

Mobile Model-Based Bridge Lifecycle Management Systems

Amin Hammad^{*}, Cheng Zhang, Yongxin Hu & Elaheh Mozaffari

Concordia Institute for Information Systems Engineering

1425 René Lévesque Boulevard, Montreal, Quebec, Canada H3G 1T7, Canada

Abstract: *This paper discusses the requirements for developing Mobile Model-based Bridge Lifecycle Management Systems (MMBLMSs). These new systems should link all the information about the lifecycle stages of a bridge (e.g., construction, inspection and maintenance) to a 4D model of the bridge incorporating different scales of space and time in order to record events throughout the lifecycle with suitable levels of details. In addition, MMBLMSs should support distributed databases and mobile location-based computing by providing user interfaces that could be used on thin clients, such as PDAs and tablet PCs, equipped with wireless communications and tracking devices, such as a GPS receivers. A prototype system developed in Java language is used to demonstrate the feasibility of the proposed methodology for realizing these systems.*

1 INTRODUCTION

This paper discusses the requirements for developing Mobile Model-based Bridge Lifecycle Management Systems (MMBLMSs). These new systems should link all the information about the lifecycle stages of a bridge (e.g., construction, inspection and maintenance) to a 4D model of the bridge incorporating different scales of space and time in order to record events throughout the lifecycle. In addition, MMBLMSs should support distributed databases and mobile Location-Based Computing (LBC) by providing user interfaces that could be used on thin clients, such as PDAs and tablet PCs, equipped with wireless communications and tracking devices, such as a GPS receivers.

The paper starts by reviewing conventional BMSs

and recent trends in 4D models, mobile computing and LBC. This is followed by an analysis of the requirements of the proposed MMBLMSs. Special consideration is given to the spatial and temporal issues, such as the requirements to support different levels of detail (LoDs), and to adopt available standards for interoperability. Then, a prototype system developed in Java language to demonstrate the feasibility of the proposed methodology is discussed in detail including the system architecture and the database and user interface designs. A case study about Jacques Cartier Bridge in Montreal is also demonstrated.

2 REVIEW OF BMS'S

Bridge lifecycle management aims to perform the management functionalities related to bridges from the conception stage to the end of their useful life, through the design, construction and operation and maintenance stages. The major tasks in bridge management are: (1) collection of inventory data, (2) inspection (3) assessment of condition and strength, (4) repair, strengthening or replacement, and (5) prioritizing the allocation of funds.

BMSs are a means of managing information of bridges to assure their long-term health, and that maintenance programs can be formulated in line with budgetary constraints and funding limitations. BMSs include four basic components: data storage, cost and deterioration models, optimization models for analysis, and updating functions (Czepiel, 2004; Ryall, 2001). The core part of a BMS is the database which is built up of information obtained from the regular inspection and maintenance activities. Bridge database management includes the collection, updating, integration, and archiving of the following information: (1) bridge general

^{*}To whom correspondence should be addressed. E-mail: hammad@ciise.concordia.ca

information (location, name, type, load capacity, etc.), (2) design information and physical properties of the elements, (3) inventory data, (4) regular inspection records, (5) condition and strength assessment reports, (6) repair and maintenance records, and (7) cost records. A simplified BMS structure is demonstrated in Figure 1.

New approaches in BMSs try to introduce new information technologies to facilitate mobile data collection and manipulation. For example, a system developed by the University of Central Florida for the Florida Department of Transportation (FDOT) (Kuo et al, 1994) consists of both a field and office set up with a pen-based notebook computer used to collect all field inspection data. The Massachusetts Highway Department is using a system called IBIIS to store and manage all of their bridge documents (Leung, 1996). As part of this system, inspectors are equipped with a video camcorder to take video and still photographs and a notebook computer to enter the rating data for each bridge and commentary. A more recent, Personal Digital Assistant (PDA) based field data collection system for bridge inspection is *Inspection On Hand* (IOH) (Trilon, 2004). IOH helps inspectors capture all rating information, commentary and sketches using hand-held, pen-based PDAs, and share data with Pontis bridge management system. In addition, The Digital Hardhat (DHH) is a pen-based computer with special multimedia facility reporting system software that allows the field worker to save multimedia information, such as text, sound, video and images, into a database. DHH technology enables dispersed inspectors to communicate information and to collaboratively solve problems using shared multimedia data (Stumpf et al., 1998).

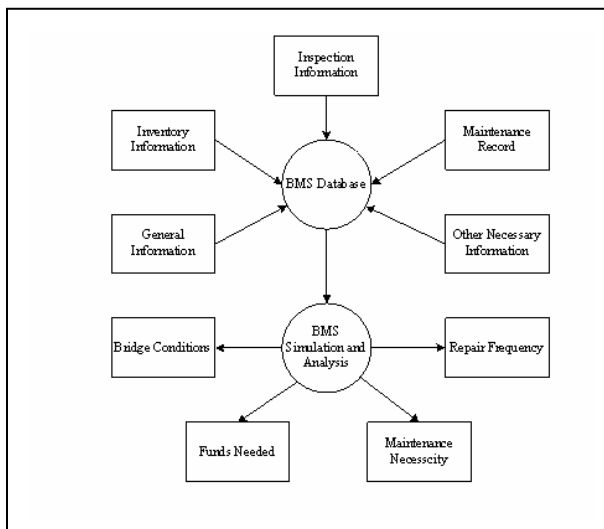


Fig. 1. General structure of BMSs

The proposed approach for MMBLMSs presented in this paper makes the first attempt to integrate 4D bridge models with BMSs and to make the resulting information accessible to mobile on-site workers. Although 4D models have already been built to support construction planning and scheduling (Zhang et al., 2000), these models are not integrated with Facilities Management (FM) or infrastructure management systems. In addition, there is no available system architecture to support interaction with these models in mobile situations.

3 REQUIREMENTS OF MMBLMS'S

Using mobile and wearable computers in the field under severe working and environmental conditions requires new types of interaction that increase the efficiency and safety of field workers. Research on systems aiming to provide information related to infrastructure at different stages of their lifecycle to mobile workers has been undertaken. Garrett *et al.* (2002) discussed the issues in delivering mobile and wearable computer-aided inspection systems for field users. Sunkpho *et al.* (2002) developed the Mobile Inspection Assistant (MIA) that runs on a wearable computer and delivers a voice recognition-based user interface. They also proposed a framework for developing field inspection support systems.

Mobility is a basic characteristic of field tasks. The inspector of a bridge has to move most of the time in order to do the job at hand. The inspector walks over, under or around the bridge, or in some cases climbs the bridge. Knowing the location of the inspector with respect to the inspected elements can greatly facilitate the task of data collection by automatically identifying the elements, and potentially specifying the locations of defects on these elements. Present methods of capturing location information using paper or digital maps, pictures, drawings and textual description can lead to ambiguity and errors in interpreting the collected data.

Location-Based Computing (LBC) is an emerging discipline focused on integrating geoinformatics, telecommunications, and mobile computing technologies (Beadle et al., 1997). LBC utilizes geoinformatics technologies, such as Geographic Information Systems (GISs) and tracking methods, such as the Global Positioning System (GPS), in a distributed real-time mobile computing environment. In LBC, elements and events involved in a specific task are registered according to their locations in a spatial database, and the activities supported by the mobile and wearable computers are aware of these locations using suitable positioning devices. For example, an inspection system based on LBC would allow the bridge inspector to accurately locate the cracks on a predefined 3D model of the bridge in real time without the

need for any post-processing of the data.

The first author (Hammad et al., 2004) discussed the concept and requirements of a mobile data collection system for engineering field tasks called *LBC for Infrastructure field tasks* (LBC-Infra) and identified its system architecture based on available technologies and the modes of interaction. This paper builds on the experience gained from the development and testing of LBC-Infra to propose a new methodology for designing future MMBLMSs that will integrate the different information about the lifecycle of a bridge (e.g., construction, inspection and maintenance schedules) to the 3D model of the bridge, resulting in 4D models. The following are the main requirements of MMBLMSs.

(1) 4D modeling and spatio-temporal analysis: 4D models will allow for spatio-temporal visualization and analysis that are not possible with present BMSs. 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 the data input error; (3) Facilitating data sharing; and (4) Improving the efficiency of database management. 4D visualization can be understood more quickly and completely than the traditional construction management tools (Fischer, 2001). This requirement is the first step towards future 5D or nD concepts for bridge management, which can incorporate other factors to the model, such as cost, to achieve more comprehensive data integration.

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 problems in site layout planning, or for predicting spatial conflicts, such as conflicts between workspaces (Akinci et al., 2002). Workspace analysis aims to create different types of workspaces for crew, equipment, and other required spaces in the work site, detect conflicts between these workspaces, and then resolve these conflicts.

(2) Lifecycle data integration: A uniform bridge inspection reporting system is essential to evaluate the condition of a structure correctly and efficiently, and to establish maintenance priorities. The results of an inspection must be accurately and fully recorded so that a complete history of the structure is available at any time. If available, all of the design information such as drawings, design calculations, soil investigation reports, etc. should be used to help at the inspection and maintenance stages. Different types of inspections

(inventory, routine, damage and in-depth inspections) allow a bridge owner to establish appropriate inspection levels consistent with the inspection frequency and the type of structure and details. On the other hand, for practical purposes, it is common to subdivide the inspection of a bridge into its main constituent parts, namely the inspection of the superstructure, substructure and foundations, and then to further subdivide these into their separate elements. Condition ratings assigned to elements of a component must be combined to establish the overall component condition rating.

(3) IFC standardization: Standardization is important for facilitating data sharing and exchange between all the groups involved in bridge management at all the stages of the lifecycle. The Industrial Foundation Classes (IFC) are an open international standard managed by the International Alliance of Interoperability (IAI) (IAI, 2004). 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. ISO announced acceptance of IFC as a common language in the construction industry in 2002. The IFC2x Platform Specification is now ISO/PAS 16739. 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 *IfcBridgeStructureType*, *IfcBridgeTechnologicalElementType*, *IfcBridgePrismaticElement* and *IfcBridgeBondingElementType*. 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 IFCs are necessary to cover the later stages of the lifecycle of structures. Hassanain (Hassanain et al., 2000) proposed an IFC-based data model for integrated maintenance management. The proposed approach includes entities such as *IfcCondition*, *IfcInspection*, *IfcRiskSchedule*, *IfcResource*, *IfcCostElement*, and so on.

There are two resource 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.

(4) Requirements of space and time scales: MMBLMSs should link all the information about the lifecycle of a bridge to a 4D model of the bridge incorporating different scales of space and time in order to record events throughout the lifecycle with suitable LoDs. In the field of computer graphic, the basic idea of LoDs is to use simpler versions of an object as it makes less and less contribution to the rendered image. 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 (Shamir and Pascucci, 2001). As for the time LoDs, different types of schedules have different time units, such as month, week, day, and hours.

(5) Databases requirements: MMBLMSs should support distributed databases while providing the required security management for the access and update of the data (de la Garza and Howitt, 1998; Liu et al., 2002). 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. A good combination of the two approaches is the object-relational approach to 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, 2004).

(6) Mobile and location-based computing and user interface requirements: MMBLMSs should support mobile and location-based computing by providing user interfaces that could be used on thin clients, such as PDAs and tablet PCs (Fujitsu, 2004), equipped with wireless communications and tracking devices, such as a GPS receiver. For example, in the case of a bridge inspector equipped with a mobile or wearable computer that has a tracking device, based on the location and orientation of the inspector and the task to be achieved, the system may display information about the parts of interest within the focus of the inspector or navigation arrows to the locations where cracks are most likely to be found. The spatial database of the bridge and the surrounding environment, and the tracking devices attached to the inspector, make it possible to locate structural elements and detected problems and provide navigation guidance to these objects. In addition, all newly collected information can be tagged in space.

Tacking technologies can be grouped into four categories (Karimi and Hammad, 2004): (1) active source systems; (2) passive source systems; (3) dead reckoning systems; and (4) hybrid systems (Azuma, 1997). Active source systems require powered signal emitters and sensors specially placed and calibrated. The signal can be magnetic, optical, radio, ultrasonic, or from the GPS satellites. The main passive source systems are electronic compasses, sensing the earth magnetic field, and vision-based systems that depend on natural light. Electronic compasses are small, inexpensive, and accurate. However, like magnetic sensors, they have the problem of magnetic distortion when in proximity to metals. Vision-based systems use video sensors to track specially placed marks. Dead reckoning systems do not depend on any external signal source. For example, an inertial system measures the linear accelerations and rotation rates resulting from gravity using linear accelerometers and rate gyroscopes, respectively. Hybrid systems use multiple measurements obtained from different sensors to compensate for the shortcomings of each technology when used alone. One possible hybrid system is to measure position by differential GPS and inertial tracking, and orientation by a digital compass and tilt sensors.

(7) Decision-support requirements: Bridge management tasks are in general knowledge-intensive tasks demanding specialized study and practical training. A simple “help” functionality is not suitable for MMBLMSs because the users are in mobile situations and will not have the time to browse the documents provided by such functionality. Therefore, the knowledge necessary for each task should be knowledge-engineered in a way that it could be readily accessible and applicable in a certain situation based on the task. Rule-based expert systems can be used to organize the knowledge pertaining to each group of tasks, e.g., inspection or maintenance, and these rules could be automatically activated in certain situations based on the context of the task that is executed using agents technology (Mizuno et al., 2002; Russell and Norvig, 2003).

4 PROTOTYPE SYSTEM DEVELOPMENT AND CASE STUDY

To demonstrate the feasibility and usefulness of the proposed methodology, a prototype system is developed and is discussed in detail in this section. In order to allow for information sharing on the Internet, Java programming language is used to build the system. Java is a platform-independent and versatile language, enabling developers to create applets that can be downloaded and run within a web browser while interacting with server-side applications. Java 3D is used to implement the 3D graphics

of the system (Walesh and Gehinger, 2001). Java 3D is a runtime API for developing portable applications and applets that can run on multiple platforms and multiple display environments.

This prototype system is designed to achieve the requirements discussed in Section 3 and to realize the following major functions: (1) Representing the 4D model of bridges with different LoDs; (2) Designing a window-based user-friendly interface with user access control; (3) Developing a comprehensive bridge databases including construction, inspection, and maintenance records.

The system integrates a 3D model of a bridge with object-relational database, GIS and tracking components, and multimedia equipment to develop a 4D model for BMS that can be used on-site in mobile situations for retrieving and updating information. IFC standard is referenced in the definitions of spatial and temporal objects. Using the 4D model, the user can directly interact with the system to get information on a certain stage of the lifecycle of a bridge. In addition, the system provides a rule-based expert system to support the decision-making related to inspection activities. Because of the large scope of the system, the discussion in the rest of the paper will be limited to the main features of the system focusing on the overall architecture and user-interface design.

4.1 Case Study

Jacques Cartier Bridge is chosen as the subject of the case study. Jacques Cartier Bridge is a five-lane bridge with about 2.7 km in length, spanning the St. Lawrence River between the cities of Montreal and Longueuil (Figure 2) (PJCCI, 2004). The bridge has a steel truss frame combined with prestressed concrete (PC) decking structure system. The super structure is seated on RC piers. Inaugurated in 1930, this bridge carries about 43 million vehicles per year with an annual increase rate of 2.4%, making it one of the busiest bridges in North America when considering traffic volumes per lane. Over the last 70 years, the bridge deck had suffered seriously from the increase of the number and load of trucks and the de-icing salts used extensively since the 1960s. Consequently, the deck has been replaced in 2001-2002. This replacement project is the most significant restoration project ever undertaken on a Canadian bridge. During two construction seasons in 2001 and 2002, the bridge underwent complete re-decking of the five lanes. 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.



Fig. 2. Jacques Cartier Bridge

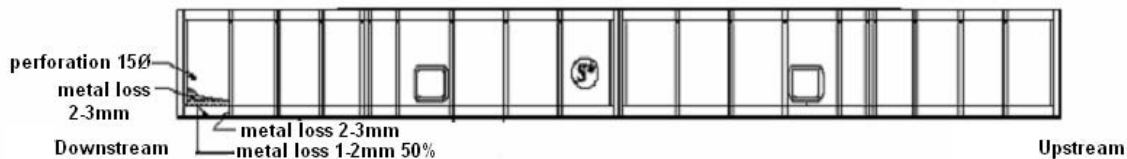


Fig. 3. Example of floor-beam inspection information

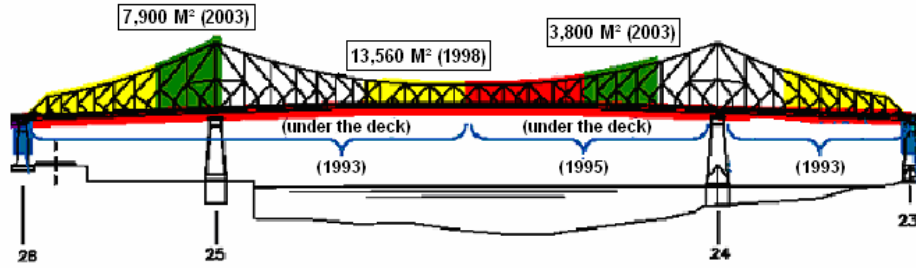


Fig. 4. Bridge painting history

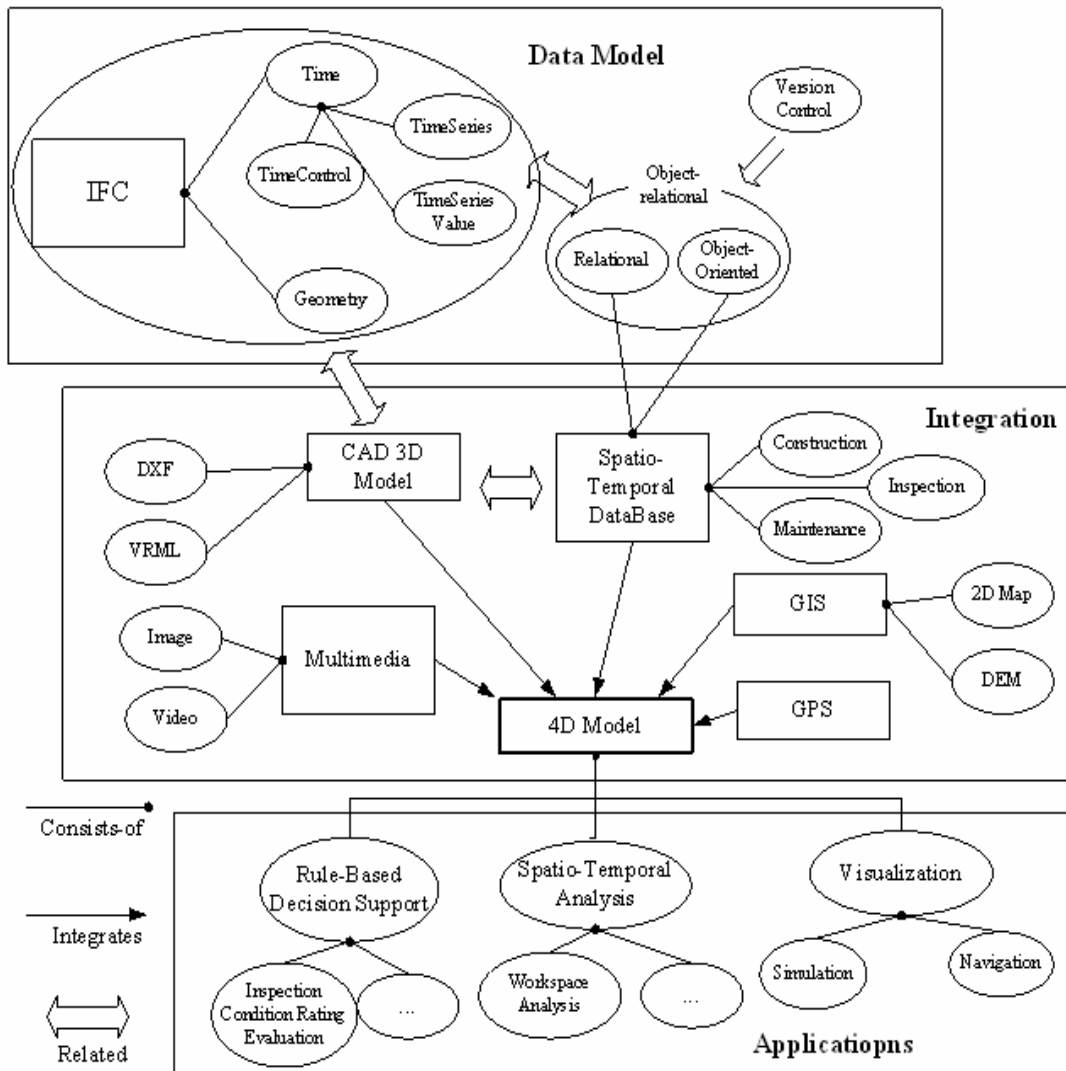


Fig. 5. General Structure of the prototype system

The bridge data were acquired from the bridge management authorities (Jacques Cartier and Champlain Bridges Incorporated) (PJCCI, 2004; Zaki and Mailhot, 2003). The data include AutoCAD drawings, deck rehabilitation schedules and inspection and maintenance records. Figure 3 shows part of the inspection data of a floor-beam including metal loss and perforation. Figure 4 shows main span painting maintenance history of the bridge until 2003. These data have been used in the development of the prototype system. We created several 3D models with different LoDs by converting the DWG file of the bridge into DXF (Data eXchange Format) and VRML (Virtual Reality Modeling Language) formats and extracting the information about the geometry and topology of the bridge elements into our database. We built the database to include data about different stages of construction, rehabilitation, and inspection. We also developed some applications to demonstrate the usefulness of the 4D approach, such as displaying elements with different colors according to construction, painting, or rehabilitation periods. In addition, we acquired the digital map and the DEM (Digital Elevation Model) data of Montreal to generate 2D and 3D maps (Clément, 2004).

4.2 Structure of the System

The data model in the system can be described as object-relational data model. It is developed using relational database and combined with Java, which is an object-oriented programming language. Data are stored in hierarchy from most detailed element, such as the deck panel to the main structure, such as the superstructure. Each object table is related to the sub or super tables. Apart from the structure, activities occurring during the lifecycle are linked with related objects' table to add detail about time, type of the activity, etc. The time entities in the database are defined based on the time resources definitions of IFC. Also definitions from IFC geometric model resources are used to create multi-representation of the 3D bridge model with different LoDs. A 3D map is created in the system to permit the computations based on the location of the users. The model of the bridge is based on geographic global coordinate system using Modified Transverse Mercator Projection (MTM) and the bridge model. The 3D map consists of a 2D layers draped on the DEM model 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 explained in Section 3. The location of the user is reflected on the map

as well as in the 3D browser. A digital video camera can be also connected with the system to facilitate image and video capturing.

With the integrated 4D model, the system can provide many functions, such as visualization, analysis, and decision-making support. Visualization has powerful functions for interacting with the system in a virtual reality or augmented reality modes (Hammad et al., 2004). Users can query the database through the GUI or by picking on 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 provided in the system. Other important applications are the 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 and resolved using a rule-based expert system approach. The expert system approach is also used in a decision support sub-system for inspection condition rating using Java Expert System Shell (JESS) (Friedman-Hill, 2003).

4.3 Database Design

Java Database Connectivity (JDBC) is a programming framework for Java developers writing programs that access information stored in databases. Our system has options to connect with several database management systems (DBMS) like Oracle, Informix, Microsoft Access, MySQL, etc. The commands to be executed by the DBMS on a database are based on SQL (Structured Query Language). In addition, in order to allow the system to interoperate with other applications, we use an IFC data structure representing bridge 3D objects.

The database of the 4D model is designed with Microsoft Access XP to present the information of all the truss and deck components. The name, type, dimensions, location, properties, and the starting and ending dates of the construction or maintenance activities of each member are defined in the corresponding tables.

In order to avoid security restrictions resulting from the applet accessing the database directly, we use three-tier solution where the applet is only responsible for display, and will introduce mid-tier for all application logics that related to data retrieving/updating, etc. The mid-tier can be realized using a servlet or a middleware application (CORBA or EJB) between the applet and the database. Another method to bridge between the front-end application and IFC or XML data is Web services where all data are saved in a central database. A web service application will provide services to query 3D bridge objects. The users can license the 3D model API to hook up their applications with the web services.

4.4 GIS and Tracking Components

A GIS sub-system is created using MapObjects Java Edition (ESRI, 2004). The purpose of adding the 2D map of Montreal is to provide information to the users of the system (e.g., bridge inspectors) about their positions and the environment around them. The map includes several layers related to Montreal City, such as a boundary layer and layers for the roads, rivers, and administrative areas. The GIS has the main functions for zooming and retrieving information about the attributes of different layers. In addition, to locate the bridge model on the map, we added the same map of Montreal and the DEM to our 3D browser.

Finding the location of the user is achieved using Differential GPS (DGPS) or video tracking. DGPS is based on correcting the effects of the pseudo-range errors caused by the ionosphere, troposphere, and satellite orbital and clock errors by placing a GPS receiver at a precisely known location (base station). The pseudo-range errors are considered common to all GPS receivers within some range. DGPS has a typical 3D accuracy of better than 3 m and an update rate of 0.1-1 Hz. Real-time kinematic GPS (RTK-GPS) receivers with carrier-phase ambiguity resolution can achieve accuracies better than 1 cm (Kaplan, 1996). We are testing the system with Trimble 5700 RTK GPS receiver. We are also using an Augmented Reality toolkit, called ARToolKit (Hirokazu, 2000), to track markers by means of a video camera. This method has many limitations on the accuracy of tracking and the range for recognizing a marker, which varies with the marker size. For example, a marker with an edge size of about 7 inches can be recognized from a distance of about 4 feet. On the other side, the GPS can be used only under the condition of having direct line of sight to at least four satellites.

4.5 User Interface Design

4.5.1 Design approach

Several design patterns are used in developing applications, such as MVC (Model-View-Control), observer, proxy, and façade patterns. The MVC design pattern is used for the whole system design (Potel, 1996). MVC is a widely used software design pattern that enforces the separation between the input, processing, and output of an application. Each of these components handles a discrete set of tasks, enabling loose coupling and the ability to change one component without affecting the others.

4.5.2 Development environment and 4D model

The Graphical User Interface (GUI) of the system was

developed using Java Swing classes. Java is a cross-platform language and the GUI components may have different sizes depending on the platform. Therefore, creating a proper layout manager is extremely important. A layout manager controls the size and position of *Components* in a *Container*. Because the screen space of mobile computers is limited, tabbed panes are used to organize the different data items necessary at each stage of the lifecycle of bridges.

The 4D 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. At this stage, only the bridge truss and the deck panels are considered. 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 forming a tree structure in parent-child relationships. BranchGroup objects are used to form scene graphs. TransformGroup objects can be constructed by applying Transform3D objects, which represent transformations of 3D geometry such as translations and rotations.

The UI of the system is shown in Figure 6. On the right side, there is a time input interface that allows the user to query the database about events that happened during a specific period (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?). The starting and ending dates of a period can be input using a calendar interface or sliding bars, and the 3D model will present the corresponding elements with different colors representing the progress ratio. In addition, the user can directly manipulate the 3D bridge model (e.g., zooming and rotation) and select an element of the bridge by clicking on that element (picking). Upon selection, the element will be highlighted and the related information about the element will be displayed. Alternatively, the user can select an element from the database interface and the element will be highlighted in the model.

4.5.3 Navigation interface

Three navigation behaviors are currently available in the system: drive, fly, and orbit behaviors. These behaviors use the pointing device (digital stylus or mouse) to control the view platform motion. Each button on the pointing device generates a different type of motion while the button is pressed. The distance of the cursor from the center of the canvas3D controls the speed of motion. The drive behavior allows the user to move to any point in the 3D space, with pointer controls for moving forward, backward, and translation along the *X*, and *Y* axes as shown in Figure 7.

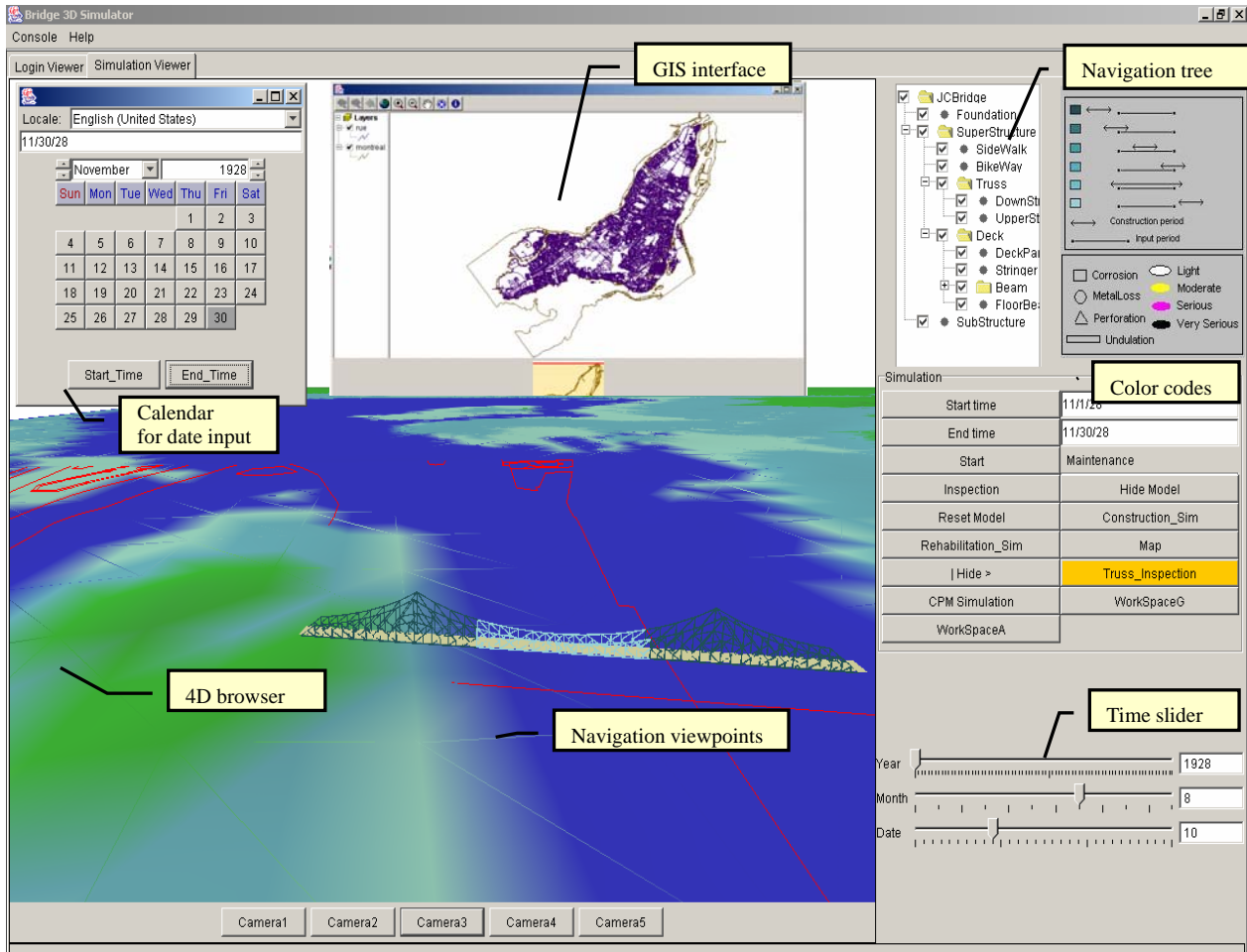


Fig. 6. Screen shot of the user interface of the prototype system

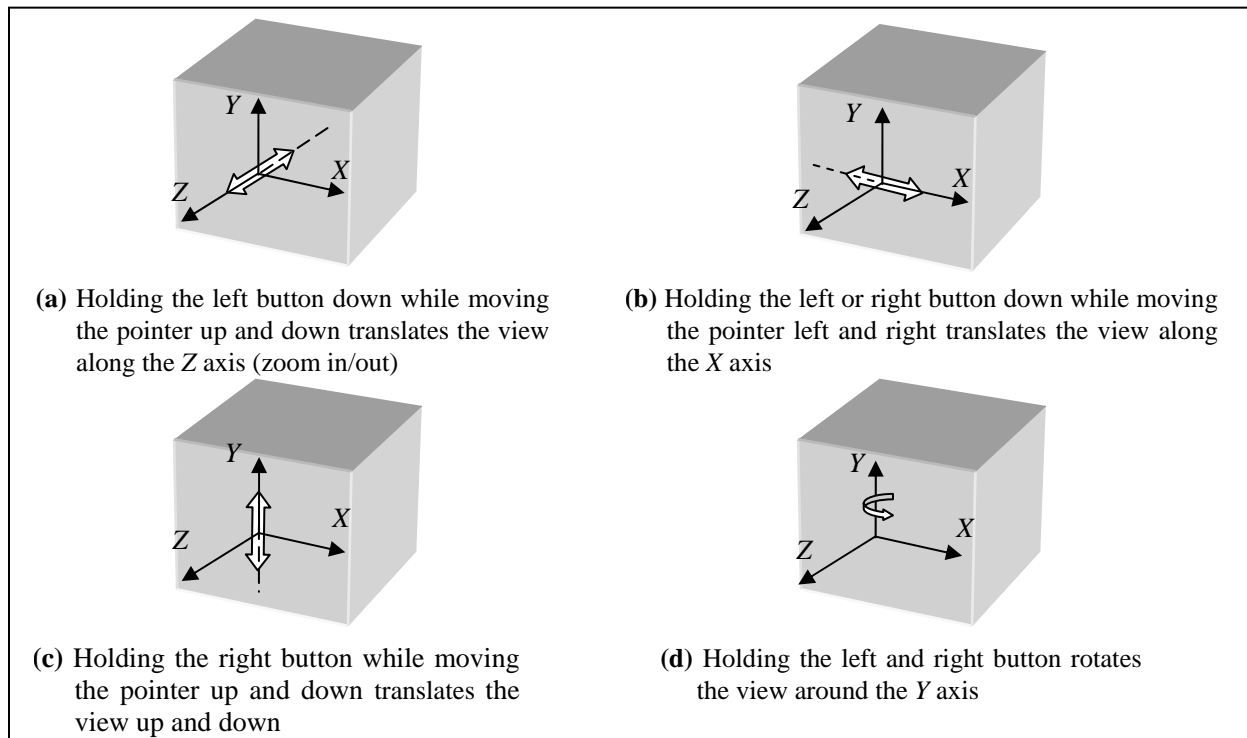


Fig. 7. Drive navigation behaviors

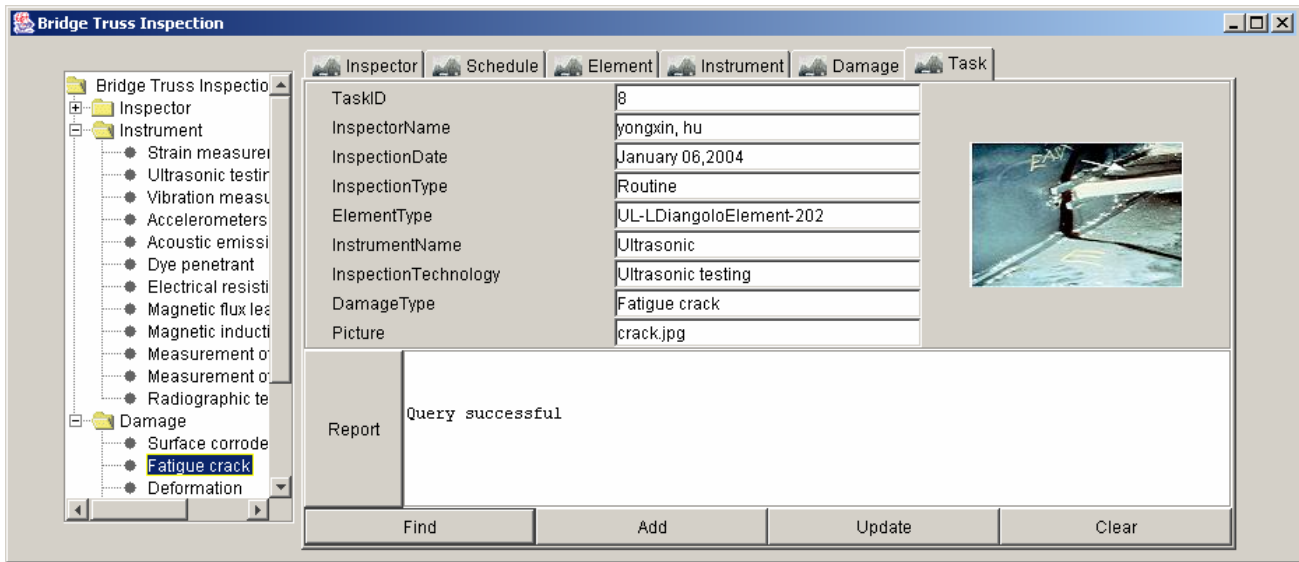


Fig. 8. Inspection task report tabbed pane

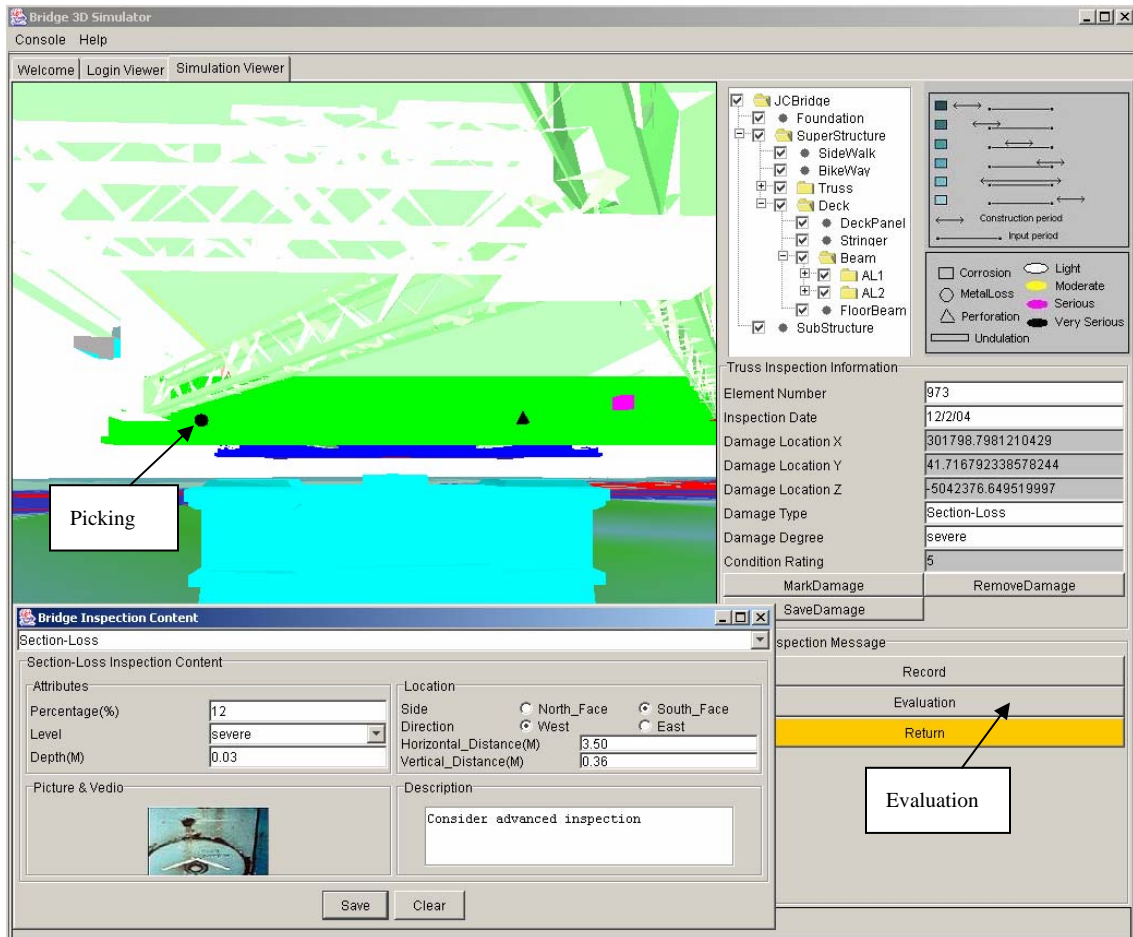


Fig. 9. Inputting the defect location by picking

4.5.4 Inspection user interface

The inspection user interface is explained here as an example of the interaction methods used in the system at different stages of the lifecycle. An inspector can apply inspection procedures through a number of ordered tabbed panes. The panes are *Inspector*, *Schedule*, *Element*, *Instrument*, *Damage*, and *Task*. In the first two tabbed panes, some general inspection information needs to be input about the inspector and schedule. The user can find, add, and update the bridge inspection data by querying the database. In the *Element* tabbed pane, inspector can choose the exact element to inspect according to customized inspection scheme by *picking* the element on the 3D model at the approximate location of the defect. In the *Instrument* tabbed panel, a suitable inspection tool can be selected depending on the type of damage. *Damage* pane is the core part of the bridge inspection interface. Video/image capture functionality is also available using Java Media Framework (JMF) API (JMF, 2004). The last tabbed pane, *Task* (Figure 8), is to summarize the previous inspection information for future assessment.

Figure 9 shows an example of *picking* a floor beam element on the 3D model at different locations to input the location of defects. Picking is the process of selecting shapes in the 3D virtual world using the 2D coordinates of the pointer on the Canvas3D. The PickCanvas class is used to turn the mouse location into an area of space, or

PickShape, that projects from the viewer through the mouse location into the virtual world. When a pick is requested, Java 3D figures out the pickable shapes that intersect with the PickShape. The pick returns a list of PickResult objects, one for each shape that is intersected by the PickShape. After getting the face that is picked, the normal vector of that shape is created and the damage on the surface is added on that surface based on the direction information obtained from the normal vector. The damage will be automatically marked on the 3D model of the element using a specific shape and color, which are defined based on the damage type and deterioration degree, respectively. For example, in Figure 9, black sphere represents severe section-loss.

Bridge inspection is a knowledge-intensive process. In order to support the inspectors using the system, the Java hyperlink functionality was added to allow the user to access inspection manuals in Hypertext Markup Language (HTML) format, such as “Bridge Inspector’s Reference Manual” (FHWA, 2002). The link is context sensitive and will extract only the relevant information. In addition, a rule-based expert system (Friedman-Hill, 2003) is used to analyze the collected damage data and to calculate the element condition rating. Because of space limitation, the details of the expert system development are not discussed in this paper. In the future, other functions such as drawing sketches and generating history reports will be added to the system.

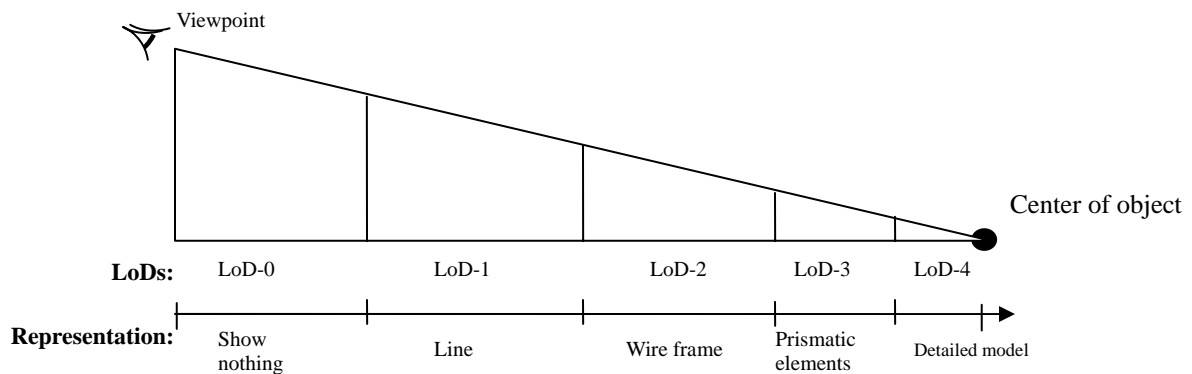


Fig. 10. Relationship between distance and LoDs

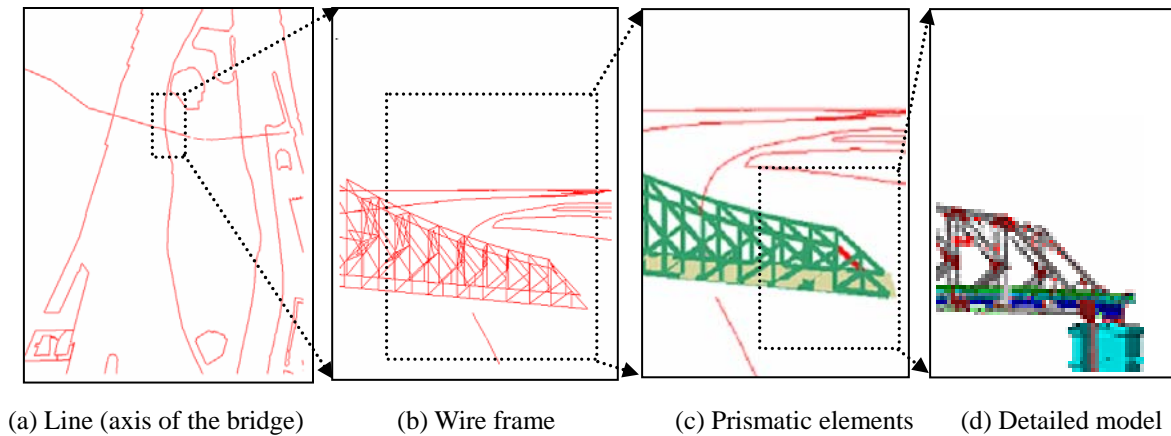


Fig. 11. Different spatial LoDs of the bridge

4.5.5 Levels of Details (LoDs)

Four different LoDs for the shape can be used in this system. Line, wire frame, prismatic elements, and detailed VRML objects are used according to the distance between the model and the viewer to optimize the performance of the system. As shown in Figures 10 and 11, 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 objects, sequentially. The concept of LoDs is also used to control the display of the damages.

Different temporal levels of detail 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 4. The inspection time is usually represented by the date of inspection. Higher time resolution is used for inspection purposes to record damages that could happen in a very short time.

5 CONCLUSIONS

This paper proposed a new type of Mobile Model-based Bridge Lifecycle Management Systems (MMBLMSs) and discussed the requirements for developing such systems. The developed prototype system integrates 3D graphics and a database to realize the 4D model of Jacques Cartier Bridge. The preliminary testing of the system and its user interface showed that it has good potential for realizing future MMBLMSs. Further development and testing of the system in practical situations are necessary to improve the functionalities and usability of the system.

ACKNOWLEDGMENTS

We would like to thank Basheer Khabir, Khaled El Ammari, Taotao Yong, XinSong He, Xiaoming Li, Tao Wang and Xiaohui Xiao from Concordia University and Yanlin Wang from Borland Software Corporation for their contribution in developing the prototype system. Also, the cooperation of Mr. Guy Mailhot from Jacques Cartier and Champlain Bridges Incorporated, Mr. Adel Zaki and Mr. Raymond Coté from SNC Lavalin in providing the data about Jacques Cartier Bridge and Mr. A. Clément from the IT department of Montreal Municipality in providing the GIS data of Montreal are greatly appreciated.

REFERENCES

- Akinci B., Fischer M., Levitt R., Carlson R. (2002). Formalization and Automation of Time-Space Conflict Analysis, *ASCE, Journal of Computing in Civil Engineering*, **16**(2), 124-133.
- Azuma, R.T. (1997). A Survey of Augmented Reality, *Presence: Teleoperators and Virtual Reality*, **6**(4), pp. 355-386.
- Beadle, H.W.P., Harper, B., Maguire, G.Q., Judge, J. (1997). Location Aware Mobile Computing, *Proc. IEEE/IEE International Conference on Telecommunications*, (ICT97), Melbourne, April, 1319-1324.
- Clément, A. (2004), «Normes de production de la géobase régionale des tronçons de voies publiques avec fourchettes d'adresses» IT Department, Ville de Montréal, Internal Document, March 2004.
- Czepiel, E., Bridge Management Systems, Literature Review and Search http://www.iti.northwestern.edu/publications/technical_reports/tr11.html
- ESRI (2003). "Getting Started with Mapobjects_Java."

- <www.esri.com/software/mojava/download.html>
- FHWA (2000), 'Bridge Inspector's Reference Manual' Publication No. FHWA NHI 03-001.
- Introduction to 4D Research, Fischer, M. (2001) <<http://www.stanford.edu/group/4D>>.
- Friedman-Hill, E. (2003). "Jess in Action: Java Rule-based Systems", ISBN 1930110898.
- Fujitsu web site: <<http://www.fujitsu.com>>.
- Garrett, J.H. Jr., Bürgy, C., Reinhardt, J. and Sunkpho, J. (2002). An Overview of the Research in Mobile/Wearable Computer-Aided Engineering Systems in the Advanced Infrastructure Systems Laboratory at Carnegie Mellon University. *Bauen mit Computern*, Bonn, Germany. VDI Verlag GmbH, Duesseldorf, Germany.
- de la Garza, J.M. and Howitt, I. (1998). Wireless Communication and Computing at the Construction Jobsite, *Automation in Construction*, Vol. 7, 327-347.
- Hammad, A., Garrett, J.H. and Karim, H. "Location-Based Computing for Infrastructure Field Tasks," (2004). In Karim, H. and Hammad, A. (editors), 'Telegeoinformatics: Location-Based Computing and Services,' CRC Press.
- Hassanain, M. Froese, T. and Vanier, D. (2000). "IFC-based Data Model for Integrated Maintenance Management". 8th International Conference on Computing in Civil and Building Engineering (ICCCBE-VIII), Stanford University, California, USA, August, 14-17.
- Hirokazu K., Mark B., Ivan P. (2004). ARToolKit version 2.33: A Software Library for Augmented Reality Applications, <http://www.hitl.washington.edu/artoolkit/>
- IAI, (2004), <http://www.iai-international.org/iai_international>
- IFC-BRIDGE (2004). <<http://www.iai-france.org/bridge>>
- Java Media Framework API web site <<http://java.sun.com/products/java-media/jmf/index.jsp>>.
- Kaplan, E.D. (1996). *Understanding GPS: Principles and Applications*, Artech House.
- Karim, H. and Hammad, A. (editors) (2004). "Telegeoinformatics: Location-Based Computing and Services," CRC Press.
- Kuo, S.S., Clark, D.A. and Kerr, R. (1994). Complete Package for Computer-Automation Bridge Inspection Process. *Transportation Research Record*, No. 1442, 115-127.
- Leung, A. (1996). Perfecting Bridge Inspecting. *Civil Engineering Magazine*, 59-61.
- Liu, D., Cheng, J., Law, K.H. and Wiederhold, G. (2002). An Engineering Information Service Infrastructure for Ubiquitous Computing, *CIFE Technical Report #141*, Stanford University.
- Mizuno Y., Abe M., Fujino Y., Abe M. (2002). Development of Interactive Support System for Visual Inspection of Bridges, International Forum on Infrastructure Maintenance, November, 2003. Tokyo, Japan.
- Object-Relational Mapping (2004). <http://www.service-architecture.com/java-databases/index.html>.
- PJCCI (2004). The official website of Jacques Cartier and Champlain Bridges Incorporated. <<http://www.pjcci.ca/English/jacques-cartier/>>.
- Potel, M. (1996). MVP: Model-View-Presenter, The Taligent Programming Model for C++ and Java, <<ftp://www6.software.ibm.com/software/developer/library/mvp.pdf>>
- Ryall, M.J. (2001). "Brdige Management," ISBN 0 7506 5077X.
- Russell, S. and Norvig, P. (2003). Artificial Intelligence, A modern Approach, Prentice Hall.
- Shamir A. and Pascucci, V. (2001). Temporal and Spatial Level of Details for Dynamic Meshes, ACM, VRST01, November 2001, Banff.
- 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.
- Sunkpho, J., Garrett, J., and McNeil, S. (2002). A Framework for Field Inspection Support Systems Applied to Bridge Inspection, Proceedings of the 7th International Conference on the Applications of Advanced Technologies in Transportation, Cambridge MA, August 5-7, 417-424.
- Trilon, Inc. web site: <<http://www.trilon.com/>>.
- Walesh, A. and Gehringer, D. (2001). Java 3D, API Jump-start, Prentice Hall PTR.
- Zaki, A., R. and Mailhot, G. (2003). Deck Reconstruction of Jacques Cartier Bridge Using Pre-cast Pre-stressed High Performance Concrete Panels. *PCI Journal*, September-October, 2-15.
- Zhang, J., Anson, M., and Wang, Q. (2000), A New 4D Management Approach to Construction Planning and Site Space Utilization. *Computing in Civil and Building Engineering*, ASCE, 15-22.