Spatio-temporal Visualization and Analysis for Infrastructure Management Systems

Cheng Zhang

A Thesis

in

The Department

of

Building, Civil and Environmental Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Applied Science (Building Engineering) at
Concordia University
Montreal, Quebec, Canada

November 2005

Your file Votre référence
ISBN: 0-494-14245-6
Our file Notre référence
ISBN: 0-494-14245-6

ABSTRACT

Spatio-Temporal Visualization and Analysis

for Infrastructure Management Systems

Cheng Zhang

There is much information needed to manage the activities that occur throughout the lifecycle of an infrastructure system, such as a bridge, including design, construction, inspection and maintenance activities. Conventional infrastructure management systems (IMSs) provide only limited support for representing, visualizing and analyzing the spatio-temporal relationships in this information, which could result in spatial constraints being ignored. Therefore, the spatio-temporal data should be integrated in the IMSs and be visualized and analyzed to improve the efficiency and usability of these systems.

The objective of this research is to propose new approaches and methods for visualizing and analyzing spatio-temporal information during the lifecycle of infrastructure systems. By integrating 4D modeling with several information technologies, the new approaches facilitate spatio-temporal representation in a more accurate and easily understood way. Two types of spatio-temporal analysis are applied: one is workspace representation and conflict detection; the other is cell-based discrete-event simulation of construction processes. Combinations of different 3D shapes are used to represent the workspaces for different activities, which is more accurate than the simple prismatic elements used in previous research to represent workspaces. Conflicts related to space and time are detected based on the defined workspaces during a specific period. Furthermore, cell-

based simulation is used to represent space resources, which enables conflict analysis and visualization of the worksite and the occupation of space. The optimal resource combination can be found not only based on resource constraints, but also the availability of space.

A framework has been developed to investigate the feasibility of the proposed approaches. In addition, a prototype system has been developed based on a case study of Jacques Cartier Bridge in Montreal. The results proved that the proposed methods have a good potential to be integrated in future IMSs.

ACKNOWLEDGEMENTS

First and foremost, I wish to express my deepest gratitude and sincere appreciation to my supervisor Dr. Amin Hammad for his expert guidance, unparallel wisdom, and invaluable support throughout my study at Concordia. I have learnt a lot from him not only about how to do research, but also how to stick to my goal and never give up. His instructions will guide me in the rest of my life.

I would also like to extend my appreciation to Dr. Tarek Zayed for his valuable guidance in construction simulation modeling and Dr. Gabriel Wainer from Carleton University for providing us with CD++ software and for his kind help in developing the cell-based model.

I would like to thank my family, my husband Jian, my daughter Yiyue, and my parents Xiaoxia and Xiaotian for encouraging me to continue my education, and for their unconditional support and understanding. I appreciate their courage in bearing the distance that separated us during my studies.

I would like to thank my friends from Telegroinformatics and Infrastructure Management Group: Yongxin Hu, Elaheh Mozaffari, Basheer Khabir, and Khaled El Ammari for their contribution in working as a team and helping in my research.

I would like to thank Mr. Guy Mailhot from Jacques Cartier and Champlain Bridges Incorporated, Mr. Adel Zaki, Mr. Raymond Coté, and Mr. Germain Cardinal from SNC

Lavalin for providing the data about Jacques Cartier Bridge and evaluating the prototype system, and Mr. André Clément from the GIS Department of Montreal Municipality for providing the GIS data of Montreal.

At last, but definitely not least, I would like to thank all my friends in Montreal who shared my happiness and sadness during this portion of my life. Without their encouragement and help, I could not keep on moving to achieve my goals.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

2D Two-dimensional

3D Three-dimensional

4D Four-dimensional

AEC Architecture, Engineering, and Construction

BMS Bridge Management System

CAD Computer-Aided Design

CDED Canadian Digital Elevation Data

CPU Central Processing Unit

CSG Constructive Solid Geometry

DEVS Discrete-EVents systems Specification

DEM Digital Elevation Model

DXF Data eXchange Format

FM Facilities Management

FMIS Facilities Management Information System

GIS Geographic Information System

GPS Global Positioning System

GUI Graphical User Interface

IAI International Alliance of Interoperability

ISO International Organization for Standardization

IFC Industry Foundation Classes

IMS Infrastructure Management System

LBC Location-Based Computing

LoD Level of Details

MTM Modified Transverse Mercator

NAD83 North American Datum of 1983

OODB Object-Oriented DataBase

SQL Structured Query Language

STEP STandard for the Exchange of Product model

TIN Triangulated Irregular Network

VR Virtual Reality

VRML Virtual Reality Modeling Language

XML eXtensible Markup Language

CHAPTER 1 INTRODUCTION

1.1 GENERAL BACKGROUND

Modern infrastructure is key to the prosperity of our cities and the health of our communities (Infrastructure Canada 2002). Infrastructure systems are usually large and have long life. There is much information needed to manage the activities and events that occur throughout the lifecycle of an infrastructure system, including planning, design, construction, operation, inspection and maintenance activities.

The traditional approach to represent information about the infrastructure is to build data-bases, including Computer-Aided Design (CAD) drawings, inspection and maintenance databases, images, and the documents related to the design, construction, maintenance, etc. In many cases, maps of the area of the project are added to show the location of the infrastructure. Various tools are available to create and record these data, such as CAD tools, which are used to create the 2D and 3D drawings of the infrastructure; Geographic Information Systems (GISs), which are used to create the maps of the environment around the infrastructure; project management software packages, which are used to create the schedules of the construction and maintenance activities; Database Management Systems (DBMSs), which record the data of design, construction, maintenance and inspection, and generate all kinds of reports; and Infrastructure Management Systems (IMSs), which help in the decision making related to infrastructure systems. However, the tools and functions available in current IMSs provide only limited support for representing, visualizing and analyzing the spatio-temporal relationships within the lifecycle, and the integrated information is not enough to create a clear visual model to demonstrate the changes that take place during the lifecycle of the infrastructure. Therefore, a new method is needed to explicitly represent the spatio-temporal data and to facilitate spatio-temporal analysis in IMSs, thus improving the efficiency and usability of future IMSs. The objective of spatio-temporal information integration in IMSs is to digitally integrate spatial data, such as a 3D digital model of an infrastructure, with the temporal information of the activities and the events that occur at certain time or during a certain period, and finally to facilitate spatio-temporal analysis, such as workspace conflict detection.

Among all kinds of infrastructure, bridges are expensive public properties, and they are used to serve the public for tens or even hundreds of years as part of the transportation network. Keeping the bridges in healthy condition is important responsibility of the bridge authorities to protect public safety and enhance traffic flow. During their long service life, bridges deteriorate under the effect of loads and environment impacts, which results in cracks, corrosion from deicing salt, etc. The only effective means to protect bridges from these damages is performing proper bridge management to provide the bridges with suitable maintenance and repair work. Because of the importance and complexity of bridges, Bridge Management Systems (BMSs) are selected as the main focus of this research to investigate the feasibility and the potential benefits of the integration of spaio-temporal data representation and analysis.

1.2 RESEARCH OBJECTIVES

The main objective of this research is to investigate spatio-temporal visualization and analysis methods during the lifecycle of a birdge. Several approaches are proposed to represent and visualize the spatio-temporal information, then spatio-temporal analysis is applied to find potential conlicts that may occur during construction or rehabilitation activities (e.g., conflicts between equipment workspaces), or to find the optimal combination of resources by simulating the work processes. Our research objectives are:

(1) Applying 4D modeling throughout the lifecycle of a brige. The roadmap for the model-based Information and Communications Technology (ICT) suggests methods for integration based on 4D models (3D + time) (Figure 1.1).

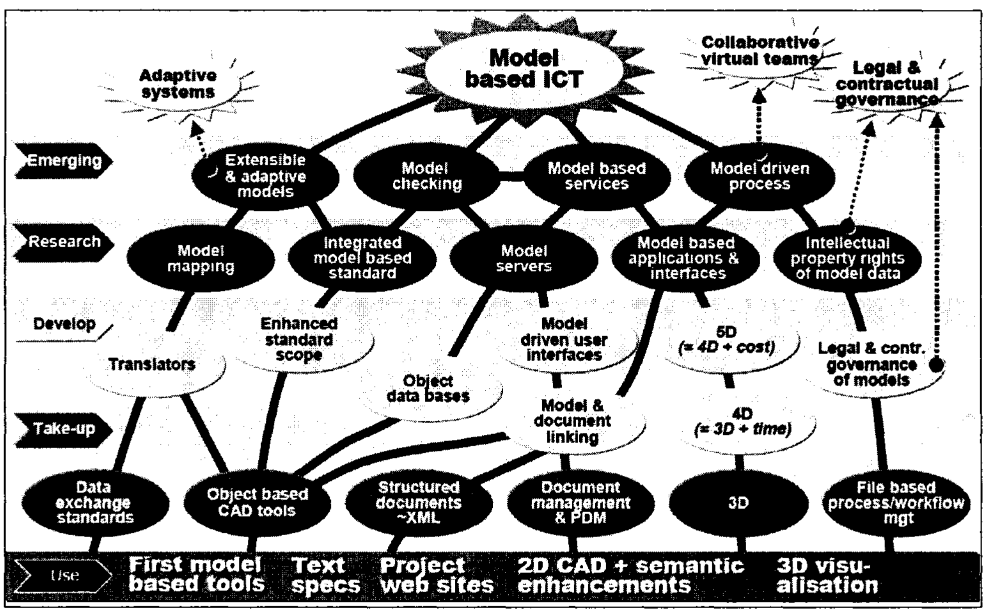


Figure 1.1 Roadmap for model-based ICT (ROADCON 2003)

In this research, we are going to integrate spatio-temporal information with the 3D model in BMSs to build a 4D model, which will facilitate spatio-temporal visualization and analysis. Based on these 4D models, 5D or nD models can be developed in the future by integrating other information, such as cost, to achieve the vision of the model-based ICT.

(2) Analyzing spatio-temporal conflicts during construction. This objective will focus on workspace conflict detection and work processes simulation considering spatio-temporal constraints. It aims to extend the previous research on workspace representation and analysis in the case of large infrastructure projects. A specific representation of equipment workspaces using composite shapes will be investigated and used to detect workspace conflicts. Furthermore, cell-based simulation tools will be used to investigate dynamic spatio-temporal representation and analysis.

CHAPTER 2 LITERATURE REVIEW

2.1 REVIEW OF BRIDGE MANAGEMENT SYSTEMS

The collapse of the Silver Bridge of the United States in 1967, where 46 people died in the accident, called the legislation for creating bridge management systems (BMSs) and developing the National Bridge Inspection Standards (NBIS) and the National Bridge Inventory (NBI) (Shepard and Johnson 2001). Since 1970s, the NBI is used as the primary source of data for bridge management. In the 1990s, two bridge management systems were developed in the United States, i.e., Pontis and BRIDGIT. Pontis is the predominant system employed in the U.S. In addition, many modern BMSs (Table 2.1) have entered into practice focusing on collecting more data about element condition, cost, traffic, and historical data (Frangopol et al. 2001). These BMSs assist the manager to determine the optimal time to execute improvements actions on a bridge and use the limited resources to balance lifetime reliability and lifecycle cost.

A BMS is a rational and systematic approach for organizing and carrying out all the activities related to managing network-level bridges (Itoh et al. 1997). It is more than a collection of facts; it is a system that looks at all the information concerning all the bridges and is able to make comparisons between them in order to rank them according to their importance within the overall infrastructure with regards to safety and budgetary constrains (Ryall 2001). For any BMS to work effectively, it has to have as much pertinent input information about the bridge as possible. Basically all systems will have modules dealing with inventory, inspection, maintenance, and finance. Analyzing and processing all of this information will allow the management to have better control over the bridges.

Table 2.1 Partial list of BMSs (Ryall 2001)

Country	Authority	BMS	Year
USA	Pennsylvania Department of Transportation (PenDOT)	The Pennsylvania BMS	1987
Finland	The Roads and Waterways Administration Department(RAW)	Finnish RWA, BMS	1990
UK	The Department of Transport	HiSMIS system	1990
USA	Federal Highway Administration FHWA	PONTIS	1992
USA	NCHRP & National Engineering Technology Corporation	BRIDGIT	1993
Canada	Ontario's Ministry of Transportation	Ontario BMS	1999
UK	Surrey Country Council (SCC)	Modified COSMOS	2000
South Africa	National Roads Agency (NRA)	South Africa BMS	2000

The bridges' information in a particular transportation network is stored in a database which forms the heart of a BMS. Bridge database management includes the collection, updating, integration, archiving the following information:

- Bridge general information (bridge location, name, type, live load capacity, etc.)
- Design information (physical properties of the elements)
- Collection of inventory data
- Regular inspection records
- Condition and strength assessment reports

- Repair and maintenance records
- Cost records, etc.

All these data are used to estimate deterioration and performance models for bridges, predict the performance of the bridge population, and optimize the maintenance strategies (Agile Bridge Manager 2005).

At the early stage of the development of BMSs, the basic requirements are: data storage, cost and deterioration models, and optimization models for analysis and updating functions. However, the requirements of modernizing infrastructure management and the availability of information technology make it necessary and possible to develop more sophisticated and flexible systems.

In the mid 1980s, with the availability of powerful Personal Computers (PCs) and software systems, modern BMSs could be built to overcome the limitation of paper-based systems in dealing with a large number of bridges. Compared with paper-based BMSs, the new systems have many advantages such as powerful data updating, information processing, information sharing, computer-generated decision making models, etc. Itoh et al. (1997) described a network-level bridge life-cycle management system that integrates geographic information, design information, inspection and maintenance information, image information, and earthquake information. Several applications of this system are discussed, which include location selection, maintenance plan optimization, and spatial retrieval.

In addition, computer graphic and image processing techniques enable integrating images and drawings of the bridges with the BMS to help the decision-makers get more information about the geometry of the structure. During a routine bridge inspection, information can be recorded at a high level of photographic detail (Jauregui et al. 2005). Photographic images can be moved from the flat, 2D world into a more immersive, 3D environment complete with interactive components. This makes it possible for an inspector or engineer to explore and examine a bridge's physical condition as if he/she was actually at the site, with the simple use of a computer and mouse in lieu of virtual reality equipment such as goggles, headsets, or gloves (Apple 2003). The advantages of using computer graphic techniques compared with conventional tools are: more realistic data representation, easier data collection and updating, less data input errors, more direct interaction with the system, and the possibility to visually integrate time and cost information. This new integration is one of the critical steps for the achievement of the application of model-based Information and Communications System (ICT) in the construction industry (ROADCON 2003).

The development of modern BMSs also requires making the information accessible to mobile on-site workers and update the information with less human errors, data loss, and misinterpretation of data caused by transferring data from one format to another. Hammad et al. (2005) proposed Mobile Model-based Bridge Lifecycle Management Systems (MMBLMSs) to support distributed databases and Location-Based Computing (LBC) by providing user interfaces that could be used on thin clients equipped with wireless communications and tracking devices. LBC adds the possibility of monitoring the execution of the work in real time using tracking techniques, such as the Global Positioning System (GPS), while comparing the location of the workers and equipment with the corresponding workspaces as planned in the construction schedule (Navon and Goldschmidt 2003). The monitoring could be used to measure the efficiency of the work, check safety requirements, or control the equipment to automate the construction work. In summary, more and more information technologies have been and will be used in BMSs to enable efficient and effective management.

2.2 SPATIO-TEMPORAL REPRESENTATION

As an important part of the information that should be included in BMSs and other IMSs, spatio-temporal information is used to show the dynamic changes occurring during the lifecycle of bridges. Many tools and methods are used to represent this information in the Architecture, Engineering, Construction and Facilities Management (AEC/FM) industry.

2.2.1 Spatial representation

(1) Computer-Aided Design (CAD)

CAD technology has been one of the most influential IT innovations. The use of CAD has quickly spread among AEC firms (Kale and Arditi 2005). 2D and 3D drawings can be created using CAD software, such as AutoCAD, MicroStation, and so on. They are the basic tools that can represent spatial information of structures and are widely used in design, construction and operation. The CAD systems provide a variety of data representation schemes (e.g., wire frames, surfaces, and solid models) and data exchange

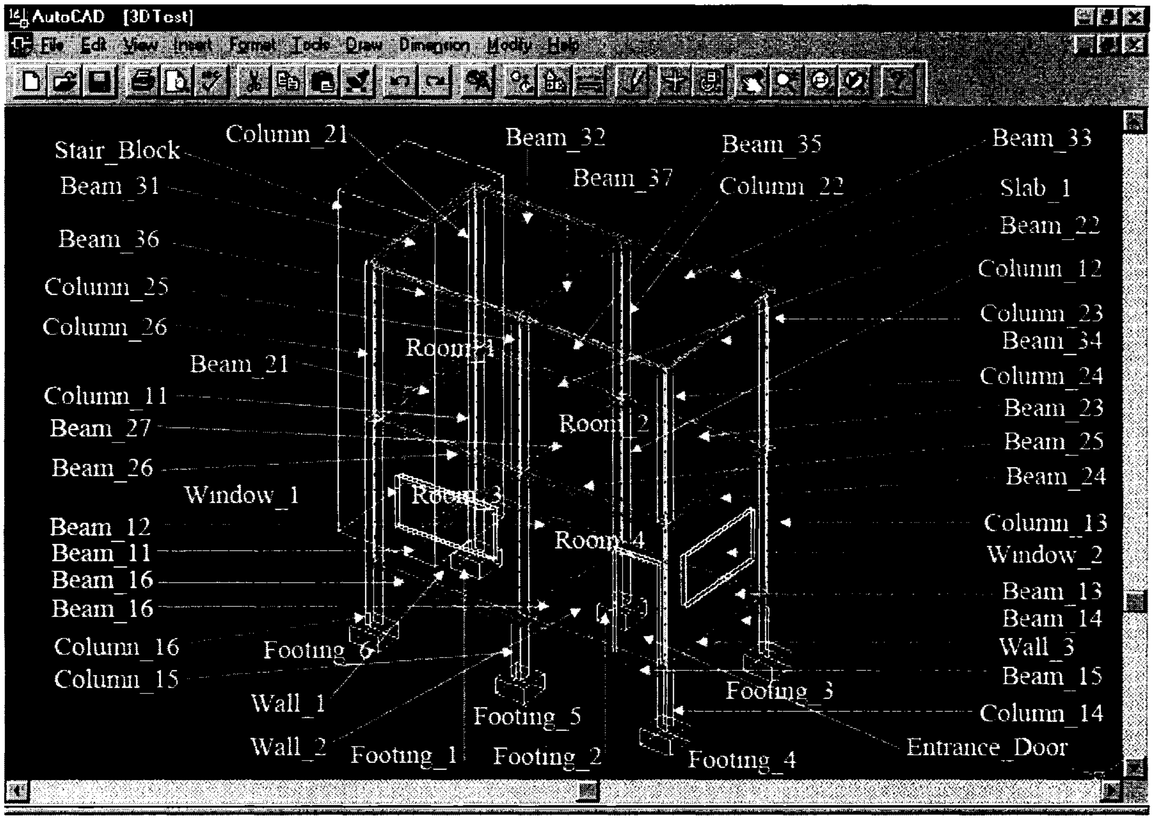


Figure 2.1 3D model of a building (Nguyen and Oloufa 2002)

protocols. In these CAD systems, topological information describing the spatial relations between the components of the structure can be retrieved to support constructability analysis, construction planning, and so on (Nguyen and Oloufa 2002). A typical 3D model of a building is shown in Figure 2.1.

Constructive Solid Geometry (CSG) is a method of representing and processing graphical solid objects. A CSG solid is constructed from a few *primitives* with *Boolean* operators. Spheres, cones, cylinders and rectangular solids are shapes commonly used as primitives. A CSG solid is represented as a function of the primitives and operations used in its composition. Its structure is represented by a tree, called CSG tree, where these operations are hierarchized. Each applied operation is represented as an internal node, and each primitive as a leaf node. The Boolean set operations are intuitive ways to combine solids based on the set operations. The three main operations are:

- Union ($\cup$): The resulting solid corresponds to all the volume from the operand solids.
- Intersection ($\cap$): The resulting solid corresponds to the coincident volume from the operand solids.
- Difference ($-$): The resulting solid corresponds to the volume from one of the operand solids outside the other ones.

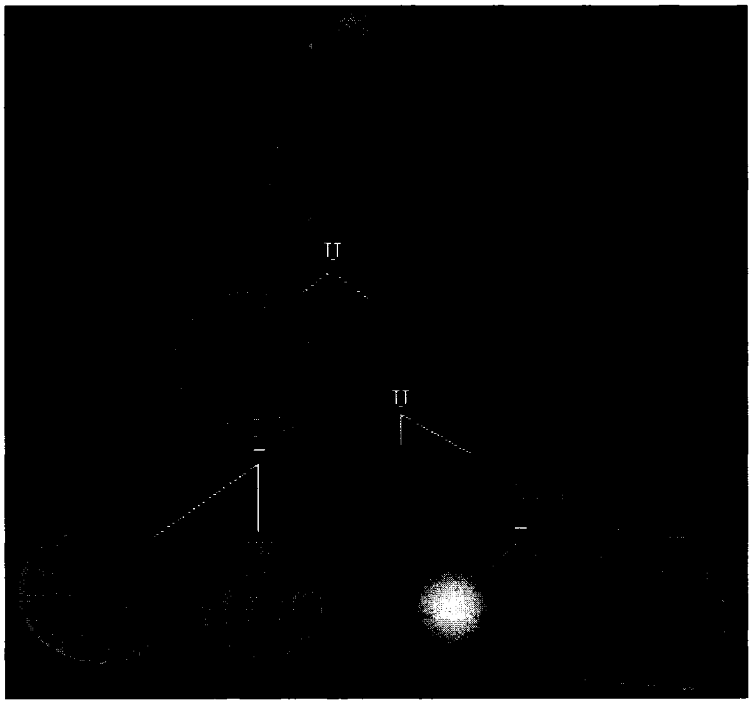


Figure 2.2 CSG representation (CSG API 2004)

As shown in Figure 2.2, the shape on the top is the combination of several primitives using the three operations. Due to its simplicity and efficiency, the Boolean set operations were implemented on most of the 3D modeling tools, being used as a resource to compose 3D models. Thus, a CSG solid can be represented by a set equation and can also be considered a design methodology.

(2) Virtual Reality and 3D Animation

Virtual Reality (VR) is a technology that creates a virtual three-dimensional space in a computer to visually reproduce the shape, texture and movement of objects (Miyamoto and Konno 2005). One of the benefits of the virtual environment is the ability to present users with a realistic simulation for a given application domain, thus making them more understandable.

VR can be used in IMSs to evaluate the proposed upgrading of existing facilities because it can help in the presentation of complex ideas, modeling concepts, and simulating results, and may also facilitate communication between evaluation analysts and individuals involved in the decision making process (Gracanin and Collura 2001). Jauregui et al. (2005) proposed an approach using QuickTime Virtual Reality (QTVR) to document bridge inspection data in a virtual reality format at a high level of photographic details. This makes it possible for an inspector to explore and examine a bridge's physical conditions as if he/she was actually at the site. Also, VR can be used for education purposes. For example, an education system about damage processes is developed for educating bridge maintenance engineers about deteriorating bridges (Miyamoto and

Konno 2005). In addition, virtual reality can be used for space planning in construction. For example, Heesom et al. (2003) proposed a VR system that allows the allocation and visualization of construction workspaces.

Similar to VR, 3D animations are used in construction research. Kamat and Martinez (2004) presented research about visualizing simulated construction operations in 3D using a system called VITASCOPE (Figure 2.3). 3D animation can be used to show the construction process based on simulation result and help in detecting any discrepancies during the development of the simulation model (Nassar et al. 2003).

Figure 2.3 VITASCOPE animation snapshot of a construction site (Kamat 2003)

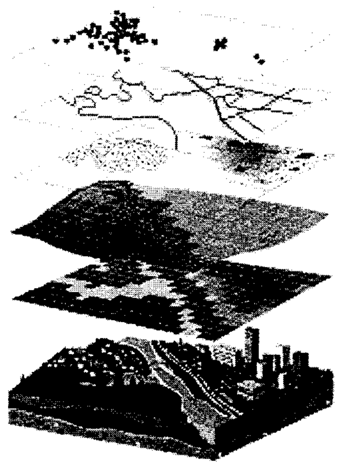


Figure 2.4 Spatial modeling in GIS (GIS 2005)

(3) Geographic Information Systems (GISs)

GIS is a technology that manages, analyzes, and disseminates geographic knowledge. It is an organized collection of computer hardware, software, geographic data, and personnel designed to efficiently capture, store, update, manipulate, analyze, and display all forms of geographically referenced information (GIS 2005). GIS represents reality as a set of map layers and relationships between them (Figure 2.4). The major advantage of a GIS is that it allows identifying the spatial relationship between map features.

Nowadays, software integrated with GIS data is broadly used for design, construction, and maintenance. GISs are also attracting attention in infrastructure management. Early in 1996, visual IMS, a GIS-based and multimedia-integrated infrastructure management system was developed, which covered three main parts of infrastructure system: roads, water supply, and wastewater. In this system, GIS is used to find the infrastructure that share the same location to facilitate carrying out the Maintenance, Rehabilitation, and

Replacement (M, R&R) in the most efficient and economical sequence (Hudson et al. 1997). The traditional documents and 2D/3D drawings could not help the user to understand the situation quickly and easily because they do not adequately handle topology and semantic properties and do not offer the GIS functionality (Groger et al. 2004). Itoh et al. (1997) described a network-level bridge life-cycle management system that integrates a GIS module and an object-oriented database (OODB) module to represent the relationship between the bridges and other urban elements affecting the bridges, such as roads and rivers.

With the rapidly increasing needs of 3D information, 3D GIS functionalities are becoming more available in commercial software and can provide extended tools for 3D navigation, animation and exploration. However, many of these systems are still lacking full 3D geometry representation (Zlatanova et al. 2002). In large-scale infrastructure projects, GISs are inevitably needed for generating information that relates to locations. A 3D map of the area covered by a BMS is needed to permit the computations based on the location of the users. Using this map, the models of bridges can be based on geographic coordinates. In order to create a 3D map, 2D maps can be draped on the Digital Elevation Model (DEM) of the same area.

DEMs have a major role in digital mapping. They are used in GIS for land-management applications. Moreover, they play the same role as contours and relief shading on conventional paper maps but offer greater analytical potential (Canada3D product standard 2001). Canada3D is a DEM produced by the Canadian Forestry Service, Ontario region. The DEM consists of an ordered array of ground elevations providing coverage of the entire Canadian landmass. It has been derived from the cells of the Canadian Digital Elevation Data (CDED) at the 1:250000 scales. Canada3D is available in two forms: grids regularly spaced at 30 or 300 arc-seconds recorded in ASCII file format. The elevation values are expressed in meters with respect to mean sea level, in accordance with the North American Datum of 1983 (NAD83). CDED Level 1 (CDED1) consists of an ordered array of ground elevations at regularly spaced intervals. CDED1 is based on National Topographic Data Base (NTDB) digital files at scales of 1: 50,000 and 1: 250,000 or various scaled positional data acquired from the provinces and territories, according to the National Topographic System (NTS). The coverage for every file corresponds to half an NTS map, which means that there are western and eastern parts to the CDED1 for each NTS map.

2.2.2 Temporal representation

Time can be considered as the fourth dimension when representing the changes over time of a structure 3D model. Adding temporal information to the spatial representation allows for managing information about the states of objects at particular times and places, and also the properties of what happened, when, how, and why (Worboys and Duckham 2004). There are a number of techniques that can be used to schedule project activities. The first scheduling method used in construction was the Gantt chart (Kerzner 1995). A Gantt chart provides a graphical representation of the project plan. It includes the following elements: the activities that comprise the project, the estimated duration for each activity, and the order of occurrence of the activities. The Gantt chart is a useful tool for monitoring the progress towards accomplishing objectives. It is easy to prepare and understand.

2.2.3 Spatio-temporal representation

(1) 4D modeling

Traditional design and construction planning tools, such as 2D drawings and network diagrams, do not support the timely and integrated decision making necessary to move projects forward quickly. They do not provide the information modeling, visualization, and analysis environment necessary to support the rapid and integrated design and construction of facilities. Extending the traditional planning tools, visual 4D models combine 3D CAD models with construction activities to display the progress of construction over time (Emerging Construction Technologies 2005).

At the Center for Integrated Facility for Engineering at Stanford University, Fischer (2001) has lead research projects related to 4D CAD since 1994. Their research has shown that more project stakeholders can understand a construction schedule more quickly and completely with 4D visualization than with the traditional construction management tools. Figure 2.5 shows a typical 4D model (Koo and Fischer 2000).

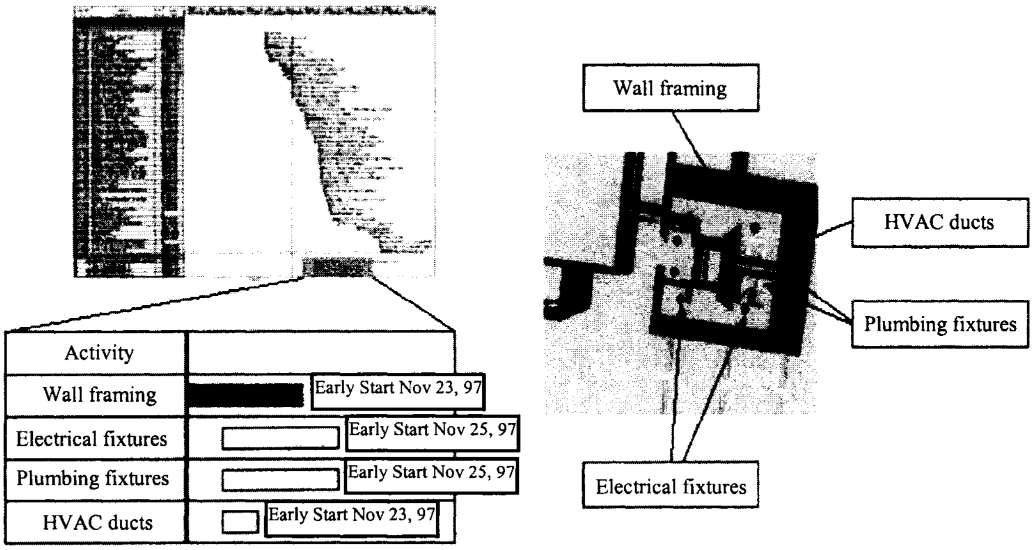


Figure 2.5 4D model showing concurrent activities (Koo and Fischer 2000)

4D models help project stakeholders to: (1) Understand the relationship between construction activities and facility operation for retrofit projects; (2) Understand and improve the use of work, access, and staging areas over time; (3) Identify spatial conflicts among crews and other production elements; (4) Analyze activity sequencing; (5) Improve constructability; (6) Improve work flow for subcontractors; and (7) Visualize the construction work to be done for a work zone, time period, or subcontractor. The Stanford research team has been formalizing the construction knowledge necessary to build 4D models and has developed a methodology that guides project planners in generating 4D models from 3D product models. Moreover, several visualization techniques, such as highlighting and overlaying were developed to visually communicate relationships between project information (Liston et al. 2000). Furthermore, the Industry Foundation Classes (IFC) interoperability standards were tested in a project called Product Model and

Fourth Dimension (PM4D) which constructed and maintained object-oriented product models with explicit knowledge of building components, spatial definitions, material composition, and other parametric properties. The approach allowed the project team to utilize visualization tools to review spatial designs in virtual walk-through, compare lighting schemes in photo-realistic renderings, and comprehend construction sequences in 4D animations, all leveraging the same electronic design information (Fischer and Kam 2002).

Many software companies are also interested in developing commercial 4D modeling software. Rischmoller et al. (2001) from Bechtel Group, Inc., described 4D Planning and Scheduling (4D-PS) as Computer Advanced Visualization Tools (CAVT) and their successful application in the case study of a mining facilities project in Chile. 4D-PS allows simulating and interacting with construction sequences through interactive graphic display devices. Recently, Stageworks system (Stageworks 2005), developed by Bechtel, has proved that 4D visualization is helpful during construction. Stageworks combines the planned work sequences from all contractors and trades on the site, and builds a 4D graphic visualization (a clash report), which highlights conflicts between different contractors planning different activities at the same time and in the same locations. The Navigator software developed by Bentley also applies 3D model review, animation and 4D simulations (Bentley 2005). The user can simulate the schedule, interact with graphical and non-graphical objects, and detect the interference between objects.

One of the limitations of current 4D modeling tools is that they have single level of detail, which means that the models do not support seamless aggregation, elaboration, and refinement of model details. Another limitation is that they do not support computer-based analysis of cost, safety, and other performance metrics (Emerging Construction Technologies 2005). Furthermore, although 4D models have already been built to support construction planning and scheduling (Zhang et al. 2000), these models are not integrated with IMSs (Hammad et al. 2005). 4D models will allow spatio-temporal visualization and analysis that are not possible with present BMSs and other IMSs. This integration of space and time will result in the following advantages: (1) Visualizing different types of data, e.g., displaying the changes in a bridge 3D model at a specific time or during a specific period of its lifecycle; (2) Providing a user-friendly interface which can reduce data input errors; (3) Facilitating data sharing; and (4) Improving the efficiency of database management.

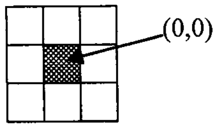
(2) Cell-based modeling

In 1948, John Von Neumann and Stephan Ulam defined a modeling formalism, called Cellular Automata (CA), suited to defining spatial systems and to allowing the description of cell-based models by using simple rules (Wolfram 1986). In CA, space is represented by a uniform grid with each cell containing a few bits of data. At each step, each cell computes its new state from that of its nearby neighbors. CA are well suited to describing spatial distribution of resources. The cells in the lattice are updated according to a local rule in a simultaneous and synchronous way, using a local computing function. This function considers the state of the present cell and a finite set of nearby cells (called the neighborhood). Unfortunately, these CA have the constraint of being synchronous.

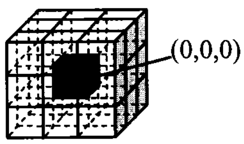
This fact reduces the timing precision for the models, and, consequently, the computations related to time resources cannot be fully used. Moreover, it should be considered that in most cell spaces there are a large number of quiescent cells. CA have attracted the attention of researchers in geography, ecology, and other environmental sciences because of their ability to model and visualize complex spatially distributed processes (Ahmad and Simonovic 2004).

In the 1970s, Bernard Zeigler (1976) defined a theory for Discrete-EVents systems Specification (DEVS). It is a formal approach for building models using a hierarchical and modular approach. This approach allows the developer to build a Model Base permitting easy reuse of models that have been validated. A real system modeled with this paradigm can be described as several sub-models coupled into a hierarchy. Each model can be behavioral (atomic) or structural (coupled), consisting of a time base, inputs, states, outputs and functions to compute the next states and outputs. The basic idea is that each model uses input/output ports in the interface to communicate with other models.

Zeigler (1976) also defined a cell space model that consists of an infinite set of geometrically defined cells, each cell containing the same computational apparatus as all other cells and connected to other cells in a uniform way. As shown in Figure 2.6 (a), for a cell located at point (0,0), the nearest neighbors would be those located at: (0,1) (1,0) (0,-1) (-1,0), which are at a distance of one cell away orthogonally and (1,1) (-1,1) (-1,-1) (1,-1) which are at a distance of one cell away diagonally (Zeigler 1976). The cell model could be two dimensional or three dimensional.



(a)



(b)

(-1,-1,1)	(-1,0,1)	(-1,1,1)	
(0,-1,1)	(0,0,1)	(0,1,1)	Upper layer
(1,-1,1)	(1,0,1)	(1,1,1)	

(-1,-1,0)	(-1,0,0)	(-1,1,0)	
(0,-1,0)	(0,0,0)	(0,1,0)	Current layer (0,0,0) is the current cell
(1,-1,0)	(1,0,0)	(1,1,0)	

(-1,-1,-1)	(-1,0,-1)	(-1,1,-1)	
(0,-1,-1)	(0,0,-1)	(0,1,-1)	Lower layer
(1,-1,-1)	(1,0,-1)	(1,1,-1)	

(c)

Figure 2.6 Cell representations:
(a) 2D representation of cells; (b) 3D representation of cells; and (c) Neighborhood cells in 3D

Figures 2.6 (b) and (c) show the generalization of the cell neighborhood representation in three dimensions. For 2D models, one layer is enough; while for 3D models, we may have as many layers as necessary to cover the physical space of the problem. In simple models, one value per cell could be sufficient. However, cell modeling in engineering applications often needs several attributes for each cell. For simplicity, these attributes could be attached to several interrelated layers so that the corresponding cells in different layers will contain one attribute each. In this case, we may have as many layers as the number of the attributes.

Based on the concept of cell spaces, Wainer (1998) developed an approach called Cell-DEVS (Wainer and Giambiasi 2002). Cell-DEVS describes cell spaces as discrete event models using the DEVS formalism, including delay functions to have a simple definition of the timing of the cell. The proposal of the Cell-DEVS paradigm considers each cell of a CA as a hierarchical and modular discrete events model (Wainer and Giambiasi 2001). In this way, complex models can be defined using a continuous time base. It also allows associating several kinds of delays for each cell, easily allowing the definition of complex models. The cell state changes according to a local function that uses the present cell state and a finite set of nearby cells. Many applications of Cell-DEVS have been developed including applications for surface tension analysis, studies of ecological systems, and a specification language used to define traffic simulations (Davidson and Wainer 2005).

2.2.4 Space and time scales

As discussed in Section 2.2.3, one of the limitations of current 4D modeling tools is using a single level of details (LoDs), which means the same model is displayed for different purposes of management. However, a simpler model could be enough in the case of planning, while a more detailed one is needed when doing inspection on site. Also, the trade-off between graphic quality and rendering time is another reason for using different LoDs. In the field of computer graphics, the basic idea of LoDs is to use simpler versions of an object as they make less and less contribution to the rendered image. When the viewer is far from an object, a simplified model can be displayed to speed up the rendering. Due to the distance, the simplified version looks approximately the same as the more detailed version (Shamir and Pascucci 2001). A common way of selecting a LoD is to associate the different LoDs of an object with different ranges: r_1, r_2, r_3,... (Figure 2.7) (Akenine-Moller and Haines 2002). LoD algorithms consist of three major parts: generation, selection, and switching. Generation is generating different representations of a model with different details. Selection is choosing a LoD model based on some criteria. Switching is changing from one LoD to another. As for the time LoDs, different types of schedules have different time units, such as month, week, day, and hour. Although LoD techniques are common in computer graphics, they have not been applied in BMSs and specialized methods should be developed to satisfy the special needs of these systems.

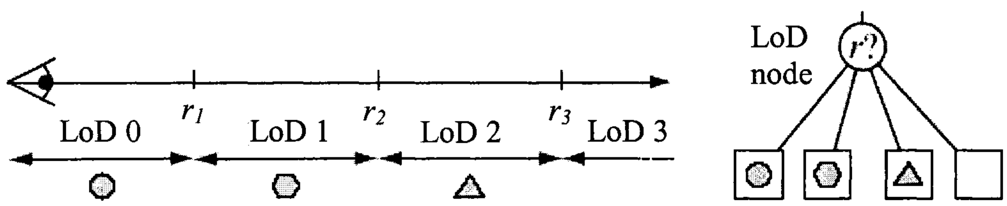


Figure 2.7 The concept of LoDs (Akenine-Moller and Haines 2002)

2.3 SPATIO-TEMPORAL ANALYSIS

Spatio-temporal analysis is the process of extracting or creating new information about a set of geometric or geographic features at a certain point of time. This type of analysis is useful for evaluating the suitability of a certain location, such as problem in site layout planning, or for predicting spatial conflicts, such as conflicts between workspaces.

2.3.1 Spatial analysis

(1) Spatial analysis in CAD

Spatial analysis in CAD mainly focuses on the design stage to find the conflicts between different systems, such as the structural system, electrical system, and Heating, Ventilation, and Air Conditioning (HVAC) system (Santos et al. 2001). A lot of research has been done in building design to provide spatial analysis tools in order to match the spatial characteristics with intended uses and offer a range of physical definitions of habitable space. The analysis level ranges from informal inspection to a more careful checking of sight lines and boundaries (Do and Gross 1997). For example, Kunigahalli et al. (1995) presented an algorithm for extracting topological relationships from a wire-frame CAD model of a concrete floor.

(2) Spatial analysis in GIS

There are two types of analysis in GIS: raster and vector analysis. For example, *ArcView Spatial Analyst* software is specialized in spatial analysis, which helps the user discover and better understand spatial relationships in the data. *ArcView Spatial Analyst* uses raster analysis, which determines locations based solely on cell locations and the output data are usually computed on a cell-by-cell basis where surrounding cells do not influence the output. Several functions are provided, such as distance mapping, density function, surface function, and neighborhood-analysis functions, to enable the user develop specific applications, such as minimizing environmental impact by modeling potential landscape and hydrologic changes due to development (ArcView 1996). GIS analysis has been recognized in the majority of civil engineering disciplines as a beneficial technology (Miles and Ho 1999). A detailed review of GIS applications in civil engineering and environmental modeling is done by Miles and Ho (1999).

2.3.2 Temporal analysis

As discussed in Section 2.2.2, Gantt charts are easy to prepare and understand; however, as construction projects have become more complex and fast-tracking delivery method has become popular, other scheduling and control methods have been developed to further analyze the time dimension in construction.

(1) Scheduling tools

In the 1950s, a technique called Line of Balance (LOB) was developed by the United States Navy to assist with production schedule control in manufacturing plants. Using this technique, it became possible to predict in advance the temporal changes in raw material production needed to meet a specific target production of the final product. This technique was found to be applicable to any work where many units of similar type are manufactured, assembled, and installed in sequence. In the construction field, it has been successfully used to control housing projects, high-rise buildings, highways, and large civil and municipal work such as tunneling and utilities pipe installation (Kerzner 1995).

In the late 1950s, the Critical Path Method (CPM) and the Program Evaluation and Review Technique (PERT) appeared. CPM is a technique based on network theory for calculating a schedule's critical path using the durations and precedence of activities. PERT was proposed by the United States Navy's Special Project Office and Bose Alen Hamilton Co. for use in the development of Polaris missiles. PERT introduced three-point estimates: optimistic, most probable, and pessimistic, based on the assumption that the required duration of an activity follows a beta distribution. On the other hand CPM uses a one-point estimate (Kerzner 1995).

These scheduling tools are concerned only with time and are often integrated with other tools during construction (e.g., workspace availability, resource allocation, cost consideration) to find the constraints and balance the trade-offs between different factors.

(2) Simulation tools

Simulation is the execution of a model represented by a computer program that gives information about the system being investigated. The simulation approach of analyzing a model can be used instead of the analytical approach, where the method of analyzing the system is purely theoretical. Simulation attempts to duplicate "real-life" engineering problems with computer models using actual data from the engineering situations being examined.

The objectives of simulation are usually to determine the impact of a change of input on the whole system or to local parts of the system. Simulation can be used before the construction of a system for performance evaluation, to estimate system throughput, system delay, and to identify any potential bottlenecks in the system. Simulation can be also used to determine the optimum layout and the capital and overhead requirements for a system. It is widely used for manufacturing systems, material handling and computer systems.

Simulation has been used in construction for process planning and resource allocation. Several simulation software for construction were developed to model and analyze the process and help decision-making. MicroCYCLONE (Halpin 1977), Simphony (Hajjar and AbouRizk 1999), and Stroboscope (Martinez 1998) are the most popular software used in the construction area. They are proved to be effective and efficient in simulating various construction projects. AbouRizk et al. (1999) modeled and analyzed the tunneling process using the Special Purpose Tunnel Template developed with Simphony. Zayed and Halpin (2001) applied simulation to concrete batching operations to analyze alternative solutions and resource management using MicroCYCLONE. Martinez et al. (2001) applied the simulation of air-side airport operations.

However, due to the specific characteristics of workspaces, it is not easy to detect spatial conflicts without an explicit representation of space in the construction simulation. Early in the research of Halpin (Halpin and Riggs 1992), it was mentioned that location or space-type flow units usually constrain the access to certain work processed and thus constrain the movement of other type units. However, in MicroCYCLONE (Halpin and Riggs 1992), space is not represented based on any spatial modeling. Instead, space is represented implicitly as any other resources using abstract symbols and it is limited to the operation spaces of equipment, excluding other workspaces, such as moving paths. Kamat (2003) proposed detecting conflicts between any pair of mobile or static objects on a construction site based on collision detection methods implemented within visualization tools of discrete event simulators. However, this approach is based on visualizing the results of the simulation rather than considering spatial issues in the simulation itself.

Some research has been done to investigate the space visualization in construction simulation. Zhang et al. (2002) used 2D icons to represent the resources, which can move along the path between activities. However, this research did not clearly represent the spatial relationships between different activities although there are icons moving from one activity to another. It is difficult to understand the worksite situation with links between activities only. Zhong et al. (2004) developed the GIS-based Visual Simulation System (GVSS) to offer planning, visualizing, and querying capabilities of complex construction processes. However, workspace conflicts are not discussed in their research. This could result in spatial constraints being ignored in the simulation, and the output may not reflect the real situation of the construction site. Spatial problems need to be studied in a general way that is easy to understand, and a model should be built in a way that the space can be represented explicitly. Furthermore, the space representation can improve the accuracy of calculating the duration of some activities, such as the transportation between two locations, with the real representation of the geometrical relationships between different points on the work site instead of using approximate average values.

2.3.3 Spatio-temporal analysis

(1) Workspace analysis

Workspaces represent the space used for construction activities. These spaces are reserved for the crew and their equipment, moving paths, material storage areas, etc. (Akinci et al. 2002a). Workspace analysis aims to create different types of workspaces for crew, equipment, and other required spaces in the work site, to detect conflicts between these workspaces, and then to resolve these conflicts.

Workspace conflicts are one of the important problems that can delay construction activities, reduce productivity, or cause accidents that threaten the safety of workers (Guo 2002). Mallasi and Dawood (2004) have discussed the idea that workspace interference can result in decreasing work productivity by about 40%. Early research in this area focused on work patterns, which are important for space planning. In order to better understand workspaces, Riley and Sanvido (1995) defined repeatable patterns that describe how typical activities use space over time. They also presented a planning method, which is represented by a process model defining a logical order and priorities for decisions about construction space plans (Riley and Sanvido 1997). The development of 3D visualization techniques in the construction industry has greatly improved the efficiency of space planning. However, 3D models lack the temporal information needed for scheduling. Furthermore, scheduling software lacks the 3D geometric information to represent workspaces. 4D models include both the temporal and the 3D geometric information necessary to display the time and location of activities. Kamat (2003) have proposed detecting conflicts between any pair of mobile or static objects on a construction site based on collision detection methods implemented within visualization tools of discrete event simulators. Akinci et al. (2002b) worked on the automation of time-space conflict analysis using 4D models. They have developed a prototype system – 4D WorkPlanner Time-Space Conflict Analyzer (4D TSConAn) using 4D models and classified spaces into three categories: macrolevel spaces, microlevel spaces, and paths. Heesom et al. (2003) have developed a dynamic virtual reality system for visualizing construction space usage focusing on the workspaces required within the proximity of the components being installed, such as workspaces for crews, equipment, and hazardous and protected areas. However, in previous research, simplified shapes (rectangular prisms) were used to represent different workspaces (Figure 2.8) (Akinci et al. 2002b). This simplification has limitations in representing complex equipment workspaces, such as cranes.

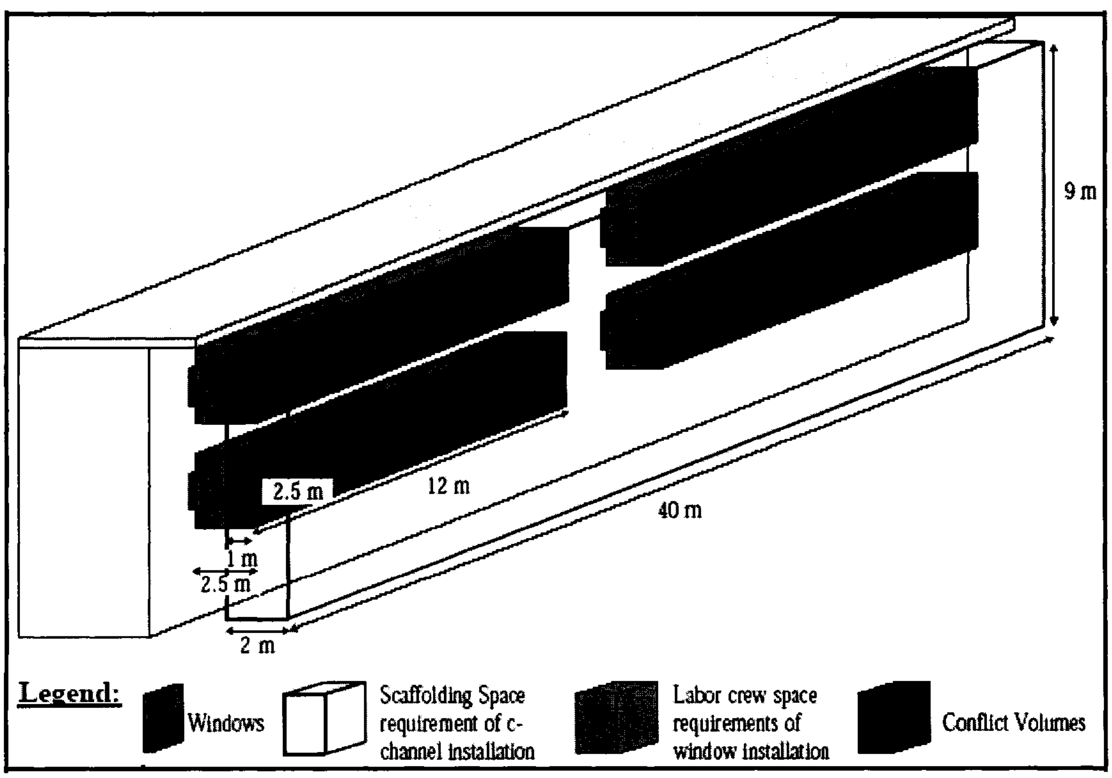


Figure 2.8 Representing workspaces using box shapes (Akinci et al. 2002b)

(2) Site layout analysis

The objective of site layout is to determine what kind of temporary works are required and to position them in space and time throughout the project to improve the construction process. A number of studies were conducted in order to improve site layout planning in construction projects. These studies adopted a wide range of methodologies and development tools including neural networks, simulation, knowledge-based systems, and genetic algorithms. Many factors were considered to find the optimal site layout, such as safety and cost (Elbeltagi et al. 2004; El-Rayes and Khalafallah 2005), temporary facilities locations (Mawdesley et al. 2002), and transportation cost (Zouein et al. 2002). In addition, dynamic site layout planning considering time is investigated in some research (Elbeltagi et al. 2004).

Several expert systems and knowledge-based systems were developed to integrate the knowledge of experts and assist in site layout planning tasks (Kumara et al. 1988; Hamiani 1989; Tommelein et al. 1991; Tommelein and Zouein 1993). Marasini and Dawood (2002) used simulation to develop a model that assists managers in designing and managing the layout of stockyards. Genetic algorithms were also used in several studies to optimize the layout of construction sites (Tam 2001; Li and Love 1998; Hegazy and Elbeltagi 1999).

Due to the vast number of trades and interrelated planning constraints in site layout planning, some particular construction methods were studied to unfold its complexity. For example, Tam et al. (2001) developed a site layout genetic algorithm model for optimizing supply locations around a tower crane. Also, new methods were investigated to help the manager in site layout planning. An automated site layout system for construction materials was integrated with a GIS-based cost estimation system (Cheng and Yang 2001). In recent years, space scheduling has received more attention because schedule compression leads to increased spatial interference among resources on-site (Zouein and Tommelein 2001). As shown in Figure 2.9, temporal project data are represented by a schedule, and associated resources are represented by the patterned rectangles inside the activity boxes. These data are used in constructing site layout over time. Other research focused on space needs of activities inside buildings, e.g., space used to store, fabricate, and assemble materials or building components just prior to or while performing work (Zouein and Tommelein 1994; Thabet and Beliveau 1994).

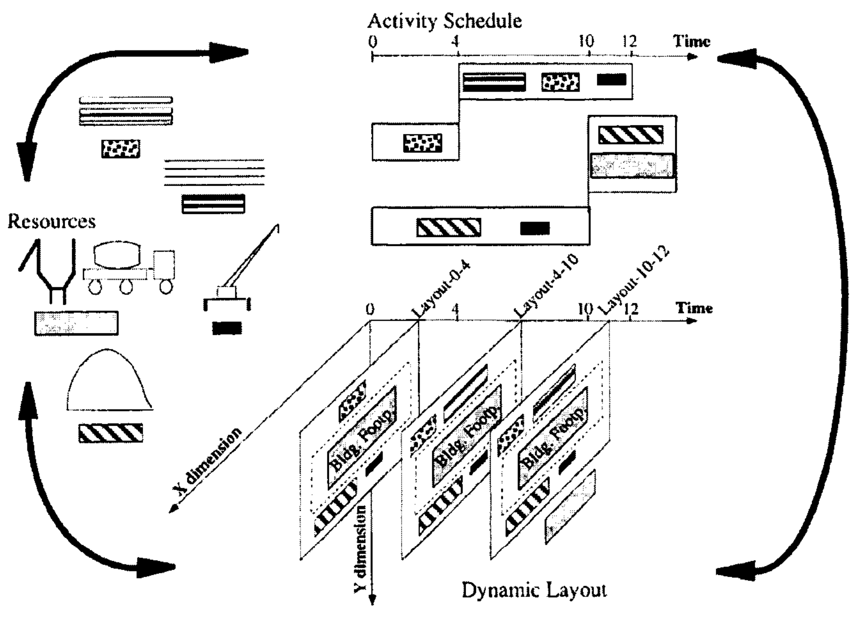


Figure 2.9 Space scheduling (Zouein and Tommelein 2001)

However, the space dynamics in staging and lay-down areas used for temporary storage of material and equipment on-site is quite different. For example, the long-term use of lay-down space by resources awaiting relocation or use by production activities can tolerate discretization of time. Changes in space availability and needs must be checked only at selected points in time. Also, distances traveled by resources between staging or work areas on a construction site are bigger than distances traveled by resources inside buildings.

(3) Cell-based spatio-temporal analysis

As introduced in Section 2.2.3, Cell-DEVS was proposed to describe cell spaces as DEVS models with timing delays, improving the definition of the models by using explicit delays. Research about outlining city sections as cell spaces has been conducted to analyze urban traffic (Tartaro et al. 2001). The cell-based model allows elaborated study of traffic flow according to the shape of a city section and its transit attributes. A static view of the city section can be easily described, including definitions for traffic signs, traffic lights, etc. The modeler can concentrate on the problem to solve instead of being occupied with defining a complex simulation. In addition, the movements of vehicles are defined by applying several rules. For example, a truck trying to change the lane must check if there is enough space to do that and should check that no vehicles are trying to occupy the same cells. These rules avoid space conflicts when vehicles are moving and make the simulation more realistic and accurate.

Cell-based analysis has been also applied in construction site analysis although it is not clearly defined. Elbeltagi et al. (2004) represented each facility as a number of small grid units that can take irregular shapes (Figure 2.10). A similar representation of construction site was applied to decompose and aggregate work zones based on triangular meshes that can represent activity workflow, project spatial hierarchy, and activity state information on a component at any given time (Akbas and Fischer 2002). However, cell-based spatio-temporal analysis has not been studied fully and more detailed analysis should be applied based on new technologies in computer science.

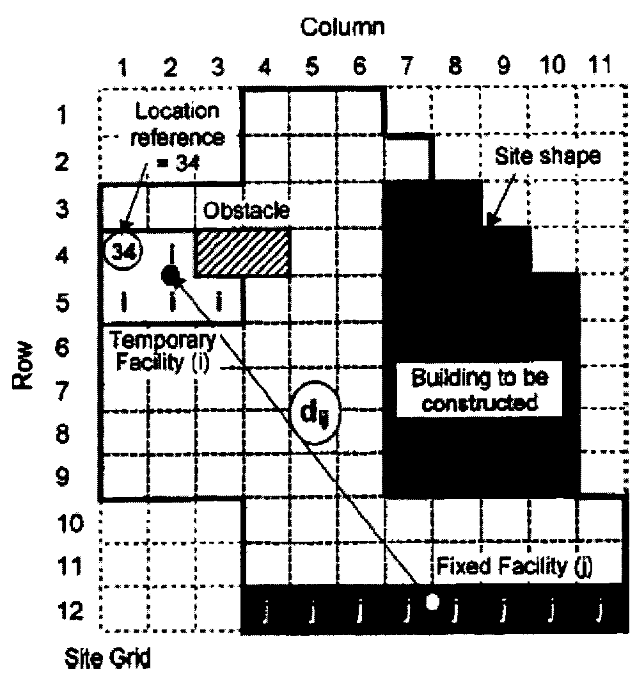


Figure 2.10 Site and facilities representation (Elbeltagi et al. 2004)

2.4 STANDARDIZATION AND INTEGRATION

The interoperability of the BMSs is of paramount importance because of the need to develop and use them by a large number of groups in a spatially and temporally distributed fashion. Standardization is important for facilitating data sharing and exchange between all the groups involved in bridge management at all the stages of the lifecycle of the bridge. There are different standards widely used in the AEC/FM industry. The International Organization for Standardization (ISO) is a global network that identifies what international standards are required by business, government and society, develops them in partnership with the sectors that will put them to use, adopts them by transparent procedures based on national input, and delivers them to be implemented worldwide (ISO 2005). International standards, such as Industry Foundation Classes (IFC) (IAI 2004) and STandard for the Exchange of Product model data (STEP) (STEP 2003), consider both the product and process models to export/import data between applications. In addition, eXtensible Markup Language (XML) is an emerging standard for modeling the information that is widely used in Internet-based Business-to-Business (B2B) applications (XML 2005). These standards can be used to retrieve metadata of the framework of IMSs of different types of infrastructure, e.g., BMSs and FMISs (Sunkpho and Garrett 2003).

(1) STEP is a comprehensive ISO standard (ISO 10303) that describes how to represent and exchange digital product information. Many STEP Application Protocols (AP) were built on the same set of Integrate Resources (IRs), so they all use the same definitions for the same information. For example, Part 225 is the AP of Structural

Building Elements Using Explicit Shape Representation, which is specific for the exchange of building element shape, property, and spatial configuration information between AEC software systems using explicit three-dimensional shape representations (ISO 10303-225 1999). Figure 2.11 shows an example of Structure_enclosure_elements, which is a type of Building_element that is an identifiable part of a building that contributes to the basic form or function of the building.

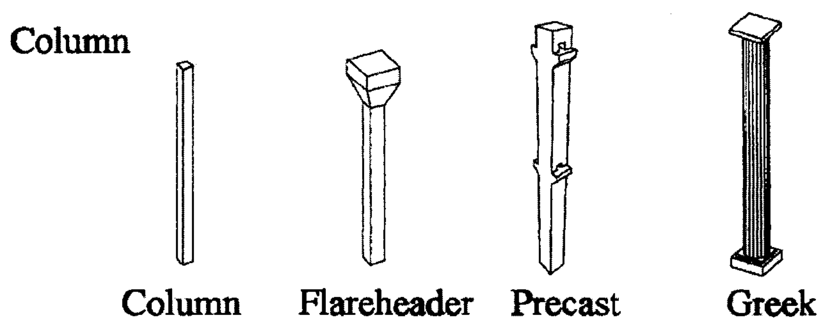


Figure 2.11 Structure_enclosure_elements – column (ISO 10303-225 1999)

(2) IFC is an open international standard managed by the International Alliance of Interoperability (IAI). IAI is a global, industry-based consortium for the AEC/FM industry, and it builds on earlier work from ISO STEP. It uses STEP technology such as the EXPRESS language and STEP Part 21 file exchange formats and follows from building construction work done within STEP. IFC is a common language used to assemble sets of computer models that capture common project information. The global scope of IFC includes: project lifecycle, geography, work disciplines, and technical applications. The current version is IFC 2X2. The ISO announced acceptance of IFC as a common language in the construction industry in 2002. The

IFC 2X Platform Specification is now ISO/PAS 16739. In IFC2x2 the concept of visual presentation of geometric items has been added to the IFC model. Any object in IFC that has a geometric representation has two attributes: *ObjectPlacement* and *Representation*. The representation capabilities have two purposes: to add the explicit style information for the shape representation of products, and to add additional annotations to the product shape representations. There are two resources related to time in the Resource layer of IFC: IfcDateTimeResource and IfcTimeSeriesResource. In IfcDateTimeResource, calendar date and local time are defined and functions about validity are also created. IfcTimeSeriesResource is new in IFC2x2. It defines two types of time points and related values: regular time and irregular time. In regular time series, data are updated predictably at predefined intervals. In irregular time series some or all time stamps do not follow a repetitive pattern and unpredictable bursts of data may arrive at unspecified points in time. A typical usage of these entities is to handle data collected from sensors in a bridge health monitoring system.

Many projects and research related with AEC/FM are applied based on IFC. The IFC-Bridge project aims to extend ISO/PAS 16739 by defining a standard representation for bridge life cycle management (IFC-BRIDGE 2004). Examples of new entities defined in IFC-Bridge are IfcBridge, IfcBridgeElement, IfcBridgeStructureElement, IfcBridgePrismaticElement, IfcBridgeSegment, IfcBridgeBondingElementType, and so on (Figure 2.12).

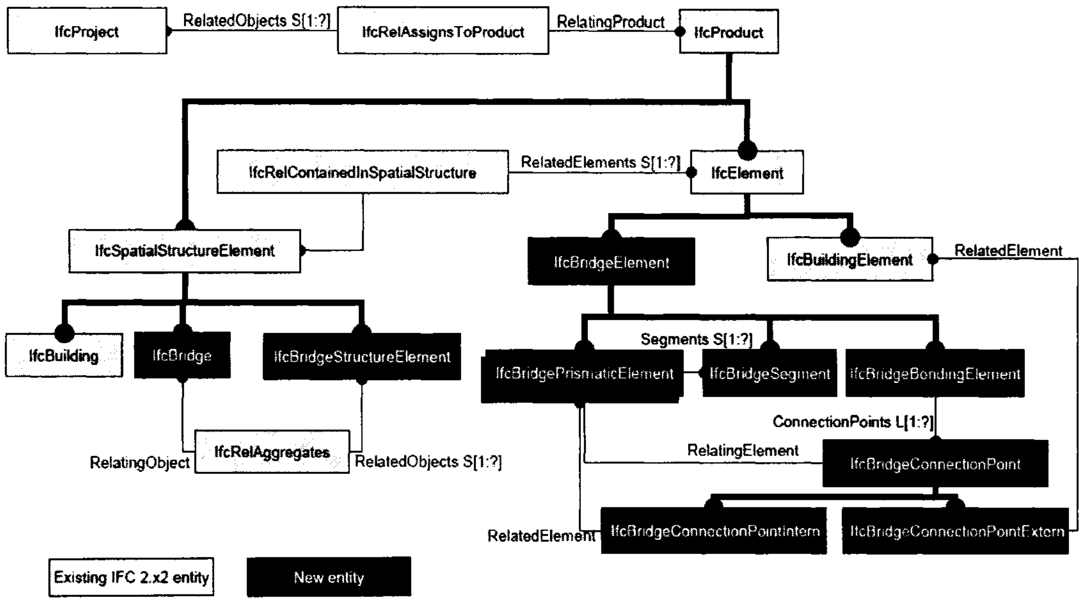


Figure 2.12 IFC-Bridge general structure (IFC-BRIDGE 2004)

As IFC-Bridge is still in the early stage of development, many details are missing. For example, the truss type is not included in the definition of IfcBridgeStructureType. Several extensions of IFC are necessary to cover the later stages of the lifecycle of structures.

One of the most popular software for planning and scheduling -- MS Project investigated and mapped the relationship with IFC (Seren and Karstila 2001). They described the specification of how typical project management information maps to target entities from different versions of the IFC-model (IFC R1.5.1, R2.0 and R2x). The MS Project internal application model is used as a typical source definition. The main focus is on how task scheduling information and associations to building elements are managed. One of the mapping specifications (IFC R2.0) is used as a basis for the 4D-Linker application.

In addition, a new project named IFG is undergoing with the purpose of providing GIS information through IFC. The development of the project will focus on making the zoning plan and building plan submission process more efficient. The GIS information in a central building and property registry will be integrated with AEC/FM information about the individual buildings that are registered. The project will make it possible to communicate relevant intelligent information from various GIS standards to CAD systems using IFC (IFG 2005).

Moreover, in recognition of the impact of XML in other IT domains, the fifth and sixth releases of the IFC have also included XML schema definition language representations, which are known as ifcXML1 and ifcXML2, respectively. The goal of this work is to make it possible to exchange IFC data files alternatively as XML documents, and to enable the reuse of IFC content and structure within XML-based initiatives for data exchange and sharing in the AEC/FM industries.

2.5 SUMMARY AND CONCLUSIONS

In this chapter, literature about the current situation and future trend of BMSs was reviewed to find the possibility for information technology integration. As the main focus of this research, spatio-temporal issues were reviewed in two aspects: visualization and analysis, to find potential research target and define our objectives.

CHAPTER 3 INTEGRATION AND VISUALIZATION OF SPATIO-TEMPORAL INFORMATION

3.1 INTRODUCTION

As introduced in Chapter 2, future BMSs and other IMSs are going to gain significant benefit more from computer graphics techniques, Location-Based Computing (LBC), and other new information technologies. New approaches for spatio-temporal visualization and analysis are needed for designing future BMSs. The information about the lifecycle of a bridge can be integrated with the 3D model of the bridge, resulting in 4D models which can display the changes on the 3D model at a specific time or during a specific period of the lifecycle. Based on the spatio-temporal information, analysis can be applied to help decision-making, such as selecting appropriate equipment according to the spatial constraints, or finding the conflicts between workspaces during construction.

In this chapter, a framework is built to integrate and visualize spatio-temporal information of a bridge, and several computational aspects of the framework are discussed. This framework is used to develop a prototype BMS, which will be introduced in Chapter 5. Furthermore, workspace analysis methods, which will be introduced in Chapter 4, are also based on the proposed framework.

3.2 FRAMEWORK FOR INTEGRATION AND VISUZLIZATION OF SPATIO-TEMPORAL INFORAMTION

The general structure of the framework is shown in Figure 3.1 (Hammad et al. 2005). This framework is based on developing an object-relational data model, integrating a number of technologies and then using the data model and the integrated technologies to develop applications, such as visualization, analysis, and decision-making support.

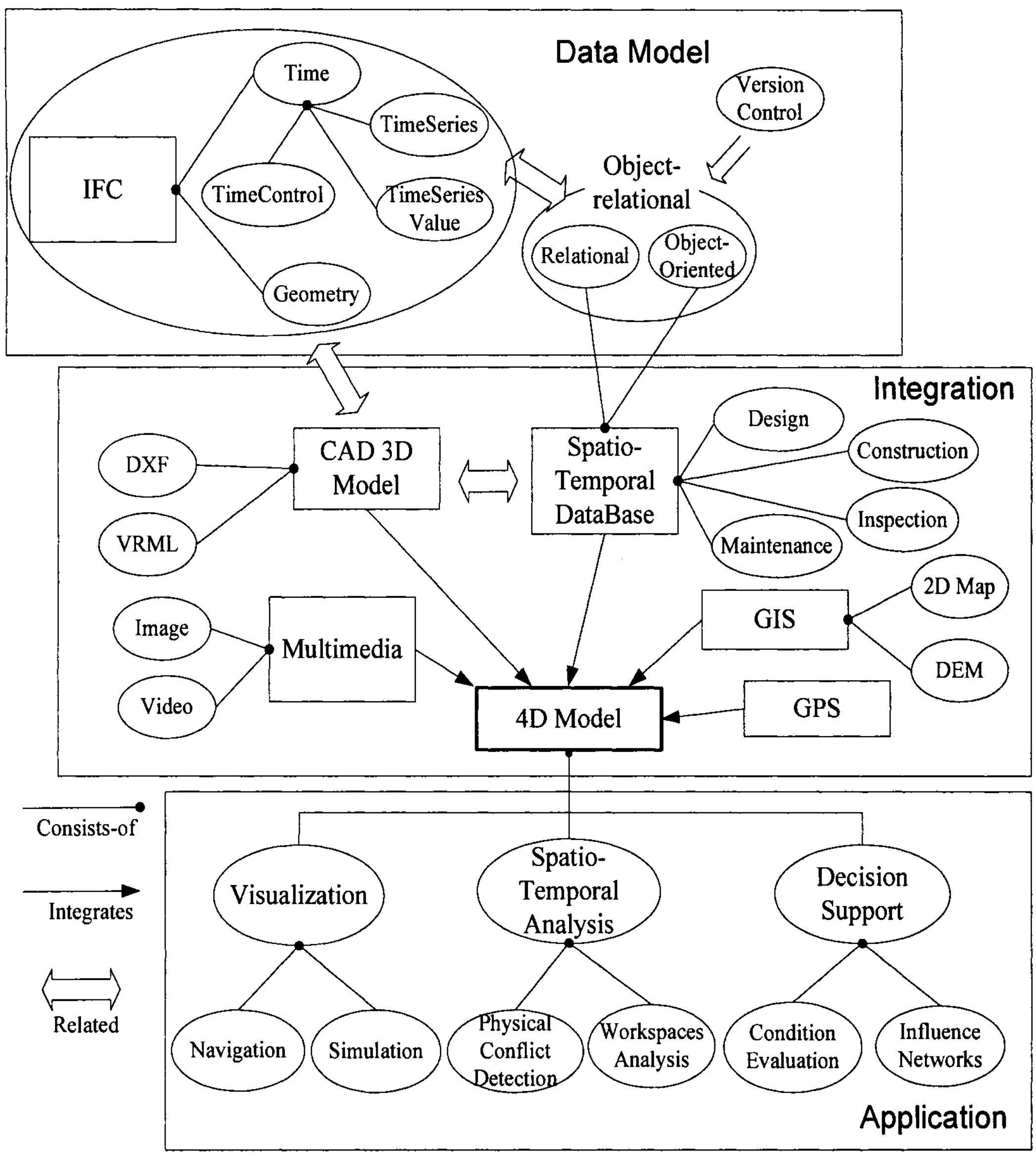


Figure 3.1 General structure of the framework (Hammad et al. 2005)

The core of the framework is a 4D model that integrates a spatio-temporal database covering the different phases of the lifecycle, and CAD 3D models of the bridges. Further integration is necessary with GIS, tracking technologies and multimedia information. A 3D map of the area covered by the BMS is needed in the framework to permit the computations based on the location of the users. Using this map, the models of bridges can be based on geographic global coordinates. In order to create a 3D map, 2D layers can be draped on the Digital Elevation Model (DEM) of the same area. In addition, the location of the user can be tracked using GPS and/or other tracking methods, and this location is used to navigate the user (e.g., to find the location of the next element to inspect) or to extract some information from the database (e.g., information about the inspection history of an element at certain location) using the concepts of LBC (Beadle et al. 1997). The location of the user is reflected on the 2D map and the 4D browser. Multimedia information, including images and videos, can be captured and automatically added to the database using the concept of the Digital Hardhat (Stumpf et al. 1998).

Visualization has powerful functions for interacting with the system in a virtual reality or augmented reality modes (Hammad et al., 2005). Users can query the database through the Graphical User Interface (GUI) or by picking a specific element, and get the results as visual feedback in the 4D model, e.g., information about the painting or rehabilitation history. Users can easily navigate in the 3D space using navigation tools (will be explained in detail in Section 3.5.2). Other important applications are spatio-temporal analysis applications, such as workspace analysis. The different workspaces for each activity can be generated and conflicts between these workspaces can be detected.

3.3 4D MODELING IN INFRASTRUCTURE LIFECYCLE MANAGEMENT

As discussed in Chapter 2, one of the limitations of the current 4D modeling tools is using a single LoD, which means the same model of the structure will be displayed regardless of the purpose of using the model, such as retrieving general information of the site layout, or applying detailed inspection for certain elements. Therefore, our modeling method focuses on developing several LoDs of the structure model to meet different management requirements.

Another significance of our modeling method is that all the lifecycle data are integrated with the model instead of limiting the scope to the construction stage. A BMS requires as much information as possible to provide an efficient management tool during all the stages of the lifecycle of bridges. The 3D model of a bridge can be used to record the inspection results in a mobile working situation. Defects detected on certain elements can be shown directly on the 3D model using different shapes and colors representing different types and levels of defects. In addition, historical defects recorded on specific elements can be shown on the model to help the decision maker evaluate the condition of the structure and take appropriate measures to maintain the functionality of the structure.

(1) Object-relational data model

The data model in the framework is an object-relational data model, which combines the relational data model with object-oriented development tools. Data are stored in a hierarchy from most detailed elements, such as a deck panel, to the main bridge structures. Each object table is related to the sub- or super-tables. Apart from the bridge structures, activities occurring during the lifecycle are linked with related objects' tables to add details about time, type of the activity, etc. The time entities in the database are defined based on the time resources definitions of IFC. Figure 3.2 shows the entity relationship diagram of the bridge database.

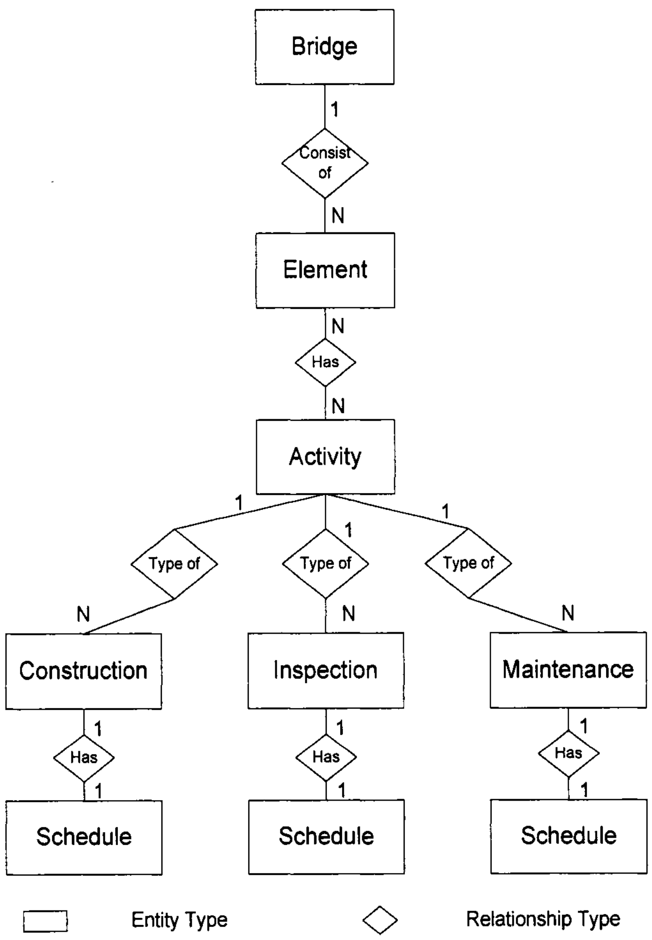
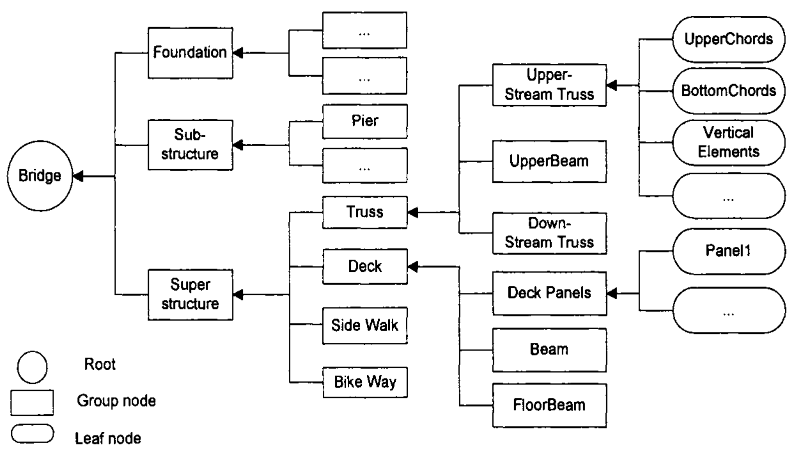


Figure 3.2 Database entity-relationship diagram

The data stored in the database about the structure of the bridge are read automatically and a logical tree is created based on the structure. Figures 3.3 (a) and (b) show an example of an object tree representing a bridge structure and its table representation, respectively. The relationship between each *group node* and the *element nodes* branching from it is a *part-of* relationship. Through querying the database, the root of the tree is found and the root node is created. Then, queries are applied recursively to find other element nodes based on the data stored in the table.



(a)

Group nodes	Element nodes
root	Bridge
Bridge	Foundation
Bridge	Sub-structure
Bridge	Superstructure
Superstructure	Truss
Superstructure	Deck
Superstructure	Side Walk
Superstructure	Bike Way
Truss	Down-Stream Truss
Truss	Upper-Stream Truss
...	...

(b)

Figure 3.3 Object tree: (a) Example of the object tree; and (b) Its table representation

(2) Creating 3D objects of bridge components

To construct the 3D model of a bridge that can be used for facilitating interaction and spatio-temporal analysis of each component, the bridge should be decomposed into several sections as individual elements according to the structural hierarchy. An example of the hierarchy of the sections of a truss bridge is shown in Figure 3.3 (a). The information of each element of the structure (e.g. dimensions, material, and cross section type) is stored in a database. Each 3D element is created according to its dimensions, and the spatial relations between elements are extracted based on the local coordinates of the start and end points of each element. The two points are used to calculate the rotation angle of the element, based on which certain transformation (i.e., translation and rotation) is applied to construct the shape of the structure. Different appearance attributes, such as color, material, and texture, can be defined for every element to meet the visual requirements. In addition, each element is given an ID number to identify which section it belongs to, for example, *TUS011_UC001* means that this element belongs to truss section 11, and it is an upper chord with the number 001. This unique ID is used to link the 3D model with the database for information querying. The model created for the bridge is used as one of the levels of details (prismatic elements) that will be discussed in the next section.

(3) Levels of Details (LoDs)

The basic idea of LoDs is to use simpler versions of an object to meet different precision needs and improve the image rendering performance. When the viewer is far from the object, a simplified model can be used to speed up the rendering. Due to the distance, the simplified version looks approximately the same as the more detailed version. As introduced in Section 2.2.4, LoDs algorithms consist of three major parts: *generation*, *selection*, and *switching*. *Generation* is generating different representations of a model with different detail. *Selection* is choosing a LoDs model based on certain ranges for the distance. *Switching* is changing from one representation to another. When the user moves, this event is detected and the distance between the user and the object is calculated. Based on this distance, the corresponding switch will be selected and the model that should be displayed in this range is rendered. Figure 3.4 shows the Unified Modeling Language (UML) interaction diagram of the LoDs behavior.

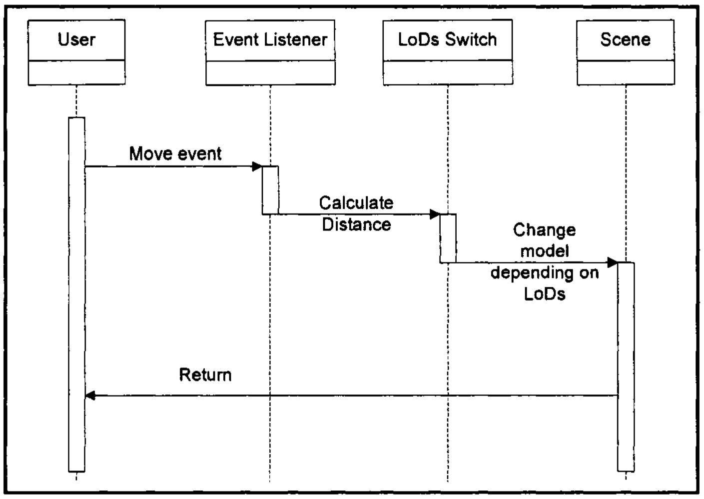
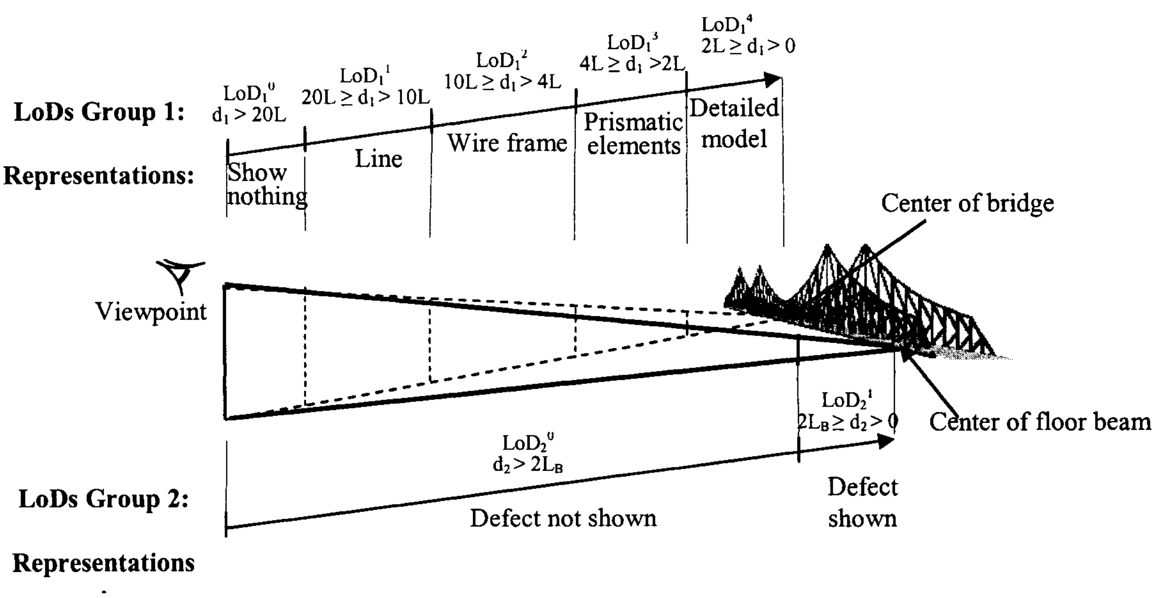


Figure 3.4 UML interaction diagram of the LoDs behavior

Four LoDs are selected to build the bridge model. The first level is representing the bridge as a 2D line, which is the axis of the bridge. This simple representation is used to overview the bridge location on the map and basic information such as traffic volume. The second level is the wire frame representation, which shows the outline of the shape of the bridge. The third level is the prismatic elements representation, which means every element is represented as prism shape and some basic analysis can be applied on this model, such as changing the color of elements to show different construction states. The most detailed level is created using Virtual Reality Modeling Language (VRML), which can show details of every components based on CAD data. The VRML model meets the requirements of higher accuracy, for example, recording and marking defects on the 3D model of the bridge.



d_1: Distance between center of bridge and viewpoint
d_2: Distance between center of beam and viewpoint
L : Length of bridge L_B : Length of beam

Figure 3.5 Relationship between distance and LoDs for the bridge and the defects on a floor beam

In addition, LoDs can be used in parallel with respect to different objects in the same system, such as the bridge elements and the defects on the elements. Each LoDs group uses different referential center point and distance range and operates only on objects related to that group. As shown in Figures 3.5 and 3.6, two LoDs groups can be used in parallel. LoDs *Group-1* is for the whole bridge model, which includes five different cases: nothing shown, line, wire frame, prismatic elements, and detailed VRML objects.

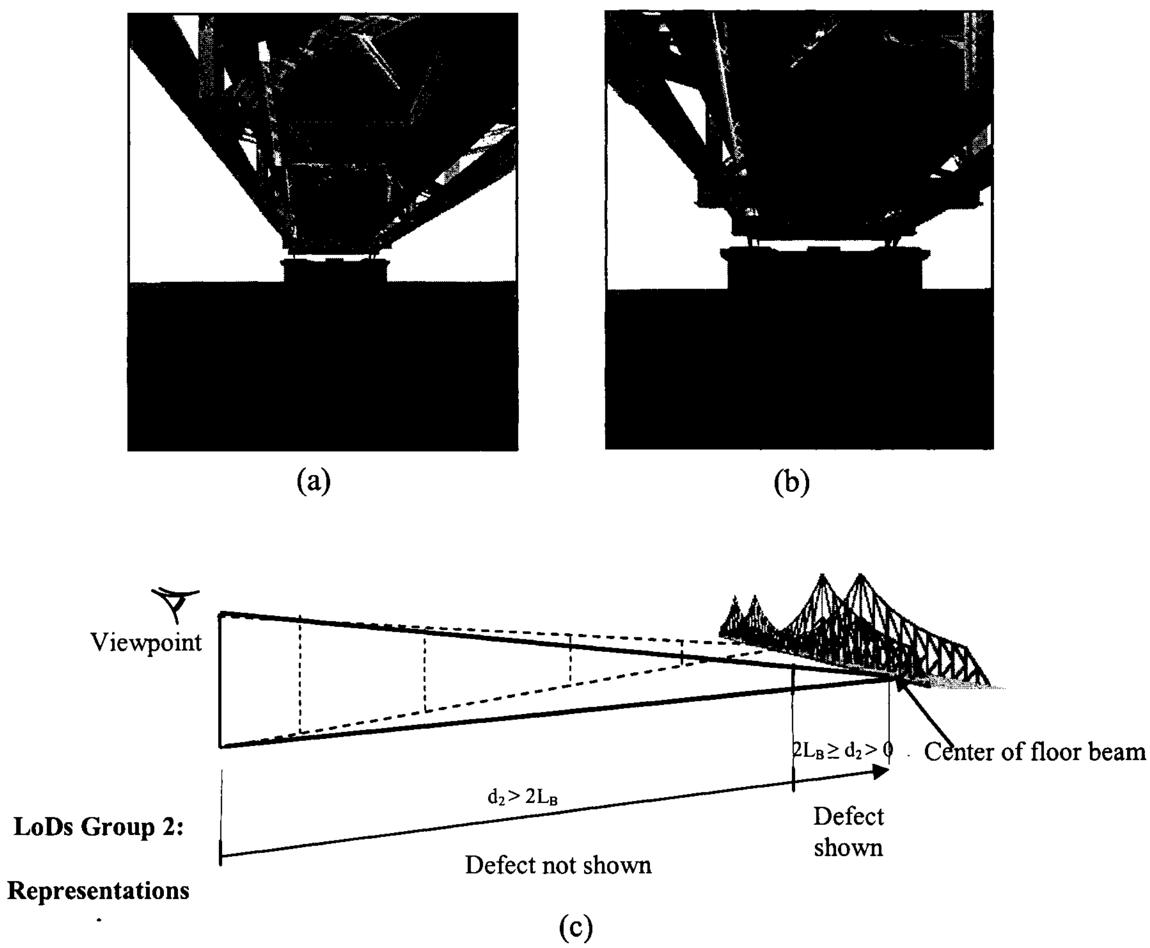


Figure 3.6 LoDs for the defect on the beam:
(a) Defects not shown on the beam; (b) Defects shown on the beam; and (c) Relationship between distance and LoDs

The distance d_1 is measured between the viewpoint and the center of the bridge. The distance range is defined in general depending on the bridge length (L). In this example, the visible range is from 0 to 20 times of the bridge length. If d_1 is between $20L$ and $10L$, only one line representing the bridge axis is shown on the map. If d_1 is between $10L$ and $4L$, the wire frame model is rendered. If d_1 is between $4L$ and $2L$, the prismatic elements model is rendered. If d_1 is between $2L$ and 0, the VRML model is rendered (Figure 3.5). LoDs *Group-2* is for the defects on a floor beam, which includes two cases: show or not show the defects. The distance d_2 is measured between the viewpoint and the center of the beam (Figure 3.6).

3.4 LOCATING THE STRUCTURE MODEL ON A 3D MAP

The purpose of adding the 3D map of the area is to provide information to the users of the system (e.g., bridge inspectors) about their positions and the environment around them. The map is created using GIS data, including several layers, such as a boundary layer and layers for the roads, rivers, and administrative areas. The GIS component has the main functions for zooming and retrieving information about the attributes of different layers. To display the terrain of the area, a 3D map is created based on the DEM. Different colors can be used to represent the range of elevation and to give realistic display. The geographic coordinates of the structure location can be found from the map, which can be used to calculate the rotation and displacement that should be applied on the structure model to locate it on the 3D map. In addition, a 2D map interface can be also added to the user interface to help the user know his location.

3.5 INTERACTION WITH THE 4D MODEL

3.5.1 User interface design

In order to show the 4D model on the small screen of a tablet PC used in a mobile situation, a simple but efficient user interface should be designed focusing more on navigation and interaction functions. The main area of the interface should be used to show the 4D browser. Navigation is facilitated by adding a structure tree of the bridge to guide the user to find certain elements in the 3D model. In addition, different viewpoints are created to navigate the user to some specific locations. Time is shown using a calendar interface and time sliding bars to meet different LoDs requirements. The start and end dates of a period can be input using the calendar to show the changes during a specific period, while time sliders (i.e., year, month, and day) can be used to show the changes dynamically, for example, by moving each slider, the sequence of the construction activities can be shown.

The user can query the database about activities that took place during a specific period. A clear color coding is needed to explain the different states of the elements having that specific color. For example, different colors represent different construction progress stages (e.g., has been built, under construction, or has not been built) of the corresponding elements of the bridge. The design of the interface is shown in Figure 3.7.

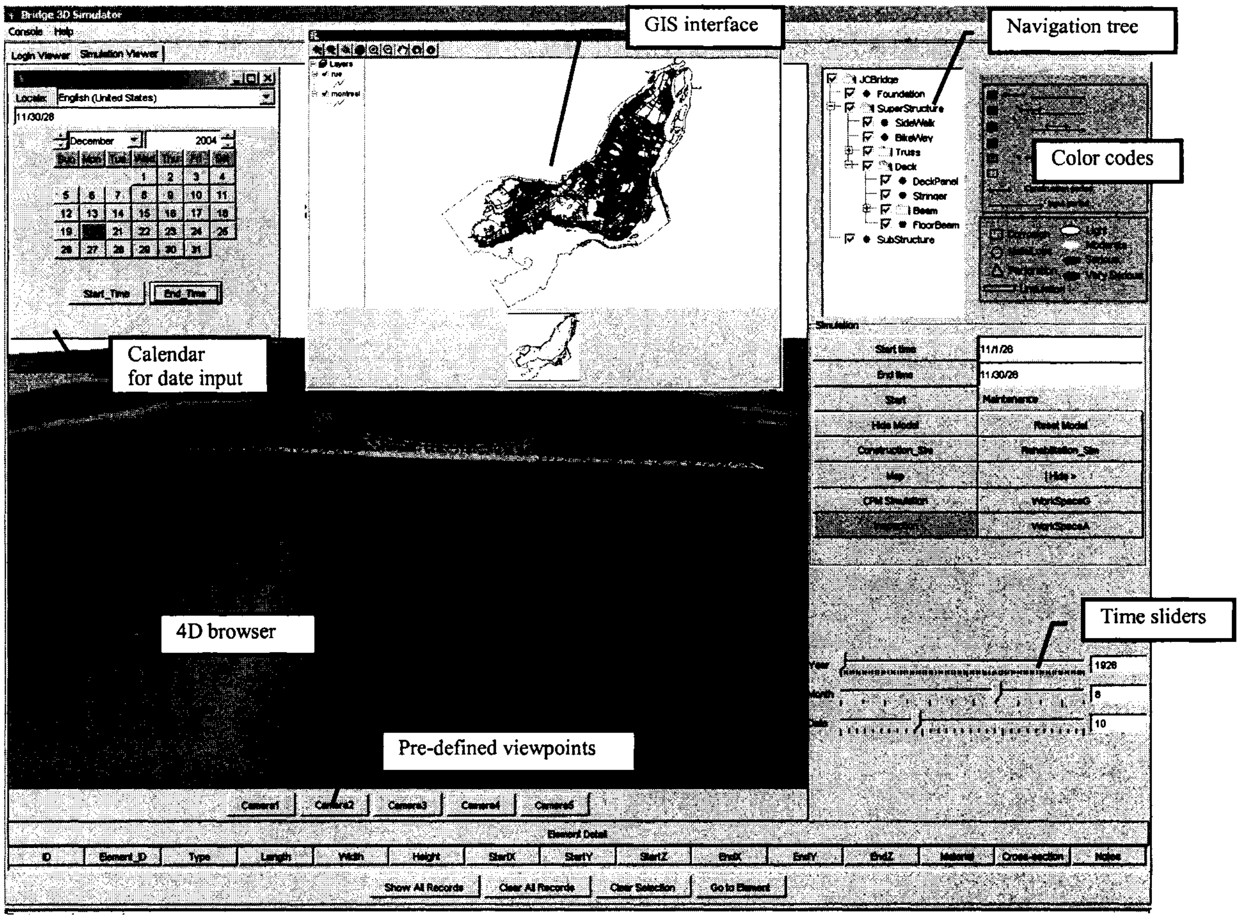


Figure 3.7 Design of the user interface of the prototype system

3.5.2 Viewpoints and navigation tree

To help the user move within the 3D environment without losing his/her way, viewpoints and a navigation tree are created. Specific viewpoints, such as the side view and top view of the bridge, are created to guide the user to these specific locations. Two points should be given as start point and end point to specify the location and direction of the viewpoint. The structure tree discussed in Section 3.3 is used as a navigation tree to identify the spatial location of each component of the bridge. The data stored in the database about the structure of the bridge can be used to automatically generate the logical tree. Through querying the database, the root of the tree is found and the root node is created. Then, queries are applied recursively to find other element nodes based on the data stored in the table. The flowchart of creating the tree is shown in Appendix A. The hierarchical tree of the bridge structure is shown on the right side of the main interface (Figure 3.7). Each tree node has a check box, which facilitates showing or not showing that element in the 3D model.

3.5.3 4D visualization of the lifecycle data

As introduced in the interface design, a calendar and sliding bars are used to specify a date or a period of time and the time step, representing the temporal LoDs, to be used in a simulation. Different temporal LoDs are needed during construction and maintenance periods. The year or the specific date of the maintenance action can represent the time of maintenance. For example, the painting of the main span is done in several years. The inspection time is usually represented by the date of inspection. Using queries, the user can directly interact with the 4D model to get information on a certain stage of the lifecycle (e.g., Which parts of the bridge were constructed by the end of 1928? What is the sequence of replacing the deck panels in 2001?).

Maintenance activities can be visualized on the 4D model, such as the painting history of the truss. In Figure 3.8 (a), the painting activity in 2000 is selected to show which parts of the truss were painted during that year. Furthermore, construction activities are simulated to demonstrate the dynamic changes on the 4D model, such as displaying elements sequentially with different colors according to construction or rehabilitation periods. Figure 3.8 (b) shows the visualization of the deck replacement sequence on the 4D model. In addition, inspection history data can be retrieved and shown on the 3D elements (Figure 3.8 (c)).

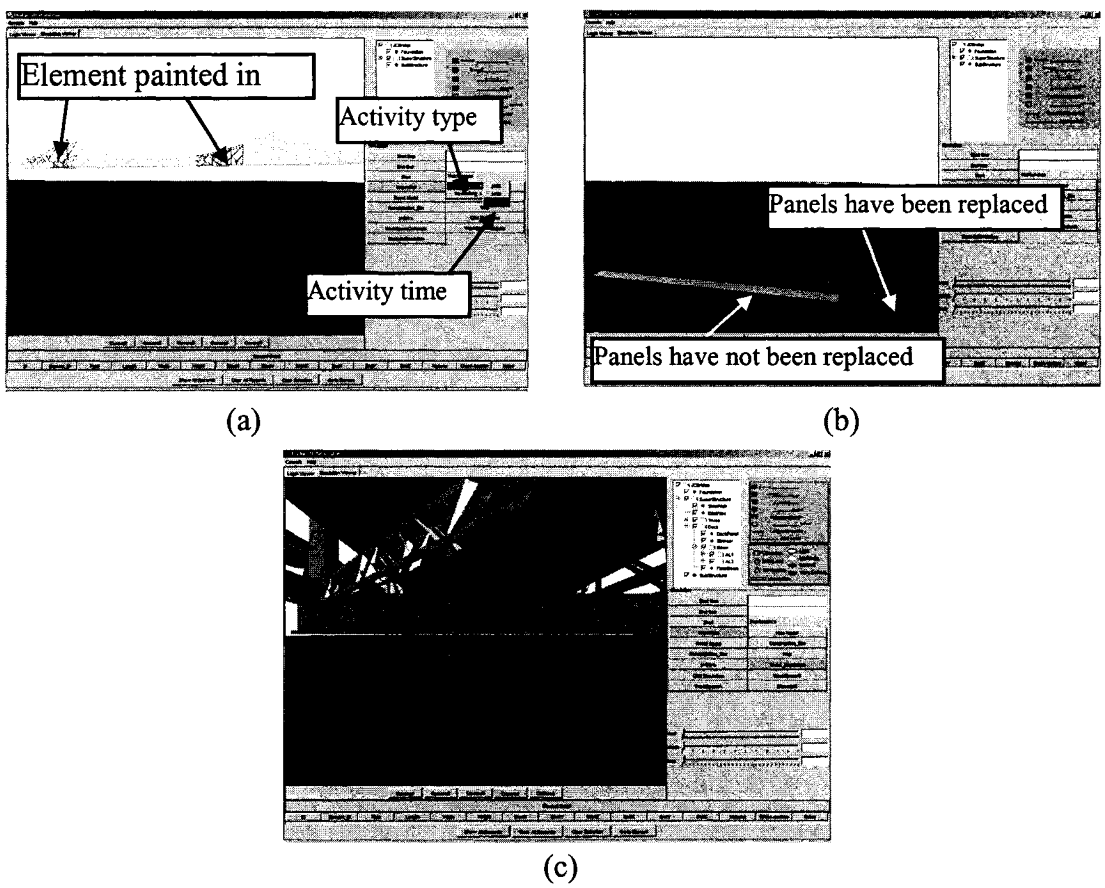


Figure 3.8 4D visualization of lifecycle data: (a) Painting history visualization; (b) Deck replacement visualization; and (c) Visualization of defects on the beam

3.5.4 Picking behavior for retrieving and updating information

Interaction with the model is mainly facilitated by picking the elements of the model. Picking is the process of selecting shapes in the 3D virtual world using the 2D coordinates of the picking device. A pick shape is selected as the picking tool. The pick shape could be a ray, segment, cone, or cylinder. The pick shape extends from the viewpoint location, through the picking device location and into the virtual world. When a pick is requested, pickable shapes that intersect with the pick shape (e.g., pick ray) are computed. The pick returns a list of objects, from which the nearest object can be computed. Each element in the 3D environment has a predefined ID, which is related with the data stored in the database. Upon selection, the element will be highlighted and a query is activated to retrieve the matching information in the database. The information about the element can be displayed on the model as discussed in Section 3.5.1. Alternatively, the user can select an element from the database interface and the element will be highlighted in the model.

Another function that can be facilitated by picking is to add new data to the 3D model. In that case, it is important to know the location and the orientation of that element in the 3D environment of the virtual model. Figures 3.9 and 3.10 show the flowchart and an example of the picking behavior, respectively. After the closest object (*O*) is found, the surface (*F*) that faces the user can be identified to display suitable feedback. Through the calculation of the distance between the picking device position and the intersection points, the nearest intersection point (*P*) can be found as well as the geometry of the face

(F) that contains (P). The normal vector (N) of surface (F) can be calculated based on the current coordinates.

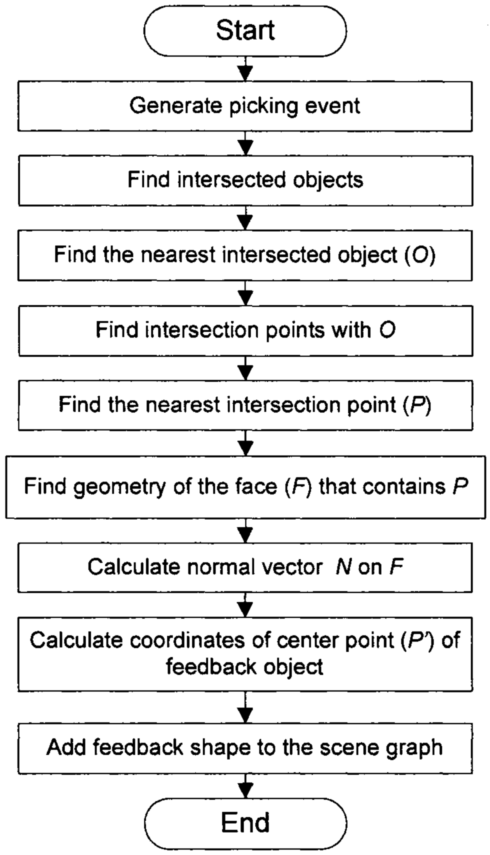
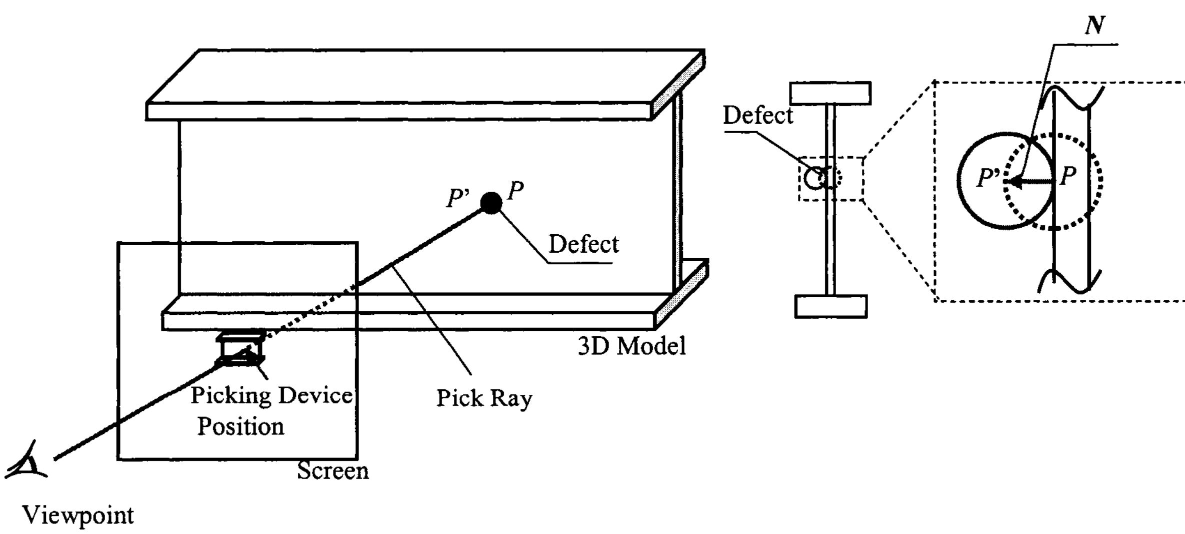


Figure 3.9 Flowchart of picking and adding defects



(a) (b)

Figure 3.10 Example of picking the 3D model for marking defects: (a) 3D sketch; and (b) side view

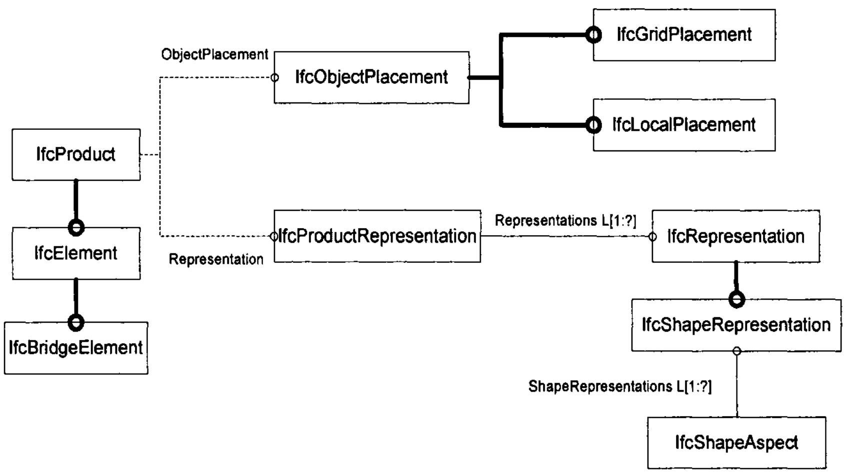
The normal vector is used to represent the orientation of that face. Based on *P* and *N*, the shape representing the feedback can be created and inserted in the scene graph at point *P'* with an offset distance from the surface *F* proportional to the size of the shape. The vector representing point *P'* can be found using the following equation:

$$\vec{P}' = \vec{P} + \text{offset} \times \vec{N} \qquad \text{Equation (3.1)}$$

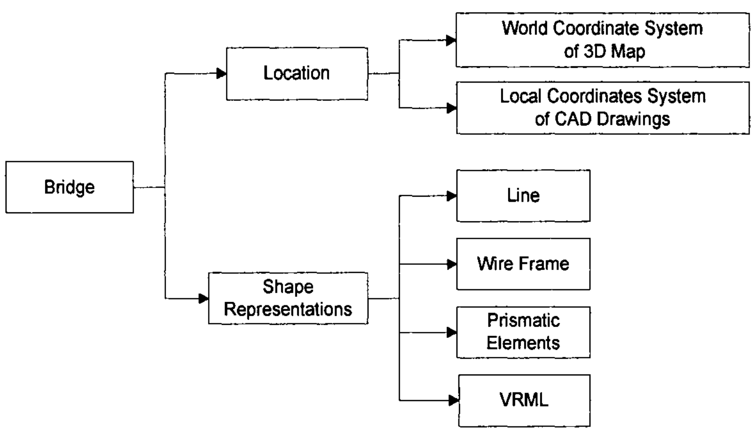
In the case of inspection, the user is allowed to directly add a defect, which is represented by a 3D shape, on the surface of the inspected element. The location of the defect is represented by the point (*P*) of the picking. However, to show this defect on the surface, the center point of the 3D shape of that defect should be moved in the direction of the normal vector on that surface (*N*) with a small offset distance based on to the size of the 3D shape as shown in Figure 3.10 (b). Otherwise, the defect on a thin element, e.g. the web of a steel beam, may appear on both surfaces of the web due to the small thickness of the web. The center point of the defect representation can be calculated using Equation (3.1).

3.6 IFC MAPPING

As introduced in Chapter 2, the IFC-Bridge is under development and it will take time to be applied in practice. In our research, we focused more on the spatio-temporal aspects of the project and the main application of IFC will be limited to mapping IFC concept with our model. The modeling of bridges and their performance involves data about the lifecycle. Such data cover a large spectrum, from schedules of all kinds of activities to time of unpredictable events. Properly representing this time-related information is essential for their proper understanding and use.



(a)



(b)

Figure 3.11 Mapping of IFC spatial representation:
(a) IFC definition; and (b) Representation in the model

(1) Spatial representation mapping

In IFC, any object that has a geometric representation has two attributes: *ObjectPlacement* and *Representation*. The placement can either be *absolute* (relative to the world coordinate system), *relative* (the object placement relative to another object), or *by grid reference*. A representation is one or more representation items that are related in a specified representation context as the representation of some concept. The shape representation is a specific kind of representation that represents a shape. For the same object placement, a product may be represented by a single or multiple shape representations (Figure 3.11 (a)).

In our 3D modeling, a geographic coordinate system is used in creating the maps and the bridge is located at its real location. Meanwhile, a local coordinate system is used to create the bridge structure to reduce computation time following the coordinates of the CAD drawings. Multiple shape representations are reflected in the LoDs. In the world coordinate system, four models with different LoDs are created to represent the bridge according to the distance range (Figure 3.11 (b)).

(2) Temporal representation mapping

There are two resources related to time in the resource layer of IFC: *IfcDateTimeResource* and *IfcTimeSeriesResource*. In *IfcDateTimeResource*, calendar date and local time are defined. *IfcTimeSeriesResource* defines two types of time points: regular time (*IfcRegularTimeSeries*) and irregular time (*IfcIrregularTimeSeries*). Corresponding attribute values (*IfcTimeSeriesValue*) are assigned to each time. Time control is facilitated by *IfcTimeSeriesSchedule*. It defines a time-series that is applicable to one or more calendar dates (Figure 3.12 (a)).

In our model, a calendar interface is created to input the time period, which can be selected based on different time zones. Information of the time and schedule about the activities is stored in the database. Construction activities are irregular, so the time schedule for construction is irregular time series. While inspection and maintenance activities can be regular or irregular, and the corresponding time schedule can be regular or irregular time series. The *Activity* table in the database stores time (*StartTime*, *FinishTime*, and *Duration*) and tasks.

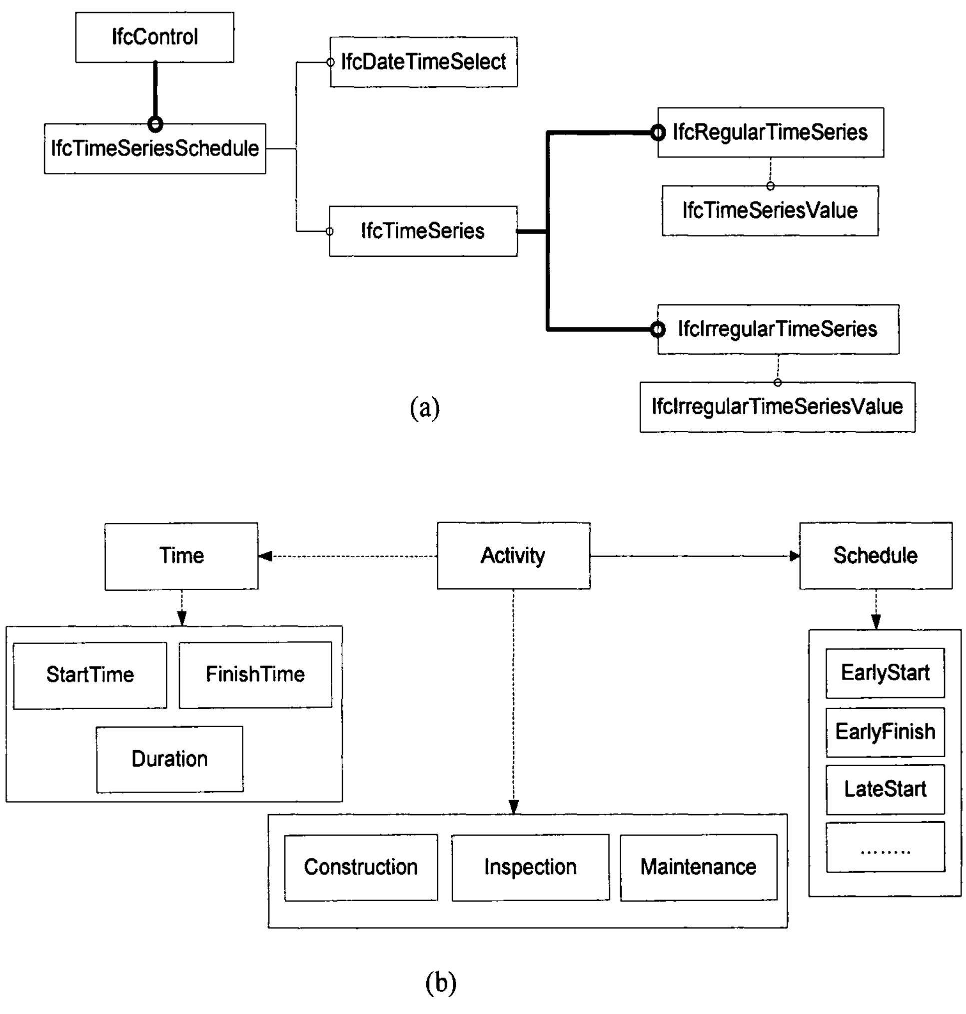


Figure 3.12 IFC mapping of temporal issue
(a) IFC definition; and (b) Representation in the model

Different activities require different time controls. Longer time is needed for construction and maintenance because each task may take several days and the sequence of tasks should be well planned and organized; while inspection may be finished in one day and the schedule is not so tight. Due to this difference, a *Schedule* table is created to provide details of each task, such as *EarlyStart* and *EarlyFinish* (Figure 3.12 (b)). Other IFC mappings are shown in Appendix B.

3.7 SUMMARY AND CONCLUSIONS

This chapter discussed spatio-temporal information visualization and integration in BMSs. Conventional BMSs provide only limited support for representing, visualizing and analyzing the spatio-temporal relationships throughout the lifecycle of the bridges. Therefore, this chapter proposed a method that integrates 4D modeling with several information technologies to facilitate space and time visualization and analysis. A 4D bridge model is integrated with a GIS to represent accurate spatial information and to make the results of the spatial analysis more reliable and understandable. The method links all the information about the lifecycle of a bridge incorporating different LoDs of space and time. A framework for BMSs was discussed including creating an object-relational data model, technology integration and applications development. Several computational issues for realizing the framework were investigated, such as navigation, picking behavior and LoDs. This approach makes the first attempt to integrate 4D bridge models with BMSs making the information accessible to mobile on-site workers.

CHAPTER 4 SPATIO-TEMPORAL CONFLICT ANALYSIS OF CONSTRUCTION ACTIVITIES

4.1 INTRODUCTION

After the integrated 4D model of a bridge is built and visualized as explained in Chapter 3, new methods for spatio-temporal analysis are investigated. As mentioned in Chapter 2, spatio-temporal analysis can help construction managers find spatio-temporal conflicts, and thus to avoid these conflicts or resolve them by taking suitable measures, such as changing an equipment that did not fit the spatial constraints, or changing the schedule of the tasks that caused these conflicts.

In this chapter, two spatio-temporal analysis approaches are proposed: one is workspace representation and conflict detection, the other is cell-based simulation for construction processes. Combinations of different 3D shapes are used to represent the workspaces for different activities, which is more accurate than the simple prismatic elements that were used in previous research to represent workspaces. Conflicts related to space and time are detected based on the defined workspaces during a specific period. The purpose of workspace analysis is to investigate the workspace representation and conflict detection in the case of bridge construction and rehabilitation projects where heavy equipment is required. Furthermore, cell-based simulation is used to represent space resources used in the whole work site, which enables conflict analysis and visualization of the worksite and the occupation of space. Simulation research that provides an explicit method to investigate possible space conflicts is still limited. Therefore, the new modeling approach is compared with MicroCYCLONE as a representative of conventional construction simulation tools to identify the advantages and limitations of each method in spatial resource representation. The optimal resource combination can be found not only based on resource constraints, but also the availability of space. The general procedures of workspace analysis and cell-based modeling are discussed in this chapter to demonstrate the methods of realizing the new proposed approaches.

4.2 WORKSPACE REPRESENTATION AND CONFLICT DETECTION

The workspace of equipment is more complex than other types because the shape and size of the workspace of equipment, such as cranes, may change depending on the specific activity. For example, the location and shape of the lift of a crane will define the required lifting zone that can be covered by changing the length and angle of the boom of the crane. Furthermore, spatial constraints may put limitations on the length and angle of the boom, thus reducing the lifting capacity of the crane. In many cases, these constraints impose a certain construction method that would not have been used otherwise, such as using two cooperative telescopic cranes instead of one larger straddle crane. These complex relationships between the lifting capacity of a crane and its dimensions necessitate careful consideration of the workspaces used in the spatial analysis of projects involving cranes. Previous research simplified the shape of workspaces by assuming simple box shapes (rectangular prisms). This simplification is useful in general for representing the workspaces of equipment, such as trucks. However, in the case of cranes and earth moving equipment (e.g., bulldozers and excavators), the workspaces should be defined based on a range of the dimensions occurring with the dynamic mechanism of different components. In order to represent more complex workspaces, it is proposed to use shapes other than boxes, e.g., cylinders, cones, and spheres, or a combination of these shapes to represent more realistic workspaces.

4.2.1 Representation of equipment workspaces using composite shapes

CSG (details can be found in Section 2.2.1) is used to combine primitive shapes to represent more complex workspaces, such as the workspace of the telescopic crane shown in Figure 4.1.

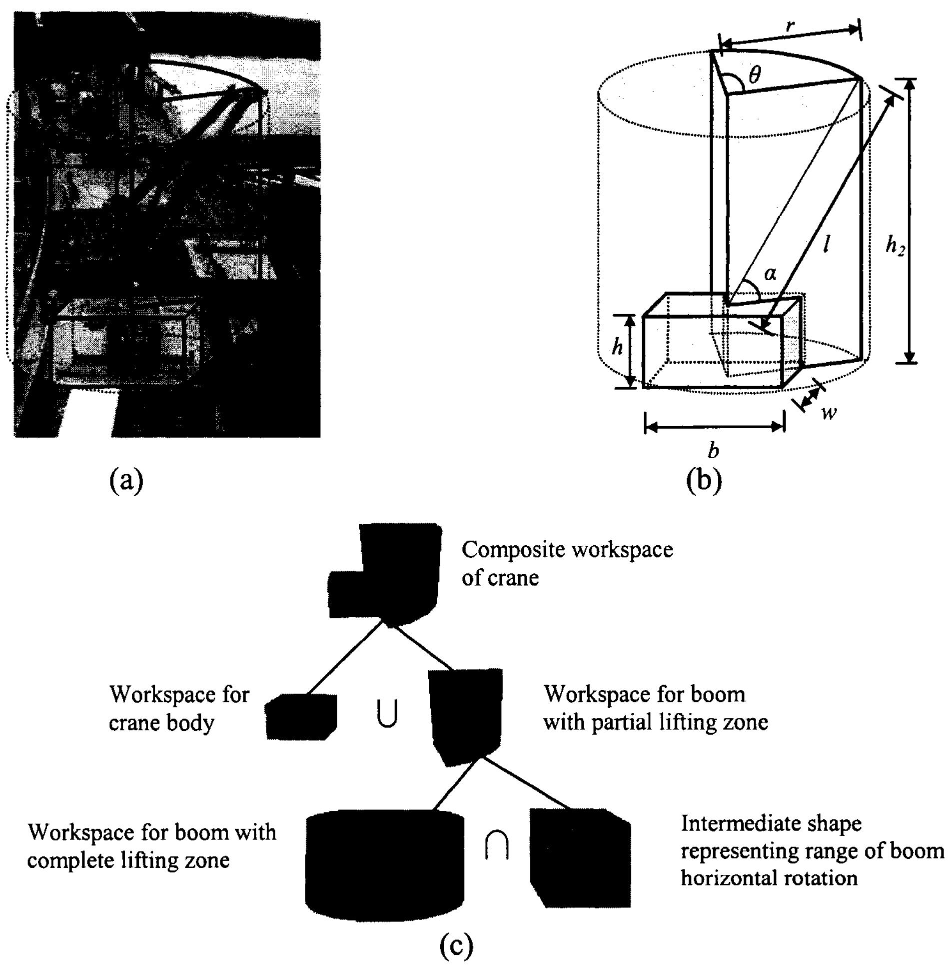


Figure 4.1 Example of workspace of a telescopic crane:
(a) Workspace of a crane superimposed on a picture of construction site;
(b) Parameters used in generating the workspace; and
(c) Generating the composite shape of the crane workspace using CSG

In this example, the composite shape of the crane workspace is created by computing the union of a box representing the workspace for the crane base (including the outriggers) and the workspace of the boom with partial lifting zone. The workspace of the boom is computed as the intersection of the complete lifting zone (cylinder) and an intermediate shape representing the angle of the boom's horizontal swinging (Figure 4.1 (c)). This method requires several parameters to define the workspace. For example, as shown in Figure 4.1 (b), the workspace of the telescopic crane is represented according to the dimensions of the base (*h, b* and *w*), the length (*l*), the angle to the ground (α), and the horizontal swinging angle (θ) of the boom. One advantage of this representation of the workspace is that it includes the lifting path.

4.2.2 General procedure for workspace analysis

The general procedure for workspace analysis described in this section identifies the main steps and algorithms needed in a computerized system for the generation and analysis of equipment workspaces. This system integrates information from, and add information to, the following databases and models (see Figure 4.2):

(1) Activity database: this database includes information about all the activities in the construction project, such as the start and finish times of each activity, the target physical components and their attributes (e.g., the deck section to be replaced), and the types of equipment required in that activity. For example, in the case study of the replacement of a bridge deck that will be discussed in detail in Chapter 5, a typical activity is the replacement of an old section of the deck with a pre-fabricated panel. In this example, each activity includes the start and finish times, the ID number of the

target section, and the required equipment such as cranes, trucks, etc.

(2) Equipment databases: Equipment manufacturers and large construction companies usually have databases of the various equipment used in their work. These databases include the specifications about the different models of a certain type of equipment. D-Crane is a good example of such databases (Al-Hussein 1999).

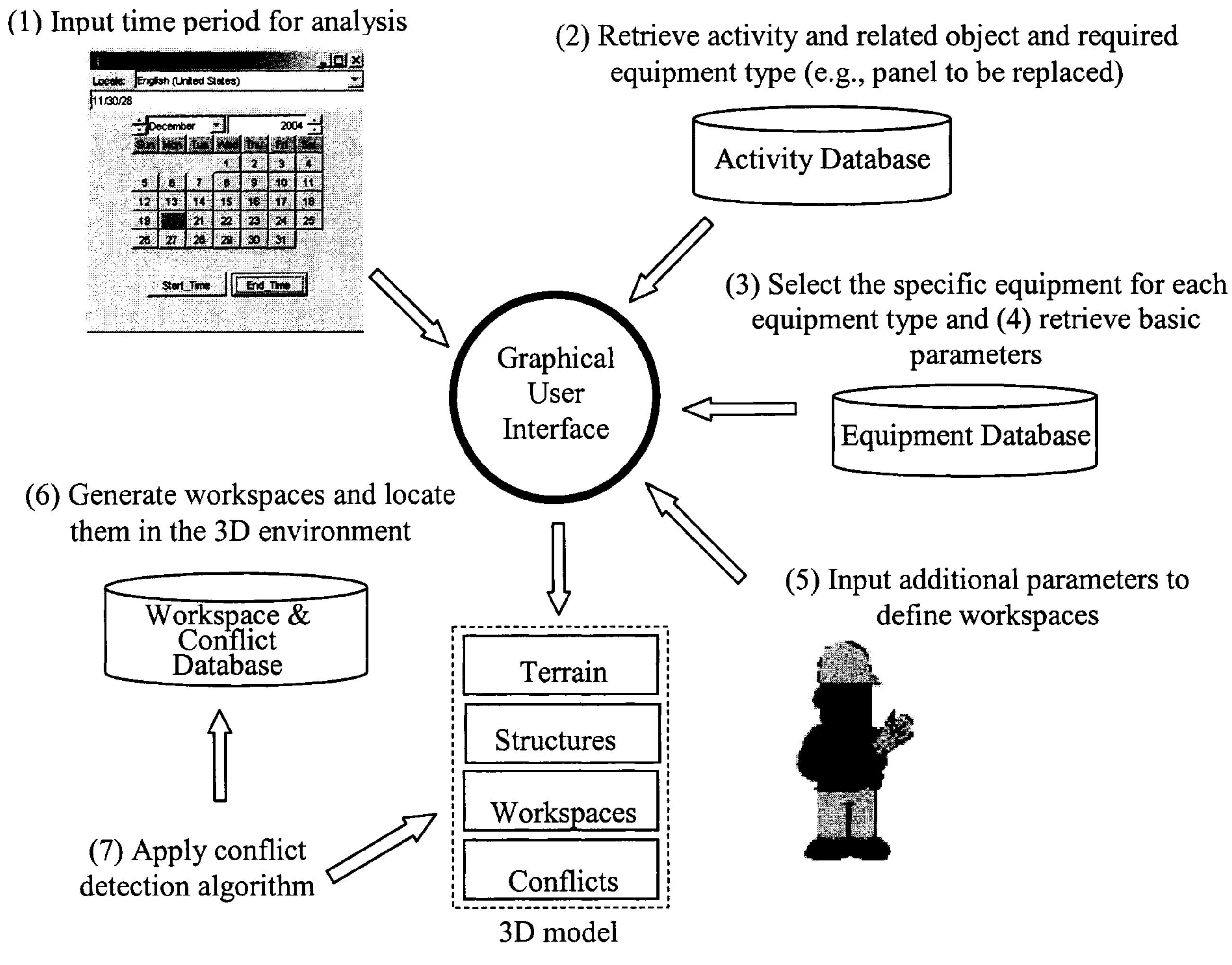


Figure 4.2 General procedure for workspace analysis

(3) Workspace and conflict database: This database has the schemata representing the attributes of workspaces and spatio-temporal conflicts. As was explained in Section 4.1, workspaces can be represented by composite shapes, and each basic shape is described by a number of parameters. Conflict information is calculated using the conflict detection algorithm (details are discussed in Section 4.2.3). It includes reference to the conflicting workspaces and the duration for which the conflict exists in addition to the attributes of the intersection shape of the workspaces, such as its volume. This database is specific for each project and can be used by the construction project manager to resolve the conflicts.

(4) 3D model of the site: This model integrates the digital terrain model of the construction site and the 3D CAD models of the surrounding structures. In addition, 3D shapes representing the workspaces are generated and added to this 3D model as explained below. These shapes are used to visualize and detect conflicts between the 3D elements representing the structures and the workspaces, or among the workspaces themselves, using the conflict detection algorithm.

Using the above databases and the 3D model, the following procedure can be applied for workspace analysis (Figure 4.2):

(1) The user starts by selecting the main activity to be considered in the workspace analysis.

(2) The system retrieves the information about this activity and all other overlapping activities from the activity database. The information includes the related objects and the required equipment types.

(3) Then, feasible equipment is selected for each required type from the corresponding

equipment database. It should be noted that selecting the optimal equipment is beyond the scope of this research.

(4) The next step is to retrieve the basic equipment parameters necessary to define their workspaces. For example, in Figure 4.1 (b), the following parameters should be retrieved from the crane database: b, w, h, l, and α. Because of the interdependency between the equipment type and its location, steps 3 and 4 may require using a specialized algorithm (Al-Hussein 2001).

(5) Other parameters that are necessary for creating workspaces and that are related to the specific site layout are input manually using the user interface of the system. For example, in Figure 4.1 (b), the boom horizontal swinging angle (θ) can be specified by the user interactively. In addition, the relative location of equipment on site can be defined with respect to the reference object that was retrieved from the activity database in step 2, such as the section of the bridge deck to be replaced (Figure 4.3). The parameters include the orientations and the offset distances between the workspace and the object.

(6) In this step, workspaces are generated using the parameters introduced in steps 4 and 5. To locate a workspace in the 3D model, the absolute location of the reference object is retrieved and combined with the relative location of the workspace to generate the absolute location of the workspace.

(7) After all the workspaces have been generated and located in the 3D model, the conflict detection algorithm is applied on each pair of workspaces, or on a workspace vs. a physical component of the structure. The conflict is also represented as a 3D shape in the 3D model. In addition, the dimensions of the conflict and other related

information are saved in the workspace and conflict database.

4.2.3 Computational aspects

(1) Locating workspaces

As was explained in step 6 in Section 4.2.2, the absolute location of the object related to the activity selected in step 2 should be retrieved and combined with the relative location of each workspace to generate the absolute location of the workspace in the world coordinate system. The transformation matrices and normal vectors of the object are computed based on information extracted from the scene graph of the 3D model and used to locate the workspaces at the right location in the world coordinate system. The normal vectors of the surfaces of objects are used to represent the orientation of the workspace, and offset distances along those vectors are used to define its relative location. For example, Figure 4.3 shows side view and top view of a simplified 2D example for locating a workspace above, to the left, and in front of a reference object with offset distances in the directions of the normal vectors N_x, N_y, and N_z, respectively.

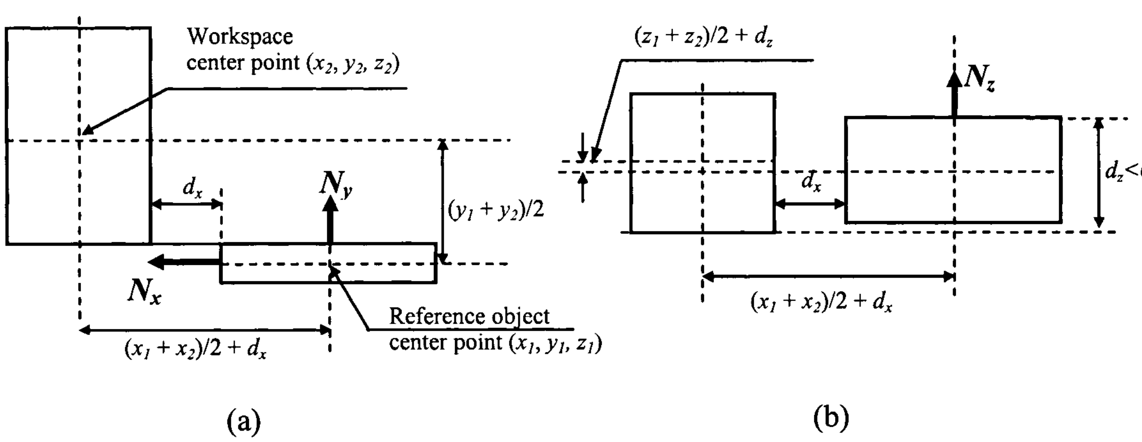


Figure 4.3 Translation distance for locating a workspace
(a) Side view of a case of *above* and *left* with offsets of zero and d_x, respectively; and
(b) Top view of a case of *left* and *in front* with offsets of d_x and d_z, respectively

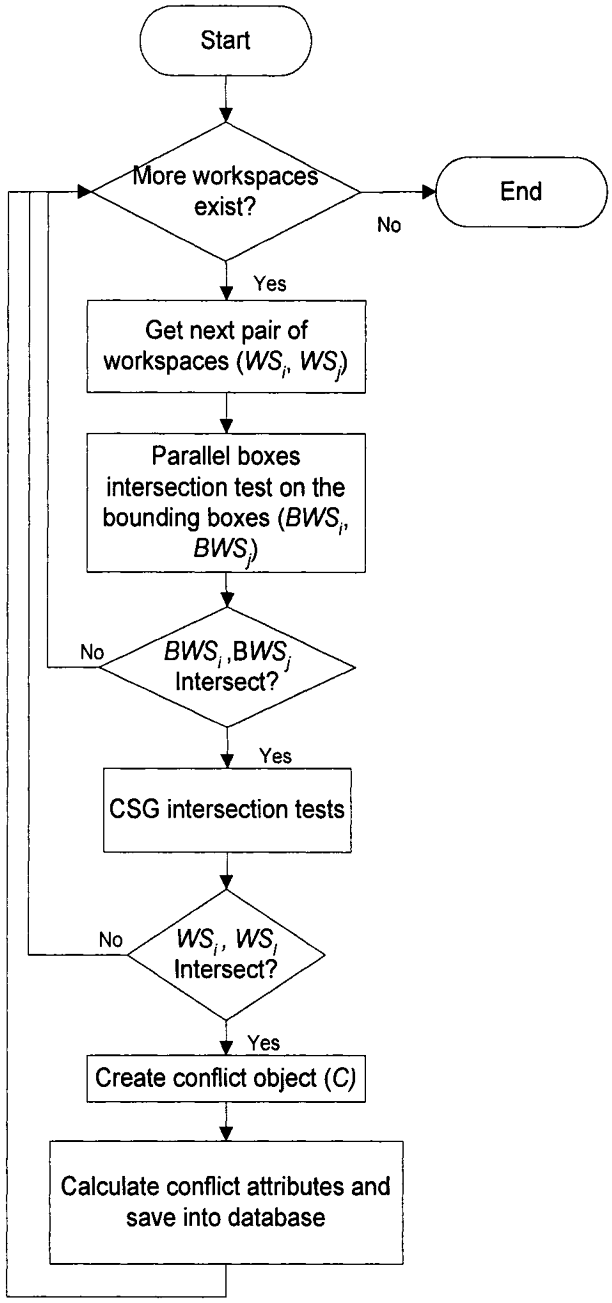


Figure 4.4 Flowchart of conflict detection algorithm

(2) Workspace conflict detection

Figure 4.4 shows the flowchart of the conflict detection algorithm (step 7). After all the workspaces *{WS}* are created, conflict detection is applied to pairs of workspaces that have temporal overlap (WS_i and WS_j) (or a workspace and a physical component). Two tests are considered in this step: (1) the test of the intersection of the bounding boxes of the workspaces (BWS_i and BWS_j), which are two rectangular prisms with parallel faces, and (2) the general intersection tests using CSG. The reason for this is that CSG is computationally intensive when considering the very large number of objects (workspaces and physical components) for which interference may have to be checked. In the first test, a conflict can be detected simply by comparing the distance between the center points of the two rectangular prisms with the dimensions of the prisms. If a conflict exists, the CSG test is applied to confirm the existence of the conflict based on the detailed representations of the workspaces and to find the accurate shape of the intersection. The following two paragraphs explain the algorithms used in each case:

a) **Bounding box collision detection algorithm:** To simplify the description of this algorithm, we will assume that both bounding boxes have their edges parallel to the axes of the coordinate system (Figure 4.5 (a)). The distances between the center points of the boxes are calculated in three dimensions ($\triangle x$, $\triangle y$, and $\triangle z$). If $\triangle x < (a_1+a_2)/2$ and $\triangle y < (b_1+b_2)/2$ and $\triangle z < (c_1+c_2)/2$ (where a_1, b_1, c_1 and a_2, b_2, c_2 are the dimensions of the two boxes) then there is an overlap between the two boxes and a conflict exists. For the purpose of simplicity, the 2D cases of intersection and containment are shown in Figures 4.5 (b) and (c), respectively. In Figure 4.5 (b), $\triangle x > abs((a_1-a_2)/2)$ and, therefore, the dimension of the conflict box along the X axis is $a_c = (a_1 + a_2)/2 - \triangle x$. Figure 4.5 (c) shows that $\triangle x < abs((a_1-a_2)/2)$ and therefore $a_c = min(a_1,a_2) = a_2$.

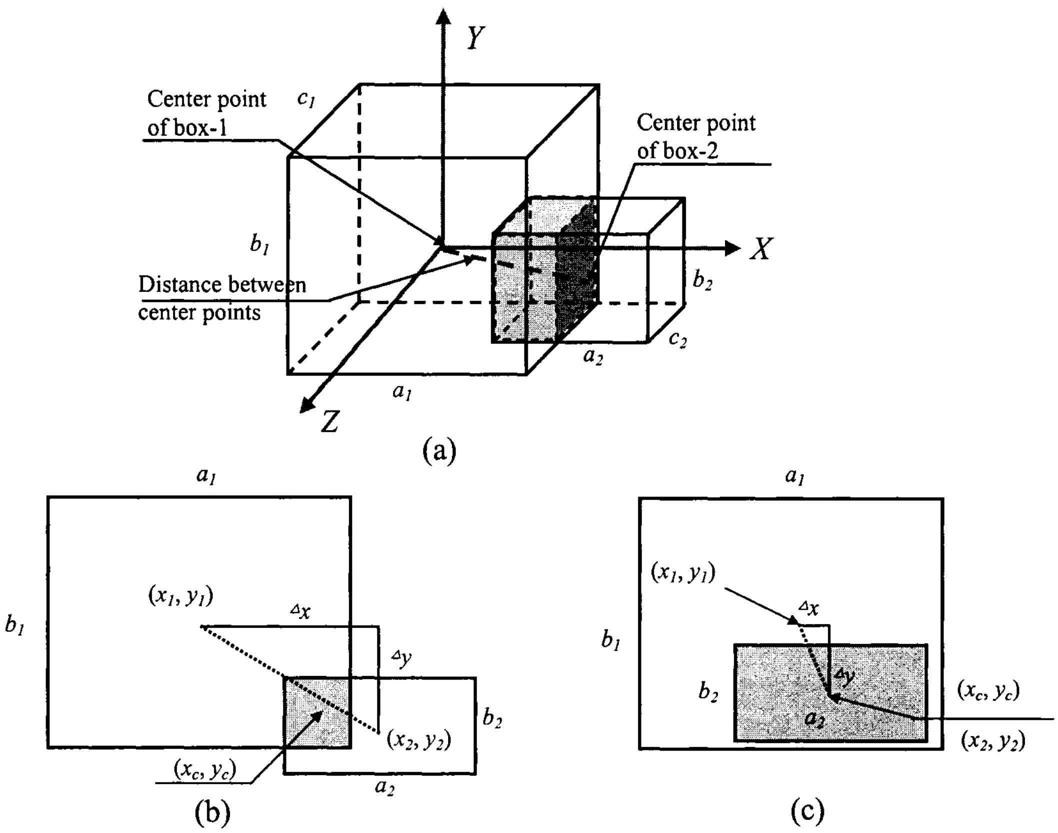


Figure 4.5 Collision detection between two parallel boxes:
(a) Distance between center points of two boxes;
(b) Intersection case; and (c) Containment case

b) **CSG collision detection algorithm:** For complex shapes created using CSG operators, a common exact collision detection algorithm is used (Watt 2000). Three tests are applied and the success of any of them implies that a collision has occurred. For the case of the two convex polyhedra *P* and *Q* shown in Figure 4.6, the first test is to check if any of the vertices of *Q* is contained in *P* and vice versa (Figure 4.6 (a)). Each vertex of *Q* has to be checked against every face of *P*, and a collision is detected if any vertex is on the inward side of all the faces of *P*. Thus for each vertex $\boldsymbol{v}_i$ of *Q* and for each face *j* of *P* the dot product $(\boldsymbol{v}_i - \boldsymbol{u}_j).\ \boldsymbol{N}_j$ is evaluated, where $\boldsymbol{u}_j$ is any vertex of face *j* and $\boldsymbol{N}_j$ is its outward normal (Figure 4.6 (c)). If this dot product is negative then the vertex v_i is on the inward side of face j. In the second test, the edges of Q are tested for penetration against the faces of P and vice versa (Figure 4.6 (b)). The intersection of an edge (v_i, v_j) of Q with the planes containing the faces of P is calculated. For any plane k of P, an edge intersects it if the perpendicular distance from each vertex to the plane changes sign. The intersection point $\boldsymbol{x}$ can then be calculated as follows:

Equations (4.1)

$$\begin{cases} d_i = (\boldsymbol{v}_i - \boldsymbol{u}_k) \cdot \boldsymbol{N}_k \\ d_j = (\boldsymbol{v}_j - \boldsymbol{u}_k) \cdot \boldsymbol{N}_k \\ t = | d_i | / (| d_i | + | d_j |) \\ \boldsymbol{x} = \boldsymbol{v}_i + t(\boldsymbol{v}_j - \boldsymbol{v}_i) \end{cases}$$

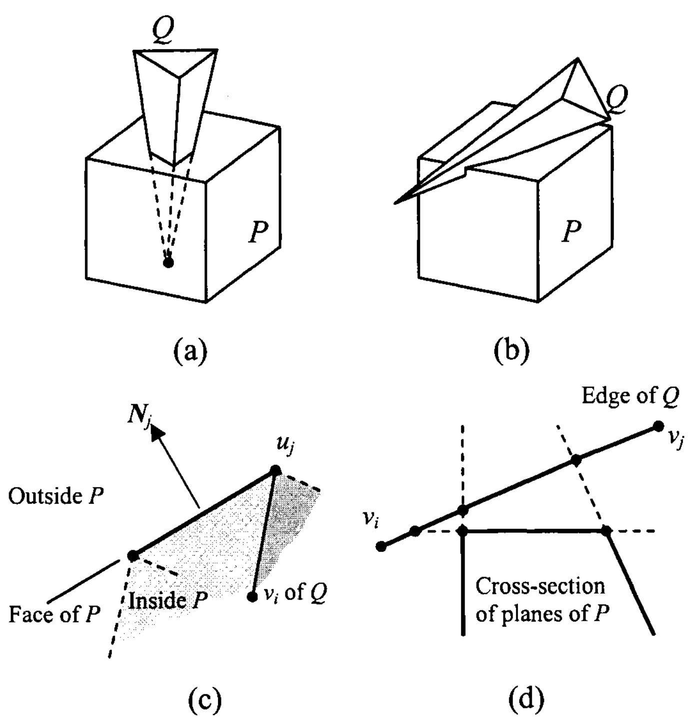


Figure 4.6 Collision detection tests for convex polyhedra (Watt 2000):
(a) A vertex of Q contained in P; (b) An edge of Q penetrates a face of P;
(c) A vertex of Q contained by P test; and (d) An edge of Q cutting a face of P

This gives, in general, a number of intersection points along the edge. Those for which $t \notin [0, 1]$ are discarded and the remaining points are sorted in order of their t values. These form a sequence of potential intersections from one vertex to the other (Figure 4.6 (d)). Finally, to check for intersection, the midpoint of each pair can be substituted into the first test. The third test is to check the infrequent case of two identical polyhedra with faces perfectly aligned.

4.2.4 Visualization of workspaces and conflicts

The advantage of visualization is that the user can simulate and check the interferences that may happen in reality between the 3D physical elements and virtual workspaces or among virtual workspaces. In many infrastructure projects, such as the case study discussed in Chapter 5, a physical model of the project is built to check workspace interferences, which is more costly than developing a digital 4D model to facilitate spatio-temporal analysis. Information about workspaces and conflicts can be shown on the model using different colors and transparency levels (Table 4.1).

Table 4.1 Color coding used in workspace analysis

Objects		**Color**	**Transparency**
Physical Elements	Bridge truss	Dark green	Opaque
	Bridge deck	Yellow	
Virtual Objects	Crane workspace	Light green	Transparent
	Truck workspace	Red	
	Emergency path	Purple	
	Conflict	Faces have same colors as workspaces causing conflict	Opaque

Elements of the structure can be shown as solid shapes with opaque colors while workspaces can be shown as transparent shapes with different colors to represent different types of workspaces. Then, conflicts can be shown as shapes with faces bearing the colors of the respective conflicting objects. All the digital representations are located in the world coordinate system to give the user a realistic view of the surrounding environment. Spatial conflicts would not be understood fully if they were not linked to geographical locations as perceived in the real world (Zlatanova et al. 2002). This requirement can be achieved by integrating the CAD models with GIS, which is considered and discussed in Chapter 3. Spatial information about the elements can be retrieved directly from the 3D environment and used to create the related workspaces and check for the existence of conflicts.

4.3 CELL-BASED SIMULATION OF CONSTRUCTION ACTIVITIES

Space is one of the resources that may cause crucial problems during construction. As reviewed in Section 2.4, discrete event simulation has been widely used in construction to allocate resources and improve productivity or mitigate conflicts. However, simulation research that provides an explicit method to investigate possible space conflicts is still limited. In this section a cell-based modeling approach is suggested to represent space resources in construction simulation, which enables conflict analysis and visualization of the worksite and the occupation of spaces. The new modeling approach is compared with MicroCYCLONE as a representative of conventional construction simulation tools to identify the advantages and limitations of each method in spatial resource representation.

4.3.1 Limitations of available simulation models

There are several software packages available to develop construction simulation models. MicroCYCLONE (Halpin 1997) and Stroboscope (Martinez 1998) are two popular programs used in construction simulation to assist the manager in making decisions about resource allocation and conflict prediction. These two programs use similar representations of activities and resources. MicroCYCLONE is used to demonstrate how space resources are represented and to identify the limitations of available construction simulation tools.

A case study of the Jacques Cartier Bridge re-decking project (details of the case study will be discussed in Chapter 5) is used to discuss the limitations of available simulation models and the feasibility of the proposed approach. The deck of this bridge was replaced in 2001-2002. The new deck is constructed of precast, prestressed and post-tensioned panels made of high performance concrete, which were prefabricated in a temporary plant installed near the south end of the bridge. The case study focuses on the two activities of removing existing deck sections and installing new panels in the main span of the bridge. These two activities were critical for the success of the project from the point of view of spatial and temporal constraints. The existing deck was removed by saw-cutting the deck into 410 sections similar in dimensions to the new panels being installed. Each existing deck section was removed by two telescopic cranes, and a new panel was lifted from a truck and lowered onto the new bearing assemblies. Old sections were transported to a dumping area near the bridge. New panels were transported from the plant located at the south end of the bridge. Table 4.2 shows the tasks involved in these two activities and their durations.

Table 4.2 Task durations and IDs used in MicroCYCLONE

Activity	Task ID	Task Description	Triangular Distribution of Durations (min)
Remove old sections	4	Cut old section	15, 18, 30
	9	Load old section	12, 15, 20
	10	Truck with old section travels to dumping area	5, 7, 8
	13	Dump old section	4, 5, 7
	14	Empty old-section truck returns to bridge	4, 5, 7
Install new panels	22	Load new panel	10, 14, 15
	23	Truck with new panel travels to bridge	6, 7, 8
	28	Install new panel	23, 26, 28
	31	New-panel truck returns to plant	4, 5, 6
	30	Team repositioning	15, 18, 20

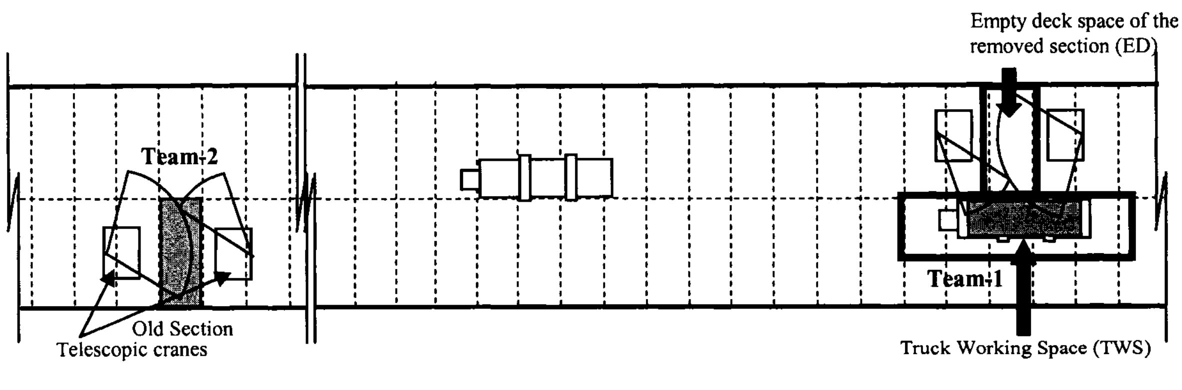


Figure 4.7 Worksite layout of the bridge re-decking

Figure 4.7 shows a schematic representation of the worksite layout during the deck replacement. Most of the time, two teams worked in parallel on different parts of the bridge. Each team used two telescopic cranes located at both sides of the section to be replaced. Two types of semi-trailer trucks were used to transport the old sections and the new panels. As shown in Figure 4.8, active-state rectangular nodes represent tasks and include the triangular time distribution for each task. Idle-state circles represent delays or waiting positions for resource entities; and directional flow arrows represent the path of resource entities as they move between idle and active states. Tasks can be executed only when all the queues (resources) needed are available.

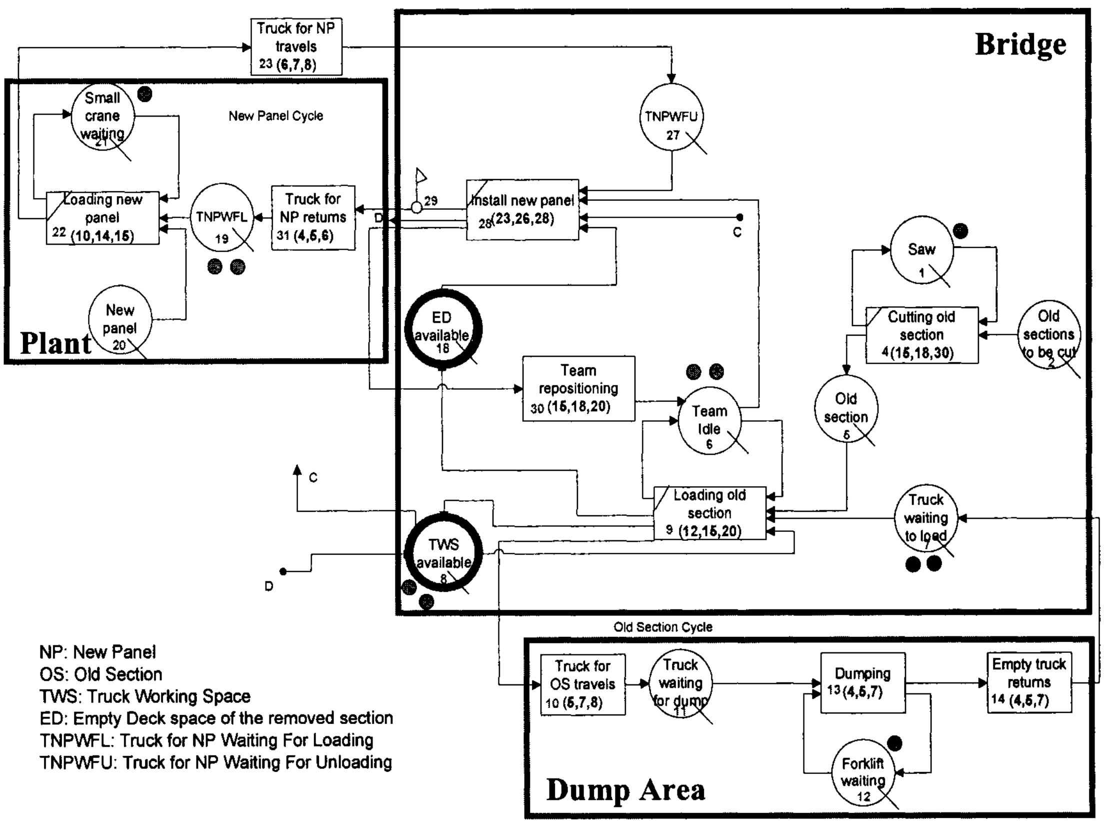


Figure 4.8 MicroCYCLONE model of Jacques Cartier Bridge re-decking project

For example, task "Dumping" (Task 13) is the task of dumping an old section of the deck in the dump area. It takes an average time of 5 minutes and needs two resources: "Truck waiting for dump" (Task 11) and "Forklift waiting" (Task 12). After the dumping of one section is finished, these two resources are released: the truck returns to the bridge (Task 14), and the forklift goes back to the idle state (Task 12). The procedure for modeling these processes involves four basic steps: (1) Flow unit identification; (2) Development of flow unit cycles; (3) Integration of flow unit cycles; and (4) Flow unit initialization. The details of MicroCYCLONE modeling techniques are available in Halpin and Riggs (1992).

To investigate the space representation in this model, a conceptual mapping of the space is applied. In Figure 4.8, spaces are represented as abstract symbol-queues. The site layout can be approximately divided into several areas according to the geographic locations, including *Bridge*, *Plant*, and *Dump Area*. There are two spaces implicitly represented as queues in this model (Figure 4.8): (1) Empty deck space of a removed section (ED); and (2) Truck working space (TWS). Other spaces, such as the moving paths of trucks, are considered to be available all the time and are not represented in this model. In addition, workspaces can change from one task to another, and it is the responsibility of the modeler to identify these spaces in MicroCYCLONE.

Moreover, in MicroCYCLONE and other conventional simulation tools, spatial conflicts are not discovered except by visualizing the results using post processing applications (Kamat 2001). In fact, our observation is that most of the time space resources are attached to other resources (equipment, materials, etc.) and are heavily dependent on the site layout. Therefore, our proposed approach is based on linking spatial resources with other resources.

4.3.2 Procedure of cell-based modeling

As discussed in the Chapter 2, space can be divided into cells and every cell is a discrete event model. A cell can change its state according to its own time delay and external events. A dynamic information exchange can be achieved during the simulation period. Conflict detection can be simplified by checking the state of each cell and avoiding an occupied cell being used by other objects. Based on this idea, a cell-based modeling approach is used to investigate the spatial issues in construction simulation. It should be noted that in spite of the availability of cell-based simulation tools developed for research purposes, this is, to the best of our knowledge, the first trial to investigate the applicability of these tools in construction simulation. In the rest of this section, the process used in building cell-based models will be introduced using several examples related to the case study of Jacques Cartier Bridge re-decking project. Cell-DEVS modeling techniques for representing spatial constraints in construction simulation will be discussed in detail. A full description of the implementation of the case study will be given in Chapter 5. Figure 4.9 shows the general procedure of cell-based modeling.

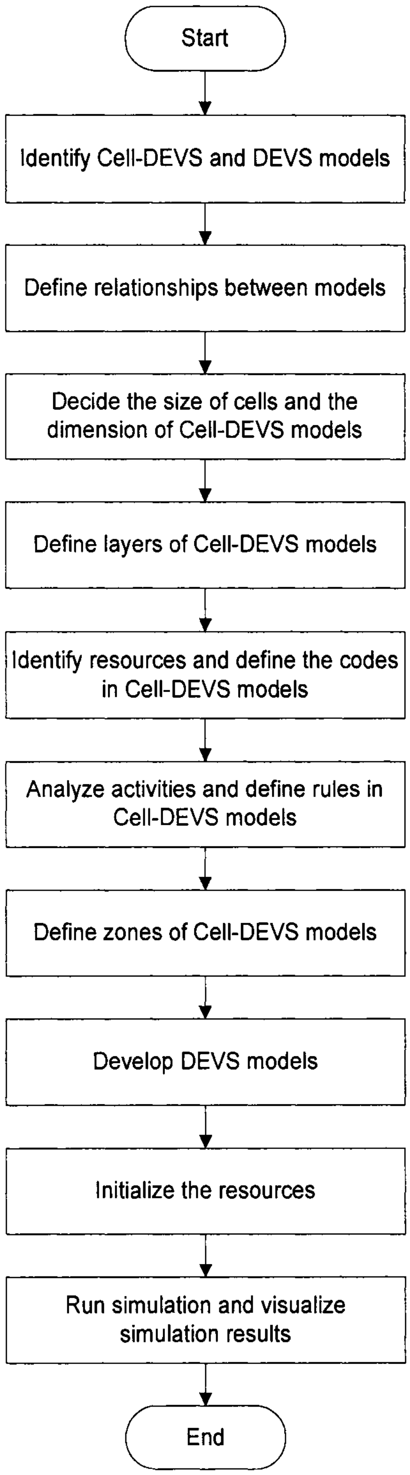


Figure 4.9 Flowchart of general procedure of cell-based modeling

This procedure has ten steps:

(1) **Identify Cell-DEVS and DEVS models.** Cell-DEVS models are used where the spatial representation is important; while the DEVS models are used to simulate the parts that do not need spatial representation or could not be represented using cells. For example, in our case study, a Cell-DEVS model could be extended to cover the whole working area, including the bridge, the dump area and the plant. However, representing all the roads between the bridge and the dump area or the plant as cells is not necessary because there are fewer spatial constraints on these parts. Therefore, only the three main working areas need to be represented as Cell-DEVS models: *Bridge, Plant* and *Dump Area.* DEVS models are used to build the models of queues and transports between the above Cell-DEVS models.

(2) **Define the relationships and information exchange between models and connect them using input and output ports.** For example, whenever there is an old section to be replaced, a request for a truck to go to the *Bridge* model will be sent to the queue of waiting trucks. More details about the implementation of this step are given in Section 5.

(3) **Decide the suitable size of cells and the dimensions of each Cell-DEVS model.** For example, Figure 4.10 shows the cell representation of the bridge model. The cell size is assumed to be 3*3 meters. The main span of the bridge can be approximately represented by 200*6 cells.

(4) **Define the layers of each model.** As explained in Section 2.2.3, several layers are needed to represent the attributes related to a Cell-DEVS model.

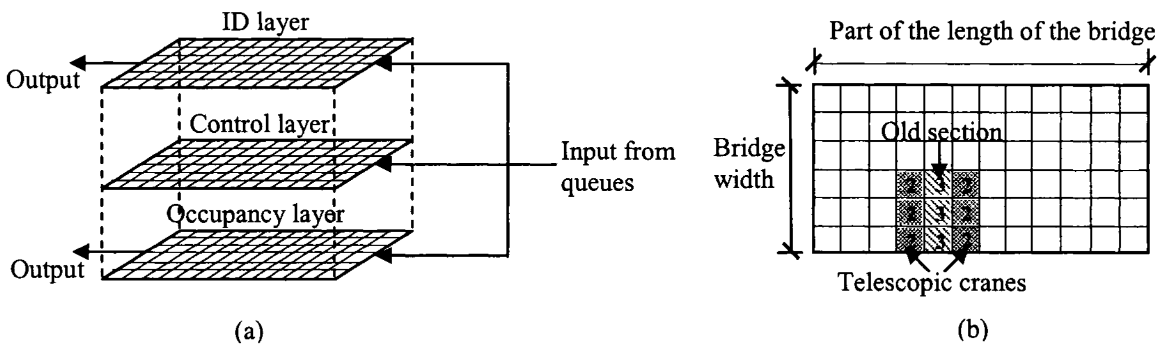


Figure 4.10 Cell representation of the *Bridge* model:
(a) The three layers used to model occupancy, direction, and ID information; and
(b) Cell representation of the *occupancy layer*

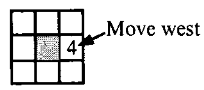
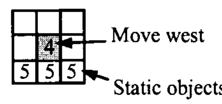
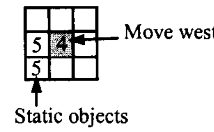
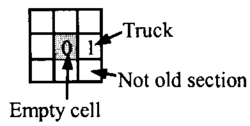
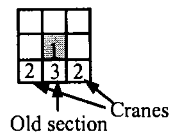
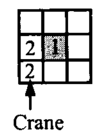
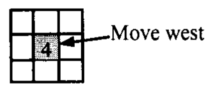
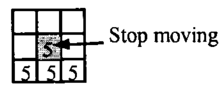
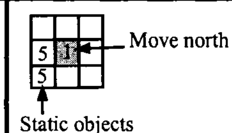
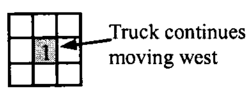
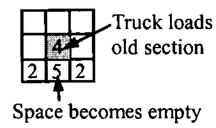
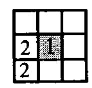
Based on our experience in cell-based modeling (Zhang et al. 2005), three layers should be created for the *Bridge* model to represent the occupancy, mobility conditions and IDs of the objects occupying the cells (Figure 4.10 (a)). The first and main layer is the *occupancy layer*, which has the occupancy states of the cells (e.g., the type of equipment occupying a cell). The second layer is the *control layer*, which decides the mobility state and moving direction, detects conflicts between objects, and sets the priority of moving depending on the types of objects as defined in the *occupancy layer*. The cells in the *control layer* will detect the corresponding cells in the *occupancy layer* and set the priority based on certain rules (as will be explained in step 6). The *ID layer* contains the ID numbers of each piece of equipment. Combining the information about a certain location (cell) collected from the corresponding cells in the three layers gives a triplet of attributes for that location: <occupancy, mobility, ID>. For example, a cell can be occupied by a truck moving west and having an ID of "2".

(5) Identify the resources needed and define the codes that will be used in Cell-DEVS models. Different encodings can be used to represent equipment states, the occupancy of cells, and the IDs of equipment, such as those given in Table 4.3. For example, Figure 4.10 (b) shows an old section (2) and two cranes (3) beside it, which occupies 3 cells each. For simplicity, cells with zero value are shown as blank cells. The same code can be used in different layers with different meanings. In the *control layer* of the *Bridge* model, different mobility conditions are represented as (1): move north; (2): move south; (3): move east; (4): move west; and (5): static object. Any number can be used in the *ID layer* to represent the IDs of equipment.

Table 4.3 Codes used in the *occupancy layer* of different Cell-DEVS models

Model	Code	Descriptions
Bridge	0	Empty cell
	1	An empty truck for old sections
	2	Crane (occupies 3 cells)
	3	Old section on the bridge (occupies 3 cells)
	4	A truck carrying an old section
	5	Empty space after removing the old section
	6	New installed panel
	8	A truck carrying a new panel
	9	An empty truck for new panels
Dump Area	1	An empty truck for old sections
	4	A truck carrying an old section
	7	Forklift
Plant	6	Small crane
	8	A truck carrying a new panel
	9	An empty truck for new panels

Table 4.4 Examples of rules of truck movements

		Rule 1	Rule 2	Rule 3
Conditions	Control layer			
	Occupancy layer			
Actions	Control layer			
	Occupancy layer			
Time delay		Time to move one cell	Triangle distribution (12, 15, 20) min.	Time to move one cell

(6) **Analyze the activities and develop rules for each Cell-DEVS model.** Cells can communicate with each other through rules that detect the state changes of a cell's neighborhood and that change the cell's own state accordingly. Rules governing the interactions between layers will guarantee the coupling of the attribute triplets introduced in step 4 and the consistency across layers. There are several types of rules that could be applied for simulating construction activities, such as rules for moving trucks, conflict detection, truck generation, direction changes, etc. Each rule has a condition part, an action part, and a time delay. The conditions and actions may involve the attributes defined in one or more layers. Examples of rules that could be used in the *Bridge* model are shown in Table 4.4. The first rule represents an empty truck for loading an old section. From the *occupancy layer*, the truck should move

into the next cell if it is not facing an old section. From the *control layer*, both the condition part and the action part show that the direction of the truck movement is towards the west (4). This means that if a truck is moving in certain direction at a certain time step, it will continue moving in the same direction in the next time step. The time delay for this action is the time needed to move one cell that can be calculated from the average speed of the truck. The second rule represents a truck arriving at the location of an old section. In the *occupancy layer*, the truck should stop to load the old section. After the time delay needed for this task, it will change its occupancy state to (4), which means the truck is now carrying the old section. In the *control layer*, the mobility state is changed to static object (5) when the truck is loading the old section. After the loading is done, the mobility state changes to (4) and the truck will move west again (assuming that this is the direction towards the dump area). The third rule is about conflict detection. The cells representing static objects, such as old sections and cranes are set to the highest value (5 = static object), which means that a moving object (e.g., truck) with a lower value should change its direction to avoid collision with these static objects. When the truck is moving west (the cell value is (4)), the rule in the *control layer* checks whether the cell to the west is occupied. In this case, conflict analysis will be applied to decide the new direction (i.e., move north or south) until there is no obstacle in the way of the truck. Then the direction will be changed back to (4) and the truck will continue moving west.

(7) **Define the zones of each Cell-DEVS model.** In general, rules in a Cell-DEVS model are applied to all the cells of that model. However, in many cases, it is necessary to specify a part of the model as a zone in order to define local rules that

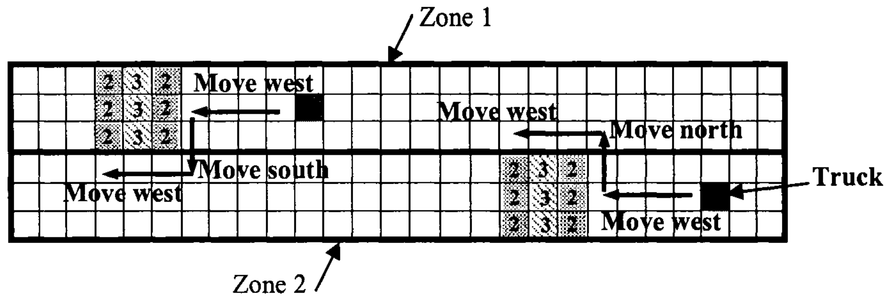


Figure 4.11 Zone division and the rules for changing directions in each zone

apply only within that zone. In the *Bridge* model, both the *occupancy* and *control layers* should be divided into two zones to define different moving directions when a conflict is detected (Figure 4.11). The main difference between the rules of these two zones is that a truck has to take a different direction when it meets an obstacle. For example, when the *control layer* detects a conflict between a truck and a crane, the direction of the truck will be changed to south or north to avoid the obstacle, depending on whether the crane is in *zone 1* or *zone 2*, respectively. This example shows the limited intelligence that can be represented by rules for controlling direction changes.

(8) Develop the DEVS models. DEVS models can be developed based on the functions that they perform, such as queues or transport functions. Examples of these DEVS models are given in Section 5.

(9) Initialize the resources. The number of resources should be initialized before running the simulation. For example, the number of trucks can be initialized in the *Dump Area* and the *Plant* models. Multiple teams can be initialized in the *Bridge* model by defining the location of each team (Figure 4.11).

(10) Run the simulation and visualize the simulation results. The simulation tool will generate the discrete changes of states in each model. These results can be visualized as an animation that gives a quick method for checking the results.

As an example of the overall mechanism of the simulation process, Figure 4.12 shows the state changes in the three layers at different time steps that will be generated based on rules similar to those explained above. At time t_1 an empty truck (1) is moving west with an ID number "2". At time t_2, it meets an obstacle, which is a crane (2). The *control layer* detects that the mobility state of the crane is static (5), thereby indicating that the crane should be given the highest priority. Therefore, the direction of the truck is changed to (1) to move north because it is in *zone 2*. After checking the neighboring cells again at t_3, the moving direction is changed back to (4) and the truck is moving west again (t_4). At time t_5, the truck arrives at the location of an old section that will be removed. It stops and the *control layer* changes the corresponding cell state to (5). After some delay, the truck occupancy state will change to (4), which means it carries an old section and is ready to move. At the same time, the cells that represent the old section change their occupancy state to (5), which means the space is empty and ready for the installation of a new panel.

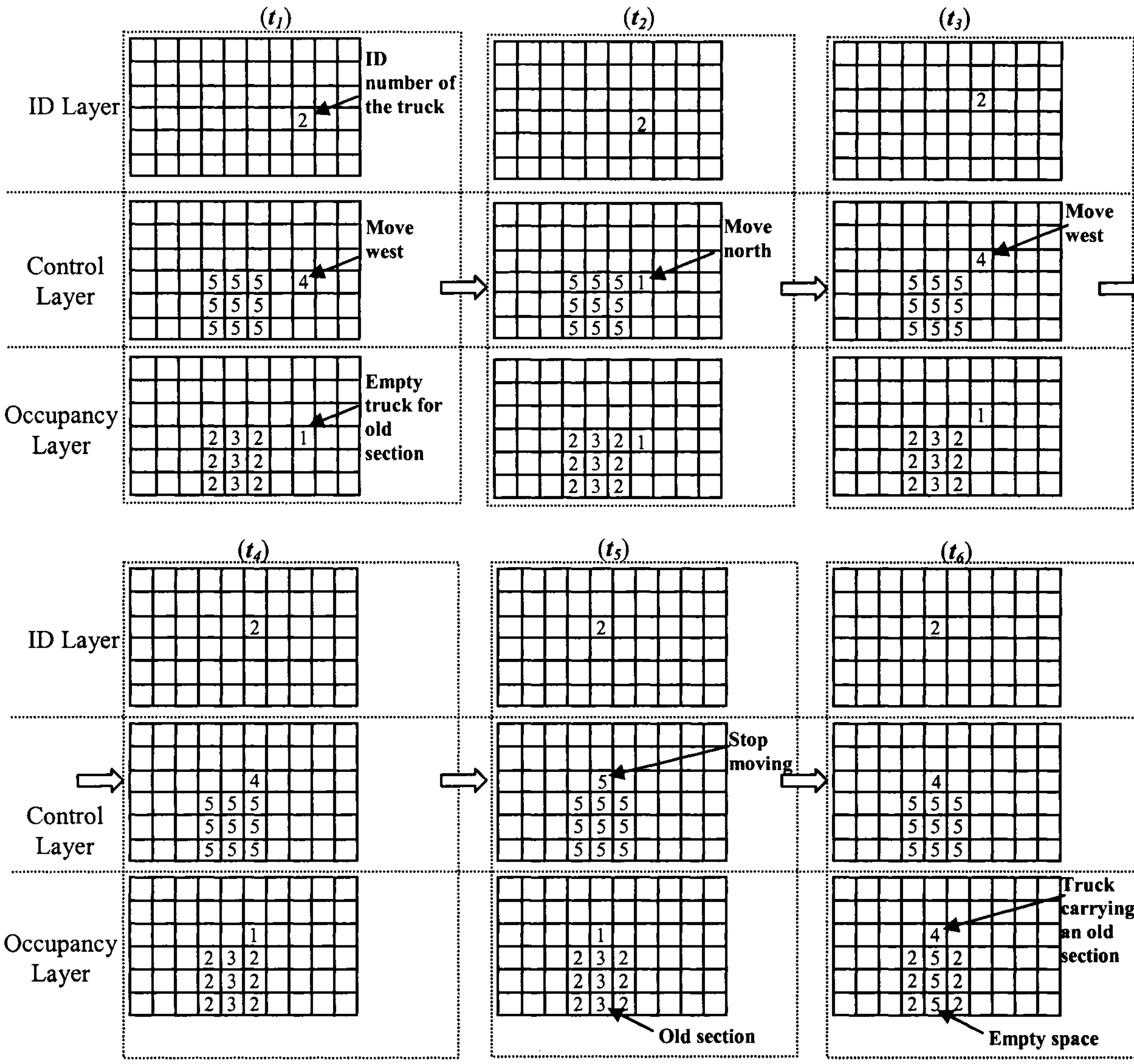


Figure 4.12 States changes in the three layers at different time steps

4.3.3 Comparison of cell-based modeling and MicroCYCLONE

The cell-based model and the MicroCYCLONE model of the case study are compared to investigate their spatial representation. The three Cell-DEVS models explicitly represent the three major areas that are implicitly identified in the MicroCYCLONE model. The physical space and the occupation situations on the bridge, the plant, and the dump area can be clearly shown using the Cell-DEVS models. Transport models are similar to the transportation tasks in the MicroCYCLONE model, such as "Truck with new panel travels to bridge" activity (Task 23). Queue models are similar to the queues in the MicroCYCLONE model, such as "Truck for old section" queue (Task 15). *Reposition* model is similar to the task of "Team repositioning" (Task 30).

The advantages and limitations of each model are shown in Table 4.5. The advantages of using cell-based simulation are the following: (1) The space can be represented explicitly and the simulation models can be visualized so that the occupation of the workspace and other spatial information about the construction environment can be understood more easily than using conventional models. The physical resources can be initialized to simulate several site layouts and thus to find the optimal layout based on the availability of workspaces; (2) The conflicts between spaces can be detected based on the site layout;

Table 4.5 Comparison of cell-based simulation and conventional simulation tools

Simulation method	**Cell-based simulation**	**Conventional simulation tools**
Spatial representation	Explicit	Implicit
Spatial conflict detection	Based on site layout	Not available
Duration of transportation activities	Based on predefined delay times and the site layout	Based on predefined durations only
Practical usability	Possibility of reusing models in similar cases	Special purpose functionalities for construction simulation

(3) The accuracy of the duration of activities is expected to improve, especially for situations where spatial conflicts are present; and (4) The models created for a project can be reused in other similar projects because the rules are flexible and easy to modify. However, there are also some limitations of the cell-based modeling, such as the difficulty representing complex layouts and mobility on curved roads. The simplified version of the model may not exactly match the worksite. Furthermore, there are some limitations of the cell-based simulation tools themselves, such as the complexity of defining rules. In addition, there are many functions readily available in the conventional tools, such as productivity calculation and sensitivity analysis, which have to be implemented in the cell-based modeling tools.

4.4 SUMMARY AND CONCLUSIONS

Two proposed approaches for spatio-temporal conflict detection and analysis were discussed in this chapter. The first approach focused on workspace analysis where different types of workspaces for crew, equipment, and other required spaces in the work site are created to detect conflicts between them for specific tasks. This approach extends the previous research on workspace analysis in the case of large infrastructure projects focusing on the representation of workspaces related to heavy equipment such as cranes, and the computational methods to define the composite shapes of these workspaces using CSG. The second approach investigats the feasibility of explicit space representation in construction simulation using cell-based modeling to further analyze the spatio-temporal conflicts and to achieve the dynamic detection of these conflicts during construction. The spatial information of the construction environment can be understood more easily and the optimal resources combination can be found not only based on resource constraints, but also the availability of workspaces, which is one of the main reasons that cause delays. This is the first step in investigating the feasibility of applying cell-based simulation to construction projects.

The general procedure of these two approaches showed the specific steps for generating workspace, detecting workspace conflict, and simulating construction processes using cells. The following conclusions can be stated: (1) Spatio-temporal information can be visualized and analyzed in a more understandable way by applying new approaches of workspace analysis and cell-based simulation; (2) Composite shapes with different levels of complexity can be used to represent workspaces of equipment using CSG techniques. Compared with the simple shape of workspace used in previous research, this new approach proposed a more realistic representation and the conflict detection is more accurate based on a geographic coordinate system; (3) Cell-based simulation can be used as a general method to represent the worksite layout and to analyze and visualize the activities that may be affected by space constraints; and thus the spatial conflicts that may occur during construction can be detected. Compared with conventional simulation tools, cell-based modeling has more advantages in spatial representation; (4) The proposed procedures for these two approaches can integrate the attributes of a construction activity, the attributes of the equipment used in this activity, and the geometry information of the project to represent the space explicitly and detect conflicts.

CHAPTER 5 IMPLEMENTATION AND CASE STUDY

5.1 INTRODUCTION

Based on the discussion in Chapters 3 and 4, a prototype system is developed following the framework architecture of integration and visualization to investigate the feasibility and usability of spatio-temporal information visualization and analysis. In addition, a cell-based model is developed and implemented to simulate the construction process. A case study of Jacques Cartier Bridge in Montreal is used to demonstrate the prototype systems.

5.2 SELECTION OF DEVELOPMENT TOOLS

Java is a general purpose programming language with a number of features that make the language well suited for use on the World Wide Web. Java is chosen to develop the prototype system for four reasons: Object orientation, safety, simplicity, and breadth of the standard library (Horstmann 2004). Object orientation enables programmers to spend more time on the design of their programs and less time coding and debugging. Furthermore, graphics, user interface development, database access, multithreading, and network programming are all parts of the standard library. The Java 3D Application Programming Interface (API) allows the programmer to describe the 3D scene using coarser-grained graphical objects, as well as by defining objects for elements such as appearances, transforms, materials, lights, etc. Compared with Open GL, the code is more readable, maintainable, reusable, and easier to write (Selman 2002).

Furthermore, MapObjects-Java Edition is available to build custom applications that incorporate GIS and mapping capabilities or to extend the capabilities of existing applications (MapObjects-Java 2005). MapObjects-Java Edition helps the user build applications that perform a variety of geography-based display, query, and data retrieval based on four tiers architecture: client, presentation, Web, and server tiers.

The database of the 4D bridge model is designed with Microsoft Access to represent the information of all the components of bridge. Although relational database management systems are still the norm in BMS practice, object-oriented modeling and programming tools are widely used in software engineering and can greatly enhance the quality of the software because of their flexible data structure. A good combination of the two approaches is the object-relational approach for database development, which can relate the information in the relational database with the data structure of bridge components as described in object-oriented programs (Object-Relational Mapping 2004). The two development tools: Access and Java language can meet the requirements perfectly. Java Database Connectivity (JDBC) is a programming framework for Java developers writing programs that access information stored in databases.

5.3 BACKGROUND OF THE CASE STUDY

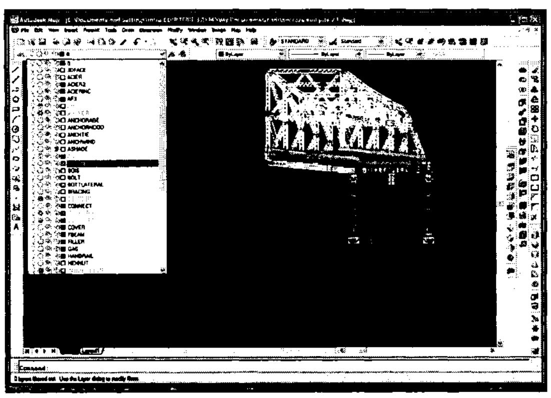
The Jacques Cartier Bridge in Montreal is chosen as the subject of this research. The Jacques Cartier Bridge is a five-lane bridge with a steel truss frame about 2.7 km in length, spanning the St. Lawrence River between the cities of Montreal and Longueuil (Figure 5.1) (Zaki and Mailhot 2003).

Figure 5.1 Main span of Jacques Cartier Bridge

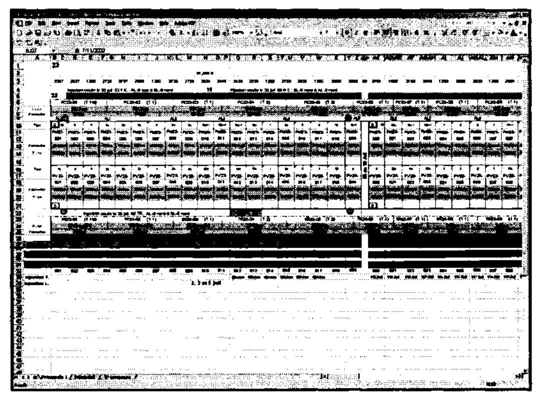
This bridge was originally named the "Harbor Bridge" at its inauguration on May 24, 1930. Then in 1934, the name was changed to the "Jacques Cartier Bridge" in tribute to the explorer who discovered Canada in 1534. In Montreal, Jacques Cartier Bridge is not only a means of transportation, but also a landmark of the city. In 1962, an estimated 18 million vehicles crossed the Jacques Cartier Bridge annually. With an increasing rate of 2.4% annually, now this bridge carries some 43 million vehicles every year, making it one of the busiest bridges in North America when considering traffic volumes per lane.

Over the last 70 years, the old reinforced concrete bridge deck had suffered seriously from the increase of the number and load of trucks and from the de-icing salts used extensively since the 1960s. Consequently, the deck was replaced in 2001-2002. The new deck is constructed of pre-cast, pre-stressed and post-tensioned panels made of high performance concrete which were prefabricated in a temporary plant installed near the south end of the bridge. Due to the spatial and temporal constraints, deck replacement had to be done during the night of weekdays from 8:30 p.m. to 5:30 a.m. The deck replacement involved six principal types of construction activities (Zaki and Mailhot 2003): (1) Steel work that included floor beam repair, strengthening work, and installation of new bearing assemblies to support the panels; (2) Precasting of new deck panels; (3) Removal of existing deck sections; (4) Installation of new panels; (5) Joint mortar placement and post-tensioning; and (6) Installation of expansion joint armors, cast-in-place expansion joint dams, and waterproofing and paving work. The existing deck was removed by using a saw to cut the deck into sections with dimensions similar to those of the new panels being installed. Each existing deck section, which included the slab, steel stringers, barriers and railings, was removed and a new panel (weighing between 22 and 42 tons) was lifted from a flat-bed semi-trailer truck and lowered onto the new bearing assemblies, which were installed by other crews working in advance during the day. At peak production, the contractor was able to replace eight panels per crew per night. A total of 1680 precast deck panels were installed, including 410 panels for the main span.

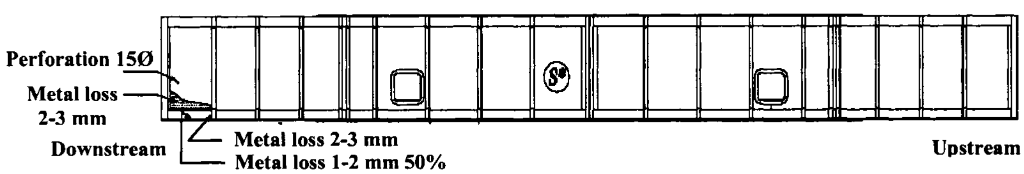
The bridge data were acquired from the bridge management authority (Jacques Cartier and Champlain Bridges Incorporated) (PJCCI 2004; Zaki and Mailhot 2003). The data include AutoCAD drawings, deck rehabilitation schedules, inspection records, and maintenance history data (Figure 5.2). The prototype system was developed based on these data. Several 3D models with different LoDs were created by converting the DWG file of the bridge into DXF (Data eXchange Format) and VRML (Virtual Reality Modeling Language) files and extracting the information about the geometry and topology of the bridge elements into our database. Data about the different stages of design, construction, rehabilitation, and inspection were stored in the database. In addition, the digital map and the DEM data of Montreal were acquired to generate 2D and 3D maps. The simulation model focused on the deck-rehabilitation project, and the task duration is shown in Table 4.1, including the two main activities: removing the old section, and installing the new panel. Pictures of the project are shown in Appendix C.



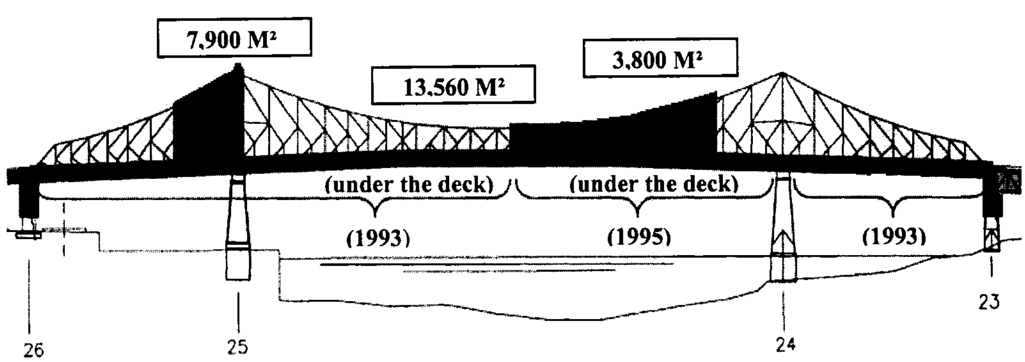
(a)



(b)



(c)



(d)

Figure 5.2 Data used in the case study: (a) Structure data; (b) Schedule data; (c) Inspection data; and (d) Maintenance history (painting of steel truss)

5.4 IMPLEMENTING THE 4D INTEGRATED MODEL

5.4.1 Structure of the prototype system

In order to easily group classes and interfaces that are related, the relevant classes and interfaces are organized in different packages. A package is a collection of related classes and interfaces providing access protection and namespace management. The general structure of the packages developed in the prototype system is illustrated in Figure 5.3. The *bridge3d.core* package creates the bridge shape including the LoDs representation of truss elements and deck. The *bridge3d.model* package includes the interfaces to the database. The *bridge3d.ui* package has the main GUI classes. The *gis* package has classes for manipulating 2D and 3D map objects. The *dem* package has classes to load and integrate the DEM file into the system.

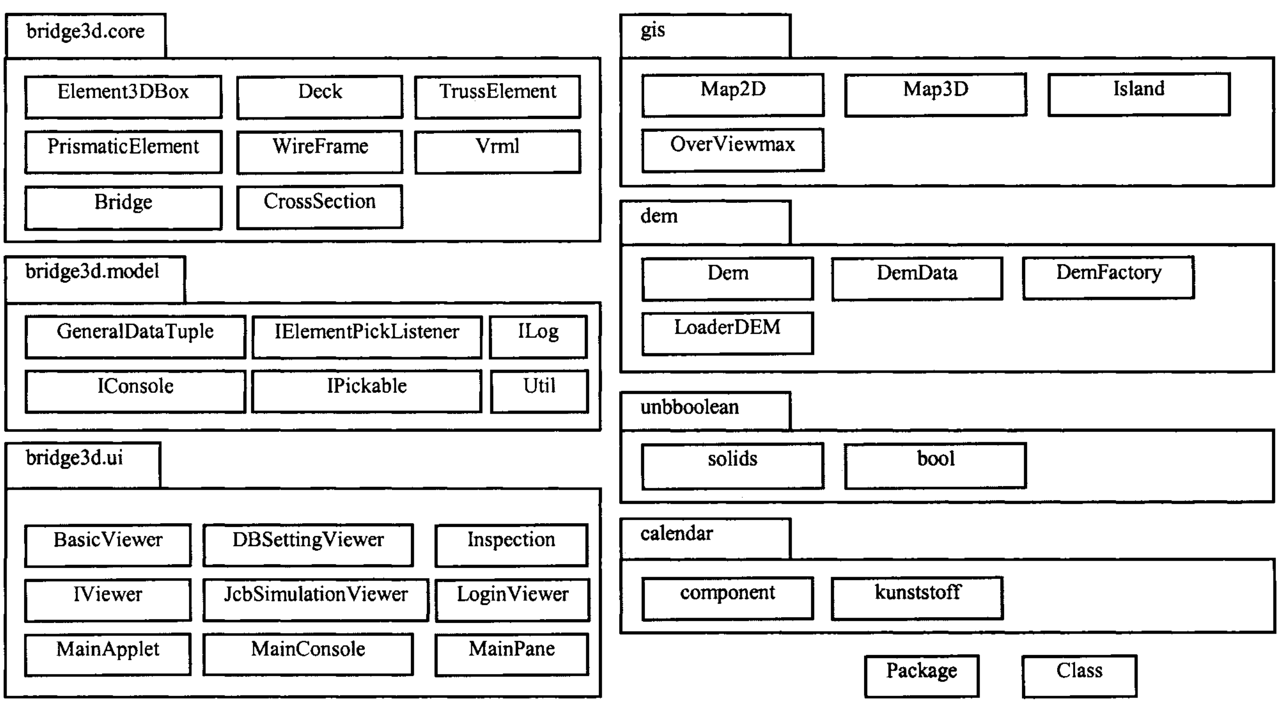


Figure 5.3 Structure of the packages developed in the prototype system

The *unbboolean* is the API to apply Boolean set operations for Java3D to create CSG shapes (CSG API 2004). The *calendar* package has classes for date input.

5.4.2 Creating the 3D prismatic element model of the main span

Virtual universes in Java 3D can be created from *scene graphs*. Scene graphs are assembled from objects to define geometry, location, orientation, and appearance of objects. Java 3D scene graphs are constructed from node objects using *BranchGroups* to form a tree structure based on parent-child relationships (Figure 5.4). *TransformGroup* objects can be constructed by applying *Transform3D* objects, which represent transformations of 3D geometry such as translations and rotations (Walesh and Gehinger 2001).

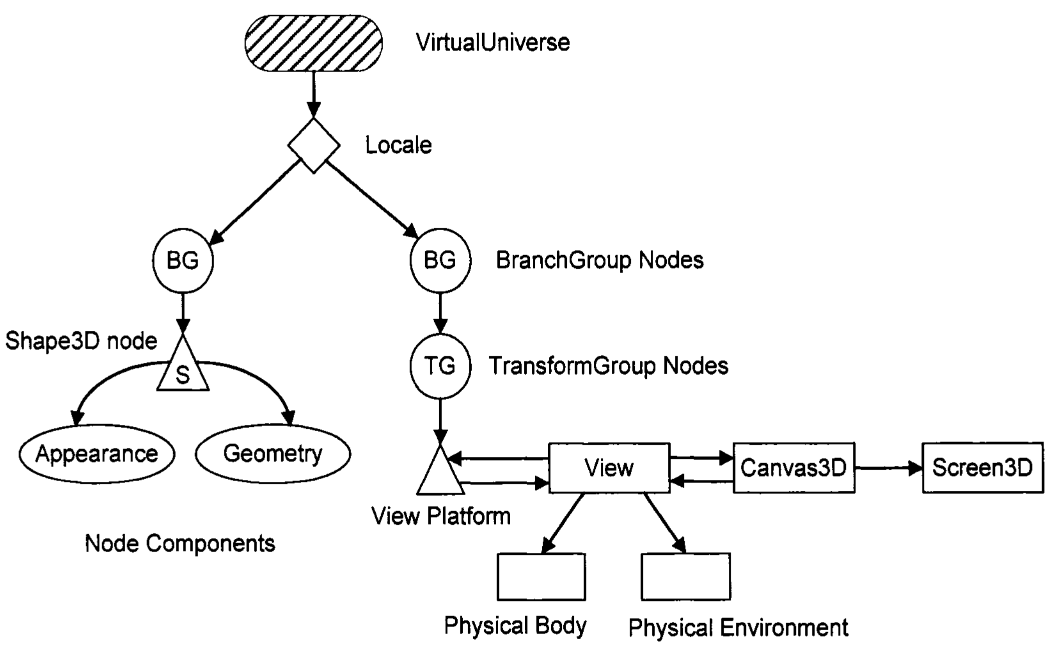


Figure 5.4 Scene graph (Walesh and Gehinger 2001)

The 3D model is built using Java 3D based on the CAD drawings of the main span of Jacques Cartier Bridge and other data about the original construction and re-decking schedules. Only the bridge truss and the deck panels of the main span are considered.

The 3D prismatic elements are first created at the origin based on the start point of an element. Then the coordinates of the eight vertices (*v0* to *v7*) are calculated based on the width, length, and the height of the element (Figure 5.5). Six faces of the prismatic element can be constructed using these vertices, e.g., the left face consists of *v0*, *v3*, *v4*, and *v2*. The local coordinates of the start point and the end point of an element are read from the DXF file, which can be used to calculate the rotation angle of the element in x-y plane and the translation value. The 3D prismatic elements are first created in the x-y plane by applying the transformation including the above translation and rotation to each element. Then, the translations and rotation in z direction are applied to the *BranchGroup* that contains these elements to locate the model in the 3D environment.

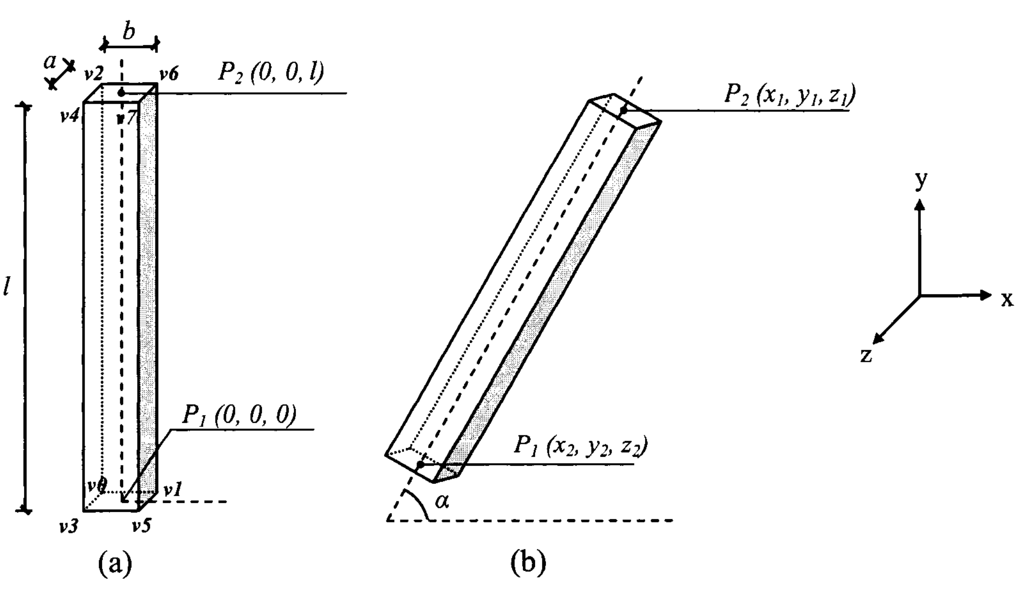


Figure 5.5 Prismatic element: (a) At the origin; and (b) After transformation

Different appearance, such as color, material, and texture, can be defined as attributes of each element. In addition, each element is assigned a unique ID equal to the ID of the member in the database and registered in a *HashMap*, which has a key and associated element. The *HashMap* allows the user to quickly find an existing element based on a key value. Following the method described above, the simplified 3D model of the main span is created (Figure 5.6).

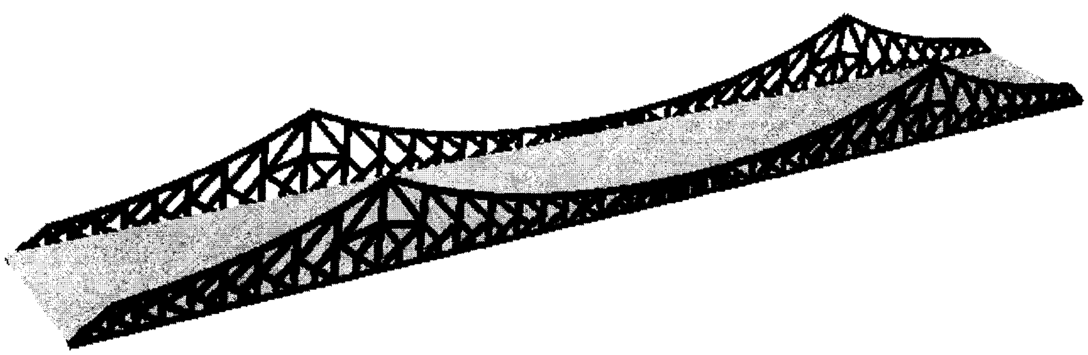


Figure 5.6 3D model of the main span

5.4.3 Creating different LoDs of the model

As discussed in Section 3.3, four levels of details are applied in the system, which are line, wire frame, prismatic elements, and VRML models. The line model is created by simply drawing a 2D line representing the axis of the main span using tow points. The wire frame model is created based on the topology of the main span truss extracted from the DXF file. The local coordinates of each truss element are read from the DXF file and 3D lines are generated in the 3D environment. Transformations including translation and rotation are applied to these lines to locate the model on the map. The prismatic elements model is constructed as discussed in the previous section. The VRML model is loaded using a VRML loader to import the bridge detailed model to the scene graph. The same transformations are applied to locate the model on the map. In addition, two floor beams: AL-1 and AL-2 supporting the deck are selected to show a parallel LoDs. Defects on a beam can be shown or not shown depending on the distance between the viewpoint and the center of the beam. The threshold distance is defined as 1.5 times of the length of the beam.

In Java3D, a class called *DistanceLOD* provides LoD behavior based on the distance from a graphic object to the viewer. Each LoD behavior has one or more *Switch* objects as a target. A *Switch* object is a special group that includes zero, one, or more of its children in the scene graph for rendering. With a *DistanceLOD* object, the selection of the child of the target *Switch* object is controlled by the distance of the *DistanceLOD* object to the view based on a set of threshold distances. The threshold distances are specified in an array beginning with the maximum distance for which the first child of the switch target(s) will be used. The first child is typically the most detailed visual object. When the distance from the *DistanceLOD* object to the view is greater than this threshold distance, the second child of the switch is used.

The whole structure of LoDs is shown in Figure 5.7. The target *Switch* object is a child of a *TransformGroup* object and referenced by the *DistanceLOD* object. LoD_1 is for the whole bridge, and LoD_2 and LoD_3 are for two beams. These three LoDs are used in parallel, and in this way, a complex visual model can be represented.

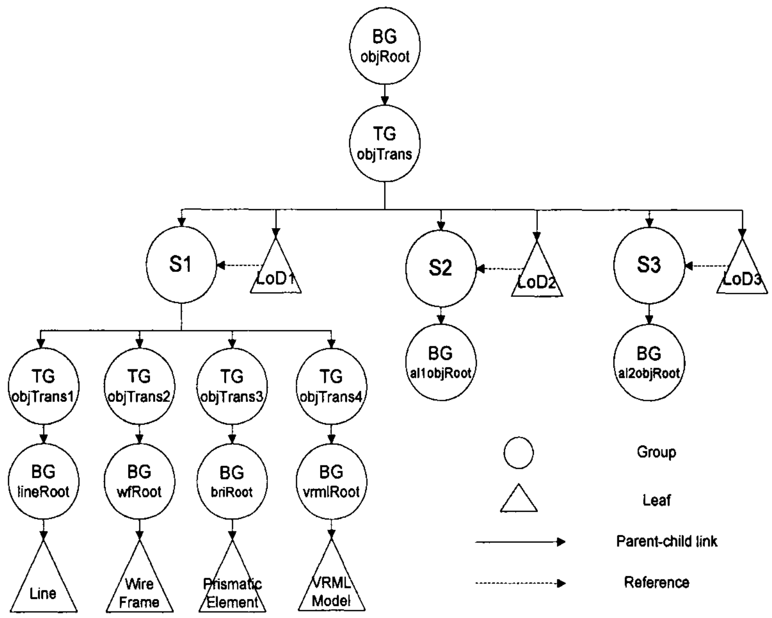
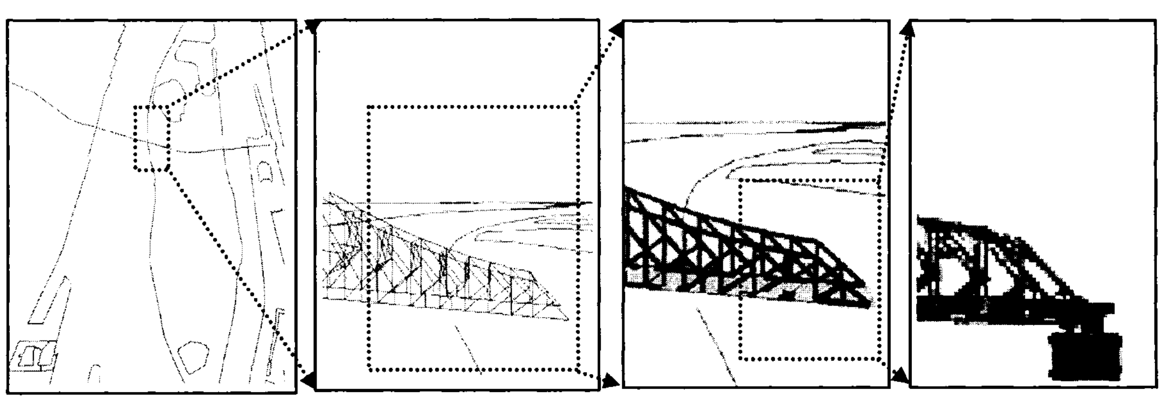


Figure 5.7 LoDs representation in Java 3D

As shown in Figure 5.8, when the viewpoint is far from the bridge, the user can see only one line representing the axis of the bridge. When the viewpoint comes nearer, the user can see the wire frame, prismatic elements and the detailed VRML objects, sequentially. At the same time, when the user moves near the location of the two beams AL-1 and AL-2, the defects on the beams will be shown or not shown depending on the distance, as shown in Figure 3.6.



(a) Line (axis of the bridge) (b) Wire frame (c) Prismatic elements (d) Detailed model

Figure 5.8 Different spatial LoDs of the bridge

A calendar and sliding bar interfaces are used to specify a date or a period of time, representing the temporal LoDs, to be used in a simulation (Figure 5.9). Different temporal LoDs are needed during construction and maintenance periods. The year or the specific date of the maintenance action can represent the time of maintenance. For example, the painting of the main span was done in several years as shown in Figure 5.2. The inspection time is usually represented by the date of inspection. Higher time resolution is used for inspection purposes to record defects that could happen in a very short time.

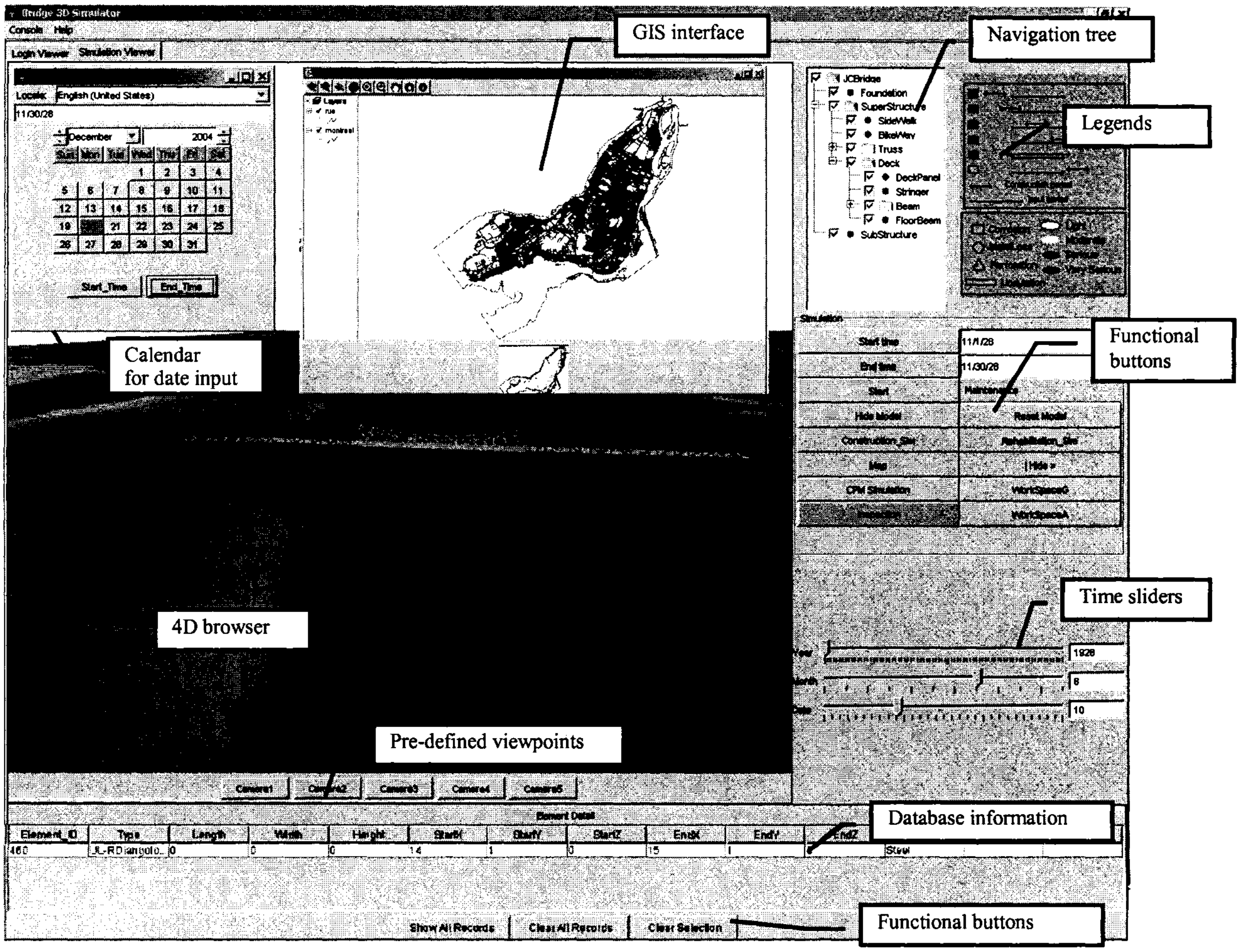


Figure 5.9 Screen shot of the user interface of the prototype system

5.4.4 GIS integration

A GIS sub-system is created using MapObjects Java Edition (ESRI, 2004). The map includes several layers related to Montreal City, such as a boundary layer and other layers for the roads, rivers, and administrative areas (Figure 5.10). The Modified Transverse Mercator Projection (MTM) was used because it is the standard projection used by the local government. In addition, the DEM model of the area was added to the scene graph.

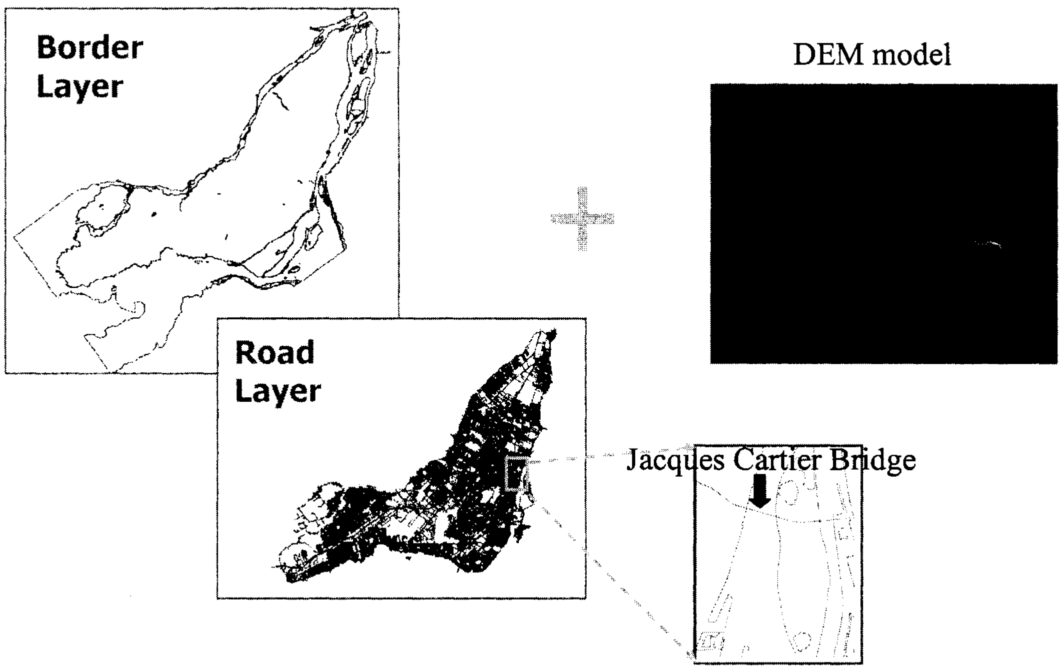


Figure 5.10 GIS information

The DEM data source is the Canadian Digital Elevation Data (CDED), which was introduced in Section 2.2.1. The CDED consists of an ordered array of ground elevations (recorded in meters) at regularly spaced intervals. The grid spacing is based on geographic coordinates at a maximum and minimum resolution of 0.75, 1.5 and 3 arc seconds for the scale of 1:50,000, and 3, 6 and 12 arc seconds for the scale of 1:250,000 respectively, depending on the latitude. In our prototype system, the 1: 50,000 CDED is used. For this scale, the grid spacing is always 0.75 arc seconds along a profile in the South-North direction and varies from 0.75 to 3 arc seconds in the West-East direction, depending upon the three geographic areas (Figure 5.11). Montreal is in the A area, where the grid spacing in the west-east direction is 0.75 arc seconds. To integrate this elevation information in our prototype system, the geographic coordinates have to be transferred to the world coordinates and match the 2D map of Montreal. Two CDED files, which cover the West and East parts of Montreal, are transferred to Triangulated Irregular Network (TIN) model, and then projected to Modified Transverse Mercator (MTM) file using the parameters from Table 5.1.

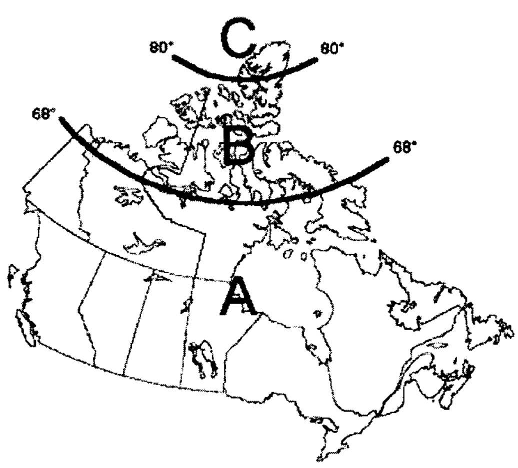


Figure 5.11 Coverage of the three geographic areas

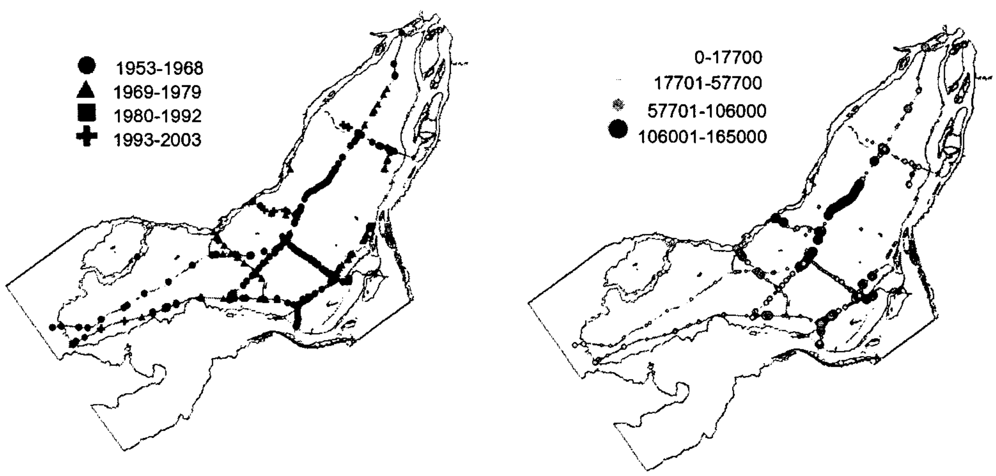
GEOGRAPHIC AREA	LATITUDE		SPACING (latitude and longitude in arc seconds)			SPACING (in metres, approximate)			CELL COVERAGE (latitude - longitude)		
	from	to	lat.		long.	N.-S.		E.-W.			
A	—	68°	0.75"	x	0.75"	23 m	x	16-11 m	15'	x	15'
B	68°	80°	0.75"	x	1.5"	23 m	x	17-8 m	15'	x	30'
C	80°	90°	0.75"	x	3"	23 m	x	17-8 m	15'	x	1°

Table 5.1 1:50,000 CDED cell coverage according to the three geographic areas

In addition, a color coding is defined to display different areas of the terrain according to the elevation values. Most of the elevation values of Montreal are below 100 and different ranges are defined to approximately represent the terrain following the color coding given in Dlgv32 pro software (2004).

5.4.5 User interface development

Following the GUI discussed in Section 3.5.1, the main user interface of the system is shown in Figure 5.9. On the left-hand side, a 4D browser is displayed to show the 3D environment and the bridge. The user can navigate the 3D bridge model and select an element of the bridge by picking that element using the mouse or other picking devices. Upon selection, the element will be highlighted and the related information about the element will be displayed. A logical tree of the bridge structure is shown on the right-hand side. Each tree node has a check box, which facilitates showing or not showing that element in the 3D model. Beside the navigation tree, legends are shown to explain the color coding. Under these two items, there is a time input text filed that allows the user to query the database about events that happened during a specific period. The start and end dates of a period can be input using the calendar interface, which is shown on the upper left corner. Several buttons are listed below the time input field to active functions, such as *Workspace Generation* and *Workspace Analysis* buttons. Three time sliders are displayed under the buttons, which can be used to get dynamic feedback from the 3D model. Some predefined viewpoints are displayed under the 4D browser, which can change the viewpoint and navigate the user to some specific location in the 3D environment. At the bottom of the interface, data retrieved form the database are displayed following the structure of the database. In addition, a GIS interface is created to show the 2D map of Montreal and some basic GIS functions are provided, such as showing the bridges of the area and defining different categories to group these bridges. As shown in Figure 5.12, 329 bridges in Montreal are displayed using different symbols according to the construction year and the traffic volume. It is easy to gain a direct visual understanding of this basic information.



(a) (b)

Figure 5.12 Bridge attributes distribution in Montreal:
(a) Construction year; and (b) Annual average daily traffic volume (vehicles/day)

5.4.6 Date information representation and querying

The date information can be input from the user interface and used to query the database for associated information. However, the types of date data used in the database and Java language are not exactly the same. The type used in the database is DATE, which has the format of mm/dd/yyyy. While in Java language, there are two types of date: *util.Date* and *sql.Date*. The *sql.Date* can be used to exchange information with the DATE from the database. When the user input the date information in the text field, it is read as a String.

A function named *parseDate* is created to transfer the String to *util.Date*, and then to *sql.Date* to compare with the date information stored in the database when executing the query.

To show different colors for each element depending on the input period, six conditions are defined to find corresponding elements. The user input start and end dates of a time period to know the construction stages for each element during this period. By comparing the information stored in the database of the construction activities with the query period, an element of the bridge can be categorized into six groups depending on its construction period as shown in Figure 5.13: (1) query period is after the construction period, (2) query start date is within construction period, (3) query period contains construction period, (4) query end date is within construction period, (5) construction period contains query period, and (6) construction period is after query period. Each group is assigned different color and refreshed in the scene graph. An example is shown in Figure 5.9. The same concept is used in the query generated using the three time sliders (year, month, and day). Every time a slider is moving one time step, a query will be executed and the colors of the members of the bridge change continuously.

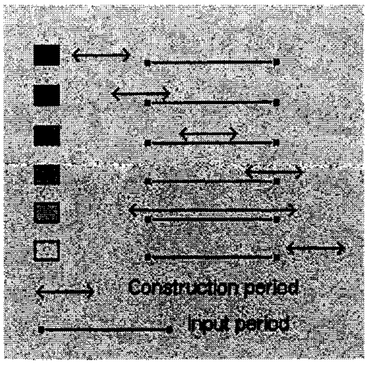


Figure 5.13 Legend of relationship between construction period and query period

5.5 WORKSPACE ANALYSIS OF BRIDGE RE-DECKING PROJECT

5.5.1 Description of the re-decking project

As mentioned in Section 5.3, the deck of Jacques Cartier Bridge has been replaced in 2001-2002. The deck replacement involved two main activities that were critical for the success of the project from the point of view of spatial and temporal constraints: (1) Removal of existing deck sections; and (2) Installation of new panels. These two activities are very similar from the workspace point-of-view. Therefore, it is enough to consider one of them to detect spatial conflicts. The purpose of workspace analysis is to help the manager select suitable equipment that satisfies spatio-temporal constraints.

Cranes are among the most important equipment used to lift old sections and install new panels. Low clearance below the cross-frames of the through truss was a major constraint in selecting the cranes. The contractor built a physical model of the bridge to check the conflicts that could appear when using different crane types (Figure 5.14 (c) and (d)). In the initial plan for replacing the deck panels, the contractor contemplated the usage of an industrial straddle crane (Shuttlelift SL80 Gantry Crane) with a maximum lifting capacity of 80 tons and an overall height, width and length of 8.13 m (26' 8"), 9.14 m (30') and 9.14 m (30'), respectively (Figure 5.14 (a)). It was found that two straddle cranes (used by two teams) cannot meet at the same location on the bridge because of the limited road width. In addition, a straddle crane would block the emergency lane, which is necessary at all times. Because of these spatial conflicts, the contractor decided to use 60.5-ton self-propelled telescopic cranes (Grove TR 700E Crane) instead of the straddle cranes (Figure

5.14 (b)). Furthermore, telescopic cranes are more available, easier to operate and to mobilize at the beginning and end of a shift.

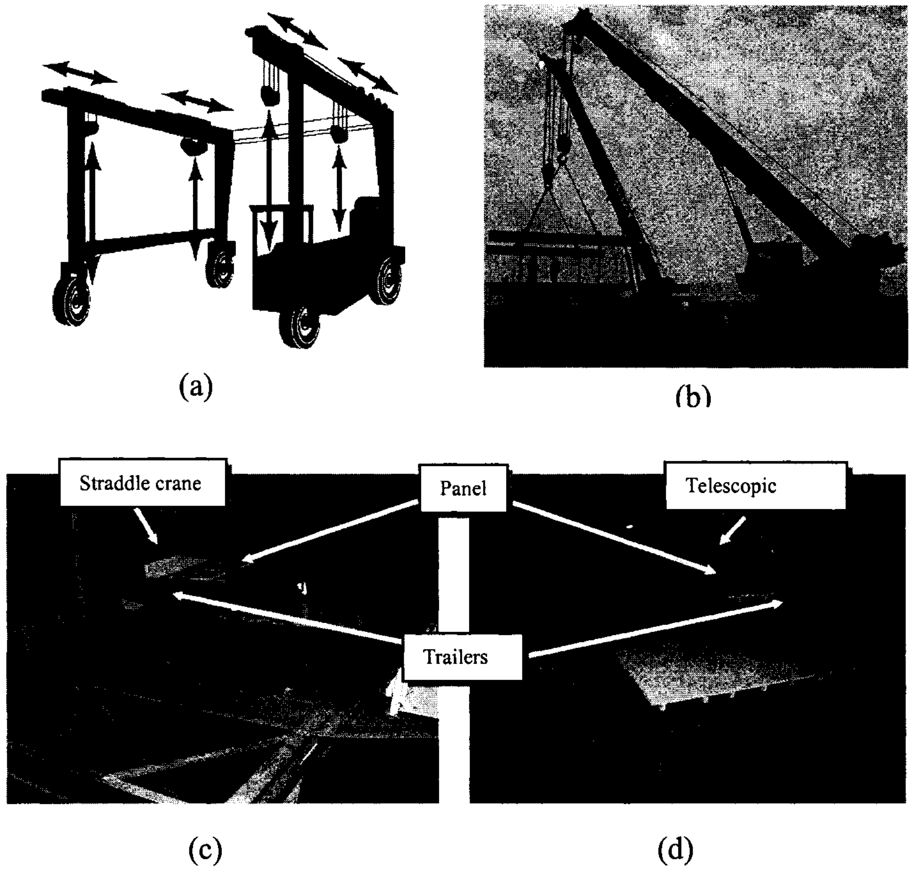


Figure 5.14 Cranes used in the case study: (a) Straddle crane; (b) Telescopic cranes; (c) and (d) Partial models of the bridge with straddle and telescopic cranes, respectively

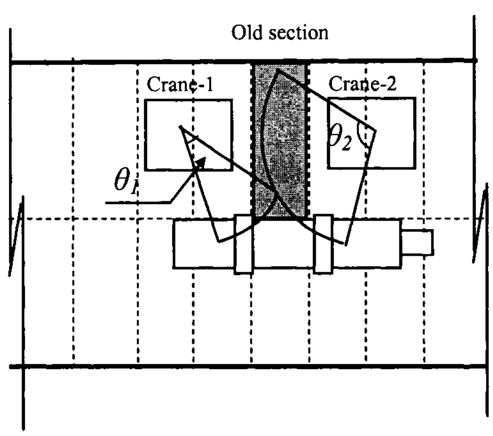
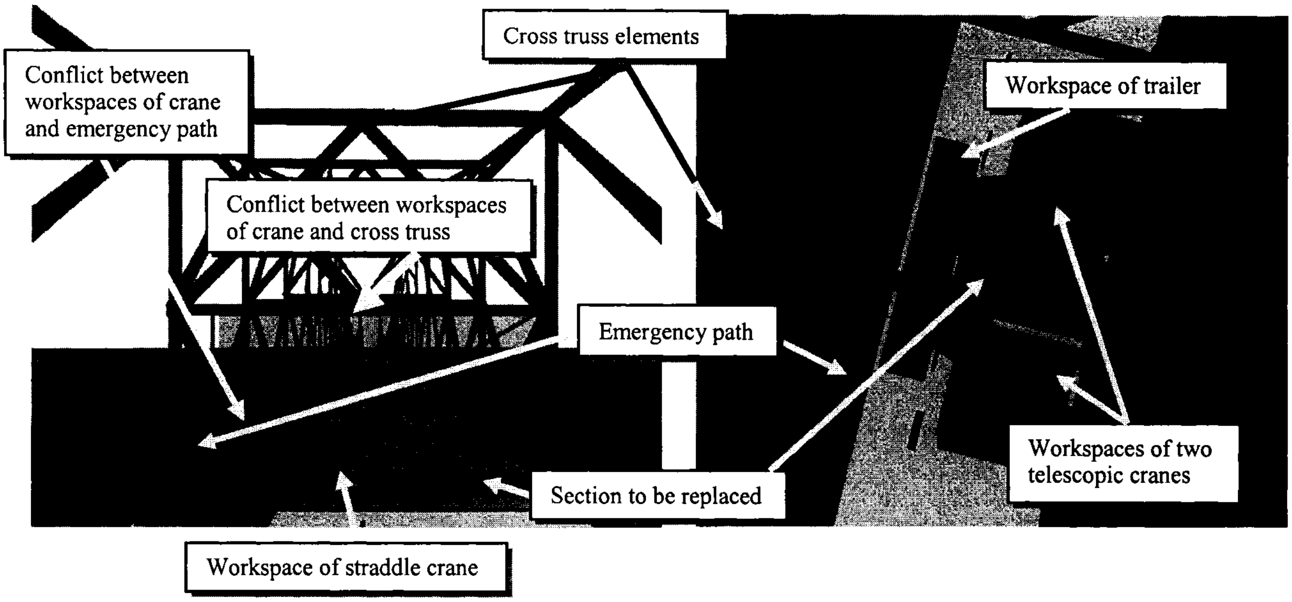


Figure 5.15 Locations of equipment in the case study

However, because of the low clearance problem, the crane had to work with a near-flat boom ($\alpha = 15°$), thereby, decreasing its lifting capacity by more than 50%. Consequently, this resulted in using two telescopic cranes instead of one to match the lifting requirements of the panels. Figure 5.15 shows the positions of the cranes and the trailer serving one team. Two telescopic cranes are positioned on both sides of the section to be replaced.

5.5.2 Conflict detection between workspaces

Workspaces are generated based on the procedure explained in Section 4.2.2. The workspaces include workspaces for cranes, trailers, and an emergency path. In the case of the straddle crane, when an old section is cut and removed by the crane, the empty area on the deck becomes a safety area and the crane moves to the other side across the road to load the old section on a trailer and then waits for unloading a new panel from another trailer (Figure 5.14 (c)). Therefore, the workspace of the straddle crane should cover the range of positions of the crane along the width of the bridge. The workspaces of the telescopic cranes are represented by composite shapes as shown in Figure 4.1. Both conflicts among workspaces and between workspaces and the components of the truss are automatically detected and analyzed. Figures 5.16 (a) and (b) show different workspaces using one straddle crane and two telescopic cranes, respectively. In Figure 5.16 (a), the workspace of the straddle crane and the emergency path are represented by two transparent boxes, and a conflict between them is detected. In addition, because of the low clearance of the main span (about 5.5 *m*), another conflict between the crane workspace and the cross truss elements is detected.



(a) (b)

Figure 5.16 Workspace conflicts detected in the case study:
(a) Straddle crane viewed from the road level; and
(b) Two telescopic cranes viewed from the top of the bridge

Figure 5.16 (b) shows a top view of the workspaces of the two telescopic cranes with near-flat booms on both sides of an old section, the trailer and the emergency path. Although there are overlapping parts between the workspaces of the cranes and the trailer, this overlap does not cause any real conflicts because the booms of the cranes always move above the trailer. No other conflicts are detected in this case between the workspaces and the components of the truss.

5.6 RE-DECKING SIMULATION USING CELL-BASED SIMULATION

A cell-based simulation model is developed using the same re-decking project. The old sections were transported to a dumping area near the bridge. The new panels were transported from the plant located at the south end of the bridge. Two types of semi-trailer trucks were used to transport the old sections and new panels. Most of the time, two teams worked in parallel on different parts of the bridge. Each team used two telescopic cranes located at both sides of the section to be replaced.

5.6.1 Structure of the cell-based model

Figure 5.17 shows the structure of the cell-based simulation model of the re-decking project of Jacques Cartier Bridge using CD++, a tool for cell-based discrete-event modeling and simulation based on DEVS formalism (Wainer 2002). This model is a combination of Cell-DEVS and DEVS models. Arrows show the input and output information flow between different models through ports. *Bridge*, *Plant*, and *Dump Area* are Cell-DEVS models. The following DEVS models are built to facilitate communications between these Cell-DEVS models: *Control Unit* model, *Reposition* model, and *Transport (T)* and *Queue (Q)* models.

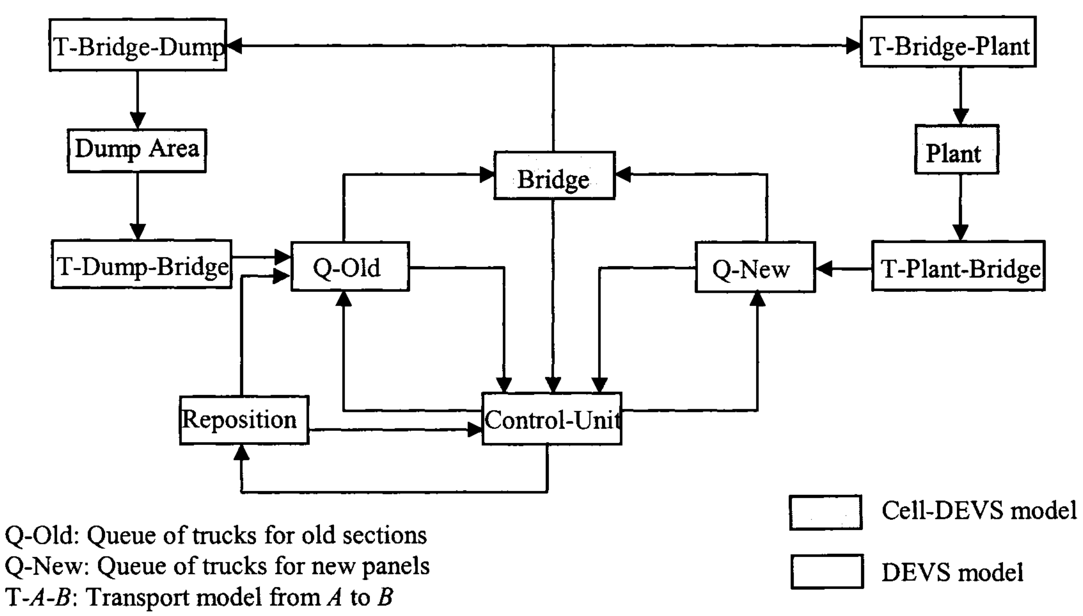


Figure 5.17 Structure of the cell-based model of Jacques Cartier Bridge re-decking project

For example, the *T-Plant-Bridge* is a *Transport* model representing the transportation of a panel from the plant to the bridge, and the *Q-Old* is a *Queue* model representing the queue of trucks that will carry the old sections. The *Control Unit* model is built to provide overall control of the system, such as permitting a queue to send a truck to the bridge when a truck is needed.

The computing functions of the cells are defined in CD++ using a set of rules with the form: VALUE DELAY {CONDITION}. This format indicates that when the CONDITION is satisfied, the state of the cell changes to the designated VALUE, and it is DELAYed for a specific time. To calculate the new value for a cell's state, the simulator takes each rule and evaluates the condition clause. If the condition is evaluated as true, then the action and delay clauses are evaluated. The result will be the new cell state and will be sent as an output after the delay. The time delays used in rules follow the duration of activities as given in Table 4.1. The above models are briefly discussed in the following sub-sections:

5.6.2 Cell-DEVS models

The discussion about the *Bridge* model given through examples in Section 4.3.2 covered most of the aspects of this model. The following simplifications are made: (1) A truck is represented by one cell (3×3m) while a telescopic crane is represented by three cells (Figure 4.10 (b)); and (2) The moving direction of all trucks on the bridge is always from east to west.

There are two input ports of the *Bridge*: *in-old* and *in-new*. The queues will send the ID number of each truck to the *Bridge* through the respective input port. The output port of the *Q-Old* is linked with the *in-old* port, while the *Q-New* is linked with the *in-new* port. Different layers act differently when receiving an ID number of a truck from a queue. The *occupancy layer* generates trucks; the *control layer* generates the directions; and the *ID layer* keeps the received value as it is and moves it with the truck.

There is one output port of the bridge, which is linked with *T-Bridge-Dump*, *T-Bridge-Plant*, and *Control-Unit* models to send the ID number of each truck. The ID numbers are initialized in the *Dump Area* or in the *Plant* models.

The techniques used in implementing the *Dump Area* and the *Plant* models are similar to those used in the *Bridge* model. About 100 rules were applied to implement all the functions of the Cell-DEVS models. It takes about 20 minutes to simulate a 9-hour construction period using this model on a computer with Pentium 4 CPU.

5.6.3 DEVS models

The DEVS models are defined using C++ language supported by the CD++ simulation software. The atomic models that were used are:

- Queues: *Q-Old*, *Q-New*
- Transport models: *T-Dump-Bridge*, *T-Plant-Bridge*, *T-Bridge-Dump*, *T-Bridge-Plant*
- *Control-Unit*
- *Reposition*

As an example of transport models, the *T-Dump-Bridge* model receives the trucks for carrying an old section from the *Dump Area* and keeps it with some time delay to represent the transportation time from the dump area to the bridge.

The *Control-Unit* model is the central part of the whole model. It communicates with other models and checks the need for trucks and then decides when to send a truck to the bridge. When the *Control-Unit* receives a truck from the *Q-Old*, it will check if a truck for carrying an old section is needed. In that case, it will send a signal to the *Q-Old* to send a truck to the bridge. The *Q-Old* will send a truck to the bridge when the queue is not empty. Similarly, when the *Control-Unit* model receives a truck from the *Q-New*, it will check if a truck for carrying a new panel is needed. In that case, it will send a signal to the *Q-New* to send a truck to the bridge. When the *Control-Unit* model receives a signal from the *Bridge*, it will check the type of the truck. If the truck is carrying an old section, which means there is an empty space available on the bridge, it will send a signal to the *Q-New* to send a truck for carrying a new panel to the bridge provided that the queue is not empty. If the signal from the *Bridge* is an empty truck that finished unloading a new panel, this means that the work cycle of a team replacing an old section with a new panel has been finished. In this case, the *Control-Unit* will send a signal to the *Reposition* model to simulate the delay of the time needed to move the equipment to the next old section to be replaced. After the time delay, the *Reposition* model will send a signal to *Q-Old* asking for a truck. At the same time, a signal will go to the *Control-Unit* to increase the number of the old sections available on the bridge. In addition, the

Control-Unit is used to accumulate the number of the removed old sections and the installed new panels and to calculate the productivity of the work.

We did the simulation of the Jacques Cartier Bridge re-decking case study using the following resources based on the data from the real project: (1) two teams, (2) two trucks for carrying old sections, and (3) two trucks for carrying new panels. The locations of the teams are initialized in the *Bridge* model, and the numbers of trucks are initialized in the *Dump Area* and the *Plant* model. The construction time that has been simulated is 9 hours, which is the real construction period from 8:30 p.m. to 5:30 a.m. the next day. The source code of the model is shown in Appendix D. The result of the simulation shows that 18 panels have been installed during this period, a number which is the same as the result obtained from MicroCYCLONE. This similarity could be explained by the simplicity of the spatial aspects of the case study in which the explicit representation of spatial constraints in the cell-based simulation did not affect the result.

The work site layout can be displayed at every time step to show the space occupancy in the *Bridge* and other Cell-DEVS models. Figure 5.18 shows part of the *occupancy layer* of the *Bridge* model, where two teams are working on the bridge in parallel. Figures 5.18 (a) and (b) show that *Truck-1* is loading an old section, while *Truck-2* is coming to unload a new panel at another location. Figure 5.18 (c) shows *Truck-1* carrying the old section to the dump area. Figure 5.18 (d) shows *Truck-3* coming with a new panel to be installed. Figure 5.18 (e) shows *Truck-3* after it changed its direction to avoid *Truck-2*.

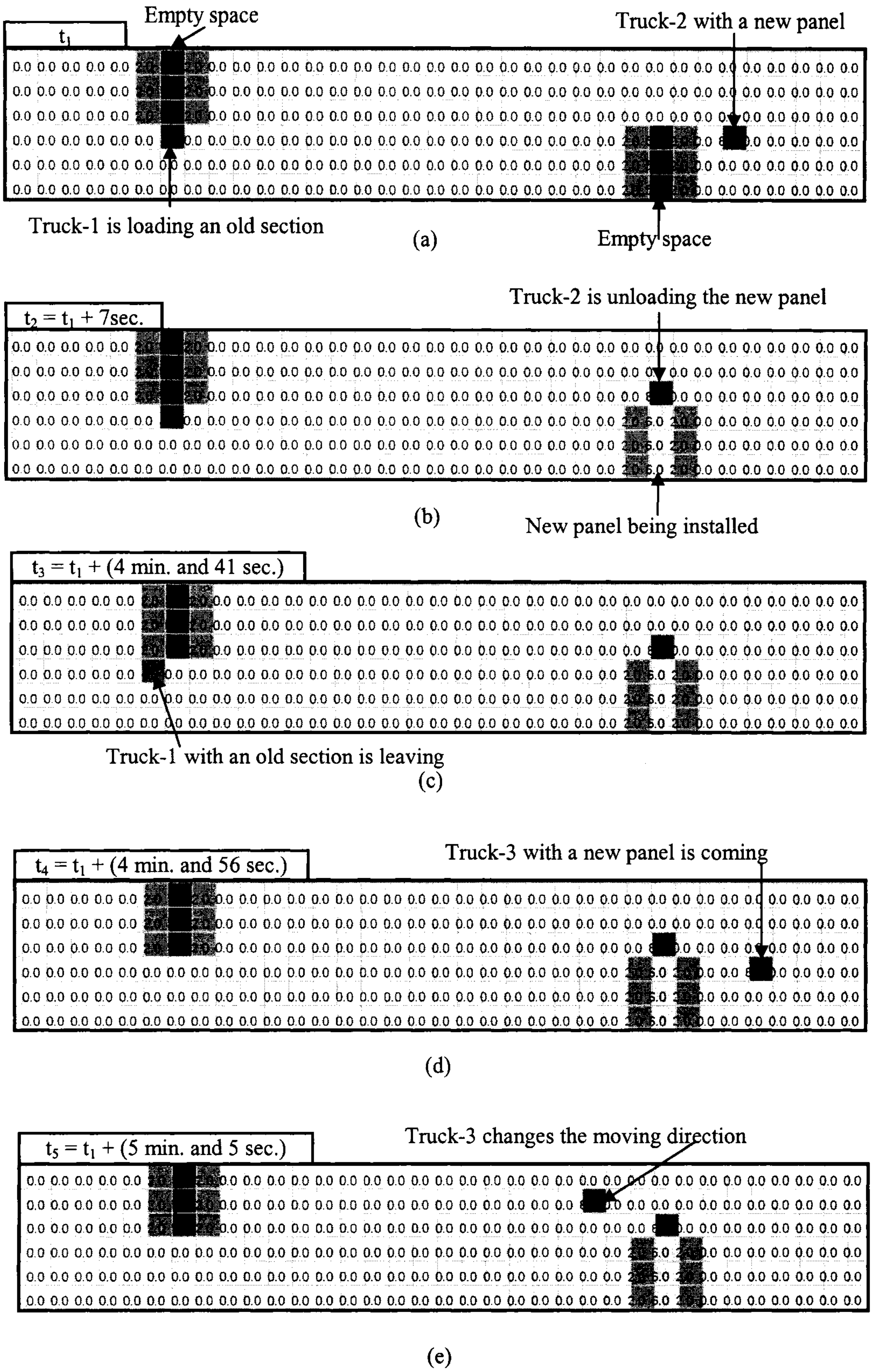


Figure 5.18 Snap shot showing part of the *occupancy layer* of the *Bridge* model

5.7 SUMMARY AND CONCLUSIONS

This chapter described the implementation of the proposed approaches and methods discussed in Chapters 3 and 4. A prototype system is developed and a case study of Jacques Cartier Bridge is used to demonstrate the feasibility of these approaches and methods. The software development tools are selected to integrate several information technologies in the prototype system. The cell-based model is developed using CD++ software based on the same case study.

Spatio-temporal visualization is facilitated by creating a 4D model of the bridge with different LoDs. Information retrieved form the structure drawings and lifecycle data, such as the construction, inspection, and the maintenance data were stored in the database and shown on the 4D model through querying the database. Based on the spatial model of the structure, virtual workspaces were generated using composite shapes. Compared with prismatic shapes, more realistic representation was achieved by combining working envelopes of each part of the equipment. Detailed conflict detection was applied to these composite shapes to find the overlapped parts for further conflict analysis. In the case of Jacques Cartier Bridge re-decking project, workspace representation and analysis helped selecting the equipment (cranes) used in the project. The prototype system was demonstrated to engineers responsible of the bridge management and they gave positive evaluation. Furthermore, the preliminary testing of the system and its user interface showed that it has good potential for realizing future BMSs because it was carefully designed and implemented to satisfy the specific requirements of these systems.

Based on the same re-decking project, a cell-based model is built to simulate the activities of the re-decking project and to realize dynamic conflict detection. Spatial resources are explicitly represented in this model using cells. Spatial models, such as the *Bridge*, *Plant*, and *Dump Area*, were created as Cell-DEVS models; while non-spatial models, such as *Transportation*, were created as DEVS models. Spatial conflicts were avoided by developing rules controlling the equipment movement on the spatial models. Accurate timing can be obtained from the simulation because it is based on a detailed spatial model of the site. The case study demonstrated the feasibility of cell-based construction simulation. In spite of the present limitations of cell-based simulation for practical uses, it has several advantages over conventional tools in representing spatial constraints and detecting spatial conflicts.

In summary, spatio-temporal visualization and analysis is useful in helping the project manager easily understand the work environment, finding spatial conflicts for specific activities within certain period, selecting suitable equipment to avoid spatio-temporal conflicts, and obtaining possible resource combination considering spatio-temporal constraints.

CHAPTER 6 SUMMARY, CONCLUSIONS AND FUTURE WORK

6.1 SUMMARY

This thesis investigated several spatio-temporal visualization and analysis issues in IMSs. Conventional IMSs provide only limited support for representing, visualizing and analyzing the spatio-temporal relationships throughout the lifecycle of the infrastructure. This research proposed new methods that integrate 4D modeling with several information technologies to facilitate spatio-temporal visualization and analysis. BMSs were used as an example to realize the proposed new approaches. A 4D bridge model was integrated with a GIS to represent accurate spatial information, thus making the results of the spatio-temporal analysis more reliable and understandable. All the information about the lifecycle of a bridge incorporating different levels of details of space and time was linked with the model to realize the proposed framework. In addition, many information technologies were integrated into the 4D model to investigate the benefits that can be achieved from these advanced tools.

Two spatio-temporal analysis approaches were proposed in this research: workspaces representation and analysis for conflict detection, and cell-based simulation of construction processes. The first approach focused on representing workspaces in a more realistic way using composite shapes. Different types of workspaces for crew, equipment, and other required spaces in the work site are created to detect conflicts between these workspaces. This approach extends the previous research on workspace analysis in the case of large infrastructure projects focusing on the representation of workspaces related to heavy equipment such as cranes, and the computational methods to define the composite shapes of these workspaces using CSG. Conflict detection is based on tasks executed during a specific time period. The second approach investigated the possibility of a cell-based simulation to further analyze the spatio-temporal conflicts and to dynamically achieve the automatic detection of these conflicts during construction. Based on the cell representation of the spatial model, the information of the construction environment can be understood more easily and the optimal resources combination can be found not only based on resource constraints, but also the availability of workspaces, which is one of the main reasons that cause delays. Spatio-temporal conflicts can be avoided by applying rules to control the movement of equipment and other objects.

A case study of Jacques Cartier Bridge in Montreal was used to develop a prototype system and demonstrate the applications using the new proposed approaches. Part of the lifecycle data of this bridge were stored in the database and linked with the prototype system using querying techniques. The spatio-temporal analysis was applied to investigate the problems that could have occurred during the re-decking project of the bridge. The preliminary testing of both approaches of workspace analysis and cell-based modeling showed good potential to integrate them in future BMSs.

6.2 CONCLUSIONS

The conclusions of this research are grouped into the following three areas:

(1) A framework of BMS was developed including creating an object-relational data model, technology integration, and applications development. Several computational

issues necessary to realize the framework were also discussed, such as navigation, picking behavior and LoDs. This approach makes the first attempt to integrate 4D bridge models with a BMS making the information accessible to mobile on-site workers. The prototype system, which was developed based on the framework, was demonstrated to engineers responsible of the bridge management and they gave positive evaluation. Furthermore, the preliminary testing of the system and its user interface showed that it has good potential for realizing future BMSs because it was carefully designed and implemented to satisfy the specific requirements of these systems.

(2) Composite shapes with different levels of complexity can be used to represent workspaces of equipment using CSG techniques. The proposed semiautomatic procedure can integrate the attributes of a construction activity, the attributes of the equipment used in this activity, and the geometry information retrieved from a 3D model of the project to generate the required workspaces, and then to apply conflict detection algorithms on them. The prototype system demonstrated the feasibility of the proposed approach in a case study about the deck replacement of a through truss bridge.

(3) Applying cell-based modeling method to simulate the work processes is the first step in investigating this method in construction. The implicit representation of space using conventional simulation tools makes it difficult to check availability of workspaces, thus the spatial conflicts could not be detected during the simulation. In contrast, cell-based simulation can be used as a general method to represent the worksite layout and to analyze and visualize the activities that may be affected by the

space constraints, and thus the spatial conflicts that may happen during construction can be detected. The proposed general procedure of developing cell-based construction simulation models showed the specific steps for defining layers and conflict detection rules and how these steps can be used in simulating construction activities. In spite of the present limitations of cell-based simulation for practical usages, it has several advantages over conventional tools in representing spatial constraints and detecting spatio-temporal conflicts.

6.3 LIMITATIONS AND FUTURE WORK

While pursuing this research, several limitations have been identified related to the requirements and the performance of the developed methods and techniques.

(1) The usability of the prototype system for BMSs needs more testing. Further development and testing in practical situations are necessary to improve the functionalities and usability of the system.

(2) The workspace representation is simplified and can be more accurate by introducing other parameters, such as the maximum and minimum radii of the crane. Moreover, the conflict detection for the CSG shapes is time consuming (several minutes) and is not suitable to detect real time conflicts.

(3) The feasibility of using the cell-based approach for site layout planning should be investigated including the creation of the rules that control the behavior of the cells (e.g., mobility rules, conflict resolution rules, etc.). There is no formal way to develop these rules in complex situations. Checking the consistency of rules is difficult, especially when the number of rules increases. For example, the current

model of the bridge used in this study includes only the main span without considering the approaches, which has more complex geometry (curves).

(4) One of the most challenging problems in this research is to resolve spatio-temporal conflicts. For example, after detecting spatial conflicts by the simulation system, these conflicts should be resolved by changing the location or the schedule of one or more activities that are involved in the conflict.

Future work will focus on representing the workspaces of more complex equipment configurations and on resolving detected conflicts based on the information in the workspace and conflict database. Furthermore, additional development is needed to support the workspace generation of different types of equipment so that the system can be used in engineering practice.

The proposed methodology for overcoming these problems is based on using Artificial Intelligence (AI) techniques, such as expert systems, neural networks, fuzzy logic, and multi-agent systems. An increasing volume of research has combined AI and simulation. However, applying AI in solving spatio-temporal conflicts in construction projects is still an emerging research area.

Another extension of the present research of spatio-temporal issues is the investigation of the spatio-temporal interactions among the equipment and crew members in a construction site. In this case, the locations of each equipment or worker can be tracked in real time using a suitable tracking method (e.g., GPS, radio frequency identification tags, etc.), and these locations can be effectively used to track the progress of the work or to predict and avoid conflicts. Agents can be created to represent every equipment, worker, and other resources, and to communicate real-time information and negotiate conditions about tasks, locations, etc., in a multi-agent environment.

REFERENCES

AbouRizk, S.M., Ruwanpura, J.Y., Er, K.C., and Fernando, S. (1999). Special Purpose Simulation Template for Utility Tunnel Construction, In Proceedings of the 1999 Winter Simulation Conference, pp. 948-955.

Agile Bridge Manager 2005 [online]. Available from http://www.trdi.com/ [cited on November 16, 2005].

Ahmad, S., and Simonovic, S.P. (2004). Spatial System Dynamics: New Approach for Simulation of Water Resources Systems, Journal of Computing in Civil Engineering, ASCE, Vol. 18, No. 4, pp. 331-340.

Akbas, R., and Fischer, M. (2002). Construction Zone Generation Mechanisms and Applications, In Proceedings of International Symposium on Automation and Robotics in Construction, 19th (ISARC), Gaithersburg, Maryland, pp. 293-298.

Akenine-Moller, T., and Haines, E., (2002). Real-Time Rendering, pp. 382-392, [online]. Available from http://www.realtimerendering.com/book.html [cited on August 8, 2004].

Akinci, B., Fischer, M., Kunz, J., and Levitt, R. (2002a). Representing Work Spaces Generically in Construction Method Models, Journal of Construction Engineering and Management, ASCE, Vol. 128, No. 4, pp. 296-305.

Akinci, B., Fischer, M., Levitt R., and Carlson, R. (2002b). Formalization and Automation of Time-Space Conflict Analysis, Journal of Computing in Civil Engineering, ASCE, Vol. 16, No. 2, pp. 124-134.

Al-Hussein, M. (1999). An Integrated Information System for Crane Selection and Utilization, Ph.D. Thesis, Dept. of Building Civil & Environmental Engineering, Concordia University.

Al-Hussein, M., Alkass, S., and Moselhi, O. (2001). An Algorithm for Mobile Crane Selection and Location on Construction Sites, Construction Innovation Journal, Vol. 1, No. 2, pp. 92-105.

Apple Computer (2003). Quicktime player [online]. Available from http://www.apple.com/quicktime [cited on November 16, 2005].

ArcView (1996). Using the ArcView Spatial Analyst, Environmental Systems Research Institute, Inc.

Beadle, H.W.P., Harper, B., Maguire, G.Q., and Judge, J. (1997). Location Aware Mobile Computing, In Proceedings of IEEE/IEE International Conference on Telecommunications, (ICT'97), Melbourne, pp. 1319-1324.

Bentley 2005 [online]. Available from http://www.bentley.com/en-us/ [cited on January 8, 2005].

Canada3D Product Standard (2001). Centre for Topographic Information [online]. Available from http://www.cits.rncan.gc.ca/ [cited on November 10, 2005].

Cheng, M.Y, and Yang, S.C. (2001). GIS-Based Cost Estimates Integrating with Material Layout Planning, Journal of Construction Engineering and Management, ASCE, Vol. 127, No. 4, pp. 291-299.

CSG API 2004 [online]. Available from http://www.geocities.com/danbalby/ [cited on October 18, 2005].

Davidson, A. and Wainer, G. (2005). ATLAS: a Specification Language for Traffic Modelling and Simulation, Simulation, Practice and Experience, Elsevier. Accepted for publication: July 2005.

Dlgv32 pro 2004 [online]. Available from http://mcmcweb.er.usgs.gov/drc/dlgv32pro/ [cited on October 18, 2005].

Do, E.Y., and Gross, M.D. (1997). Tools for Visual and Spatial Analysis of CAD Models, Computer Assisted Architectural Design Futures 1997, Kluwer Academic Publishers, Dordrecht, pp. 189-202.

Elbeltagi, E., Hegazy, T., and Eldosouky, A. (2004). Dynamic Layout of Construction Temporary Facilities Considering Safety, Journal of Construction Engineering and Management, ASCE, Vol. 130, No. 4, pp. 534-541.

El-Rayes, K. and Khalafallah, A. (2005). Trade-off between Safety and Cost in Planning Construction Site Layout, Journal of Construction Engineering and Management, ASCE, Vol. 131, No. 11, pp. 1186-1195.

Emerging Construction Technologies 2005 [online]. Available from http://www.new-technologies.org [cited on October 18, 2005].

ESRI 2004, Getting Started with MapObjects-Java [online]. Available from http://www.esri.com/software/mojava/download.html [cited on October 18, 2005].

Fischer, M. (2001). Introduction to 4D Research [online]. Available from http://www.stanford.edu/group/4D [cited on October 18, 2005].

Fischer, M. and Kam, C. (2002). PM4D Final Report, CIFE Technical Report Number 143, Stanford University.

Frangopol, D.M., Kong, J.S., and Gharaibeh, E.S. (2001). Reliability-Based Life-Cycle Management of Highway Bridges, Journal of Computing in Civil Engineering, ACSE, Vol. 15, No. 1, pp. 27-34.

GIS 2005 [online]. Available from http://www.gis.com [cited on October 18, 2005].

Gracanin, D., and Collura, J. (2001). Virtual Reality Based Interface for Simulation and Evaluation of Airport APMs, In Proceedings of the 8th International Conference on Automated People Movers, San Francisco, USA.

Groger G., Reuter M., and Plumer L. (2004). Representation of 3-D City Model in Spatial Object-Relational Databases, In Proceedings of Congress of International Society for Photogrammetry and Remote Sensing, Istanbul, Turkey, pp.13-18.

Guo, S. (2002). Identification and Resolution of Workspace Conflicts in Building Construction, Journal of Construction Engineering and Management, ASCE, Vol. 128, No. 4, pp. 287-295.

Hajjar, D., and AbouRizk, S.M. (1999). Simphony: an Environment for Building Special Purpose Construction Simulation Tools, In Proceedings of the 1999 Winter Simulation Conference, pp. 998-1006.

Hajjar, D., and AbouRizk, S.M. (2002). Unified Modeling Methodology for Construction Simulation, Journal of Construction Engineering and Management, ASCE, Vol. 128, No. 2, pp. 174-185.

Halpin, D.W. (1977). CYCLONE: Method for Modeling of Job Site Processes, Journal of the Construction Division, ASCE, Vol. 103, No. 3, pp. 489-499.

Halpin, D.W., and Riggs, L.S. (1992). Planning and Analysis of Construction Operations, Wiley Interscience, New York, 1992.

Hamiani, A. (1989). Knowledge Representation for the Site Layout Problem, In Proceedings of the 6th Conference on Computing in Civil Engineering, New York, pp. 283-289.

Hammad, A., Zhang, C., Hu, Y. and Mozaffari, E. (2005). Mobile Model-Based Bridge Lifecycle Management Systems, Journal of Computer-Aided Civil and Infrastructure Engineering, Blackwell Publishing (accepted in June, 2005).

Heesom, D., Mahdjoubi, L., and Proverds, D. (2003). A Dynamic VR System for Visualizing Construction Space Usage, In Proceedings of Construction Research Congress, Honolulu, Hawaii.

Hegazy, T.M., and Elbeltagi, E. (1999). EvoSite: Evolution-Based Model for Site Layout Planning, Journal of Computing in Civil Engineering, Vol. 13, No. 3, pp. 198-206.

Horstmann, C. (2004). Big Java, 2nd edition, John Wiley & Sons, Inc.

Hudson, W.R., Haas, R., and Uddin, W. (1997). Visual IMS: A Working Infrastructure Management System and Applications, Infrastructure Management, McGraw-Hill, pp. 345-364.

IAI 2004 [online]. Available from http://www.iai-international.org/iai_international [cited on February 7, 2005].

IFC-BRIDGE 2004. [online]. Available from http://www.iai-france.org/bridge [cited on July 18, 2004].

IFG 2005 [online]. Available from http://www.iai.no/ifg/ [cited on October 28, 2005].

Infrastructure Canada 2002: 2003-2004 Estimates, Report on Plans and Priorities (RPP).

ISO 2005. [online]. Available from http://www.iso.org/iso/en/aboutiso/introduction [cited on November 16, 2005].

ISO 10303-225 (1999). Industrial Automation Systems and Integration – Product Data Representation and Exchange – Part 225: Application Protocol: Building Elements Using Explicit Shape Representation, ISO.

Itoh, Y., Hammad, A., Liu, C., Shintoku, Y. (1997). Network-Level Bridge Life-Cycle Management System, Journal of Infrastructure Systems, ASCE, Vol. 3, No. 1, pp. 31-39.

Jauregui, D.V., White, K.R., Pate, J.W., and Woodward, C.B. (2005). Documentation of Bridge Inspection Projects Using Virtual Reality Approach, Journal of Infrastructure Systems, ASCE, Vol. 11, No. 3, pp. 172- 179.

Kale, S. and Arditi, D. (2005). Diffusion of Computer Aided Design Technology in Architectural Design Practice, Journal of Construction Engineering and Management, ASCE, Vol. 131, No. 10, pp. 1135-1141.

Kamat, V.R. (2001). Visualizing Simulated Construction Operations in 3D, Journal of Computing in Civil Engineering, ASCE, Vol. 15, No. 4, pp. 329-337.

Kamat, V. R. (2003). VITASCOPE – Extensible and Scalable 3D Visualization of Simulated Construction Operations, Doctoral dissertation, Department of Civil Engineering, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

Kamat, V.R., and Martinez, J.C. (2004). Practical 3D Animation of Multiply Articulated Construction Equipment, In Proceedings of the 2004 Winter Simulation Conference, pp.1229-1237.

Kerzner, H. (1995). Project Management, a System Approach to Planning Scheduling and Controlling, fifth edition, Wiley.

Koo, B., and Fischer, M. (2000). Feasibility Study of 4D CAD in Commercial Construction, Journal of Construction Engineering and Management, ASCE, Vol. 126, No. 4, pp. 251-260.

Kumara, S. R. T., Kashyap, R. L., and Moodie, C. L. (1988). Application of Expert Systems and Pattern Recognition Methodologies to Facilities Layout Planning, International Journal Production Research, Vol. 26, No. 5, pp. 905-930.

Kunigahalli, R., Russell, J.S., and Veeramani, D. (1995). Extracting Topological Relationship from Wire-Frame CAD Model, Journal of Computing in Civil Engineering, ASCE, Vol. 9, No. 1, pp. 29-42.

Li, H., and Love, P.E.D. (1998). Site-level Facilities Layout Using Genetic Algorithms, Journal of Computing in Civil Engineering, ASCE, Vol. 12, No. 4, pp. 227-231.

Liston, K., Fischer, M., and Kunz, J. (2000). Designing and Evaluating Visualization Techniques for Construction Planning, In Proceedings of the Eighth International Conference, Stanford, California, pp. 1293-1300.

Mallasi, Z., and Dawood, N. (2004). Workspace Competition: Assignment, and Quantification Utilizing 4D Visualization Tools, In Proceedings of Conference on Construction Application of Virtual Reality, ADETTI/ISCTE, Lisbon, pp.13-22.

MapObjects-Java 2005 [online]. Available from http://www.esri.com/software/mojava [cited on October 18, 2005].

Marasini, R., and Dawood, N. (2002). Simulation Modeling and Optimization of Stockyard Layout for Precast Concrete Products, In Proceedings of the 2002 Winter Simulation Conference, pp. 1731-1736.

Martinez, J.C. (1998). EZStrobe - General-Purpose Simulation System Based on Activity Cycle Diagrams, In Proceedings of the 1998 Winter Simulation Conference, Washington, D.C., pp. 341-348.

Martinez, J. C., Trani, A., and Ioannou, P. (2001). Modeling Air-Side Airport Operations Using General-Purpose Activity-Based Discrete-Event Simulation, Transportation Research Record: Journal of the Transportation Research Board 1744, pp. 65-71.

Mawdesley, M.J., Al-jibouri, S.H., and Yang H. (2002). Genetic Algorithms for Construction Site Layout in Project Planning, Journal of Construction Engineering and Management, ASCE, Vol. 128, No. 5, pp. 418-426.

Miles, S.B., and Ho, C.L. (1999). Applications and Issues of GIS as Tools for Civil Engineering Modeling, Journal of Computing in Civil Engineering, ASCE, Vol. 13, No. 3, pp. 144-152.

Miyamoto, A., and Konno, M. (2005). Virtual Reality Damage Process Education System for Deteriorating Bridges, In Proceedings of the 2005 Structure Congress and the 2005 Forensic Engineering Symposium, New York, USA.

Nassar, K., Thabet, W., and Beliveau, Y. (2003). Simulation of Asphalt Paving Operations under Lane Closure Conditions, Automation in Construction, Vol. 12, pp. 527-541.

Navon, R., and Goldschmidt, E. (2003). Can Labor Inputs be Measured and Controlled Automatically? Journal of Construction Engineering and Management, ASCE, Vol. 129, No. 4, pp. 437-445.

Nguyen, T.H., and Oloufa, A.A. (2002). Automation in Building Design with Spatial Information, In Proceedings of the 19th International Symposium on Automation and Robotics in Construction, Gaithersburg, Maryland.

Object-Relational Mapping 2004 [online]. Available from http://www.service-architecture.com/java-databases/index.html [cited on August 18, 2004].

PJCCI 2004: The official website of Jacques Cartier and Champlain Bridges Incorporated [online]. Available from http://www.pjcci.ca/ [cited on August 18, 2004].

Riley, D. R., and Sanvido, V. E. (1995). Patterns of Construction-Space Use in Multistory Buildings, Journal of Construction Engineering and Management, ASCE, Vol. 121, No. 4, pp. 464-473.

Riley, D. R., and Sanvido, V. E. (1997). Space Planning Method for Multistory Building Construction, Journal of Construction Engineering and Management, ASCE, Vol, 123, No. 2, pp. 171-180.

Rischmoller, L., Fisher, M., Fox, R., and Alarcon L. (2001). 4D Planning and Scheduling (4D-PS): Grounding Construction IT Research in Industry Practice, Construction Information Technology CIB W78 International Conference – IT in Construction in Africa, South Africa, pp. 34-1—34-11.

ROADCON construction ICT Roadmap Documents (2003). [online]. Available from http://www.roadcon.org/ [cited on June 18, 2004].

Ryall, M.J. (2001). Bridge Management, Butterworth-Heinemann, a Division of Reed Educational and Professional Publishing Ltd.

Santos, P., Gamito, M., and Dias, J.M.S. (2001). IDL: A Geometric Interference Detection Language, In Proceedings of the 10th Portuguese Computer Graphics Meeting, pp. 77-86.

Selman, D. (2002). Java 3D Programming, Manning Publications Co.

Seren K., and Karstila K. (2001). MS Project – IFC Mapping Specification, Version 1.0, Eurostep [online]. Available from http://www.eurostep.fi/public/4D/4D_Mapping_V1_0.pdf [cited on October 18, 2005].

Shamir, A., and Pascucci, V. (2001). Temporal and Spatial Level of Details for Dynamic Meshes, In Proceedings of the ACM Symposium on Virtual Reality Software and Technology, Banff, Alberta, Canada, pp. 77-84.

Shepard, R.W., and Johnson M.B. (2001). California Bridge, A Diagnostic Tool to Maximize Bridge Longevity, Investment, TR NEWS 215, 2001.

Stageworks 2005 [online]. Available from http://www.bechtel.com/Briefs/0305/Front_End.htm [cited on June 18, 2005].

STEP (2003) [online]. Available from http://www.tc184-sc4.org/SC4_Open/SC4_Work_Products_Documents/STEP_(10303) [cited on November 12, 2005].

Sunkpho, J., and Garrett, J.H. (2003). Java Inspection Framework: Developing Field Inspection Support Systems for Civil Systems Inspection, Journal of Computing in Civil Engineering, ASCE, Vol. 17, No. 4, pp. 209-218.

Stumpf, A., Liu, L.Y., Kim, C.S., and Chin, S. (1998). Delivery and Test of the Digital Hardhat System at U.S. Army Corps of Engineers Fort Worth District Office. USACERL ADP Report 99/16.

Tam, C.M., Tong, T.K.L., and Chan, W.K.W. (2001). Genetic Algorithm for Optimizing Supply Locations around Tower Crane, Journal of Construction Engineering and Management, ASCE, Vol. 127, No. 4, pp. 315-321.

Tartaro, M.L., Torres, C., and Wainer, G. (2001). Defining Models of Urban Traffic Using the TSC Tool, In Proceedings of the 2001 Winter Simulation Conference, Arlington, VA, USA, pp. 1056-1063.

Thabet, W.Y., and Beliveau, Y.J. (1994). Modeling Work Space to Schedule Repetitive Floors in Multistory Buildings, Journal of Construction Engineering and Management, ASCE, Vol. 120, No. 1, pp. 96-116.

Tommelein, I.D., Levitt, R.E., and Hayes-Roth, B. (1991). SightPlan Model for Site Layout, Journal of Construction Engineering and Management, ASCE, Vol. 118, No. 4, pp. 749-766.

Tommelein, I.D., and Zouein, P.P. (1993). Interactive Dynamic Layout Planning, Journal of Construction Engineering and Management, ASCE, Vol. 119, No. 2, pp. 266-287.

VITASCOPE 2003 [online]. Available from http://pathfinder.engin.umich.edu [cited on November 12, 2005].

Wainer, G. (1998). Discrete-events Cellular Models with Explicit Delays, Doctoral dissertation, Université d'Aix-Marseille III.

Wainer, G. (2002). CD++: a Toolkit to Define Discrete-event Models, Software, Practice and Experience, Wiley. Vol. 32, No.3, pp. 1261-1306.

Wainer, G., and Giambiasi. N. (2001). Application of the Cell-DEVS Paradigm for Cell Spaces Modeling and Simulation, SIMULATION Vol. 76, No. 1, pp. 22-39.

Wainer, G., and Giambiasi, N. (2002). N-Dimensional Cell-DEVS, Discrete Events Systems: Theory and Applications, Kluwer. Vol. 12, No. 1, pp. 135-157.

Walesh, A., and Gehringer, D. (2001). Java 3D API Jump-start, Prentice Hall PTR.

Watt, A. (2000). 3D Computer Graphics, 3rd Edition. Addison-Wesley Publishing Company Inc.

Wolfram, S. (1986). Theory and Application of Cellular Automata, Vol. 1, Advances Series on Complex Systems, World Scientific, Singapore.

Worboys, M., and Duckham, M. (2004). GIS: A Computing Perspective, 2nd ed. CRC Press.

XML 2005 [online]. Available from http://www.xml.org [cited on November 16, 2005].

Zaki, A. R., and Mailhot, G. (2003). Deck Reconstruction of Jacques Cartier Bridge Using Precast Prestressed High Performance Concrete Panels, PCI Journal.

Zayed, T.M. and Halpin, D. (2001). Simulation of Concrete Batch Plant Production, Journal of Construction Engineering and Management, ASCE, Vol. 127, No. 2, pp. 132-141.

Zeigler, B.P. (1976). Theory of Modelling and Simulation, Wiley-Interscience Publication.

Zhang, C., Hammad, A., Zayed, T.M. and Wainer, G. (2005). Representation and Analysis of Spatial Resources in Construction Simulation, In Proceedings of Winter Simulation Conference, Florida, U.S.A. (Accepted, reference number: hammada54350).

Zhang, H., Shi, J.J., and Tam, C. M. (2002). Iconic Animation for Activity-based Construction Simulation, Journal of Computing in Civil Engineering, ASCE, Vol. 16, No. 3, pp. 157-164.

Zhang, J., Anson, M., and Wang, Q. (2000). A New 4D Management Approach to Construction Planning and Site Space Utilization, In Proceedings of the 8th International Conference on Computing in Civil and Building Engineering, Stanford, Vol. 2, pp. 15-22.

Zhong, D., Li, J., Zhu, H., and Song, L. (2004). Geographic Information System-Based Visual Simulation Methodology and Its Application in Concrete Dam Construction Processes, Journal of Construction Engineering and Management, ASCE, Vol. 130, No. 5, pp. 742-750.

Zlatanova, S., Rahman, A.A., and Pilouk, M. (2002). 3D GIS: Current States and Perspectives, In Proceedings of the International Society for Photogrammetry and Remote Sending (ISPRS), Symposium on Geospatial Theory, Processing and Applications, 2002, Ottawa, Canada, Commission IV, WG IV/1.

Zouein, P. P., Harmanani, H. and Hajar, A. (2002). Genetic Algorithm for Solving Site Layout Problem with Unequal-Size and Constrained Facilities. Journal of Computing in Civil Engineering, ASCE, Vol. 16, No. 2, pp. 143-151.

Zouein, P.P., and Tommelein, I.D. (1994). Time-space Tradeoff Strategies for Space-schedule Construction, In proceedings of the 1st Congress of Computing in Civil Engineering, New York, pp. 1180-1187.

Zouein, P.P., and Tommelein, I.D. (2001). Improvement Algorithm for Limited Space Scheduling, Journal of Construction Engineering and Management, ASCE, Vol. 127, No. 2, pp. 116-124.

APPENDIX A: Flowchart of Creating Navigation Tree

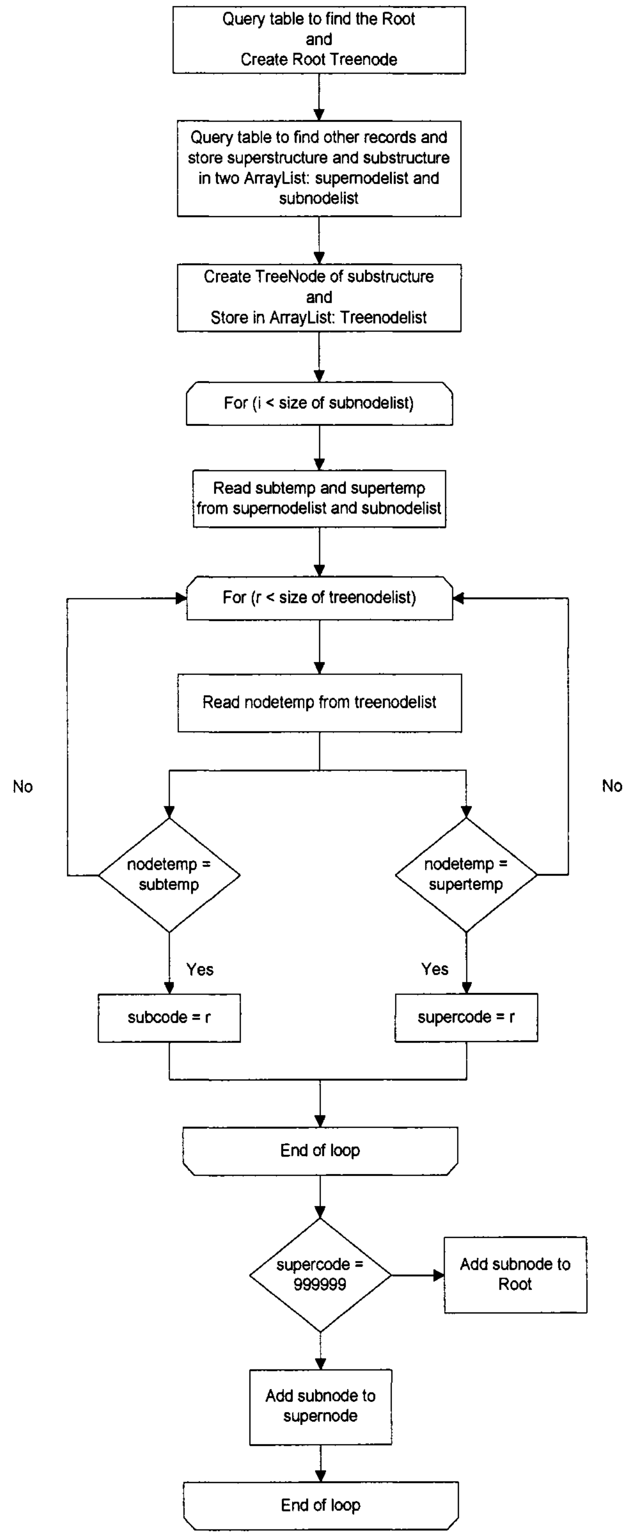


Figure A.1 Flowchart of creating navigation tree

APPENDIX B: IFC Mapping

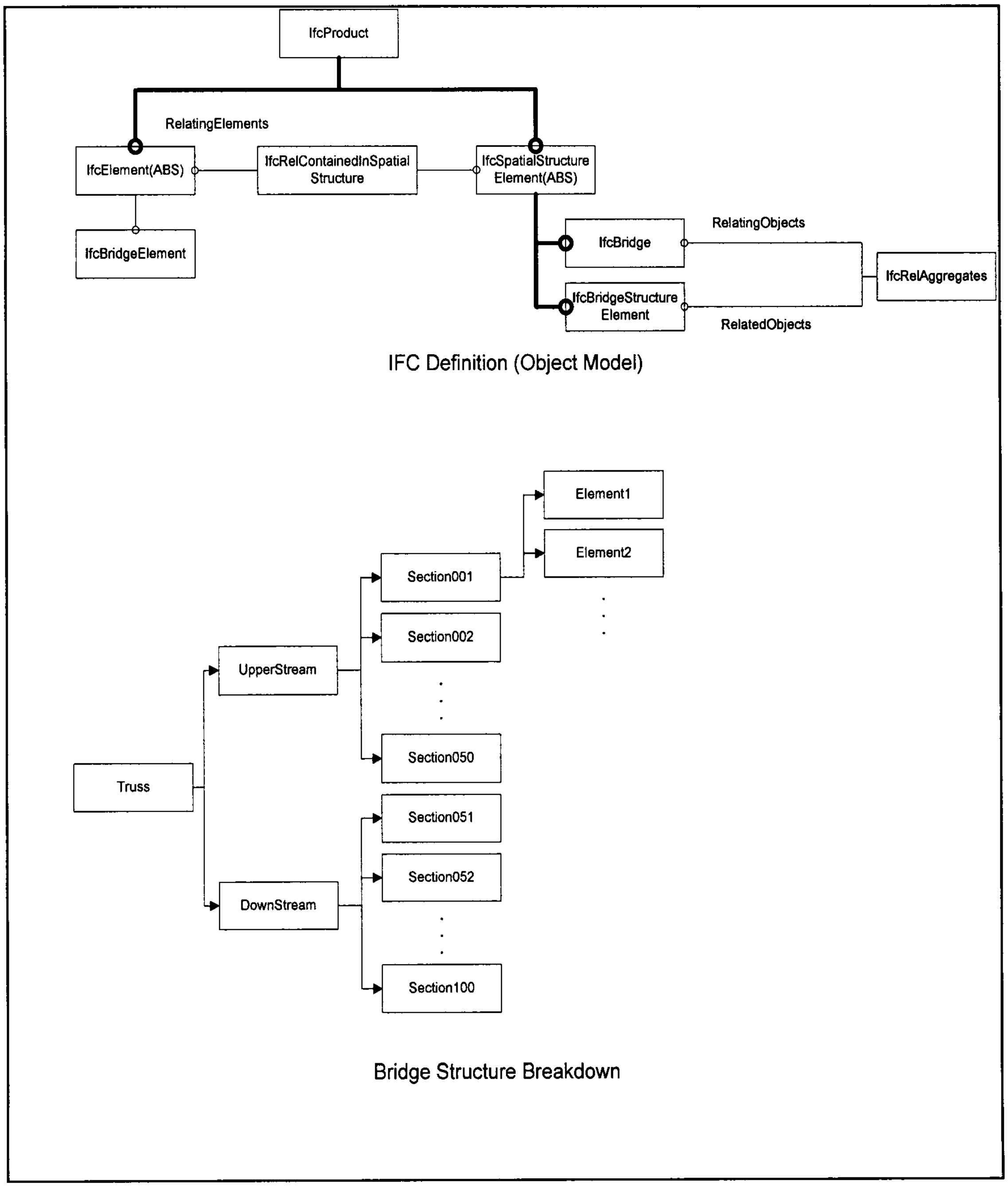


Figure B.1 IFC mapping of bridge structure

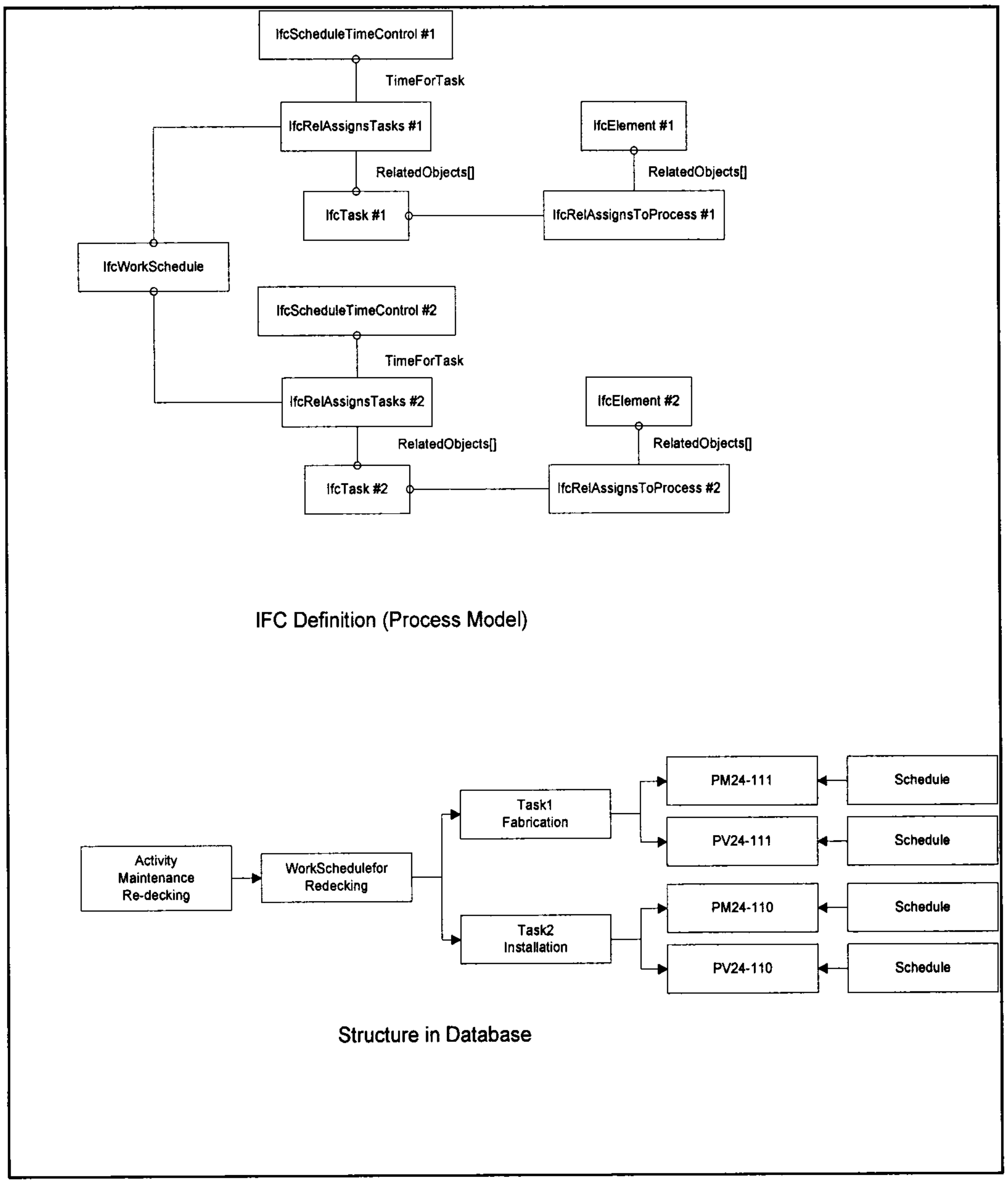


Figure B.2 IFC mapping of schedule

APPENDIX C: Jacques Cartier Bridge Re-decking Project Pictures

Figure C.1 Fabrication plant

Figure C.2 Panels transportation

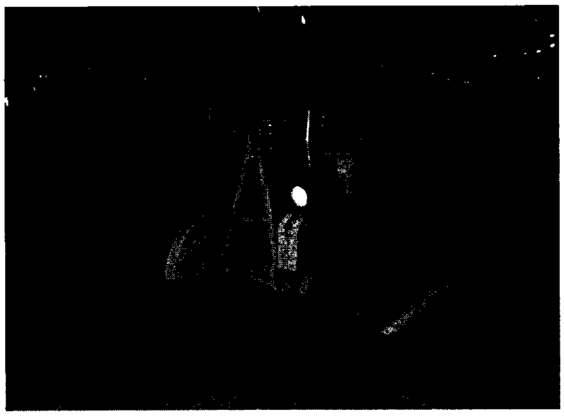


Figure C.3 Cutting old sections

Figure C.4 Installing new panels

Figure C.5 Deck replacement on the approach

APPENDIX D: Source Code of Cell-DEVS Modeling (.ma file)

```
[top]
components : bridge plant transptb@Transptb transbtp@Transbtp dumparea
transdtb@Transdtb transbtd@Transbtd
components : counter quenew@Queuen queold@Queueo uc@Controlunit
reposition@Reposition
in : in
out : out
Link : in in@plant
Link : outid@plant in@transptb
Link : out@transptb in@quenew
Link : outb@quenew innew@bridge
Link : outid@bridge in@transbtd
Link : out@transbtd in@dumparea
Link : outid@dumparea in@transdtb
Link : out@transdtb in@queold
Link : outb@queold inold@bridge
Link : outid@bridge in@transbtp
Link : out@transbtp in@plant
Link : out@bridge in@counter
Link : sendold@uc start@queold
Link : sendnew@uc start@quenew
Link : outc@queold inold@uc
Link : outc@quenew innew@uc
Link : outid@bridge in@uc
Link : outr@uc in@reposition
Link : out@reposition start@queold
Link : out@reposition inr@uc

[transptb]
delay : 00:00:00:420

[transdtb]
delay : 00:00:00:300

[transbtd]
delay : 00:00:00:420

[transbtp]
delay : 00:00:00:300

[reposition]
delay : 00:00:00:800

[uc]
num_old : 5
num_emp : 0

[bridge]
type : cell
dim : (6,200,3)
delay : transport
defaultDelayTime : 1
border : nowrapped
neighbors :                                                bridge(-1,4,2)
neighbors :                bridge(-2,0,2)                  bridge(-1,3,2)
neighbors : bridge(-1,-1,2) bridge(-1,0,2) bridge(-1,1,2) bridge(-1,2,2)
neighbors : bridge(0,-1,2)  bridge(0,0,2)  bridge(0,1,2)  bridge(0,2,2)
neighbors : bridge(1,-1,2)  bridge(1,0,2)  bridge(1,1,2)  bridge(1,2,2)
neighbors :                                                bridge(-1,4,1)
neighbors :                bridge(-2,0,1)                  bridge(-1,3,1)
```

```
neighbors : bridge(-1,-1,1) bridge(-1,0,1) bridge(-1,1,1) bridge(-1,2,1)
neighbors : bridge(0,-1,1)  bridge(0,0,1)  bridge(0,1,1)  bridge(0,2,1)
neighbors : bridge(1,-1,1)  bridge(1,0,1)  bridge(1,1,1)  bridge(1,2,1)
neighbors :                                               bridge(-1,4,0)
neighbors :                bridge(-2,0,0)                 bridge(-1,3,0)
neighbors : bridge(-1,-1,0) bridge(-1,0,0) bridge(-1,1,0) bridge(-1,2,0)
neighbors : bridge(0,-1,0)  bridge(0,0,0)  bridge(0,1,0)  bridge(0,2,0)
neighbors : bridge(1,-1,0)  bridge(1,0,0)  bridge(1,1,0)  bridge(1,2,0)
neighbors :                                                  bridge(-1,4,-1)
neighbors :                bridge(-2,0,-1)                   bridge(-1,3,-1)
neighbors : bridge(-1,-1,-1) bridge(-1,0,-1) bridge(-1,1,-1) bridge(-1,2,-1)
neighbors : bridge(0,-1,-1)  bridge(0,0,-1)  bridge(0,1,-1)  bridge(0,2,-1)
neighbors : bridge(1,-1,-1)  bridge(1,0,-1)  bridge(1,1,-1)  bridge(1,2,-1)
neighbors :                                               bridge(-1,4,-2)
neighbors :                bridge(-2,0,-2)                 bridge(-1,3,-2)
neighbors : bridge(-1,-1,-2) bridge(-1,0,-2) bridge(-1,1,-2) bridge(-1,2,-2)
neighbors : bridge(0,-1,-2)  bridge(0,0,-2)  bridge(0,1,-2)  bridge(0,2,-2)
neighbors : bridge(1,-1,-2)  bridge(1,0,-2)  bridge(1,1,-2)  bridge(1,2,-2)

initialvalue : 0
initialCellsValue : t1.val
in : inold
in : innew
out : out
out : outid
link : inold in@bridge(2,199,0)
link : innew in@bridge(3,199,0)
link : inold in@bridge(2,199,1)
link : innew in@bridge(3,199,1)
link : inold in@bridge(2,199,2)
link : innew in@bridge(3,199,2)
link : out@bridge(1,0,0) out
link : out@bridge(2,0,0) out
link : out@bridge(3,0,0) out
link : out@bridge(4,0,0) out
link : out@bridge(1,0,2) outid
link : out@bridge(2,0,2) outid
link : out@bridge(3,0,2) outid
link : out@bridge(4,0,2) outid

localtransition : move-rule
zone : move-rule1 { (0,0,0)..(2,199,0) }
zone : move-rule2 { (3,0,0)..(5,199,0) }
zone : move-check-rule1 { (0,0,1)..(2,199,1) }
zone : move-check-rule2 { (3,0,1)..(5,199,1) }
zone : move-id-rule { (0,0,2)..(5,199,2) }

portInTransition : in@bridge(2,199,0) genoldtruck-rule
portInTransition : in@bridge(3,199,0) gennewtruck-rule
portInTransition : in@bridge(2,199,1) gendir-rule
portInTransition : in@bridge(3,199,1) gendir-rule

[move-rule]
rule : 0 1 { t }

[move-rule1]
rule : { (0,1,0) } 1 { (0,0,0) = 0 and ((0,1,0) = 4 or (0,1,0) = 9 ) and
((0,1,1) = 4 or (0,1,1) = 5) }
rule : 1 1 { (0,0,0) = 0 and (0,1,0) = 1 and (0,1,1) = 4 and not(1,1,0) = 3 }
rule : 8 1 { (0,0,0) = 0 and (0,1,0) = 8 and (0,1,1) = 4 and not((1,0,0) = 2
and (1,1,0) = 5) }
rule : 8 1 { (0,0,0) = 8 and (0,0,1) = 5 }
rule : 1 1 { (0,0,0) = 1 and (0,0,1) = 5 }
```

```
rule : { (1,0,0) } 1 { (0,0,0) = 0 and (1,0,0) != 0 and (1,0,1) = 1 }
rule : { (-1,0,0) } 1 { (0,0,0) = 0 and (-1,0,0) != 0 and (-1,0,1) = 2 }
rule : 4 {round(normal(900,10))} { (0,0,0) = 1 and (1,0,0) = 3 }
rule : { (0,0,0) } 1 { (0,0,0) = 4 and (0,0,1) = 0 and (1,0,0) = 2 }
rule : { (0,0,0) } 1 { (0,0,0) = 9 and (0,0,1) = 0 and (1,0,0) = 2 }
rule : 9 {round(normal(1560,10))} { (0,0,0) = 8 and (1,0,0) = 5 }
rule : 0 1 { (0,0,0) = 1 and (0,-1,0) = 0 }
rule : 2 1 { (0,0,0) = 2 and (0,-1,0) != 2}
rule : 3 1 { (0,0,0) = 3 and (1,0,0) != 5 and (1,0,0) != 1}
rule : 5 1 { (0,0,0) = 3 and (1,0,0) = 1 }
rule : 5 1 { (0,0,0) = 3 and (1,0,0) = 5 }
rule : 5 1 { (0,0,0) = 5 and (1,0,0) != 6 and (1,0,0) != 8}
rule : 6 1 { (0,0,0) = 5 and (1,0,0) = 8 }
rule : 6 1 { (0,0,0) = 5 and (1,0,0) = 6 }
rule : 6 1 { (0,0,0) = 6 and (0,1,0) != 0 and not((0,-1,0) = 3 and (0,1,0) = 2)}
rule : 2 {round(normal(1080,10))} { (0,0,0) = 0 and (0,1,0) = 2 and (0,2,0) = 6
and (0,-1,0) != 3 and (1,2,0) = 9 }
rule : 2 1 { (0,0,0) = 0 and (0,1,0) = 2 and (1,0,0) = 2 }
rule : 2 1 { (0,0,0) = 6 and (0,-1,0) = 3 and (0,1,0) = 2 }
rule : 2 1 { (0,0,0) = 0 and (0,-1,0) = 3 }
rule : 3 1 { (0,0,0) = 2 and (0,-1,0) = 2 and (0,1,0) = 6 }
rule : 0 1 { t }

[move-rule2]
rule : { (0,1,0) } 1 { (0,0,0) = 0 and ((0,1,0) = 4 or (0,1,0) = 9 ) and
((0,1,1) = 4 or (0,1,1) = 5 ) }
rule : 1 1 { (0,0,0) = 0 and (0,1,0) = 1 and (0,1,1) = 4 and not((-1,0,0) = 2
and (-1,1,0) = 3) }
rule : 8 1 { (0,0,0) = 0 and (0,1,0) = 8 and (0,1,1) = 4 and not((-1,0,0) = 2
and (-1,1,0) = 5) }
rule : 8 1 { (0,0,0) = 8 and (0,0,1) = 5 }
rule : 1 1 { (0,0,0) = 1 and (0,0,1) = 5 }
rule : { (-1,0,0)} 1 { (0,0,0) = 0 and (-1,0,0) != 0 and (-1,0,1) = 2 }
rule : { (1,0,0) } 1 { (0,0,0) = 0 and (1,0,0) != 0 and (1,0,1) = 1 }
rule : 4 {round(normal(900,10))} { (0,0,0) = 1  and (-1,0,0) = 3 and (0,-
1,0) != 3}
rule : { (0,0,0) } 1 { (0,0,0) = 4 and (0,0,1) = 0 and (-1,0,0) = 2 }
rule : { (0,0,0) } 1 { (0,0,0) = 9 and (0,0,1) = 0 and (-1,0,0) = 2 }
rule : 9 {round(normal(1560,10))} { (0,0,0) = 8 and (-1,0,0) = 5 and (0,-
1,0) != 5}
rule : 0 1 { (0,0,0) = 1 and (0,-1,0) = 0 }
rule : 2 1 { (0,0,0) = 2 and (0,-1,0) != 2 }
rule : 3 1 { (0,0,0) = 3 and (-1,0,0) != 1 and (-1,0,0) != 5 }
rule : 5 1 { (0,0,0) = 3 and (-1,0,0) = 1 }
rule : 5 1 { (0,0,0) = 3 and (-1,0,0) = 5 }
rule : 5 1 { (0,0,0) = 5 and (-1,0,0) != 6 and (-1,0,0) != 8 }
rule : 6 1 { (0,0,0) = 5 and (-1,0,0) = 8 }
rule : 6 1 { (0,0,0) = 5 and (-1,0,0) = 6 }
rule : 6 1 { (0,0,0) = 6 and (0,1,0) != 0 and not((0,-1,0) = 3 and (0,1,0) = 2)}
rule : 2 {round(normal(1080,10))} { (0,0,0) = 0 and (0,1,0) = 2 and (0,2,0) = 6
and (0,-1,0) != 3 and (-1,2,0) = 9 }
rule : 2 1 { (0,0,0) = 0 and (0,1,0) = 2 and (-1,0,0) = 2 }
rule : 2 1 { (0,0,0) = 6 and (0,-1,0) = 3 and (0,1,0) = 2 }
rule : 2 1 { (0,0,0) = 2 and (0,-1,0) = 3 }
rule : 3 1 { (0,0,0) = 2 and (0,-1,0) = 2 and (0,1,0) = 6 }
rule : 0 1 { t }

[move-check-rule1]
rule : 4 1 { (0,0,0) = 0 and (0,1,0) = 5 and ((0,1,-1) = 4 or (0,1,-1) = 9) }
rule : 4 1 { (0,0,0) = 0 and (0,1,0) = 4 and (0,-1,0) = 0 and (0,1,-1) = 1 and
(1,1,-1) != 3 }
rule : 4 1 { (0,0,0) = 0 and (0,1,0) = 4 and (0,-1,0) = 0 and (0,1,-1) = 8 and
(1,1,-1) != 5 }
```

```
rule : 4 1 { (0,0,0) = 0 and (0,1,0) = 4 and (0,-1,0) = 0 and ((0,1,-1) = 4 or
(0,1,-1) = 9) }
rule : 2 1 { (0,0,0) = 0 and (0,1,0) = 4 and (0,-1,0) = 5 and (1,0,0) != 5 }
rule : 1 1 { (0,0,0) = 0 and (0,1,0) = 4 and (0,-1,0) = 5 and (1,0,0) = 5 }
rule : 2 1 { (0,0,0) = 0 and (-1,0,0) = 2 and (0,-1,0) != 0 }
rule : 4 1 { (0,0,0) = 0 and (-1,0,0) = 2 and (0,-1,0) = 0 and (-1,0,-1) != 0 }
rule : 4 1 { (0,0,0) = 0 and (1,0,0) = 1 and (0,-1,0) = 0 }
rule : 5 1 { (0,0,0) = 4 and (0,0,-1) = 1 and (1,0,-1) = 3 }
rule : 5 1 { (0,0,0) = 4 and (0,0,-1) = 8 and (1,0,-1) = 5 }
rule : 5 1 { (0,0,0) = 0 and ((0,0,-1) = 2 or (0,0,-1) = 3 or (0,0,-1) = 5 ) }
rule : 5 1 { (0,0,0) = 5 and (0,0,-1) != 0 }
rule : 0 1 { t }

[move-check-rule2]
rule : 4 1 { (0,0,0) = 0 and (0,1,0) = 5 and ((0,1,-1) = 4 or (0,1,-1) = 9) }
rule : 4 1 { (0,0,0) = 0 and (0,1,0) = 4 and (0,-1,0) = 0 and (0,1,-1) = 1 and
(-1,1,-1) != 3 }
rule : 4 1 { (0,0,0) = 0 and (0,1,0) = 4 and (0,-1,0) = 0 and (0,1,-1) = 8 and
(-1,1,-1) != 5 }
rule : 4 1 { (0,0,0) = 0 and (0,1,0) = 4 and (0,-1,0) = 0 and ((0,1,-1) = 4 or
(0,1,-1) = 9) }
rule : 1 1 { (0,0,0) = 0 and (0,1,0) = 4 and (0,-1,0) = 5 and (-1,0,0) != 5 }
rule : 2 1 { (0,0,0) = 0 and (0,1,0) = 4 and (0,-1,0) = 5 and (-1,0,0) = 5 }
rule : 1 1 { (0,0,0) = 0 and (1,0,0) = 1 and (0,-1,0) != 0 }
rule : 4 1 { (0,0,0) = 0 and (1,0,0) = 1 and (0,-1,0) = 0 and (1,0,-1) != 0 }
rule : 4 1 { (0,0,0) = 0 and (-1,0,0) = 2 and (0,-1,0) = 0 }
rule : 5 1 { (0,0,0) = 4 and (0,0,-1) = 1 and (-1,0,-1) = 3 }
rule : 5 1 { (0,0,0) = 4 and (0,0,-1) = 8 and (-1,0,-1) = 5 }
rule : 5 1 { (0,0,0) = 0 and ((0,0,-1) = 2 or (0,0,-1) = 3) }
rule : 5 1 { (0,0,0) = 5 and (0,0,-1) != 0 }
rule : 0 1 { t }


[gendir-rule]
rule : 4 1 { portValue(thisPort) > 0 }
rule : 0 1 { t }

[genoldtruck-rule]
rule : 1 1 { portValue(thisPort) > 0 }
rule : 0 1 {t}

[gennewtruck-rule]
rule : 8 1 { portValue(thisPort) > 0 }
rule : 0 1 {t}

[move-id-rule]
rule : { (0,1,0) } 1 { (0,0,0) = 0 and (0,1,0) > 0 and (0,1,-1) = 5 and (0,1,-2)
= 4 }
rule : { (0,1,0) } 1 { (0,0,0) = 0 and (0,1,0) > 0 and (0,1,-1) = 5 and (0,1,-2)
= 9 }
rule : { (0,1,0) } 1 { (0,0,0) = 0 and (0,1,0) > 0 and (0,1,-1) = 4 and (0,1,-2)
= 1 and (-1,1,-2) != 3 and (1,1,-2) != 3 }
rule : { (0,1,0) } 1 { (0,0,0) = 0 and (0,1,0) > 0 and (0,1,-1) = 4 and (0,1,-2)
= 4 }
rule : { (0,1,0) } 1 { (0,0,0) = 0 and (0,1,0) > 0 and (0,1,-1) = 4 and (0,1,-2)
= 8 and (-1,1,-2) != 5 and (1,1,-2) != 5 }
rule : { (0,1,0) } 1 { (0,0,0) = 0 and (0,1,0) > 0 and (0,1,-1) = 4 and (0,1,-2)
= 9 }
rule : { (0,0,0) } 1 { (0,0,0) != 0 and ((-1,0,-2) = 3 or (1,0,-2) = 3 or (-
1,0,-2) = 5 or (1,0,-2) = 5) and (0,0,-1) !=0 }
rule : { (0,0,0) } 1 { (0,0,0) != 0 and (0,0,-1) = 5 }
rule : { (1,0,0) } 1 { (0,0,0) = 0 and (1,0,0) != 0 and (1,0,-1) = 1 }
rule : { (-1,0,0) } 1 { (0,0,0) = 0 and (-1,0,0) != 0 and (-1,0,-1) = 2 }
```

```
rule : 0 1 {t}

[sendid-rule]
rule : { (0,0,0)-(0,0,0) + send(thisport, (0,0,0))} 1 { (0,0,0) > 0 }
rule : 0 0 {t}

[sendtruck-rule]
rule : { (0,0,0)-(0,0,0) + send(thisport, (0,0,0))} 1 { (0,0,0) > 0 }
rule : 0 0 {t}

[plant]
type : cell
dim : (3,15,2)
delay : transport
defaultDelayTime : 1
border : nowrapped
neighbors : plant(0,-2,1) plant(0,-1,1) plant(0,0,1) plant(0,1,1)
neighbors : plant(1,-2,1) plant(1,-1,1) plant(1,0,1) plant(1,1,1)
neighbors : plant(0,-2,0) plant(0,-1,0) plant(0,0,0) plant(0,1,0)
neighbors : plant(1,-2,0) plant(1,-1,0) plant(1,0,0) plant(1,1,0)
neighbors : plant(0,-2,-1) plant(0,-1,-1) plant(0,0,-1) plant(0,1,-1)
neighbors : plant(1,-2,-1) plant(1,-1,-1) plant(1,0,-1) plant(1,1,-1)
initialvalue : 0
initialCellsValue : t1-plant.val
in : in
out : out
out : outid
link : in in@plant(1,14,0)
link : in in@plant(1,14,1)
link : out@plant(1,0,0) out
link : out@plant(1,0,1) outid
localtransition : plant-rule
zone : plant-st-rule { (0,0,0)..(2,14,0) }
zone : plant-id-rule { (0,0,1)..(2,14,1) }
portInTransition : in@plant(1,14,0) gentruck-plant-rule
portInTransition : in@plant(1,14,1) genid-plant-rule

[plant-rule]
rule : 0 1 { t }

[plant-st-rule]
rule : 9 1 { (0,0,0) = 0 and (0,1,0) = 9 and (1,1,0) != 6 and (0,-1,0) != 9 }
rule : 9 1 { (0,0,0) = 9 and (0,-2,0) = 9 }
rule : 8 840 { (0,0,0) = 9 and (1,0,0) = 6 }
rule : 8 1 { (0,0,0) = 0 and (0,1,0) = 8 and (0,1,1) != 0 }
rule : 6 1 { (0,0,0) = 6 }
rule : 0 1 { t }

[plant-id-rule]
rule : 6 1 { (0,0,0) = 0 and (0,1,0) = 6 and (0,1,-1) != 0 and (1,1,-1) != 6
and (0,-1,-1) != 9 and (0,-1,-1) != 8}
rule : 7 1 { (0,0,0) = 0 and (0,1,0) = 7 and (0,1,-1) != 0 and (1,1,-1) != 6
and (0,-1,-1) != 9 and (0,-1,-1) != 8}
rule : 8 1 { (0,0,0) = 0 and (0,1,0) = 8 and (0,1,-1) != 0 and (1,1,-1) != 6
and (0,-1,-1) != 9 and (0,-1,-1) != 8}
rule : 9 1 { (0,0,0) = 0 and (0,1,0) = 9 and (0,1,-1) != 0 and (1,1,-1) != 6
and (0,-1,-1) != 9 and (0,-1,-1) != 8}
rule : 10 1 { (0,0,0) = 0 and (0,1,0) = 10 and (0,1,-1) != 0 and (1,1,-1) != 6
and (0,-1,-1) != 9 and (0,-1,-1) != 8}
rule : 6 1 { (0,0,0) = 6 and ((0,-2,-1) = 9 or (0,-2,-1) = 8) and (0,0,-1) != 0}
rule : 7 1 { (0,0,0) = 7 and ((0,-2,-1) = 9 or (0,-2,-1) = 8) and (0,0,-1) != 0}
rule : 8 1 { (0,0,0) = 8 and ((0,-2,-1) = 9 or (0,-2,-1) = 8) and (0,0,-1) != 0}
rule : 9 1 { (0,0,0) = 9 and ((0,-2,-1) = 9 or (0,-2,-1) = 8) and (0,0,-1) != 0}
```

```
rule : 10 1 { (0,0,0) = 10 and ((0,-2,-1) = 9 or (0,-2,-1) = 8) and (0,0,-1) !=
0}
rule : { (0,1,0) } 1 { (0,0,0) = 0 and (0,1,0) != 0 and (0,1,-1) = 8 }
rule : { (0,0,0) } 1 { ((0,0,0) = 6 or (0,0,0) = 7 or (0,0,0) = 8 or (0,0,0) =
9 or (0,0,0) = 10 ) and (1,0,-1) = 6 and (0,0,-1) = 9 }
rule : { (0,1,0) } 1 { (0,0,0) = 0 and (0,1,0) != 0 and (0,-1,-1) = 8 and
(0,1,-1) != 0 }
rule : 0 1 {t}

[gentruck-plant-rule]
rule : 9 1 { portValue(thisPort) > 5 }
rule : 0 1 { t }

[genid-plant-rule]
rule : { portValue(thisPort) } 1 { portValue(thisPort) > 5 }
rule : 0 1 { t }

[dumparea]
type : cell
dim : (3,15,2)
delay : transport
defaultDelayTime : 1
border : nowrapped
neighbors : dumparea(0,-2,1) dumparea(0,-1,1) dumparea(0,0,1) dumparea(0,1,1)
neighbors : dumparea(1,-2,1) dumparea(1,-1,1) dumparea(1,0,1) dumparea(1,1,1)
neighbors : dumparea(0,-2,0) dumparea(0,-1,0) dumparea(0,0,0) dumparea(0,1,0)
neighbors : dumparea(1,-2,0) dumparea(1,-1,0) dumparea(1,0,0) dumparea(1,1,0)
neighbors : dumparea(0,-2,-1) dumparea(0,-1,-1) dumparea(0,0,-1) dumparea(0,1,-
1)
neighbors : dumparea(1,-2,-1) dumparea(1,-1,-1) dumparea(1,0,-1) dumparea(1,1,-
1)
initialvalue : 0
initialCellsValue : t1-dump.val
in : in
out : out
out : outid
link : in in@dumparea(1,14,0)
link : in in@dumparea(1,14,1)
link : out@dumparea(1,0,0) out
link : out@dumparea(1,0,1) outid
localtransition : dump-rule
zone : dump-st-rule { (0,0,0)..(2,14,0) }
zone : dump-id-rule { (0,0,1)..(2,14,1) }
portInTransition : in@dumparea(1,14,0) gentruck-dump-rule
portInTransition : in@dumparea(1,14,1) genid-dump-rule

[dump-rule]
rule : 0 1 { t }

[dump-st-rule]
rule : 4 1 { (0,0,0) = 0 and (0,1,0) = 4 and (1,1,0) != 7 and (0,-1,0) != 4 }
rule : 4 1 { (0,0,0) = 4 and (0,-2,0) = 4 }
rule : 1 300 { (0,0,0) = 4 and (1,0,0) = 7 }
rule : 1 1 { (0,0,0) = 0 and (0,1,0) = 1 and (0,1,1) != 0 }
rule : 7 1 { (0,0,0) = 7 }
rule : 0 1 { t }

[dump-id-rule]
rule : 1 1 { (0,0,0) = 0 and (0,1,0) = 1 and ((0,1,-1) = 1 or (0,1,-1) = 4) and
(1,1,-1) != 7 and (0,-1,-1) != 4 and (0,-1,-1) != 1}
rule : 2 1 { (0,0,0) = 0 and (0,1,0) = 2 and ((0,1,-1) = 1 or (0,1,-1) = 4) and
(1,1,-1) != 7 and (0,-1,-1) != 4 and (0,-1,-1) != 1}
```

```
rule : 3 1 { (0,0,0) = 0 and (0,1,0) = 3 and ((0,1,-1) = 1 or (0,1,-1) = 4) and
(1,1,-1) != 7 and (0,-1,-1) != 4 and (0,-1,-1) != 1}
rule : 4 1 { (0,0,0) = 0 and (0,1,0) = 4 and ((0,1,-1) = 1 or (0,1,-1) = 4) and
(1,1,-1) != 7 and (0,-1,-1) != 4 and (0,-1,-1) != 1}
rule : 5 1 { (0,0,0) = 0 and (0,1,0) = 5 and ((0,1,-1) = 1 or (0,1,-1) = 4) and
(1,1,-1) != 7 and (0,-1,-1) != 4 and (0,-1,-1) != 1}
rule : 1 1 { (0,0,0) = 1 and ((0,-2,-1) = 4 or (0,-2,-1) = 1) and (0,0,-1) != 0}
rule : 2 1 { (0,0,0) = 2 and ((0,-2,-1) = 4 or (0,-2,-1) = 1) and (0,0,-1) != 0}
rule : 3 1 { (0,0,0) = 3 and ((0,-2,-1) = 4 or (0,-2,-1) = 1) and (0,0,-1) != 0}
rule : 4 1 { (0,0,0) = 4 and ((0,-2,-1) = 4 or (0,-2,-1) = 1) and (0,0,-1) != 0}
rule : 5 1 { (0,0,0) = 5 and ((0,-2,-1) = 4 or (0,-2,-1) = 1) and (0,0,-1) != 0}
rule : 1 1 { (0,0,0) = 0 and (0,1,0) = 1 and (0,1,-1) = 1 }
rule : 2 1 { (0,0,0) = 0 and (0,1,0) = 2 and (0,1,-1) = 1 }
rule : 3 1 { (0,0,0) = 0 and (0,1,0) = 3 and (0,1,-1) = 1 }
rule : 4 1 { (0,0,0) = 0 and (0,1,0) = 4 and (0,1,-1) = 1 }
rule : 5 1 { (0,0,0) = 0 and (0,1,0) = 5 and (0,1,-1) = 1 }
rule : 1 1 { (0,0,0) = 1 and (1,0,-1) = 7 and (0,0,-1) = 4 }
rule : 2 1 { (0,0,0) = 2 and (1,0,-1) = 7 and (0,0,-1) = 4 }
rule : 3 1 { (0,0,0) = 3 and (1,0,-1) = 7 and (0,0,-1) = 4 }
rule : 4 1 { (0,0,0) = 4 and (1,0,-1) = 7 and (0,0,-1) = 4 }
rule : 5 1 { (0,0,0) = 5 and (1,0,-1) = 7 and (0,0,-1) = 4 }
rule : 1 1 { (0,0,0) = 0 and (0,1,0) = 1 and (0,-1,-1) = 1 and (0,1,-1) != 0 }
rule : 2 1 { (0,0,0) = 0 and (0,1,0) = 2 and (0,-1,-1) = 1 and (0,1,-1) != 0 }
rule : 3 1 { (0,0,0) = 0 and (0,1,0) = 3 and (0,-1,-1) = 1 and (0,1,-1) != 0 }
rule : 4 1 { (0,0,0) = 0 and (0,1,0) = 4 and (0,-1,-1) = 1 and (0,1,-1) != 0 }
rule : 5 1 { (0,0,0) = 0 and (0,1,0) = 5 and (0,-1,-1) = 1 and (0,1,-1) != 0 }
rule : 0 1 {t}

[gentruck-dump-rule]
rule : 4 1 { portValue(thisPort) > 0 and portValue(thisPort) < 6 }
rule : 0 1 { t }

[genid-dump-rule]
rule : { portValue(thisPort) } 1 { portValue(thisPort) > 0 and
portValue(thisPort) < 6 }
rule : 0 1 { t }
```

APPENDIX E: User's Guide of the Prototype System

Software requirements:

(1) Borland JBuilder 2005 Enterprise: used to develop the prototype system of BMS;
(2) MS Access (MS Access XP): used to store the lifecycle data of the bridge;
(3) ArcGIS (ESRI 2004): used to create TIN model from DEM;
(4) CD++ (Wainer 2002, Carleton University): used to develop the prototype of cell-based construction simulation model;
(5) Windows XP: used as the operation system.

Java library requirements:

(1) Java Runtime Environment: JDK 1.4.2_04
(2) Java3D: JDK3D 1.3.1_09
(3) VRML97
(4) MapObjects Java Edition: MOJ21

Environment settings:

Java Database Connectivity (JDBC): Control Panel→Administrative Tools→Data Sources (ODBC)

Hardware requirements:

(1) CPU: Pentium 4
(2) RAM: 512MB or more
(3) Graphics accelerator card (NVIDIA Quadro4 280 NVS (IBM))
(4) Pointing device: mouse or pen

Main steps to run the system:

(1) Run JBuilder 2005 Enterprise;
(2) Select and run the project;
(3) Input username and password to log in the system and select the *Bridge* or *Building* (Figure E.1); Click *Login* button;

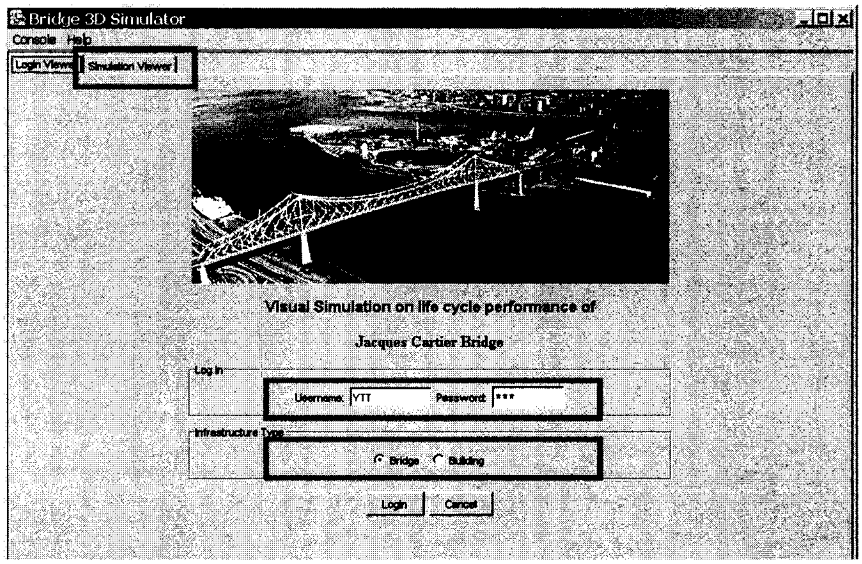


Figure E.1 Login interface

(4) Go to the main user interface by clicking the *Simulation Viewer*;

(5) Click one of the *Cameras* to some predefined viewpoints (Figure E.2);

(6) Show LoDs by moving the mouse forward and backward while holding *Alt* key and the left button of the mouse;

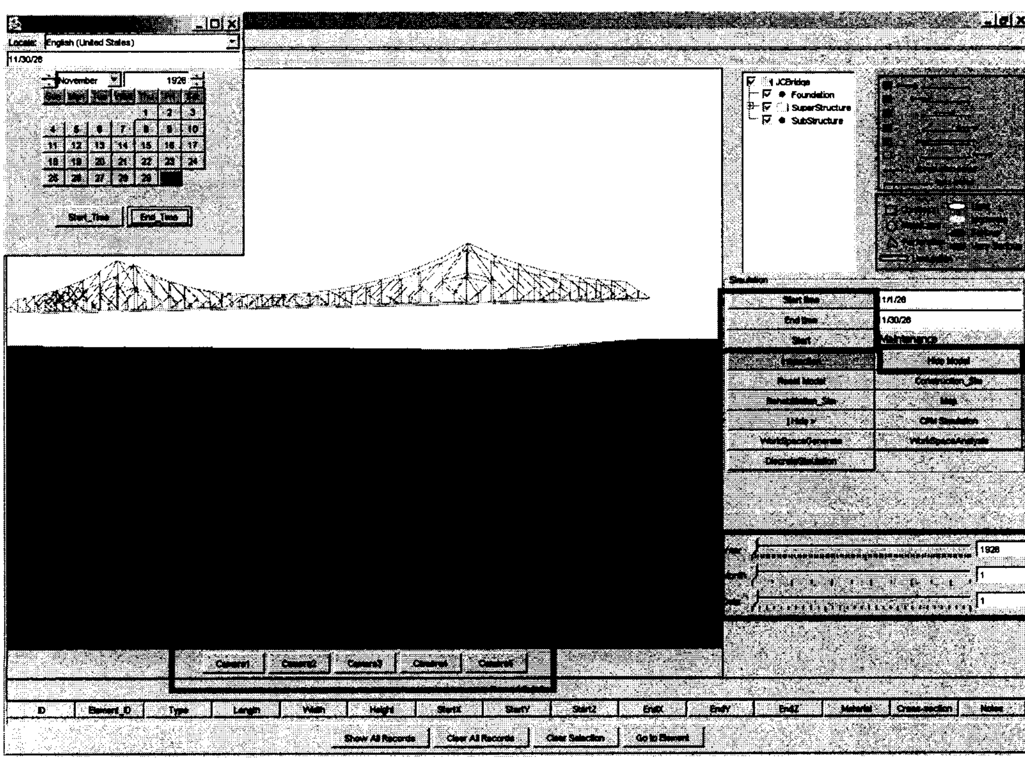


Figure E.2 Main user interface - 1

(7) Retrieve lifecycle information (Figure E.2):

Construction of Jacques Cartier Bridge: click the *Start time* button, using the popup window of the *calendar* to input the time period, which should be between March 16, 1928 and July 1, 1929. Then click *Start* button; another time input method is using the *time sliders*, which are in the right bottom corner of the interface.

Maintenance: click the *Maintenance* button and select the painting year.

Inspection: click the *Inspection* button and another interface will pop up. The part of inspection interface is not included in this thesis.

(8) Workspace generation and conflict detection (Figure E.3):

Input the activity time of the re-decking project on the main span, which should be between May 6, 2002 and August 19, 2002 (e.g. 8/12/02). Click *Workspace Generation* button and use *Camera2* to get the viewpoint from the top of the main span. Click *Workspace Analysis* button to detect conflicts between workspaces.

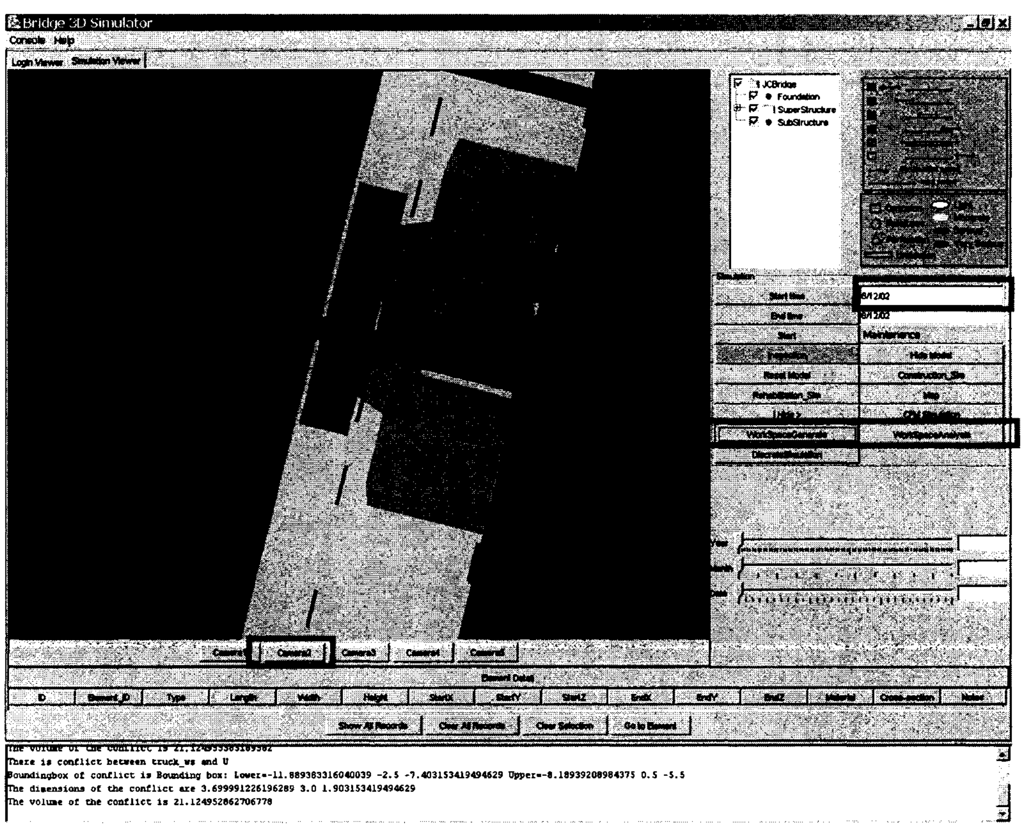


Figure E.3 Main user interface - 2

APPENDIX F: List of Publications

a. Articles accepted in or submitted to refereed journals

(1) Hammad, A., **Zhang, C.**, Hu, Y. and Mozaffari, E. (accepted in June, 2005). Mobile Model-Based Bridge Lifecycle Management Systems. *Journal of Computer-Aided Civil and Infrastructure Engineering*, Blackwell Publishing.

(2) Hammad, A., **Zhang, C.**, Cardinal, G., and Al-Hussein, M. Equipment Workspace Analysis in Infrastructure Projects. *Canadian Journal of Civil Engineering* (submitted in October, 2005).

(3) **Zhang, C.**, Hammad, A., Zayed, T., and Wainer, G. Cell-based Representation and Analysis of Spatial Resources in Construction Simulation. *Journal of Automation in Construction* (submitted in October, 2005).

b. Articles in refereed conference proceedings

(4) **Zhang, C.**, Hammad, A., Zayed, T.M., and Wainer, G. (2005). Representation and Analysis of Spatial Resources in Construction Simulation. *Winter Simulation Conference*, 2005, Florida, U.S.A. 9 pages. (reference number: hammada54350)

(5) **Zhang, C.**, and Hammad, A. (2005). Spatio-Temporal Issues in Infrastructure Lifecycle Management Systems. *In proceedings of the 1st CSCE Specialty Conference on Infrastructure Technologies*, Management and Policy, 2005, Toronto, Ontario, Canada. 10 pages.

(6) **Zhang, C.**, Mozaffari, E., El-Ammari, K. and Hammad, A. (2005). Integrated Framework for Lifecycle Infrastructure Management Systems. *International Congress of Urbistics*, 2005, Montreal, Canada. 8 pages.

(7) Mozaffari, E., Khabir, B., **Zhang, C.**, Devarakonda, P., Bauchkar, P., and Hammad, A. (2005). Interaction Models for Infrastructure Management Systems Using Virtual and Augmented Realities. *International Congress of Urbistics*, 2005, Montreal, Canada. 8 pages.

(8) Hammad, A., **Zhang, C.**, Hu, Y. and Mozaffari, E. (2004). Mobile Model-Based Bridge Lifecycle Management Systems. *In Proceedings of Conference on Construction Application of Virtual Reality*, ADETTI/ISCTE, September, 2004, Lisbon, pp. 109-120.