of construction operations." *Journal of Computing in Civil Engineering,* 19(3), 277-291.

Mohamed, Y., Borrego, D., Francisco, L., Al-Hussein, M., Abourizk, S., and Hermann, U. (2007). "Simulation-based scheduling of module assembly yards: Case study." *Engineering, Construction and Architectural Management,* 14(3), 293-311.

Moore, J. C., Rao, H. R., and Whinston, A. B. (1994). "Multi-agent resource allocation: An incomplete information perspective." *IEEE Transactions on Systems, Man and Cybernetics,* 24(8), 1208-1219.

Mukherjee, A. (2005). "A multi-agent framework for general purpose situational simulations in construction management." PhD thesis, University of Washington, USA.

Nwana, H. S. (1996). "Software Agents: An Overview." 11(3), 1-40.

Parkes, D. C., and Ungar, L. H. (2001). "An auction-based method for decentralized train scheduling." *Proceedings of the International Conference on Autonomous Agents,* ACM, 43-50.

Pinedo, M. (2008). *Scheduling: Theory, Algorithms, and Systems.* Springer, New York, USA.

Pritsker, A. B. (1986). *Introduction to simulation and SLAM II.* New York : Wiley : c1986.

Pritsker, A. B., Elliott, C. S., and Hammesfahr, R. W. (1989). *SLAM II Network Models for Decision Support.* Prentice Hall Professional Technical Reference, New Jersey, USA.

Razdan, P., Dixit, S., and Sharma, P. (2010). "An Analysis on the Effectiveness of Dynamic Clinical Framework for Information Capture in a Hospital Management

Information System." Proceedings of the ASCNT-2010, Centre for Development of Advanced Computing (CDAC), Noida, India, 172-180.

Reddy, H. R., and Varghese, K. (2002). "Automated path planning for mobile crane lifts." *Computer-Aided Civil and Infrastructure Engineering,* 17(6), 439-448.

Ren, Z., and Anumba, C. J. (2004). "Multi-agent systems in construction–state of the art and prospects." *Autom.Constr.,* 13(3), 421-434.

Rothkopf, M. H., and Harstad, R. M. (1994). "Modeling competitive bidding: a critical essay." *Management Science,* 40(3), 364-384.

Russell, S. J., and Norvig, P. (2003). *Artificial intelligence: a modern approach.* Prentice Hall/Pearson Education, Upper Saddle River, N.J.

Sadeghi, N., and Fayek, A. R. (2008). "A framework for simulating industrial construction processes." *Proceedings of the 40th Conference on Winter Simulation,* 2396-2401.

Sawhney, A. (1994). "Simulation-based planning for construction." PhD thesis, University of Alberta, Canada.

Sawhney, A., Bashford, H., Walsh, K., and Mulky, A. R. (2003). "Agent-based modeling and simulation in construction." *Proceedings of the 35th conference on Winter simulation,* IEEE, 1541-1547.

Schimmoller, B. K. (1998). "Power plants go modular." *Power Engineering Magazine,* 102(1), 14-19.

Schwindt, C. (2005). *Resource Allocation in Project Management.* Springer, .

Shahin, A. A. (2007). "A framework for cold weather construction simulation." PhD thesis, University of Alberta, Canada.

Shapira, A., and Glascock, J. D. (1996). "Culture of using mobile cranes for building construction." *J.Constr.Eng.Manage.,* 122(4), 298-307.

Shi, J., and AbouRizk, S. M. (1994). "A resource based simulation approach with application in earthmoving/strip mining." *Proceedings of the 26th conference on Winter simulation,* IEEE, 1124-1129.

Shi, J. J. (1999). "Activity-based construction (ABC) modeling and simulation method." *J.Constr.Eng.Manage.,* 125(5), 354-360.

Shoham, Y., and Leyton-Brown, K. E. (2009). *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations.* Cambridge University Press, Cambridge.

Simulation Interoperability Standards Committee (SISC), IEEE. (2001a). "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Federate Interface Specification." *IEEE Std 1516.1-2000,* i-467.

Simulation Interoperability Standards Committee (SISC), IEEE. (2001b). "IEEE standard for modeling and simulation (M&S) high level architecture (HLA)-object model template (OMT) specification." *IEEE Std 1516.2-2000,* i-130.

Simulation Interoperability Standards Committee (SISC), IEEE. (2000). "IEEE standard for modeling and simulation (M&S) high level architecture (HLA) - framework and rules." *IEEE Std 1516-2000,* i-22.

Sivakumar, P. L., Varghese, K., and Ramesh Babu, N. (2003). "Automated path planning of cooperative crane lifts using heuristic search." *J.Comput.Civ.Eng.,* 17(3), 197-207.

Smith, V. L. (1991). *Papers in Experimental Economics.* Cambridge University Press, Cambridge, England.

Song, L. (2004). "Productivity modeling for steel fabrication projects." PhD thesis, University of Alberta, Canada.

Song, L., Wang, P., and AbouRizk, S. (2006). "A virtual shop modeling system for industrial fabrication shops." *Simulation Modelling Practice and Theory,* 14(5), 649-662.

Stark, R. M., and Rothkopf, M. H. (1979). "Competitive Bidding: a Comprehensive Bibliography." *Operations Research,* 27(2), 364-390.

Taghaddos, H. (2009). "Developing a generic resource allocation framework for construction simulation." *Proceeding of the 16th Annual Canadian Construction Research Forum*.

Taghaddos, H., AbouRizk, S. M., Mohamed, Y., and Hermann, U. (Submitted). "Simulation based auction protocol for resource scheduling problems." *J.Constr.Eng.Manage.*

Taghaddos, H., AbouRizk, S. M., Mohamed, Y., and Hermann, U. (2010). "Simulation-based multiple heavy lift planning in industrial construction." *Proceedings of the 2010 ASCE Construction Research Congress(CRC)*, 349-358.

Taghaddos, H., AbouRizk, S. M., Mohamed, Y., and Hermann, U. (2009). "Integrated simulation-based scheduling for module assembly yard." *Proceeding of the 2009 ASCE Construction Research Congress (CRC),* 1270-1279.

Taghaddos, H., AbouRizk, S. M., Mohamed, Y., and Hermann, U. (2008a). "Simulation-based resource leveling in multi-project construction." *Proceeding of the Canadian Society of Civil Engineers (CSCE) 2008 Annual Conference,* GC4241-GC42410.

Taghaddos, H., AbouRizk, S. M., Mohamed, Y., and Ourdev, I. (2008b). "Distributed agent-based simulation of construction projects with HLA." *Proceedings of the 40th Conference on Winter Simulation,* 2413-2420.

Taghaddos, H., Abourizk, S., Mohamed, Y., Hendi, A., and Hermann, U. (Submitted). "Auction based simulation for industrial crane operations." *J.Constr.Eng.Manage.,* .

Taghaddos, H., Saba, F., and ElNimr, A. (2009). "An overview of the COSYE environment for construction simulation." *Proceeding of the 16th Annual Canadian Construction Research Forum*.

Tamaki, H., Sakakibara, K., Murao, H., and Kitamura, S. (2004). "Optimization model and simulation-based solution for a class of crane scheduling problems." *Proceedings of the IEEE International Conference on Control Applications,* IEEE, 902-905.

Tantisevi, K., and Akinci, B. (2008). "Simulation-based identification of possible locations for mobile cranes on construction sites." *J.Comput.Civ.Eng.,* 22(1), 21-30.

Tharachai, T. (2004). "Multiple project resource scheduling for construction." PhD thesis, University of Michigan, Michigan, USA.

Thompson, K., and Bordwell, D. (2010). *Film history : an introduction.* McGraw-Hill Higher Education, c, New York, NY.

Tommelein, I. D., Carr, R. I., and Odeh, A. M. (1994). "Knowledge-based assembly of simulation networks using construction designs, plans, and methods." *Proceedings of the 26th conference on Winter simulation,* IEEE, 1145-1152.

Tommelein, I. D. (1998). "Pull-driven scheduling for pipe-spool installation: simulation of lean construction technique." *J.Constr.Eng.Manage.,* 124(4), 279-288.

Van Tol, A. A. (2005). "Agent embedded simulation modeling framework for construction engineering and management applications." PhD thesis, University of Alberta, Canada.

Varghese, K., Dharwadkar, P., Wolfhope, J., and O'Connor, J. T. (1997). "A heavy lift planning system for crane lifts." *Microcomput.Civil Eng.,* 12(1), 31-42.

Varghese, K., and O'Connor, J. T. (1995). "Routing large vehicles on industrial construction sites." *J.Constr.Eng.Manage.,* 121(1), 1-12.

Vickrey, W. (1961). "Counterspeculation, auctions and competitive sealed tenders." *Journal of Finance,* 16(1), 8-37.

Wang, C., Ghenniwa, H., and Shen, W. (2007a). "AdSCHE: An auction-based decentralized scheduling framework." *Proceedings of IEEE SMC 2007,* IEEE, 2865-2870.

Wang, C., Ghenniwa, H. H., and Shen, W. (2007b). "A winner determination algorithm for auction-based decentralized scheduling." *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems,* ACM, New York, NY, USA, 686-693.

Wang, P. (2006). "Production-based large scale construction simulation modeling." PhD thesis, University of Alberta, Canada.

Wang, P., Mohamed, Y., Abourizk, S. M., and Rawa, A. R. T. (2009). "Flow production of pipe spool fabrication: Simulation to support implementation of lean technique." *J.Constr.Eng.Manage.,* 135(10), 1027-1038.

Weiss, G. (1999). *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence.* MIT, Cambridge, Mass.

Wellman, M. P., Walsh, W. E., Wurman, P. R., and MacKie-Mason, J. K. (2001). "Auction protocols for decentralized scheduling." *Games Econ.Behav.,* 35(1-2), 271-303.

Wilken, K., Liu, J., and Heffernan, M. (2000). "Optimal instruction scheduling using integer programming." *SIGPLAN Notices (ACM Special Interest Group on Programming Languages),* 35(5), 121-133.

William, M., and Bennet, C. (1996). "ALPS: Automated lift planning system." *Proceedings of American Society of Civil Engineers Third Congress on Computing in Civil Engineering,* .

Willis, E. M. (1986). *scheduling construction projects.* John Wiley & Sons Inc, .

Wooldridge, M., and Jennings, N. R. (1995). "Intelligent Agents: Theory and Practice." *Knowledge Engineering Review,* 10(2), 115-152.

Yasantha, R. A. J. (2001). "Special purpose simulation for tunnel construction operations." PhD thesis, University of Alberta, Canada.

Zhou, F. (2006). "An integrated framework for tunnel shaft construction and site layout optimization." PhD thesis, University of Alberta, Canada.

Appendix A[6]

Simulation-Based Resource Leveling in Multi-Project Construction

H. Taghaddos[1], S. AbouRizk[1], Y. Mohamed[1], and U. Hermann[2]

1 Department of Civil & Environmental Engineering, Hole School of Construction University of Alberta, Edmonton, Alberta, Canada

2 PCL Industrial, Edmonton, Alberta, Canada

Abstract: Scheduling multiple projects or projects with multiple complex units is a complicated process involving a number of uncertain factors and constraints. Conventional project management software applications do not provide a resource-driven schedule that optimizes the use of available resources as well as satisfying logic and space constraints that usually affect the construction process. An optimized schedule can significantly affect the resource utilization and enhance the productivity of the project. This paper introduces a simulation-based resource leveling based on a two-pass discrete-event simulation model in large

---

[6] This paper is published at proceeding of the Canadian Society of Civil Engineers (CSCE) 2008 Annual Conference, GC4241-GC42410.

construction projects. This methodology is implemented in an actual case study of a module assembly yard. This sample study offers a good representation of multi-unit projects in which every module assembly represents a complex and independent production unit with its own distinct logic network. It is shown that the method is very practical for large-scale projects and can produce significant improvement in the resource utilization of the system.

1. Introduction

Construction projects are subject to different uncertain factors and constraints based on their open production environment. Scheduling multi-unit projects is a complicated process, since these large projects are composed of a number of projects that are integrated and have an impact on each other. Conventional project management software, such as Microsoft Project or Primavera, does not easily provide a schedule output that satisfies all the available constraints and optimizes the resource utilization. A simulation-based scheduling approach is an alternative way to model a multi-unit project while satisfying available constraints and enhancing resource utilization (El-Rayes and Moselhi 1998). Through simulation, the system can be improved by leveling resource utilization (Leu and Hung 2002).

In this study, a simulation-based resource leveling based on a two- pass discrete-event simulation model in large construction projects is introduced. This proposed methodology is elaborated using an actual case study of a module assembly yard, in which every module assembly represents a project with a logic network of its own. Assembling each module is considered practically as a unique project in relation to its internal design and components.

2. Module Assembly Yard

Current construction projects in the Alberta oil sands region depend heavily on modular construction. This method minimizes the amount of time, cost, and effort needed to construct onsite in Northern Alberta's harsh weather conditions (Schimmoller 1998). Modularization offers numerous benefits for both constructor and client, including shorter project duration, reduced number of workers onsite, reduced project costs, improved safety and quality, and more flexible construction processes (Burke and Miller 1998; Davila Borrego 2004; Maru and Kawahata 2002).

"Modules" refer to a construction unit made of pre-assembled components such as structural steel frames, racks of pipes, cables, equipment, or a combination of miscellaneous components (Davila Borrego 2004).

After all the required components have been built in the fabrication shop, modules are assembled in a module assembly yard near the spool fabrication shop. A typical module yard is divided into a number of areas, which are called "lots." Each lot is composed of a number of rows, which are called "bays," and each module occupies a fraction of a bay. Figure 1 displays the layout of the module assembly yard for PCL Industrial.

Figure 1 PCL module yard layout

To assemble a module, first a suitable space in the yard must be found. The allocation of space depends on many factors, such as type, length, early start (which represents the time that all the components are available), estimated duration of the assembly process, and shipping date of the module. A number of activities must take place in order to build the module; these can be found by comparing the module with the historical data. The average duration of each activity, overlap with its predecessor activity, and the required manpower are also assigned based on historical information.

A realistic schedule should meet a number of the constraints based on the yard layout. Some of the constraints are physical, and some others are logical, as determined by yard superintendents (Mohamed et al. 2007). The major constraints are:

Fixed module yard layout.

Superintendent may decide to assemble a module in a specific bay, or across a specific set of bays, i.e. {Bay A1, Bay A2, Bay A3, Bay A4, Bay B1, Bay B2, Bay B3, Bay B4}, depending on the module type and the availability of equipment (Figure 2).

Assigning space in different lots must follow a specific order, called "option array," based on the type of the module. For example, {Lot A, Lot C, Lot B, Lot D, Lot E} dictates that the priority is to find a space in the bays of Lot A, and then Lot B if no space is found in Lot A, and so on.

Fixed maximum number of crews for any task and maximum amount of ramp up.

Module should be completed by its assigned shipping date.

Limited maximum number of shipments per day.

Modules may only be shipped out by transporter when the space in the front of the bay is empty.

Assembly bays are only accessible from one end, so whenever assembly of a module is complete, all the space in front of it must be empty to remove the module.

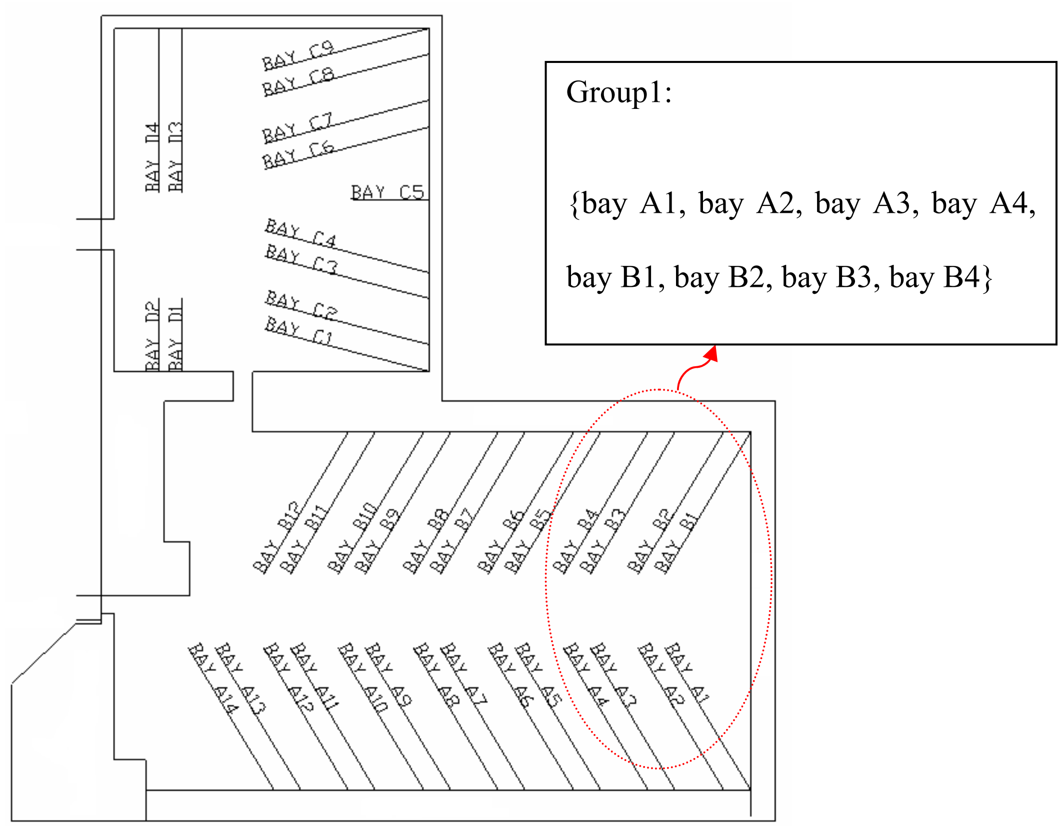


Figure 2 An example of bay grouping in the module yard

Scheduling such a system is very complicated due to the dynamic nature of the process. For instance, if there are not enough workers to perform an activity on a module that is in front of the bay, the finish time of that activity is delayed, and the modules in the back are stocked until the module in the front is finished. Therefore, using CPM-based techniques is a fairly tedious and inefficient solution. Simulation is an excellent technique to address the dynamic nature of such a system in scheduling. It helps us satisfy the constraints of the system by using priority logic and provides a flexible model that can be used to improve and optimize the system overall.

3. Model Description

Figure 3 indicates the main elements of the proposed approach. The scheduling process is performed through a two-pass discrete-event simulation model. First, the simulation model schedules the activity without considering any resource limitations; then it modifies the schedule through resource leveling. Finally, the interface provides the scheduler with different reports regarding the start and finish dates for each module and the required manpower for different activities at any point of time. It also illustrates the suggested location of the various modules at any given time.

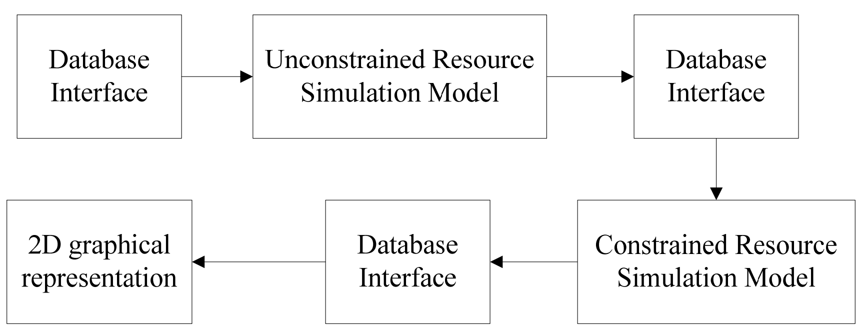


Figure 3 Main elements of the proposed simulation-based approach

4. Database interface

From the user's perspective, the database interface is the entry point for the scheduling process (Figure 4). It is implemented using Microsoft Access and contains the following information:

Site information: maximum number of shipments per day, maximum size of crew in module yard, maximum number of crews for any task, and maximum amount of ramp up;

Lots information: lot names and period that the lots are active for module assembly;

Bays information: bay names, corresponding lot names, length, and location;

Bay Group information: group number and bay's name for the group;

Lot options information: the preferred order of the lots for allocating space to a module;

Modules information: type, size, priority, total estimated durations, early starts, planned shipping dates, ID of required tasks, duration and manpower of the required tasks, and overlap with the precedent task.

Because the developed model is used for scheduling as well as tracking, information about modules that are completed or partially assembled is also recorded in the database. This information includes the location of the modules in the bay, the actual start date of the assembly process, actual start dates of the stated tasks, the actual finish dates of the completed tasks, and the shipping date on which the module is removed. After populating all the required data, the scheduling simulation model is called from the Access interface.

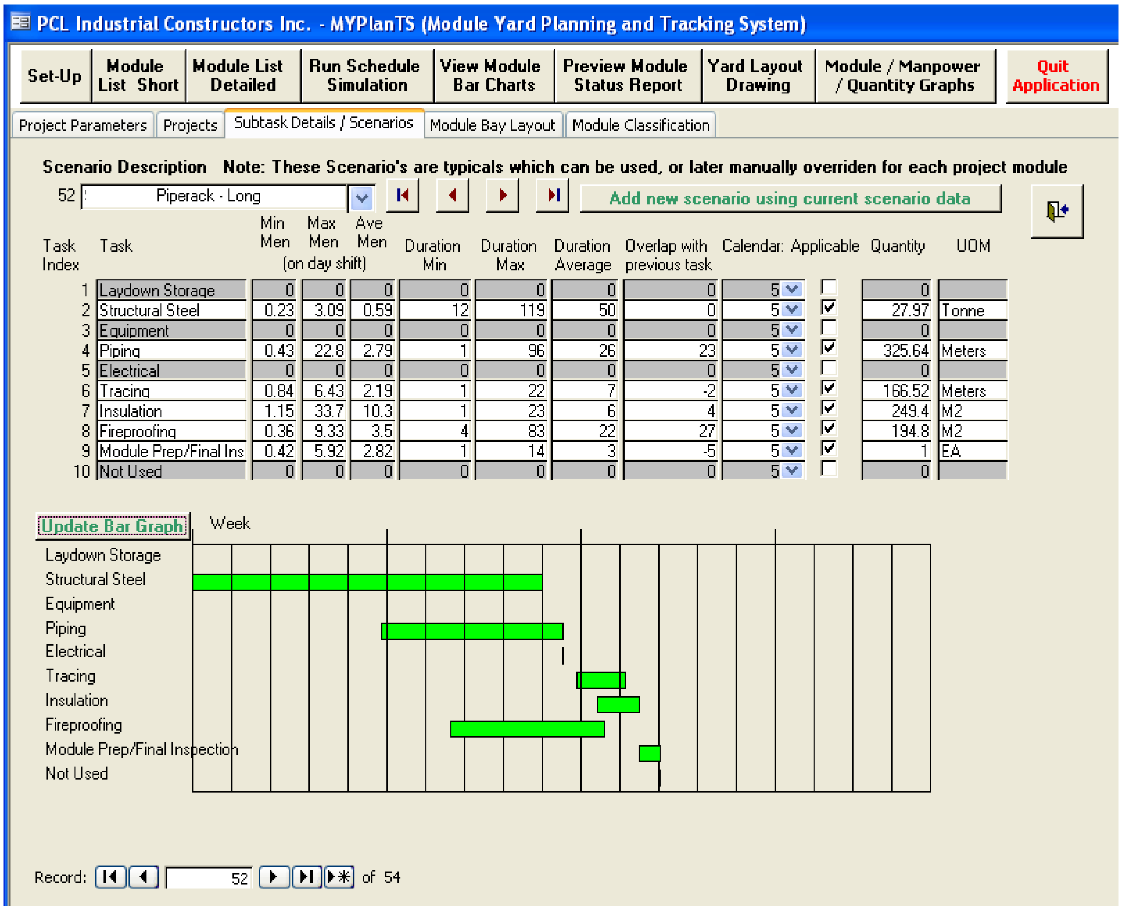


Figure 4 Scheduling interface

5. Scheduling Simulation Model

Computer simulation is defined as the process of designing a mathematical-logical model of a real world scenario and experimenting with the model on a computer (Pritsker, A. Alan B. 1986). Construction simulation is a powerful tool that can be used by a construction company for a number of tasks, such as productivity measurement, resource planning, site planning, etc. (Sawhney et al. 1998). The use of simulation in construction was put forth by Halpin (Halpin 1977) with the invention of CYCLONE (CYCLic Operation NEtwork). AbouRizk and Hajjar (1998) presented a new approach to facilitate the adoption of simulation to the construction industry. They developed powerful software, called *Simphony*, using an object-oriented programming framework. In this discrete-event model, it is possible to allocate resources to modules based on the priorities of processing of these modules, which is calculated based on planned shipping dates, early starts, and the actual date the simulation model run takes place (Mohamed et al. 2007). The model developed here was built under the Simphony.NET (version 1.1.3.4) simulation environment.

Simphony allows the user to develop a hierarchical simulation model. The logic of the entire simulation model is depicted in Figure 5. First, the model reads the required information from the database; then, an entity (corresponding to a module) is produced at the early start time (or actual start time) of each module. If the module is already completed, the corresponding entity is sent directly to the last element to populate the results into the database. Otherwise, the entity is sent to the Space allocation, Assembly, and Release elements, consecutively (Figure 6).

In space allocation elements, the time and space criteria are defined as two heuristic rules to optimize the space utilization. The time criterion is needed to place modules in order so that a module is not trapped by another module because of its late finish date. This criterion is satisfied by having a rough estimate of finish time of the modules in the bay, and putting the modules that will be finished later behind the other modules that will be finished earlier. However, the finish time at the space allocation phase is just an estimate and it will change according the resource limitations. The space criterion places modules in a way that remaining space in the back of the bay will be small enough (i.e. less than 20 feet) or big enough to be able to put another module in the back of the bay (i.e. more than 80 feet). However, if no space is found according to the space criteria, it will be ignored. These two criteria help to improve the space utilization in the yard.

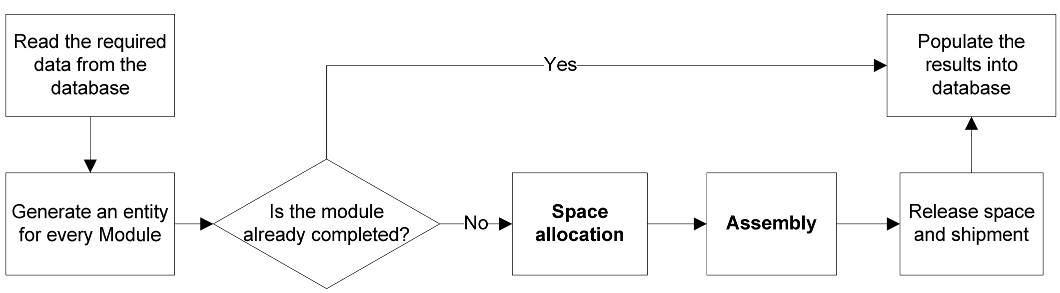
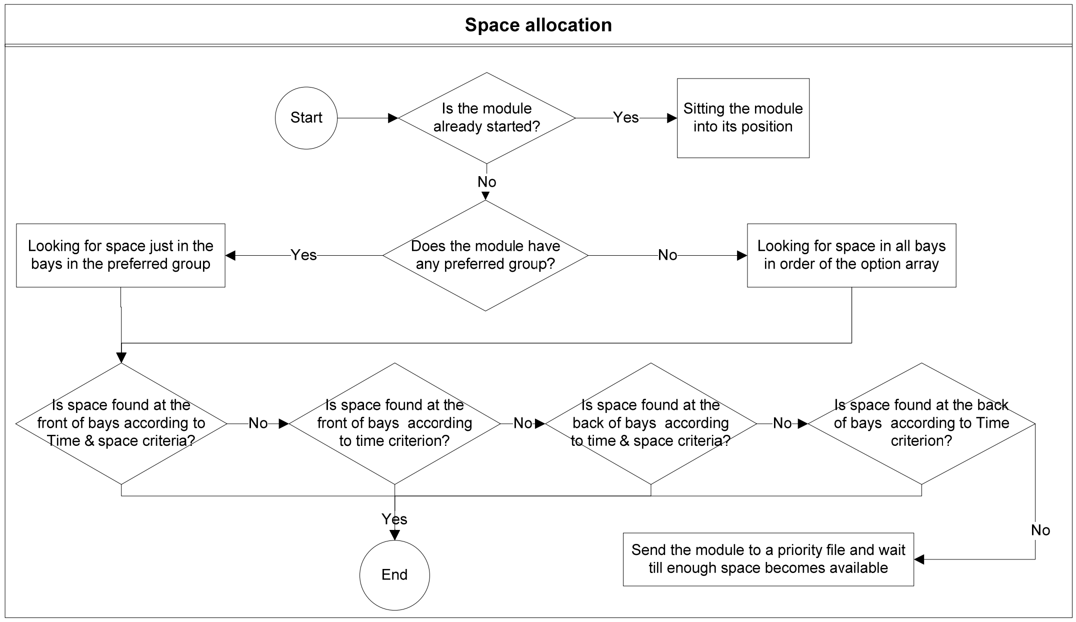
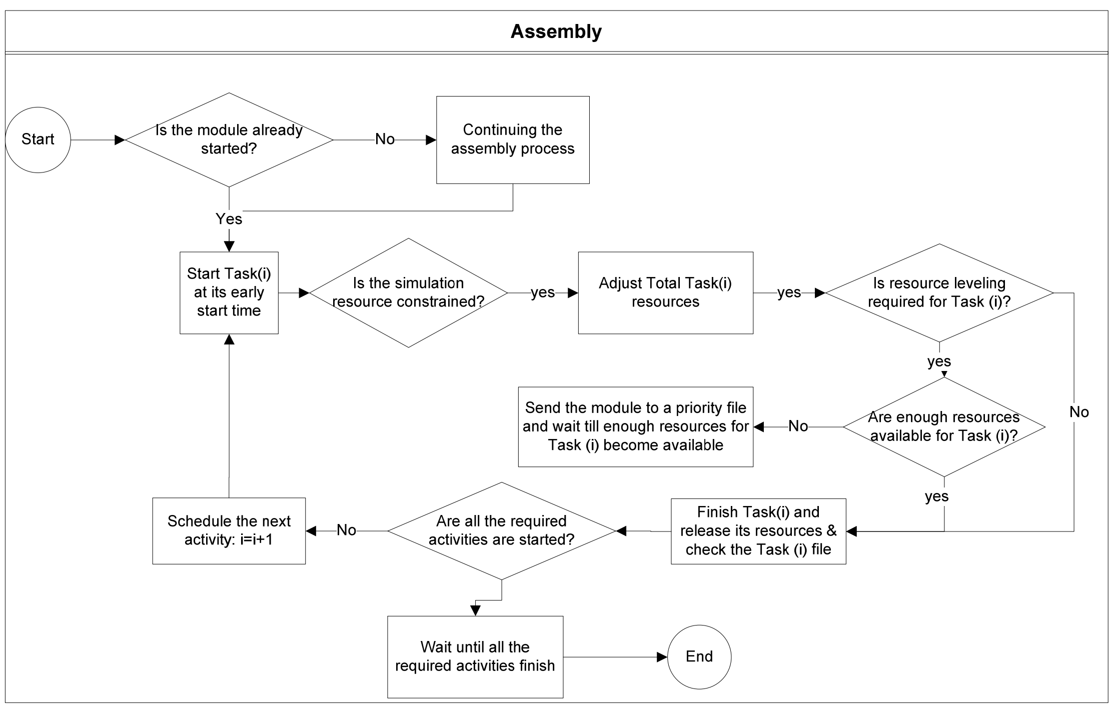


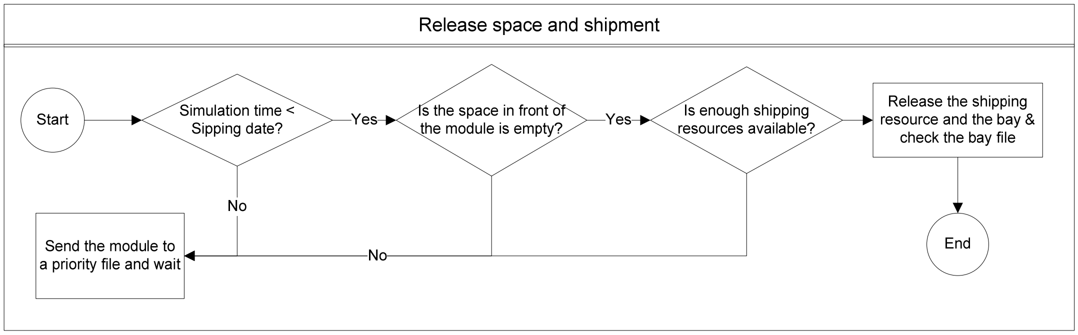
Figure 5 Module yard simulation model logic



a) Space allocation element



b) Assembly element



c) Release space and shipment element

Figure 6 Logic of three main elements of module yard simulation model

The finish time of a module is just an estimate in the space-allocating phase. The duration of a module changes based on resource (manpower) limitations. The problem of scheduling has a dynamic nature in that any resource assignment to a module may affect the finish time of a module.

6. Simulation-Based Resource Leveling

Resource leveling is a real need from the perspective of project manager to avoid day-to-day fluctuation in resource demands and to maintain an even flow of application for construction resources (Harris 1990). The main objective in a resource-leveling problem is to reduce peak resource requirements and to smooth out the resource assignment within the required project duration (Leu and Hung 2002).

Early attempts to solve resource-leveling problems employed mathematical models, including integer linear programming and exhaustive enumeration (Easa

1989). These mathematical models are mostly applicable to small-scale problems; larger problems require more extensive computational effort (Easa 1989). Researchers have expended major efforts in developing heuristic resource leveling or adopting computational optimization techniques, such as genetic algorithms and simulated annealing, to solve construction resource leveling problems (Chan et al. 1996; Harris 1990; Leu and Hung 2002; Leu et al. 2000). Nevertheless, these resource leveling approaches do not work well for dynamic multi-project problems such as module yard scheduling, where duration and resource assignment of every project module may have a significant impact on other projects.

The resource leveling approach used in this study is a simplified and practical method using two runs of the simulation model. In the first run, the problem is considered as an unconstrained resource simulation model with no limitation in providing the resources for each task. The total manpower for each individual task to be leveled is calculated at a regular basis (i.e. weekly). The manpower of the insulation activity throughout the project in Figure 6 shows cyclical fluctuation. For example, in March 2008, about 130 workers are required for insulation; one month later, only 30 workers are needed. This manpower curve should be smoothed and the simulation model should be forced to follow it as the trend line. In this study, a simple moving average method (Ellis and Parbery 2005) within a 4-week neighbourhood has been used to smooth the manpower curve. Therefore, the manpower (MP) at each point of time (i) is the average of its neighbouring points (Formula 1):

$$Tr_i = \sum_{i-4}^{i+4} MP_i/\text{n} \quad [1]$$

where n is the number of manpower points included in the average, i is the relative position of the manpower currently being considered within the total number of points, and MP_t is the manpower at time i.

Implementing this method results in a smoother curve, referred to as the trend line (Figure 6). For optimal results, the manpower curve should follow the trend curve. It is not realistic to expect a rectangular shape for the resource-leveling curve, because the workload is not uniformly distributed in time. For instance, in the current example, there is little insulation work in December 2007, while there is extensive insulation work in the middle of March 2008.

In the second run of the simulation model, it is assumed that the total number of the resources follow the trend curve. This has been implemented in the simulation model by adjusting the total number of available resources according to the trend line.

When enough resources are not available for a certain activity, the entity must wait in a priority queue until the resources become available. The priority of the modules is increased in the file when the resource waits for a while in the queue. As soon as the resources (i.e. insulation workers) become available, they will be assigned to the entity with the highest priority in the queue. This priority logic helps us to avoid late shipment of a module. However, some overrides are defined in the simulation model to allow the available resources to be neglected, and an activity to be started at a given date. This also helps avoid late shipments.

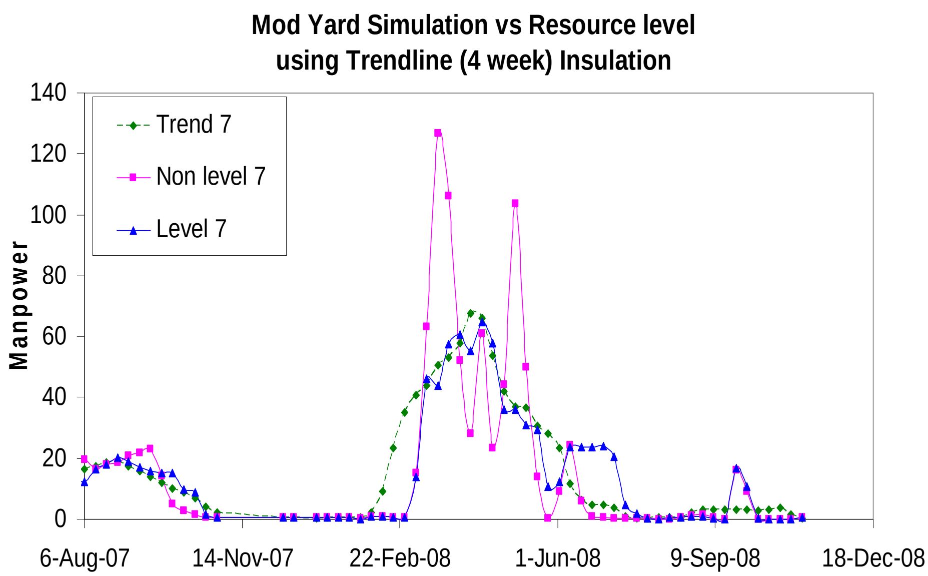


Figure 6 Insulation manpower loading: original, trend, and leveled manpower curve

When adjusting resources, if the number of resources must be increased, there is no disruption to the model. However, if the number of the resources must be reduced, the model decreases the free resources, and reduces more resources as required when some other resources complete their tasks. By comparing the trend and level curves in Figure 6, it can be seen how closely the modified manpower has reached the trend curve.

7. Conclusion

Scheduling multi-unit construction projects, such as a module yard, is a complicated task. Multi-unit construction projects usually have a dynamic nature; the completion times and resource utilization of different projects are interlinked with each other. Therefore, a number of constraints must be considered in the scheduling phase for the successful implantation of the project. Simulation is an excellent tool to model such projects and to derive a realistic schedule for the project.

This study examines the introduction of a simulation-based resource leveling based on a two- pass discrete-event simulation model into an industrial module assembly yard. First, the simulation model schedules the activities without considering resource limitations, and then the schedule is modified based on the resource allocation trend-lines. Although the method does not commit to a single, optimum solution, it produces significant improvement in the resource utilization of the system. This approach is very appealing, as it is quick, practical, and very flexible. The interface provides the scheduler with different reports regarding the start and finish dates for each module and required manpower for different activities at any point of time. It also illustrates the suggested location of the various modules at any given time, providing easy access to a range of key data.

The proposed approach smoothes out the resource utilization curve and forces the schedule to follow the modified curve. Future research will investigate clear guidelines for finding the trend curve to better standardize the method of analysis for practitioners. Additionally, surveys of other factors such as the number of simulation runs may be of importance to researchers.

8. References

AbouRizk, S. M., and Hajjar, D. 1998. A framework for applying simulation in construction. *Canadian Journal of Civil Engineering,* 25(3): 604-617.

Burke, G. P., and Miller, R. C. 1998. Modularization Speeds Construction. *Power Engineering Magazine,* 102(1): 20-22.

Chan, W., Chua, D. K. H., and Kannan, G. 1996. Construction resource scheduling with genetic algorithms. *Journal of Construction Engineering and Management,* 122(2): 125-132.

Davila Borrego, L. F. 2004. "Simulation-based scheduling of module assembly yards with logical and physical constraints." PhD thesis, University of Alberta, Canada.

Easa, S. M. 1989. Resource leveling in construction by optimization. *Journal of Construction Engineering and Management,* 115(2): 302-316.

Ellis, C. A., and Parbery, S. A. 2005. Is smarter better? A comparison of adaptive, and simple moving average trading strategies. *Research in International Business and Finance,* 19(3): 399-411.

El-Rayes, K., and Osama, M. 1998. Resource-driven scheduling of repetitive activities. *Construction Management & Economics*, 16(4): 433-46.

Halpin, D. W. 1977. CYCLONE: Method for modeling job site processes. Journal of the Construction Division, ASCE, 103(3): 489-499.

Harris, R. B. 1990. Packing method for resource leveling (pack). *Journal of Construction Engineering and Management,* 116(2): 331-350.

Leu, S., and Hung, T. 2002. An optimal construction resource leveling scheduling simulation model. *Canadian Journal of Civil Engineering,* 29(2): 267-275.

Leu, S., Yang, C.,and Huang, J. 2000. Resource leveling in construction by GA-based optimization and its DSS application. *Journal of Automation in Construction,* 10: 27-41.

Maru, A., and Kawahata, J. 2002. Hitachi Modularization Technology. Nuclear Plant Journal., 20(5): 39.

Mohamed, Y., Borrego, D., Francisco, L., Al-Hussein, M., AbouRizk, S., and Hermann, U. (2007). Simulation-based scheduling of module assembly yards: Case study. *Engineering, Construction and Architectural Management,* 14(3): 293-311.

Pritsker, A. Alan B. 1986. *Introduction to simulation and SLAM II.* Wiley, New York , NY, USA.

Sawhney, A., Abourizk, S. M., and Halpin, D. W. 1998. Construction project simulation using CYCLONE. *Canadian Journal of Civil Engineering,* 25(1): 16-25.

Schimmoller, B. K. (1998). Power plants go modular. *Power Engineering Magazine,* 102(1): 14-19.

Appendix B[7]

Simulation-Based Schedule Enhancement of Tower Cranes

E. Moghani[1], H. Taghaddos[1], M. Salehi[2], S.M. AbouRizk[3], and Y. Mohamed[4]

1 PhD students, Department of Civil and Environmental Engineering, University of Alberta, Edmonton, AB, Canada T6G 2W2

2 MSc student, Department of Civil and Environmental Engineering, University of Alberta, Edmonton, AB, Canada T6G 2W2

3 Professor, Hole School of Construction Engineering and Management, Department of Civil and Environmental Engineering, University of Alberta, Edmonton, AB, Canada T6G 2W2

4 Assistant Professor, Hole School of Construction Engineering and Management, Department of Civil and Environmental Engineering, University of Alberta, Edmonton, AB, Canada T6G 2W2

Abstract: Tower cranes can cause bottlenecks in large construction projects, where the demand for undertaking material handling exceeds the capacity of the tower crane. In practice, material handling involves a variety of uncertainties and constraints that pose a challenge for the project manager in producing an optimum schedule for the project. Skilled workers are scarce and expensive resources in

---

[7] This paper is published at the proceeding of the Canadian Society of Civil Engineers (CSCE) 2009 Annual Conference, ICS0701-ICS0709.

many parts of the world, including North America. Therefore, it is very crucial for the success of a project to keep the crew fully utilized and to consider alternative resources whenever tower cranes cannot deliver the materials on time. To guarantee the availability of the cranes, project schedule should be adjusted to keep the daily utilization of the cranes below a specified limit (e.g. 80%). In this study, a simulation-based approach is employed to evaluate the daily utilization of tower cranes throughout the life cycle of a project. Based on this approach, a special purpose simulation template is developed to simulate tower cranes' activities in similar construction projects. This model is integrated with a database to collect all the required information and populate the daily utilization results. The primary purpose of the proposed approach is to enhance the schedule by identifying the periods when a tower crane is overloaded and adjusting the schedule accordingly.

Introduction

Tower crane planning is usually a critical task for construction managers, who must consider overall cost and schedule. As a limited resource, cranes can cause a bottleneck in large construction projects, where several cranes undertake material handling. Due to the shortage of crews in some areas in North America (i.e. province of Alberta), it is very crucial for the success of a project to avoid keeping crews on site idle as they wait for the crane. Project managers in this case study were interested in verifying the preliminary schedule and estimating the utilization of the crane. They wanted to keep the utilization of the tower cranes less than 80% throughout the construction period.

In this study, a simulation-based approach is presented to evaluate the construction schedule by considering the tower cranes as project resources. For this purpose, a special purpose simulation template was developed for managing tower cranes in the project.

State of the Art

Recent studies suggest computer simulation as a potentially helpful tool to manage tower cranes on construction projects. Special Purpose Simulation (SPS) has been introduced to assist practitioners who are not knowledgeable in simulation or programming in simulating a construction domain (AbouRizk and Mohamed 2000). Examples of these SPS tools are CRUISER (Hajjar et al. 1998), EARTHMOVER (Martinez 1998), and module yard scheduling (Taghaddos et al. 2008). SPS for tower crane management was first used by Appleton et al. (2006) to model tower crane operations based on priority-rating logic. However, these researchers calculated the moving duration of work packages (modules) by entering the coordinates of source and destination for each module and calculating the radial, horizontal, and vertical moving duration for each one. Using this method would take a great amount of time to enter the coordinates of each module; also, entering all the coordinates will be less accurate because of restrictions on the job site during project execution that require the use of different locations for pick up and drop off of different lifts. In addition, the coordinates of modules are usually subject to change as the schedule evolves. Therefore, modifying the coordinates after each change would be very time consuming and impractical. Furthermore, calculating the moving time using mathematical techniques for each individual module would be very inefficient, and it would make the program very slow as it takes a great amount of time to calculate the durations for a one-year project. This approach is impractical for most construction projects, where detailed information about modules (i.e. the movement of column formworks in the building) is not available in the planning stage.

Based on the aforementioned reasons, the simulation team decided to use approximate time distributions for calculating the moving durations. The required data in this project were gathered by observing the cranes movement on a similar job site with the same tower cranes as well as through discussion with experienced superintendents. Finally, a new Special Purpose Simulation (SPS) was created for analyzing the tower crane utilization based on a case study that illustrate the tower crane operation on the "Edmonton Clinic Project" in Edmonton, Alberta by PCL Constructors, Inc.

Problem Statement

The Edmonton Clinic project is a construction project now under construction by PCL Constructors, Inc. This project is a nine-story concrete structure building in

Edmonton, Alberta. Each floor is divided into seven areas, and three tower cranes cover these areas (Figure 1). There are some areas covered by more than one tower crane.

Figure 1. Layout of the project and tower crane positions

Each floor of the building is composed of seven areas. In each area, there are a number of elements to be built, including columns, slabs, and shear walls. Columns are spread throughout the building in a grid, and these are built using five column forms (Figure 2). Slabs are built using scaffold deck forms on the first floor and high-flyer deck form on the other floors (Figure 3). Each high-flyer deck form has its own specific shape. It cannot move within the floor, only to the next floor above. There are also a number of shear walls and core forms in each floor (Figure 2). In order to build elements (i.e. columns) in an area, tower cranes are utilized to move the required modules (i.e. concrete, formwork, reinforcement) to their locations.

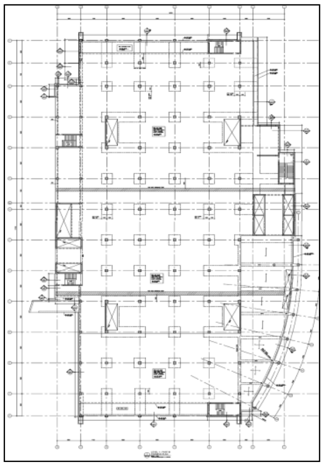


Figure 2. Project layout

Figure 3. High-flyer formwork

Due to the high cost of labour and shortage of the work force in Alberta, the project manager considers labour power as the most valuable resource and tries to utilize the workers as much as possible. Therefore, the manager is trying to find out whether or not the cranes will be able to perform all the activities assigned to them in a one-day span. In other words, the manager wants to find out how long each crane needs in a day to accomplish its assigned activities. If the required time is more than 80% of the working hours in that day, some of those activities can be performed in another way to avoid having idle labourers. For example, instead of moving the concrete in buckets, the crew can pump concrete to the building. Simulation helps the manager analyze the preliminary schedule, indicate the daily utilization of each tower crane, and identify the busy days for each tower crane.

Database Interface

The input data for the simulation model are provided through a database interface, developed in Microsoft Access 2007. A great advantage of this model is its ability to integrate with an external database, which allows the simulation model to be more generic and usable for different tower crane projects. The same database also represents the simulation results, which enable users to access the results and the related graphs directly through the database. The database includes eight different input tables described below:

1. Scheduled Data table

The Scheduled Data table is the main table of this database and contains all the information of the modules, including module and activity name, planned duration, start and finish date, number of units per pick, total picks per day, area, and floor number. All the information in this table is obtained from the Primavera schedule report of the project. The table is sorted by planned start date and finish date (Figure 4).

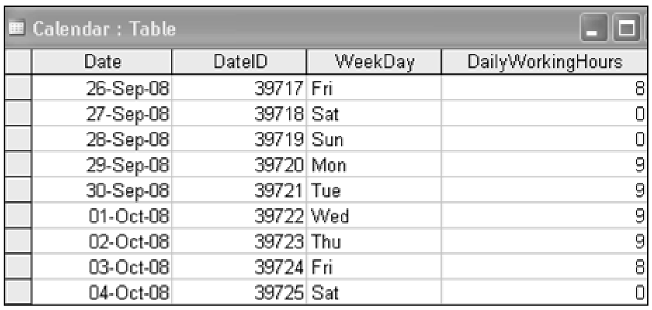
ScheduledData : Table

Module Name	Activity	Activity Code	Planned Units	Planned Duration	Planned Start	Planned Finish	Units/Picks	Floor	Area	# Picks per day
Column Forms	Columns - Form	SSC0A03110	60	4	39821	39826	10.6	0	A	1
Concrete - Column	Columns - Pour	SSC0A03120	11	1	39827	39827	1.5	0	A	7
Reinforcing Steel	Columns - Reinforce	SSC0A03130	3600	2	39825	39826	1	0	A	18
Column Forms	Columns - Strip	SSC0A03140	60	2	39828	39829	10.6	0	A	3
Shear Wall Forms	Shearwalls - Form	SSC0A03190	158	2	39836	39839	45	0	A	2
Concrete - Shearwall/Core	Shearwalls - Pour	SSC0A03200	29	1	39840	39840	1.5	0	A	19
Reinforcing Steel	Shearwalls - Reinforce	SSC0A03210	7250	1	39836	39836	1	0	A	1
Shear Wall Forms	Shearwalls - Strip	SSC0A03220	158	1	39841	39841	45	0	A	4

Figure 4. Scheduled Data table

2. Calendar table

This table contains all the dates from project start to completion, which are derived from the Primavera schedule. The table also contains the corresponding working hours for each day (Figure 5). It takes into consideration national holidays and long weekends according to the location of the project. This information is required to obtain the daily working hours throughout the project in order to calculate the daily utilization of the tower cranes.


Calendar : Table

Date	DateID	WeekDay	DailyWorkingHours
26-Sep-08	39717	Fri	8
27-Sep-08	39718	Sat	0
28-Sep-08	39719	Sun	0
29-Sep-08	39720	Mon	9
30-Sep-08	39721	Tue	9
01-Oct-08	39722	Wed	9
02-Oct-08	39723	Thu	9
03-Oct-08	39724	Fri	8
04-Oct-08	39725	Sat	0

Figure 5. Calendar table

3. Crane Assignment table

This table specifies the proportion of each area that is covered by each tower crane. In this project, there are seven areas covered by three tower cranes. Figure 6 shows Cranes 1, 2, and 3 can serve 45%, 10%, and 45% of area D by considering their reaching area.

CraneAssignment : Table

DestinationArea	Crane1 (% share)	Crane2 (% share)	Crane3 (% share)
A		100	
B		100	
C			100
D	45	10	45
E	100		
F			100
G	100		

Figure 6. Crane Assignment table

4. Duration tables

One of the main inputs for the simulation model is moving duration data for various modules. Moving duration is the sum of vertical duration, horizontal duration, and marginal duration. A marginal duration is the time it takes to lift a module to a certain elevation above the destination point, for reasons of safety. A sub-simulation model is employed to estimate the range of moving duration, in the form of triangular distribution, for every module according to crane speed and the distance between the source and destination points. Horizontal and vertical durations depend on the module type. Some of the modules including concrete column, reinforcing steel and concrete shear wall/core modules, are brought in from outside of the building every time. Other modules are moved once to the building and stay inside the building. They move to other positions inside the building, either within a floor or onto the next floor, until the end of building construction. A brief explanation of the duration tables is presented below.

The “Inside Horizontal Duration” table contains all the horizontal movement durations for the modules that are moved within the building from one floor to another (Figure 7). This table specifies the name and type of various modules (coming from outside or moving within the building). It also includes double handling feature that is beneficial for some of the modules in the early planning stage. For example, for the shear walls, if the forming and stripping activities are combined together as one activity in the database, the duration should be different from usual modules. However, in the current database, where more detailed information is available, all the activities need a single lifting. For the modules that mostly move across floors or between floors, the user also has to specify the minimum, mean, and maximum time to move between different possible places on the floor or to the next floor. Because we do not exactly know to where and how each work package moves across the floor or to the next floor, it is better to leave this to the discretion of the user to specify approximate times.

InsideHorizantalDuration : Table

maduleName	OutsideLifting	DoubleLifting	InsideHorizantalDuration
Column Forms	☐	☐	(0.2 , 0.8 , 1.00)
Concrete - Column	☑	☐	
Concrete - Shearwall/Core	☑	☐	
Core Forms	☑	☐	
Hi-Flyer Deck Forms	☐	☐	(0.40,1 .00,2 .00)
Reinforcing Steel	☑	☐	
Scaffold Deck Forms	☑	☐	
Shear Wall Forms	☐	☐	(0.70, 0.77, 0.80)
Wall Forms	☐	☐	(0.19, 0.30, 0.38)

Figure 7. “Inside Horizontal Duration” table

In the “Outside Horizontal Duration” table (Figure 8), the minimum, mode, and maximum horizontal duration for moving a module from outside the building to the specific area are presented. Minimum and maximum values represent the durations of the horizontal movement from the outside source to the closest and furthest corner of the area, and the mode value is the duration of moving from the outside source to the middle point.

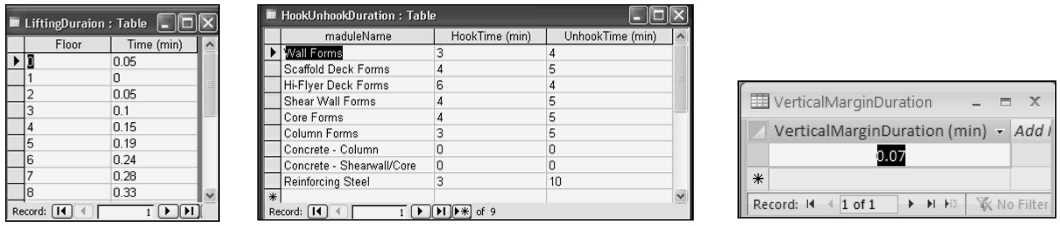
OutsideHorizontalDuration : Table

CraneNumber	HDur@A (min)	HDur@B (min)	HDur@C (min)	HDur@D (min)	HDur@E (min)	HDur@F (min)	HDur@G (min)
1	(0.0 , 0.0 , 0.0)	(0.0 , 0.0 , 0.0)	(0.0 , 0.0 , 0.0)	(0.45, 1.36, 2.19)	(0.30, 0.86, 1.45)	(0.0 , 0.0 , 0.0)	(0.02, 0.63, 1.30)
2	(0.12, 1.11, 1.95)	(0.28, 0.89, 1.52)	(0.0 , 0.0 , 0.0)	(1.08, 2.20, 3.20)	(0.0 , 0.0 , 0.0)	(0.0 , 0. 0, 0.0)	(0.0 , 0.0 , 0.0)
3	(0.0 , 0.0 , 0.0)	(0.0 , 0.0 , 0.0)	(0.78, 1.15, 1.39)	(0.78, 1.62, 2.38)	(0.0 , 0.0 , 0.0)	(0.72, 1.33, 1.87)	(0.0 , 0.0 , 0.0)

Figure 8. “Outside Horizontal Duration” table

Figure 9 shows the vertical duration for each floor. For moving a module from one floor to another, the vertical duration is the difference of the duration of the next floor and the current floor.

The hook and unhook duration are estimated according to superintendent judgment and historical information from a similar project (Figure 9).



LiftingDuraion : Table

Floor	Time (min)
0	0.05
1	0
2	0.05
3	0.1
4	0.15
5	0.19
6	0.24
7	0.28
8	0.33

HookUnhookDuration : Table

maduleName	HookTime (min)	UnhookTime (min)
Wall Forms	3	4
Scaffold Deck Forms	4	5
Hi-Flyer Deck Forms	6	4
Shear Wall Forms	4	5
Core Forms	4	5
Column Forms	3	5
Concrete - Column	0	0
Concrete - Shearwall/Core	0	0
Reinforcing Steel	3	10

VerticalMarginDuration

VerticalMarginDuration (min)
0.07

(a) (b) (c)

Figure 9. a)“Vertical Duration”, b)“Hook/Unhook duration”, and c)“Vertical Margin Duration” tables

Finally, the “Vertical Margin Duration” table includes an estimation of the duration that takes the crane to move the module a few meters higher than the height of the destination point and bring it down.

5. Result table

There is just one result table that includes the utilization of the cranes as an hourly and percentage based. In addition, the user can follow the results by analyzing the graphs in the same database. Different graphs show the utilization of the cranes during the lifetime of the project.

Proposed approach

In order to solve this problem, we have developed a general simulation template that interacts with a database to analyze the crane schedule. This general template can be utilized on similar buildings with any number of cranes. To use this template, the user should enter the required data in the database and make the model. After running the simulation, the model reads the data from the database, moves the modules on a daily basis, and exports the results to the database.

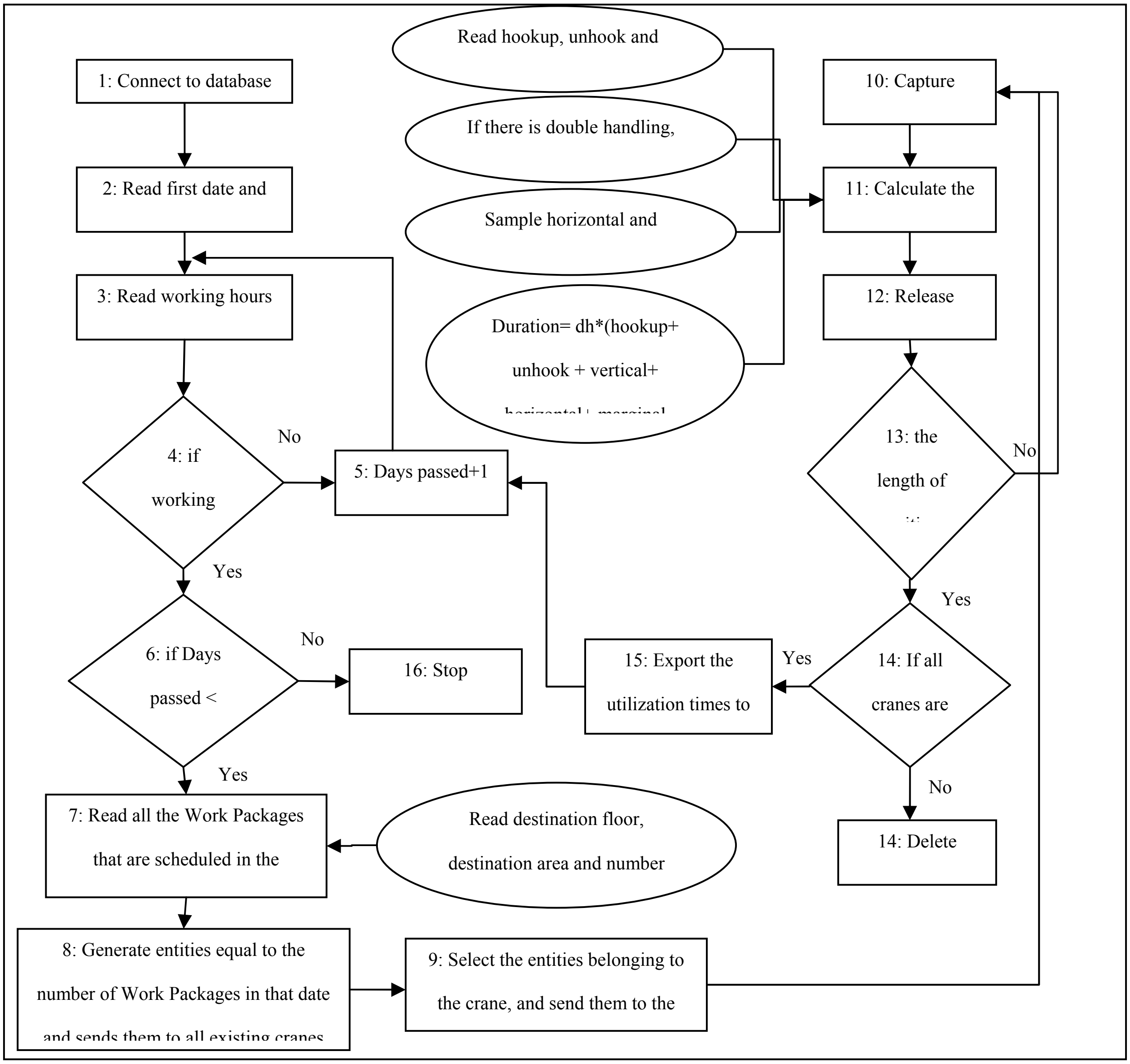


Figure 10. Logic of simulation model

Simulation Template (Elements)

This template is developed by using the basic concepts of the five-element template developed by Appleton et al. (2006). The main concern with the old template was the time required to enter the data in the simulation model and the inefficiency of calculating the moving durations. The new template consists of three elements and is easier for users to work with and enter the data in the database (Figure 11).

The *New Entity element* is used to read and store the first and last date of the schedule. It compares the current date with the final date and stops the simulation if the current date exceeds the final date. It is responsible for advancing the date when an entity comes from the results element, or the working hours are equal to zero, or the current date does not exist in the calendar. It selects the modules that are scheduled to move in the current day and identifies their attributes, such as name, number of picks, destination floor, and destination area from the database. Then, it generates entities equal to the number of picks. It then sends the entities out to the crane elements. The user must identify the database path in this element.

Crane element selects the modules that belong to the crane and sends them to the waiting file, captures the crane, reads the hook and unhook times and the appropriate lifting time, samples the suitable horizontal time for the modules, calculates the travel time and records the utilization time of the crane. After moving the modules, the *Crane* element sends it out, and sets the crane as ‘idle’ before capturing the next module in the waiting file. Because in this model only the sum of working hours is important, the entities do not have priorities. Therefore, they are served based on the first in, first out (FIFO) concept.

Results element: Every time one crane moves a module, it sends an entity into this element; consequently the element checks the status of all the existing cranes in the model. If all the cranes are idle, it exports the daily utilization time of all the existing cranes to the database and releases an entity, which returns to the *New Entity* element. Consequently, the *New Entity* element advances the simulation day, starts reading the new modules in the next day, and send them to the crane elements.

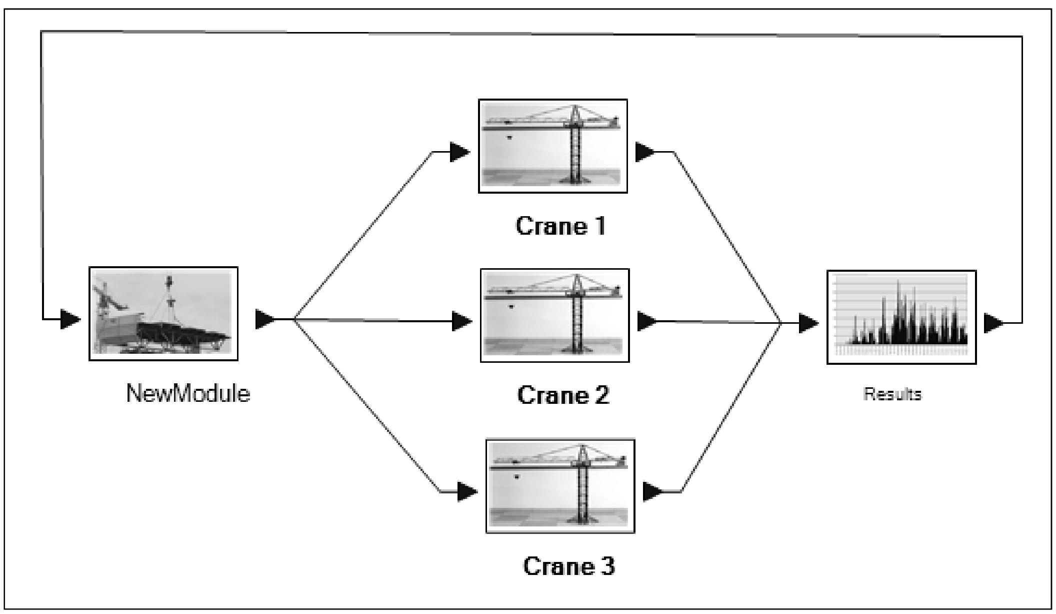


Figure 11. Tower crane model in *Simphony*

Results and Conclusions

The focus of this study is the utilization of tower cranes on a daily basis. Figure 12 displays the utilization chart of Crane 1 in this study. These charts provide a good opportunity to explore various alternatives and to find the best practical solutions. For instance during a period in which Crane 1 was over-utilized, the project manager decided to pump the concrete to the columns instead of lifting it using the crane. This strategy significantly reduced the tower crane utilization in that period. The other capability that this approach provides is in deciding the best pick points for each crane among the available alternatives.

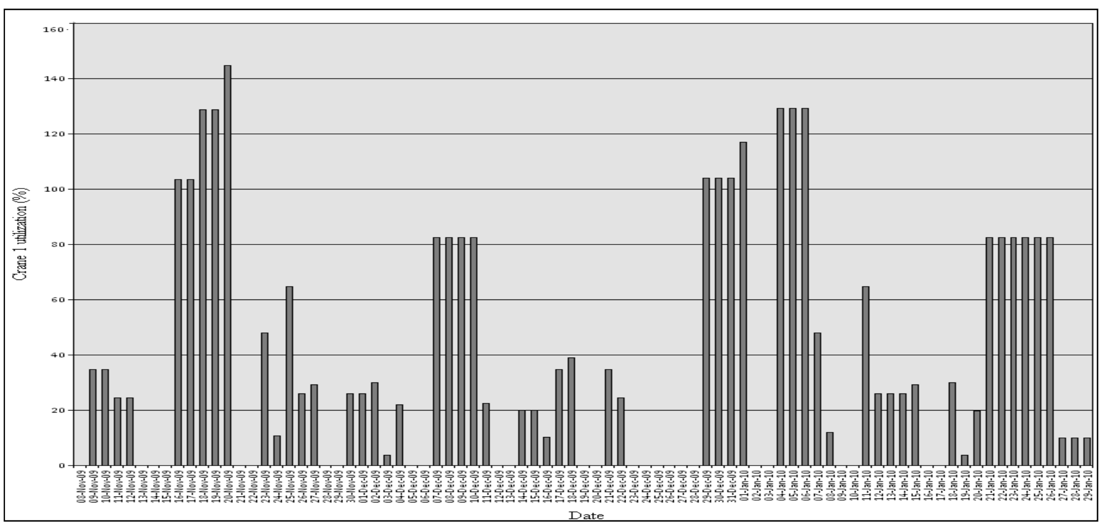


Figure 12. Crane 1 utilization

This study proposes a flexible approach for revising construction schedules according to the availability of tower cranes and enhances the project schedule.

The main purpose of the proposed approach is to revise the schedule and identify periods of bottlenecks caused by tower crane in the construction sites.

Acknowledgement

The authors wish to acknowledge and thank the PCL Construction Company for their kind collaboration.

References

AbouRizk, S.M. and Mohamed, Y. 2000. Simphony - an integrated environment for construction simulation. *Proceedings of the 32nd conference on Winter simulation*: 1907-1914

Appleton, B.J.A., Niaz, M.A., AbouRizk, S.M., Al-Hussein, M., Mohamed, Y. 2006. Priority rating logic simulation model for tower crane operations management. *Journal of Construction Research* 7(1):133-47.

Hajjar, D. and AbouRizk, S.M. 2002. Unified modeling methodology for construction simulation. *Journal of Construction Engineering and Management* 128(2):174-85.

Hajjar, D., Abourizk, S., Xu, J. 1998. Construction site dewatering analysis using a special purpose simulation-based framework. Canadian Journal of Civil Engineering 25(5):819-28.

Martinez, Julio C. 1998. Earthmover - simulation tool for earthwork planning. *Proceedings of the 30th conference on Winter simulation*: 1263 – 1272.

Taghaddos, H., AbouRizk, S.M., Mohamed, Y., Hermann, U. Forthcoming 2008. Simulation-based resource levelling in multi-project construction. *CSCE 2008 Annual General Meeting & Conference*.