Computer-Aided Civil and Infrastructure Engineering **21** (2006) 530–547

Mobile Model-Based Bridge Lifecycle Management System

Amin Hammad*

Concordia Institute for Information Systems Engineering, 1425 René Lévesque Boulevard, Montréal, Québec, H3G 1T7, Canada

Cheng Zhang & Yongxin Hu

Department of Civil, Building & Environmental Engineering, Concordia University, Montréal, Québec, H3G 1M8, Canada

&

Elaheh Mozaffari

Department of Electrical & Computer Engineering, Concordia University, Montréal, Québec, H3G 1M8, Canada

Abstract: *This article discusses the requirements for developing a mobile model-based bridge lifecycle management system (MMBLMS). This new system should link all the information about the lifecycle stages of a bridge (e.g., design, construction, inspection, and maintenance) to a 4D model of the bridge incorporating different scales of space and time to record events throughout the lifecycle with suitable levels of details (LoDs). In addition, MMBLMS should support distributed databases and mobile location-based computing by providing user interfaces that can be used on mobile computers, such as tablet PCs. A framework for MMBLMS is described and the basic computational issues for realizing it are discussed including the navigation modes, the picking behavior and the LoDs for representing bridge elements and defects. A prototype system developed in Java language is used to demonstrate the feasibility of the proposed methodology for realizing this system.*

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

1 INTRODUCTION

Bridge lifecycle management aims to perform the management functionalities related to bridges from the conceptual stage to the end of their useful life, through the design, construction, operation, and maintenance stages. This article investigates the possibility of extending the functionalities of present bridge management systems (BMSs) in two directions: (1) linking all the information about the lifecycle stages of a bridge (e.g., design, construction, inspection, and maintenance) to a 4D model of the bridge incorporating different scales of space and time to record events throughout the lifecycle; and (2) providing user interfaces that facilitate using the 4D models on mobile computers, such as PDAs and tablet PCs equipped with tracking devices, such as Global Positioning System (GPS) receivers. The proposed new system is called mobile model-based bridge lifecycle management system (MMBLMS).

The article starts by reviewing conventional bridge management systems (BMSs) and recent trends in 4D models, mobile computing and Location-Based Computing (LBC). This is followed by an analysis of the requirements of the proposed MMBLMS. Special consideration is given to the spatial and temporal issues, such as the requirements to support navigation, picking behavior, and different Levels of Details (LoDs), and to adopting available standards for interoperability. A framework of MMBLMS is described and the basic computational issues for realizing it are discussed. Then, a prototype system developed in Java language is discussed in detail to demonstrate the feasibility of the proposed methodology. A case study of Jacques Cartier Bridge in Montreal is also demonstrated.

2 REVIEW OF BMSs

The major tasks in bridge management are: (1) collection of inventory data, (2) inspection, (3) assessment of condition and strength, (4) decisions about repair, strengthening or replacement, and (5) prioritizing the allocation of funds. BMSs are means of managing information of bridges to support decision-making that assures their long-term health and to formulate maintenance programs in line with budgetary constraints and funding limitations. BMSs include four basic components: data storage, cost and deterioration models, optimization and analysis models, and updating functions (Czepiel, 2004; Ryall, 2001). The core part of a BMS is a database that 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 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.

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 setup 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 videos and pictures, 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 the Pontis bridge management system. In addition, the Digital Hardhat (DHH) is a pen-based computer with a special multimedia reporting system 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).

Based on our literature review and to the best of our knowledge, the proposed approach for MMBLMS presented in this article 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 (IMSs). In addition, there is no available system architecture to support the interaction with these models in mobile situations.

3 REQUIREMENTS OF MMBLMS

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 to do the job at hand. The inspector walks over, under or around the bridge, and 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 focusing 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 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 with minor 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 their modes of interaction. This article builds on the experience gained from the development and testing of LBC-Infra to propose a new methodology for designing future MMBLMS 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 paragraphs briefly discuss the main requirements of MMBLMS. These requirements are based on interviews with bridge managers and on our experience with previous prototypes of LBC-Infra (Hammad et al., 2004). Because of the broad range of these requirements, some of them will be considered in our future work and will not be discussed in the rest of the present article.

1. **4D modeling and spatio-temporal analysis:** 4D models facilitate spatio-temporal visualization and analysis that are not possible in present BMSs. This integration of space and time results in the following advantages: (1) visualizing different types of data, for example, 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 errors; (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). The Stageworks (Stageworks, 2005) system developed by Bechtel has proved that 4D visualization is helpful during construction. The Navigator software also applies 3D model review, animation and 4D simulation (Bentley, 2005). This requirement is the first step toward 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 in time. This type of analysis is useful for evaluating the suitability of a certain location 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 worksite, to detect conflicts between these workspaces, and then to 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, and so on should be used to help at the inspection and maintenance stages (Itoh et al., 1997). Different types of inspections (inventory, routine, defect and in-depth inspections) allow the 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 subdivide these parts 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:** The interoperability of the MMBLMS is of paramount importance because it is usually developed and used by a large number of groups in a spatially and temporally distributed fashion. Therefore, 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 standard called Industry Foundation Classes (IFC) can help in achieving the interoperability of MMBLMS. IFC is an open international standard managed by the International Alliance of Interoperability (IAI) (IAI, 2004). In IFC2×2 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 the acceptance of IFC as a common language in

the construction industry in 2002. The IFC2× Platform Specification is now ISO/PAS 16739. The IFC-Bridge project aims to extend ISO/PAS 16739 by defining a standard representation for bridge lifecycle 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 IFC are necessary to cover the later stages of the lifecycle of structures. 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 resources related to time in the Resource layer of IFC: IfcDateTimeResource and IfcTimeSeriesResource. In IfcDateTimeResource, calendar date and local time are defined. IfcTimeSeriesResource is new in IFC2×2. 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:** One of the long-term goals of MMBLMS is to link all the information about the lifecycle of a bridge to a 4D model of the bridge incorporating different scales of space and time to record events throughout the lifecycle with suitable 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). As for the time LoDs, different types of schedules have different time units, such as month, week, day, and hour.
5. **Database requirements:** A large project needs to store pertinent data for the lifecycle that can be used at every stage to help managers plan and organize their work efficiently. MMBLMS should support distributed databases while providing the required security management for accessing and updating 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 because of their flexible data structure. A good combination of the two approaches is the object-relational approach for database development (McClure, 1997) 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:** MMBLMS should support mobile and location-based computing by providing a user interface that can 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 can display information about the parts of interest within the focus of the inspector or display 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.

Tracking 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's 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 markers. 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. Differential GPS (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 1–10 Hz. Real-time kinematic GPS (RTK-GPS) receivers with carrier-phase ambiguity resolution can achieve accuracies better than 3 cm (Kaplan, 1996).

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 MMBLMS because the users are in mobile situations and do 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 is 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, for example, inspection or maintenance, and these rules can be automatically activated in certain situations based on the context of the task (Hu and Hammad, 2005; Mizuno et al., 2002; Russell and Norvig, 2003).

4 FRAMEWORK FOR MMBLMS

4.1 General structure of the framework

The general structure of the framework is shown in Figure 1. This structure 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.

1. **Object-relational data model:** The data model in the framework is an object-relational data model. 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 the tables of related objects to add details about time, type of activity, and so on. The time entities in the database are defined based on the time resources definitions of IFC. In addition, definitions from IFC geometric model resources are used to create multi-representation of the 3D bridge model with different LoDs. 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 2a 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. First, the root of the tree is found by querying the database about the group node named "Root" and the root node is created. Then, queries are applied recursively to find other element nodes based on the data stored in the table.
2. **Technology integration:** 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 MMBLMS 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. 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 DGPS 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 a certain location) using the concepts of LBC explained in Section 3. The location of the user is reflected on the 2D map and in 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.
3. **Applications:** With the integrated 4D model, the framework can be used to develop many applications, 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 a specific element, and can get the results as visual feedback in the 4D model, for example, information about the painting or

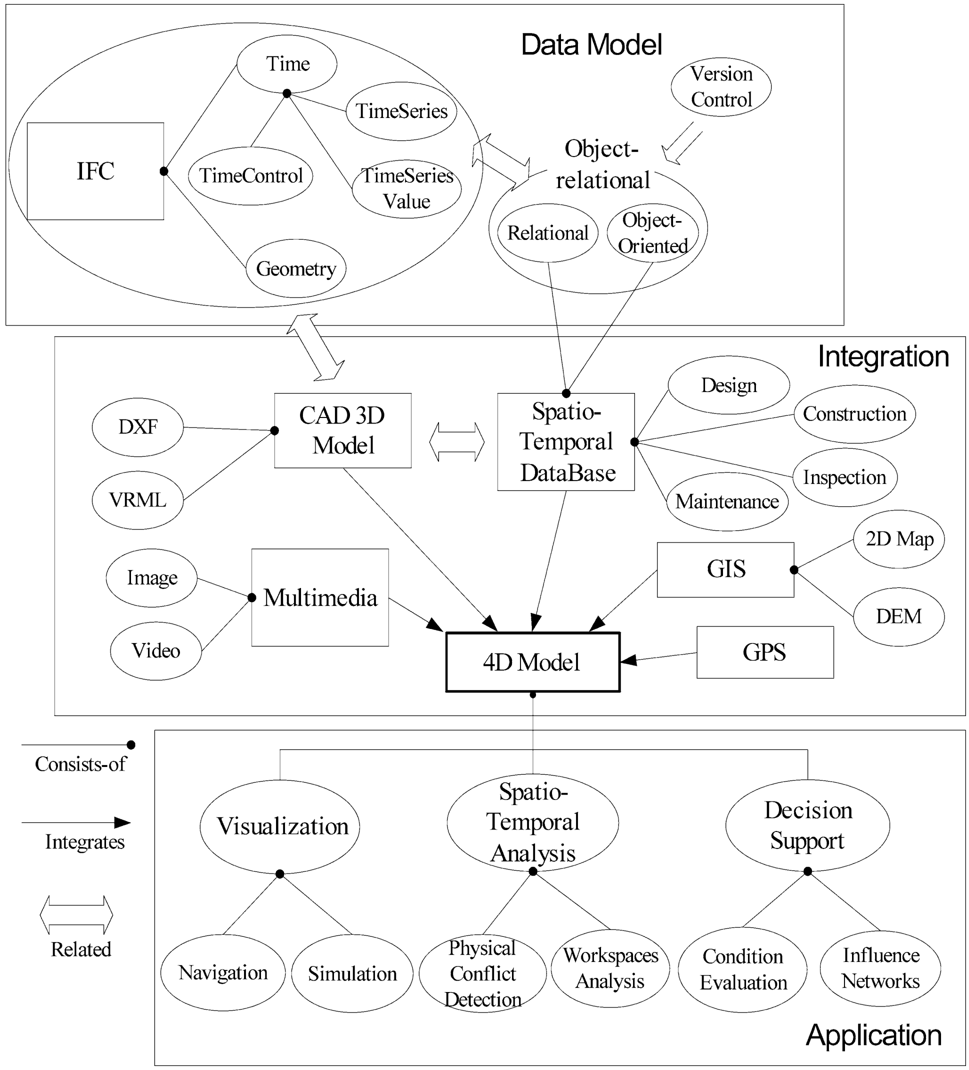


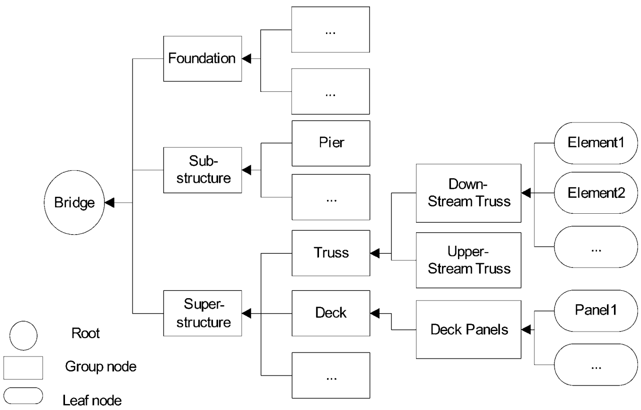
Fig. 1. General structure of the framework.

rehabilitation history. Users can easily navigate in the 3D space using navigation tools (explained in detail in Section 4.2.1). 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 and resolved using a rule-based expert system approach (Guo, 2002).

Several design patterns are used in developing the user interface, such as MVC (Model-View-Control), observer, proxy, and façade patterns. The MVC design pattern is selected for the present framework (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.

4.2 Computational aspects of the framework

4.2.1 Navigation modes. Research about navigation in 3D spaces aims to facilitate the navigation by constraining the degrees of freedom of movement, such as terrain-following algorithms (Zhao et al., 2001) or providing a tree structure representation of the model (Reinhardt et al., 2004). Two modes of navigation are suggested in our framework: logical navigation and graphical navigation. These modes take advantage of the available navigation methods and algorithms while considering the specific needs of MMBLMS. The logical navigation is represented by a hierarchical tree, which includes the



(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)

Fig. 2. Example of the object tree (a), and its table representation (b).

structure of the bridge as extracted from the object-relational database. The graphical navigation is achieved by interacting with the 3D model. Three navigation behaviors are investigated for the framework: drive, fly, and orbit behaviors. These behaviors use the pointing device (e.g., 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 coordinate system controls the speed of motion. As an example of the navigation behaviors, the drive behavior allows the user to move to any point in the 3D space, with pointer controls for translations along the X, Y, and Z axes and rotation around the Y axis as shown in Figure 3. Furthermore, constraints can be added so that the navigation is restricted to a certain surface, such as the digital terrain or the surface representing the top of the deck of a bridge.

4.2.2 Location-based automatic bridge selection. Retrieving bridge information from the BMS database in real time may not be efficient because of the large number of bridges and bridge elements. In this section, as a first step toward facilitating the automatic retrieval of relevant inspection information, a location-based automatic bridge selection algorithm integrating GIS and GPS is developed based on the distance between the user and a set of bridges represented by their center points. This algorithm can be extended in the future to the more general case of retrieving information about bridge elements based on a 3D spatial model.

Figure 4 shows a conceptual diagram of how the bridges are selected based on the distance between the inspector and a set of bridges. As the inspector moves

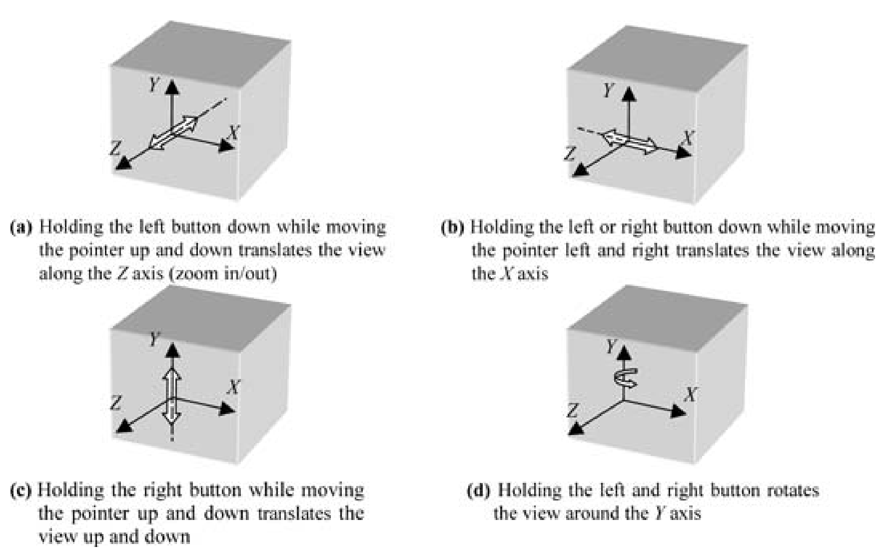


Fig. 3. Drive navigation behaviors.

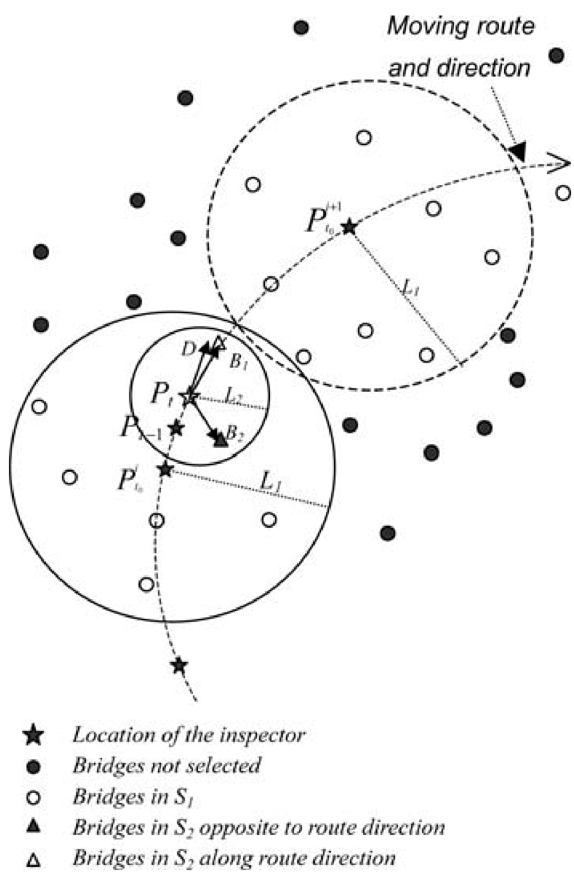


Fig. 4. Selecting the nearest bridge using GIS and tracking.

from one location to another, the set of bridges is selected in three steps: (1) A larger set of bridges (S_1) is periodically selected (every $\Delta t = T$) within a distance L_1 from the inspector's position (P_{t_0}) at initial time t_0; (2) A smaller set of bridges (S_2) is continuously selected from S_1 within a smaller distance L_2 from the inspector's position (P_t) at current time t; and (3) A final set of bridges (S_3) is selected from S_2 by choosing only those bridges that are within the field of view of the inspector. To avoid excessive and unnecessary updates of the position based on GPS data, the required frequency (f) for updating the GIS map as well as the minimum distance ($d_{\min}$) between two successive positions can be set. In addition, the quality of the position information is considered based on the Dilution of Precision (DOP). DOP is the geometric effect caused when satellites visible to the GPS receiver are too close to each other. When the satellites are further apart, the position obtained from the GPS is more accurate. The DOP can range from an ideal value of 1 to the least accurate value of 50. The maximum allowed DOP value ($DOP_{\max}$) required for updating the position can be set to assure the reliability and accuracy of the position data. The values of L_1, L_2, and T are specified depending on the speed of movement of the inspector. Assuming the inspector is moving at an average walking speed, the following values could be used: $T = 10$m, $L_1 =$ 1,000 m and $L_2 = 100$ m. The initial time t_0 is set to time t, which is the current time at the end of every period of length T. A set of bridges S_1 is selected within a distance L_1 from P_{t_0}. A new position (P_t) read from the GPS is considered only if the DOP is less than $DOP_{\max}$. The location P_t is then converted from Latitude/Longitude (LL) coordinates to the map projection coordinates. If P_t satisfies the conditions of f and $d_{\min}$, the current position is set at P_t. If the time difference ($\Delta t = t - t_0$) is greater than T, t_0 is reset to the current time and a new set of bridges S_1 within distance L_1 from the new initial position $P_{t_0}^{i+1}$ is selected. Otherwise, another set of bridges S_2 within L_2 from P_t is selected from S_1. The third step in the selection is to select only those bridges (S_3) that are in the semicircle corresponding to the direction of movement of the inspector ($\vec{D}$). These bridges can easily be found by noticing that the dot product of the vector $\vec{D}$ and a vector $\vec{B}$ connecting the present position P_t and the location of a bridge satisfying the above condition will have a positive value (Equation 1).

$$\vec{D} \bullet \vec{B} > 0 \tag{1}$$

As an example of this selection, bridge B_1 in Figure 4 will be selected in S_3, while bridge B_2 will be eliminated from S_3. The bridges in S_3 are listed in the user interface in the order of increasing distance from P_t and the information regarding these bridges is retrieved from the BMS database. This process is repeated until the tracking is stopped.

4.2.3 Picking behavior. Interaction with the 3D 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. To interactively retrieve or update information related to the picked element, it is important to know the location and the orientation of that element in the 3D environment of the virtual model. Figures 5 and 6 show the flowchart and an example of the picking behavior, respectively. A pick shape is selected as the picking tool. The pick shape can be a ray, segment, cone, or cylinder. The pick shape extends from the viewer's eye 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 has to be found. After the closest object (O) is found, the surface (F) that faces the user should 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)

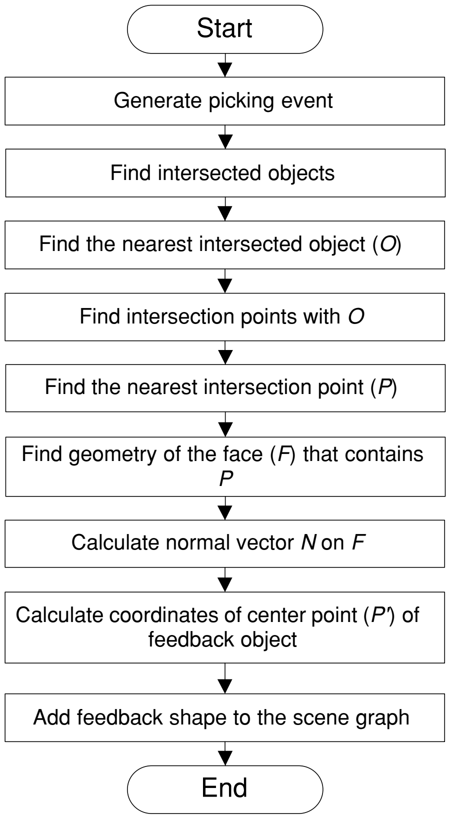


Fig. 5. Flowchart of picking and adding defects.

of surface (F) can be calculated based on the current coordinates. 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:

$$\overrightarrow{P} = \overrightarrow{P} + \text{offset} \times \overrightarrow{N} \quad (2)$$

The following example of the visual feedback based on picking is given to illustrate the method of calculation (Figure 6). In the case of inspection, the system allows the user 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 the size of 3D shape as shown in Figure 6. Otherwise, the defect on a thin element, for example, 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 (1). Different defects can be represented with different shapes and the level of the defect can be represented with different colors as shown in Figure 13 (to be explained in Section 5).

4.2.4 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. LoDs 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 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 7). Also, LoDs can be used in parallel with respect to different objects in the same system,

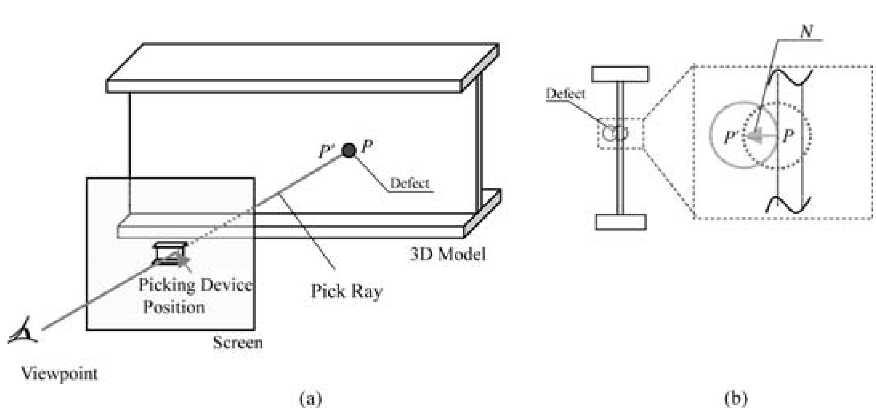


Fig. 6. Example of picking the 3D model for marking defects: (a) 3D sketch and (b) side view.

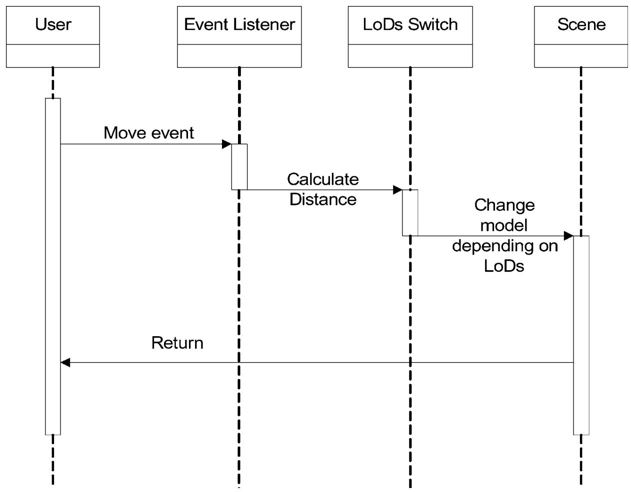


Fig. 7. UML interaction diagram of LoDs behavior.

such as the bridge element and the defects on the element. Each LoDs group uses different referential center points and distance range and operates only on objects related to that group. As shown in Figure 8, 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. 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. In this example, the visible range is from 0 to 20 times the bridge length. 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.

Although the above mentioned techniques (navigation, LoDs and picking behaviors) are common techniques in computer graphics, applying these techniques in MMBLMS requires developing specialized methods that satisfy the special needs of this system as was discussed in this section.

5 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. This prototype system is designed to fulfill the requirements discussed in Section 3 using the computational methods discussed in Section 4 to realize the following major functions: (1) representing the 4D model of bridges with different LoDs; (2) Designing a user-friendly interface with access control that can be used in mobile situations; and (3) Developing comprehensive bridge databases including design, construction, inspection, and maintenance records.

5.1 Case study

The bridges of Montreal are chosen as the subject of the case study. The Jacques Cartier Bridge is chosen as the main bridge for the detailed inspection test. 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 (PJCCI, 2004). The bridge has a steel truss frame combined with a prestressed concrete decking structure system. Inaugurated

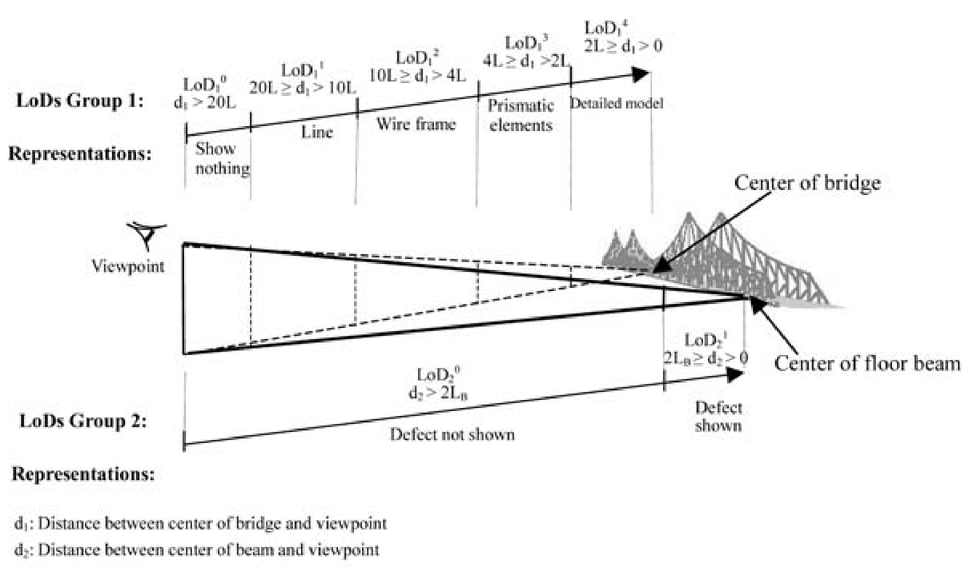


Fig. 8. Relationship between distance and LoDs for the bridge and the defects on a floor beam.

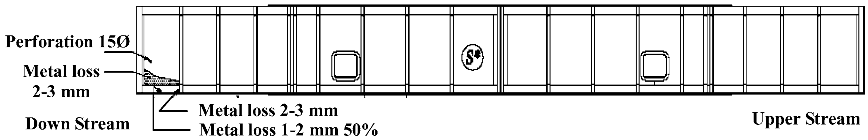


Fig. 9. Example of floor-beam inspection information.

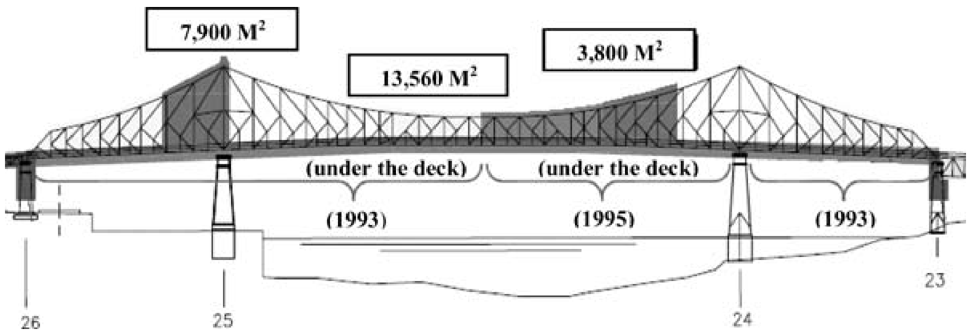


Fig. 10. Bridge painting history.

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 old reinforced concrete 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 was replaced in 2001 and 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.

The bridge data were acquired from the bridge management authority (The 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 9 shows part of the inspection data of a floor-beam including metal loss and perforation. Figure 10 shows the main span painting history of the bridge until 2003. These data have been used in the development of the prototype system. 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. The database was built to include data about the different stages of design, construction, rehabilitation, and inspection. In addition, we acquired the digital map and the DEM data of Montreal to generate 2D and 3D maps (Clément, 2004). Furthermore, a number of simulations were developed to demonstrate the usefulness of the 4D approach, such as displaying elements with different colors according to construction, painting, or rehabilitation periods.

5.2 General implementation details

The structure of the prototype system follows the framework architecture explained in Section 4.1. The system integrates a 3D model of a bridge with an 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. 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. 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 Gehringer, 2001). Java 3D is a runtime API for developing portable applications and applets that can run on multiple platforms and multiple display environments. A digital video camera is connected to the system to facilitate image and video capturing. In addition, the system provides a rule-based expert system to support the decision-making related to inspection activities using the Java Expert System Shell (JESS) (Friedman-Hill, 2003). The initial testing of the system was done using a Fujitsu LifeBook T4000 Tablet PC equipped with 1MB of RAM to improve the rendering performance. Because of the large scope of the system, the discussion in the rest of the article will be limited to the main features of the system focusing on the overall architecture and user-interface design. Further details about the system can be found elsewhere (Hu and Hammad, 2005; Mozaffari et al. 2005; Zhang and Hammad, 2005).

The graphical user interface (GUI) of the system is developed using Java Swing classes. Because Java is a cross-platform language, 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 redecking 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 the appearance of objects. Java 3D scene graphs are constructed from node objects using BranchGroups to form a tree structure based on parent–child relationships. TransformGroup objects can be constructed by applying Transform3D objects, which represent transformations of 3D geometry such as translations and rotations (Walesh and Gehinger, 2001).

5.3 Database design

Java database connectivity (JDBC) is a programming framework for Java developers writing programs that access information stored in databases. The system has options to connect with several database management systems (DBMS) such as Oracle, Informix, Microsoft Access, and MySQL. The commands to be executed by the DBMS on the database are based on structured query language (SQL). In addition, to allow the system to interoperate with other applications, we use an IFC data structure representing bridge 3D objects (IFC-BRIDGE, 2004).

The database of the 4D model is designed with Microsoft Access to present the information of all the truss and deck components of the main span of the bridge. 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.

To avoid security restrictions resulting from the applet accessing the database directly, we use a three-tier solution where the applet is only responsible for display, and will introduce mid-tier for all application logics that are related to data retrieving/updating. 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.

5.4 GIS and tracking components

A GIS sub-system is created using MapObjects Java Edition (ESRI, 2003). 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 locations and the environment around them. The map includes several layers related to Montreal City, such as a boundary layer and other layers for the roads, rivers, and administrative areas. The Modified Transverse Mercator projection was used because it is the standard projection used by the local government. 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, the same map of Montreal and the DEM were added to the 4D browser. The location of the inspector can be retrieved from the tracking devices and combined with the location of an element or a defect, which are registered in the spatial model, to help the inspector find his/her targets using virtual arrows.

Finding the location of the user is achieved using DGPS, RTK-GPS, or video tracking. We are testing the system with a Trimble 5700 RTK-GPS receiver. We are also using an Augmented Reality toolkit, called ARToolKit (Hirokazu, 2000), to track visual 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 20 cm can be recognized from a distance of about 150 cm. On the other hand, the GPS can be used only under the condition of having direct line of sight to at least four GPS satellites.

5.5 User interface design

5.5.1 General design. The main user interface of the system is shown in Figure 11. On the right-hand side, there is a time-input interface that allows the user to query the

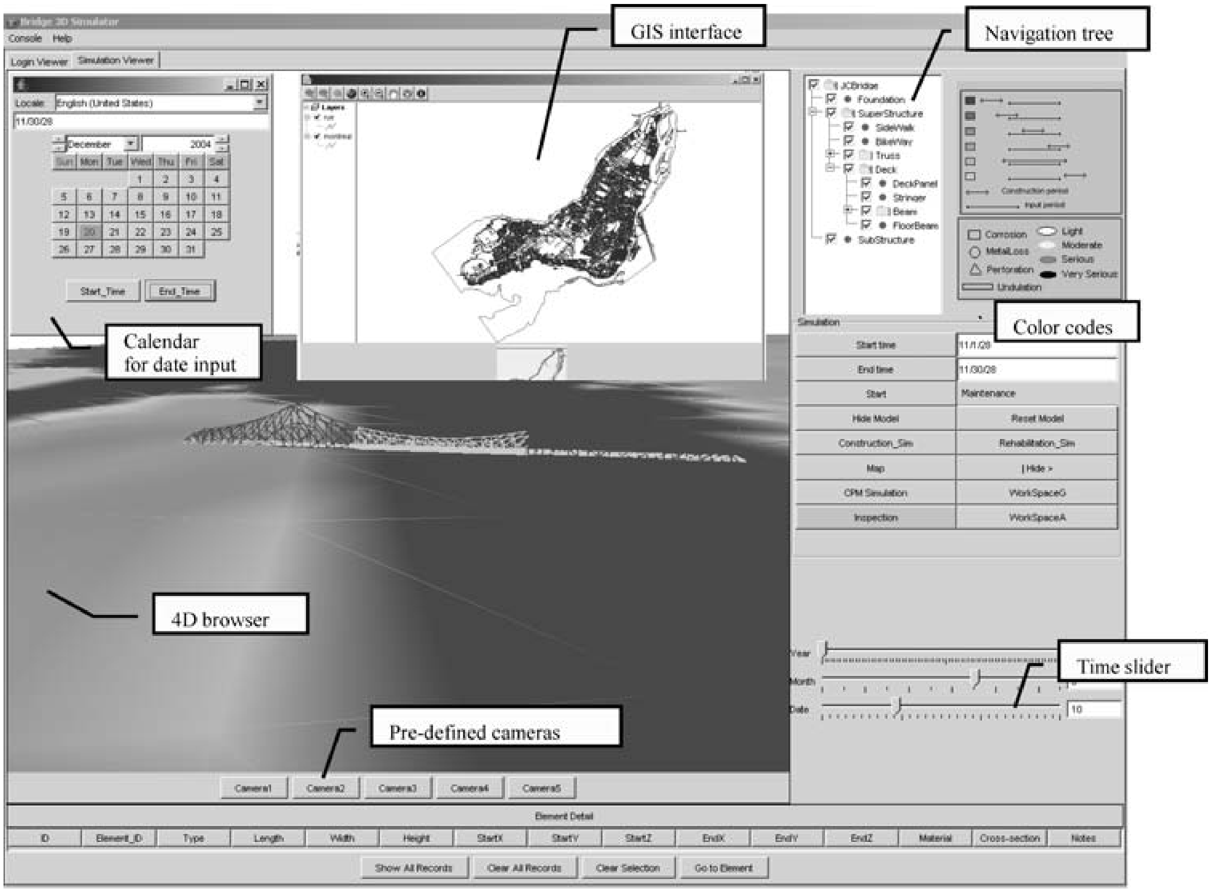


Fig. 11. Screen shot of the user interface of the prototype system.

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 start and end dates of a period can be input using a calendar interface or sliding bars, and the 3D model will reflect the corresponding elements with different colors representing the progress ratio. A logical tree of the bridge structure is also 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. In addition, the user can navigate the 3D bridge model and select an element of the bridge by picking that element. 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.

5.5.2 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* pane, the inspector can choose the exact element to inspect according to a customized inspection scheme by *picking* the element on the 3D model at the approximate location of the defect. In the *Instrument* pane, a suitable inspection tool can be selected depending on the type of defect. The *Damage* pane is the core part of the bridge inspection interface. Video/image capture functionality also has been implemented using Java Media Framework (JMF) API (JMF, 2004). The last pane, *Task* (Figure 12), is to summarize the previous inspection information for future assessment.

Figure 13 shows an example of *picking* a floor beam on the 3D model at different locations to input the location of defects. The defect will be automatically marked on the 3D model of the floor beam using a specific shape and color, which are defined based on the defect type

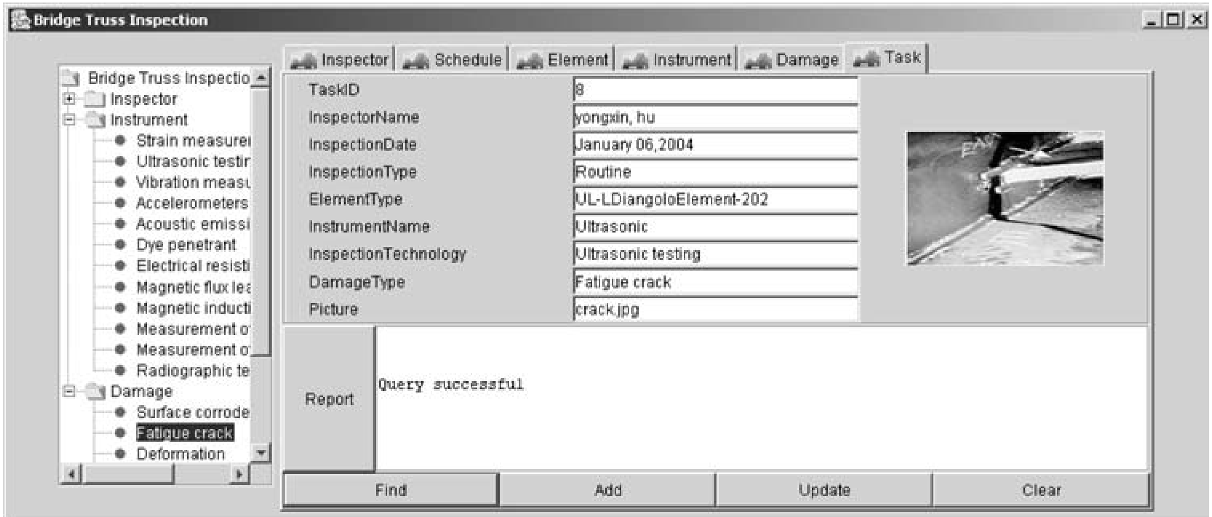


Fig. 12. Inspection task report tabbed pane.

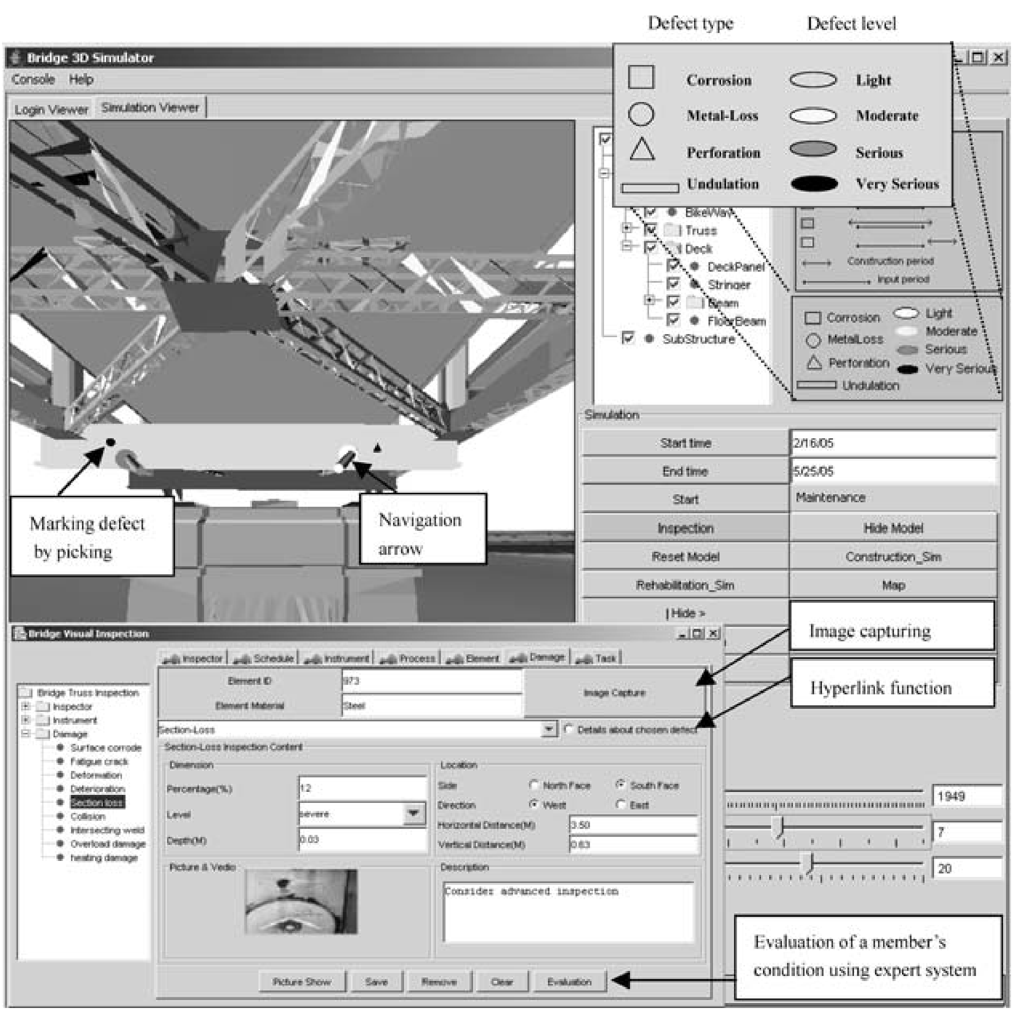


Fig. 13. Inputting the defect location by picking.

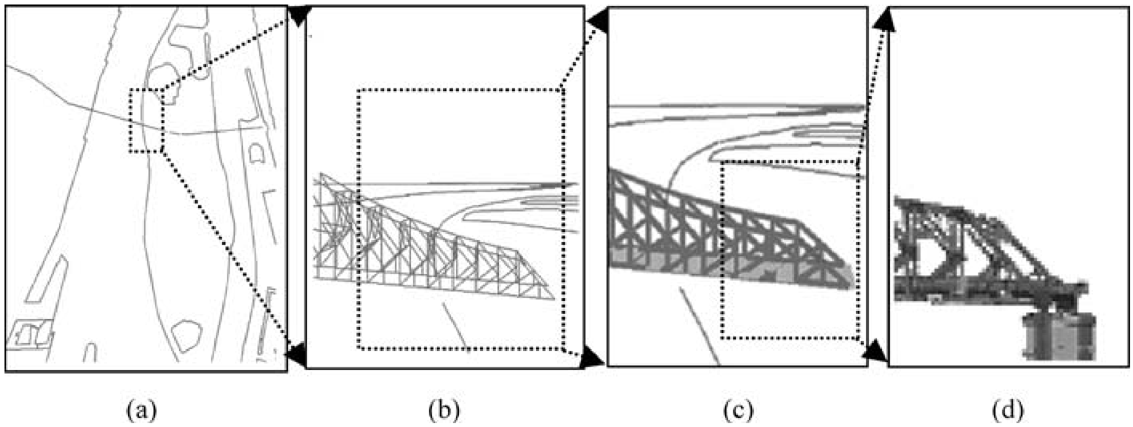


Fig. 14. Different spatial LoDs of the bridge: (a) line (axis of the bridge), (b) wire frame, (c) prismatic elements, and (d) detailed model.

and deterioration degree, respectively. For example, in Figure 13, the black sphere represents very serious metal loss.

Bridge inspection is a knowledge-intensive process. 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 developed to analyze the collected defect data and to calculate the element condition rating. The details of the expert system development are beyond the scope of this article and can be found in Hu and Hammad (2005). In the future, other functions for drawing sketches and generating history reports will be added to the system.

5.5.3 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 viewpoint and the model to optimize the performance of the system. As shown in Figure 14, 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 defects.

A calendar and sliding bar interfaces 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 (Figure 11). 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 10. The inspection time is usually represented by the date of inspection. Higher time resolution is used for inspection purposes to record defects that can happen in a very short time.

6 VERIFICATION AND VALIDATION

To test the proposed approach, the prototype system was used on August 27, 2005, in Montreal to perform two groups of inspection tasks: (1) inspection of four small bridges crossing Highway 15 North (Figure 15); and (2) inspection of a major bridge (Jacques Cartier Bridge). In the first group, the four inspected bridges, from south to north, were the following: *Jean-Talon (10815M)*, *des Jockeys (10815N)*, *Paré (10815O)*, and *Ferrier (10815P)*. The distance between two sequential bridges ranges from 150 to 250 m. The test covered the individual functionalities and the overall performance of the system in supporting inspection. The integration of GPS and GIS in the system provided accurate help in finding the inspected targets and in automatically retrieving the relevant inspection information from the database (e.g., previous bridge conditions). The bridge selection algorithm described in Section 4.2.2 as well as the GPS functionality were tested according to the algorithm explained in Figure 4. One of our main concerns was about the availability of the GPS signals because it is well known that urban canopies affect this availability. However, the GPS signals were available in RTK mode throughout the duration of the test when walking on the bridges. The visual inspection tasks were simulated using the GUI of the system on the tablet PC, and some pictures were taken and automatically added to the

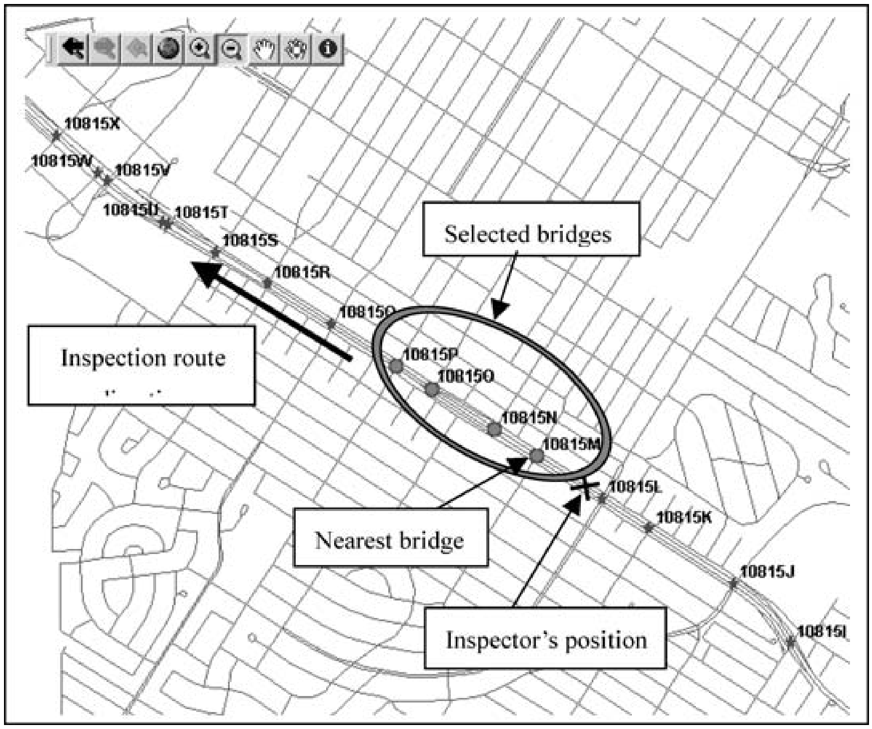


Fig. 15. Finding the nearest bridges in GIS along inspection route.

database using the digital camera attached to the hardhat of the inspector.

In the second group of inspection tasks on the Jacques Cartier Bridge, the inspection data collection was done based on the 3D model. Compared with the traditional manual data input, using the prototype system improved the efficiency of data collection with the help of directly marking defects on the bridge 3D model as explained in Figure 13. However, the GPS tracking was not available at certain locations when walking under the bridge or even on the top of the bridge because of the obstruction caused by the truss elements. Other problems were identified regarding the difficulty of reading the HMD or the display of the tablet PC because of the ambient light conditions and the small size of the characters in the GUI. Further development and testing of the prototype system are needed to improve its usability.

Table 1 shows a comparison of the approximate time required for typical inspection activities (finding elements to be inspected, inputting defect description and location, and data archiving and retrieval) using a conventional paper-based inspection approach and using the

Table 1
Comparison of the time required using a paper-based inspection approach and the proposed approach

	Paper-based approach		*Proposed approach*	
Activity	*Method*	*Time (minute)*	*Method*	*Time (minute)*
Find inspected elements	Referring to paper maps and drawings	2.0	Automatic selection/3D navigation	0.5
Input defect description	Paper forms	5.0	Electronic forms	1.0
Input defect location	Sketching	1.5	Marking elements on the 3D model	0.2
Archive data	Re-entry at the office	10.0	Automatic update of the database	0
Retrieve data	Matching paper drawings and forms	6.0	Picking elements and using electronic forms	1.0

proposed prototype system based on our initial testing. It can be noted that the proposed approach results in considerable time saving because of the possibility to directly interact with a digital 3D bridge model linked to the inspection database.

7 CONCLUSIONS AND FUTURE WORK

This article proposed a new type of MMBLMS and discussed the requirements for developing such systems. The proposed approach makes the first attempt to integrate 4D bridge models with BMSs and to make the resulting information accessible to mobile on-site workers with suitable interaction methods for navigation, picking, and LoDs. The following conclusions can be stated: (1) The requirements and a framework of MMBLMS were discussed including creating an object-relational data model, technology integration and applications development; (2) Several computational issues for realizing the framework were also discussed, such as the navigation modes, automatic spatial selection, picking behavior and LoDs; and (3) The developed prototype system integrates 3D graphics and a database to realize the 4D model of Jacques Cartier Bridge. The prototype system was demonstrated to three engineers responsible for the bridge management and they gave positive evaluations regarding the functionalities of the system. Furthermore, the preliminary testing of the system and its user interface showed that it has good potential for realizing future MMBLMS because it was carefully designed and implemented to satisfy the specific requirements of this system. One of the assumptions that may limit the practical application of MMBLMS is that building 3D models of bridges is still expensive and time consuming because the data are not always available.

Further testing of the developed components of the system in practical situations is necessary to improve the functionalities and usability of the system. It should be noted that this research is still in progress and many aspects discussed in Section 3 about the requirements of MMBLMS are still to be implemented and tested. In addition, we are in the process of modifying the system for facilities management applications (Mozaffari et al., 2005).

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. In addition, the cooperation of Mr. Guy Mailhot from The 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 evaluating the prototype system, and Mr. André Clément from the GIS 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, *Journal of Computing in Civil Engineering, ASCE*, **16**(2), 124–33.

Azuma, R. T. (1997), A survey of augmented reality, *Presence: Teleoperators and Virtual Reality*, **6**(4), 355–86.

Beadle, H. W. P., Harper, B., Maguire, G. Q. & Judge, J. (1997), Location aware mobile computing, in *Proceedings of IEEE/IEE International Conference on Telecommunications*, (ICT'97), Melbourne, 1319–24.

Bentley Software website (2005). Retrieved from: <http://www.bentley.com/en-us>.

Clément, A. (2004), *Normes de production de la géobase régionale des tronçoncs 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. Retrieved from: <http://www.iti.northwestern.edu/publications/technical_reports/tr11.html>.

ESRI (2003), *Getting Started with Mapobjects_Java*. Retrieved from: <www.esri.com/software/mojava/download.html>.

FHWA (2002), *Bridge Inspector's Reference Manual*, Publication No. FHWA NHI 03-001.

Fischer, M. (2001), Introduction to 4D research. Retrieved from: http://www.stanford.edu/group/4D.

Friedman-Hill, E. (2003), *Jess in Action: Java Rule-based Systems*, Manning Publications, Greenwich, CT.

Fujitsu website (2004). Retrieved from: <http://www.fujitsupc.com>.

Garrett, J. H. Jr., Bürgy, C., Reinhardt, J. & 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. & Howitt, I. (1998), Wireless communication and computing at the construction jobsite, *Automation in Construction*, **7**, 327–47.

Guo, S. (2002), Identification and resolution of work space conflicts in building construction, *Journal of Construction Engineering and Management*, **128**(4), 287–95.

Hammad, A., Garrett, J. H. & Karim, H. (2004), Location-based computing for infrastructure field tasks, *Telegeoinformatics: Location-Based Computing and Services*, CRC Press, Boca Raton, FL.

Hassanain, M. Froese, T. & Vanier, D. (2000), IFC-based data model for integrated maintenance management, in *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. (2000), ARToolKit version 2.33: A software library for augmented reality applications. Retrieved from: <http://www.hitl.washington.edu/artoolkit/>.

Hu, Y. & Hammad, A. (2005), Location-based mobile bridge inspection support system, in *Proceedings of the 1st CSCE Specialty Conference on Infrastructure Technologies, Management and Policy*, Toronto, Canada.

IAI website (2004). Retrieved from: <http://www.iai-international.org/iai_international>.

IFC-BRIDGE website (2004). Retrieved from: <http://www.iai-france.org/bridge>.

Itoh, Y., Hammad, A., Liu, C. & Shintoku, Y. (1997), Network-level bridge life-cycle management system, *Journal of Infrastructure Systems, ASCE*, **3**(1), 31–39.

Java Media Framework API website (2004). Retrieved from: <http://java.sun.com/products/java-media/jmf/index.jsp>.

Kaplan, E. D. (1996), *Understanding GPS: Principles and Applications*, Artech House, Norwood, MA.

Karimi, H. & Hammad, A. (2004), (eds.) *Telegeoinformatics: Location-Based Computing and Services*, CRC Press, Boca Raton, FL.

Kuo, S. S., Clark, D. A. & Kerr, R. (1994), Complete package for computer-automation bridge inspection process, *Transportation Research Record*, **1442**, 115–27.

Leung, A. (1996), Perfecting bridge inspecting, *Civil Engineering Magazine*, **66**(3), 59–61.

Liu, D., Cheng, J., Law, K. H. & Wiederhold, G. (2002), An engineering information service infrastructure for ubiquitous computing, *CIFE Technical Report #141*, Stanford University.

McClure, S. (1997), Object database vs. object-relational database, *IDC Bulletin #14821E*. Retrieved from: <http://www.ca.com/products/jasmine/analyst/idc/14821E.htm#BKMTOC13>.

Mizuno, Y., Abe, M. & Fujino, Y. (2002), Development of interactive support system for visual inspection of bridges, *International Forum on Infrastructure Maintenance*, November, 2003, Tokyo, Japan.

Mozaffari, E., Hammad, A. & El-Ammari, K. (2005), Virtual reality models for location-based facilities management systems, in *Proceedings of the 1st CSCE Specialty Conference on Infrastructure Technologies, Management and Policy*, Toronto, Canada.

Object-Relational Mapping (2004). Retrieved from: <http://www.service-architecture.com/java-databases/index.html>.

PJCCI (2004), The official website of The Jacques Cartier and Champlain Bridges Incorporated. Retrieved from: <http://www.pjcci.ca/>.

Potel, M. (1996), MVP: Model-View-Presenter, The Taligent Programming Model for C++ and Java. Retrieved from: <ftp://www6.software.ibm.com/software/developer/library/mvp.pdf>.

Reinhardt, J., Akinci, B. & Garrett, J. H. (2004), Navigational models for computer supported project management tasks on construction sites, *Journal of Computing in Civil Engineering*, **18**(4), 281–90.

Ryall, M. J. (2001), *Brdige Management*, ISBN 0 7506 5077X.

Russell, S. & Norvig, P. (2003), *Artificial Intelligence, A Modern Approach*, Prentice Hall, Upper Saddle River, NJ.

Shamir, A. & Pascucci, V. (2001), Temporal and Spatial Level of Details for Dynamic Meshes, ACM, VRST01, November 2001, Banff.

Stageworks website (2005). Retrieved from: <http://www.bechtel.com/Briefs/0305/Front_End.htm>.

Stumpf, A., Liu, L. Y., Kim, C. S. & 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. & McNeil, S. (2002), A framework for field inspection support systems applied to bridge inspection, in *Proceedings of the 7th International Conference on the Applications of Advanced Technologies in Transportation*, Cambridge, MA, 417–24.

Trilon, Inc. website (2004). Retrieved from: <http://www.trilon.com/>.

Walesh, A. & Gehringer, D. (2001), *Java 3D, API Jump-start*, Prentice Hall PTR, Upper Saddle River, NJ.

Zaki, A. R. & 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, C. & 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*, Toronto, Canada.

Zhang, J., Anson, M. & 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, **2**, 15–22.

Zhao, Y., Zhou, J., Shi, J. & Pan, Z. (2001), A fast algorithm for large scale terrain walkthrough, in *Proceedings of the International Conference on CAD and Graphics 2001*, China. Retrieved from: <http://www.cad.zju.edu.cn/home/zhaoyb/paper/2001terranwalk_release.pdf>.