Visualization, Data Sharing and Interoperability Issues in Model-based Facilities Management Systems

Khaled Hussein El-Ammari

A Thesis

in

The Department

of

Building, Civil and Environmental Engineering

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

September 2006

Your file Votre référence
ISBN: 978-0-494-20732-1
Our file Notre référence
ISBN: 978-0-494-20732-1

ABSTRACT

Facilities management requires gathering and combining a large amount of data from different sources. These data consist of product models and non-product models for describing the process information about each facility. Virtual Reality interaction methods, compared with the product models, can provide better understanding of these models. These interaction methods can be used to accommodate different building engineering applications, such as Facilities Management (FM) and construction progress monitoring. However, the users of these applications may have severe problems in exploring and interacting with the large virtual environments to accomplish specific tasks. Properly designed interfaces are critical for using these applications efficiently. Interoperability in FM industry requires supporting standards for representing the facilities lifecycle information. In this research, first a literature review is conducted about existing FM software solutions and the data representation in these applications. The research review also covers available interoperability standards that can support FM applications. Then, a new framework for the development of a model-based FM system is proposed. The framework includes a practical method for creating facilities models and several interaction and navigation methods. Furthermore, a new method is proposed for creating a generic graphical user interface based on metadata represented in a standard format. The proposed approach is demonstrated through two case studies for indoor and outdoor facilities management applications.

ACKOWNLEGEMENTS

It is definitely a great pleasure for me to express my sincere appreciation and thanks to my respectable supervisor Dr. Amin Hammad who gave me the opportunity to pursue my Master's degree and provided the initial concept for this research project. His profound knowledge and thoughtful instructions have always given me valuable support to pursue this thesis work.

Also, I would like to thank Mr. André Clément, Mr. Richard Lapointe, and Mr. Jean Pellerin from the City of Montreal. Also, I would like to thank Mr. Richard Young and Mr. Christopher Mikos from the Facilities Management Department at Concordia University for providing us with the data of the case study.

During these demanding years of study, the support of the members of Infra-group was of extreme importance. I would like to thank Elaheh Mozaffari, Cheng Zhang, Bechir Khabir and Yongxin Hu for their valuable advice to improve my programming skills. In addition, the help of all other friends, who made an excellent study environment, is highly appreciated.

Finally, I would like to express special gratitude to the Cultural Section at the Libyan Embassy in Ottawa for there support and sponsorship of my study. Their support was my main source to accomplish this work. To them and to my family back home, I dedicate this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVATIONS

Abbreviation	Description
2D	Two-dimensional
3D	Three-dimensional
4D	Four-dimensional
nD	N-dimensional
AEC	Architecture, Engineering and Construction
API	Application Programming Interface
APs	Application Protocols
AR	Augmented Reality
BIM	Building Information Model
BMS	Building Management System
CAD	Computer-Aided Design
CAE	Computer-Aided Engineering Analysis
CAM	Computer-Aided Manufacturing
CAVE	Cave Automatic Virtual Environment
CNC	Computerized Numerical Control
DB	Database
DEM	Digital Elevation Model
DOM	Document Object Model
EMS	Engineered Management System
FM	Facilities Management
FMS	Facilities Management System
GIS	Geographic Information System
GPS	Global Positioning System
GUI	Graphical User Interface
HTML	Hypertext Markup Language
HMD	Head-Mounted Display
IAI	International Alliance for Interoperability
IFC	Industry Foundation Classes
I/O	Input/Output
ISO	International Standards Organization
IT	Information Technology
JDBC	Java Database Connectivity
LAN	Local Area Network
LBC	Location-Based Computing
LoD	Levels of Detail

MBFMS	Model-Based Facilities Management System
MR	Mixed Reality
TIN	Triangulated Irregular Network
UVE	Urban Virtual Environment
VE	Virtual Environment
VR	Virtual Reality
VRML	Virtual Reality Modeling Language
WWW	World Wide Web

CHAPTER 1 INTRODUCTION

1.1 GENERAL BACKGROUND

Compared with other industries that have direct impact on the economy, building industry plays an essential role. Building industry is widely fragmented, which makes it a point of interest for many engineers, engineering management firms, and economists. This multi-billion industry has seen a significant growth in the past two decades in investment and grabbed the attention of other interested parties (e.g., software developers). This share of interest and the combined efforts are creating a new vision of management domains in the building industry, such as facilities management. For that reason, preserving a facility condition is important for improving the economy. Facilitates can be categorized into building, transportation, water supply and wastewater facilitates, among others. Due to the great amount of information required, building facilitates needed more detailed data than most of other infrastructure facilitates.

Building facilitates comprise the building envelope, building structures and subsystems, such as the HVAC, electrical and mechanical systems (e.g., elevators and escalators). Managing these building components requires Facilities Management (FM) organizations to monitor the conditions of these components and maintain the defected parts. The FM organizations are having a hard time coping with corporate directions. The incredible movement of personnel at all levels creates a great challenge to the FM departments. As long as the corporate organization is there, facility organization must be organized to serve it.

Research emphasis in recent years has been shifting from conventional methods for colleting a facility inspection data, to IT-based methods that provide more effective solutions to keep facilities in a safe and serviceable condition. The conventional approach showed many limitations (Springer, T., 2001). However, the new IT approach has promising future for the FM industry by minimizing cost, improving productivity and extending facilities lifetime in good condition. Furthermore, this new approach allows FM personnel to predict future costs by applying pre-planed maintenance services (preventive maintenance). These FM solutions have been developed to provide decision support in the quality of inspection and the allocation of the available limited funds (Sunkpho et al., 2002).

1.2 PROBLEM DEFINITION

Constructing database for Facilities Management System (FMS) is essential and considered as a core part of the system. This database is built up of facility information (i.e., product and non-product information) and other information obtained from the regular facility inspection. The FMS requires inspectors to accurately find and record defects during an inspection task. However, conventional FMS is paper-based and provides limited support for spatial visualization. Additionally, the re-entry of the data at the office results in input errors and lower efficiency. Besides, FMS is a knowledge-intensive process, which is becoming increasingly challenging due to the uncertainty issues implicated in the collected data related to the facility condition evaluation. Consequently, facility inspection evaluation results may vary depending on the accuracy of inspection data input and the inspectors' expertise.

In order to overcome these difficulties, efforts in this research study are directed to develop an accurate, practical, and generic ***Model-Based Facilities Management System*** (MBFMS) to facilitate real-time and on-site data collection and to support inspectors in evaluating building elements.

1.3 MOTIVATIONS AND OBJECTIVES

The main motivations of this research project are:

(1) Future facilities management systems require integrating lifecycle data and supporting 3D visualization and interaction models.

(2) Using these systems in mobile situations will allow on-site facilities inspectors to use mobile and wearable computers to interact with geo-referenced spatial models of the large-scale facilities.

(3) Building virtual reality models for large-scale facilities management applications is time consuming because of interoperability issues.

The objectives of this research are categorized as following:

(1) Investigating and developing a general framework for model-based application for large-scale FM.

(2) Investigating a new approach for using standard-based facility modeling to improve interoperability.

(3) Investigating interaction methods for VR applications.

1.4 THESIS ORGANIZATION

This study will be presented as follows:

Chapter 2 Literature Review: This chapter presents the current situation of FM applications in the building industry. Interoperability issues in these applications are also introduced. In addition, several approaches for representing product models are discussed including an intensive review about the most popular standards for product modeling (e.g., STEP, IFC, etc.).

Chapter 3 Modeling and visualization for model based applications in facilities management: A new data integration method is proposed for creating the VE by synthesizing information from different data sources (e.g., CAD drawings, images, etc.). The creation of IFC-based models is discussed for both the exterior and the interior of a building. In addition, linking with a database for large-scale facilities is also discussed in this chapter.

Chapter 4 Model-based navigation and model interaction: This chapter starts with introducing the inspection process through location-based technology. The design of the MBFMS's main user interface is also explained. Furthermore, two-way interaction for interacting with the VE model with the facilities database retrieval and updating is also discussed in this chapter.

Chapter 5 Implementation and Case Studies: In this chapter, the prototype system is discussed and two case studies are used to demonstrate the prototype system using the proposed approaches and methods.

Chapter 6 Summary, Conclusions and Future work: This chapter summarizes the present research work, highlights its contributions, and suggests recommendations for future research.

CHAPTER 2 LITERATURE REVIEW

2.1 INTRUDUCTION

Facilities data, as a product data, start its journey from the early stages when the project is in the design phase. At that time, the amount of data is still limited to the three dimensional (3D) geometric information and is fairly small compared with the later stages information, when the building model contains more data about the lifecycle not only the 3D geometric model. There are more information or dimensions (nD) about the building, such as cost, scheduling plans, and even quality dimension. Figure 2.1 shows these different aspects of integrating building lifecycle information.

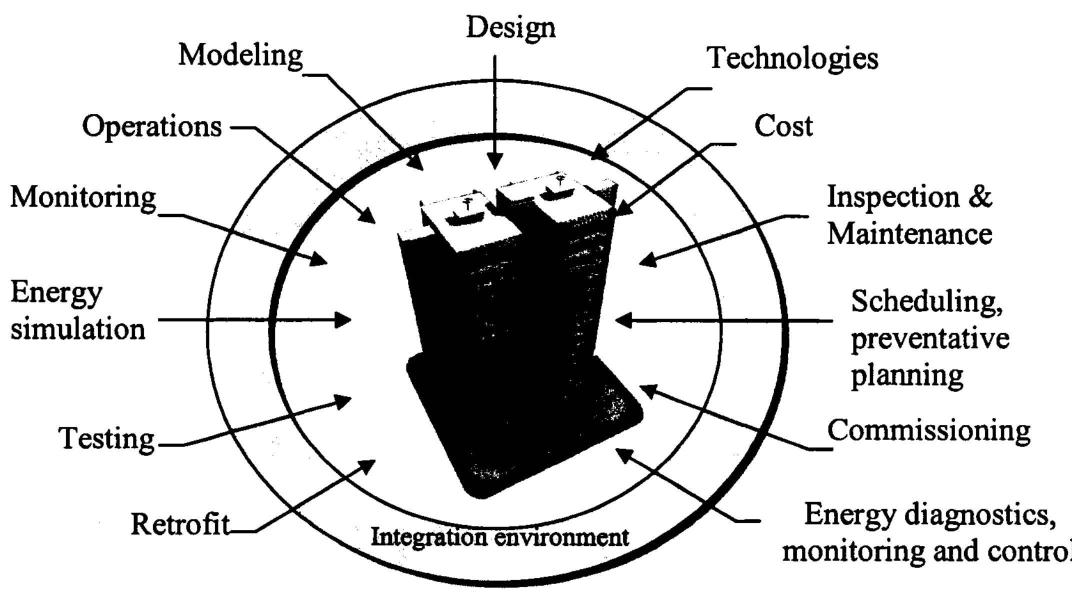


Figure 2.1 Integration in building lifecycle

This information will be valuable data for future facilities engineering systems (i.e., Model-Based Facilities Management Systems (MBFMSs)). The models used in such systems need to include the whole building lifecycle information in a rich well-represented format (e.g., IGES, STEP, IFC), which will be presented in Section 2.3.

This research reviews the state-of-the-art knowledge in the Facilities Management (FM) domain and gives a comprehensive overview on interoperability issues in this domain. This knowledge was collected by reviewing the literature, current FM software packages, and current practice in the FM domain.

2.2 REVIEW OF FACILITIES MANAGEMENT SYSTEMS

2.2.1 Definition of facilities management

FM is the integrated management of all types of built environments from commercial buildings, hotels and high-rise buildings to oil industry facilities and healthcare facilities. FM practitioners monitor the management of a building by providing a range of services including building maintenance and operations, space management, communications and energy infrastructure (Springer, 2001).

The main role of a FM department is to ensure that every facility in the organization is available for the other staff to do their work. The definition of an FM organization ranges from a small scale (e.g., an apartment building janitor may act as the FM representative) to a large scale (e.g., municipality's FM department). There are also some

FM companies providing their on-demand services to manage other companies' facilities (e.g., hotels).

In practice, FM is also defined as the management of services. These services are categorized as hard services and soft services. As an example of hard services, a building's air conditioning system should be operating efficiently, reliably, safely and legally. Soft services include ensuring that the building is cleaned properly and regularly, monitoring the performance of contractors (e.g., builders, plumbers, and electricians), etc. However, in some European countries, FM has a different and even wider definition such as the one defined by the British Institute of Facilities Management (BIFM, 2005): "Facilities management is the integration of multi-disciplinary activities within the built environment and the management of their impact upon people and the workplace".

In the FM field, many managers and maintenance staff are still performing their work activities manually using paper-based forms for inspection tasks' data collection. The inspection records in this case may include some human mistakes or may have some missing data. Therefore, applying the suitable technology and providing simple-to-use tools to automate the processes can increase productivity, profitability and improve resource utilization.

Operations management evaluation depends always on the FM and system performance. The systems can be an HVAC system, lighting system, computer system, etc. Therefore, including such systems in a FMS that presents the whole lifecycle of a facility can be a powerful tool that reduces cost and improves facility production and performance.

Areas such as project management and construction management have considerable development and there are many powerful tools available in the Architecture, Engineering and Construction (AEC) industry to support these domains (e.g., Primavera P3 in Planning and Timberline in Cost estimation). On the Other hand, the area of FM seems to be less developed and needs more effort in terms of domain-specific research and FM application development.

2.2.2 Software applications and projects in facilities management

After the remarkable growth of the FM industry and seeing more possibilities for productivity and profit improvement, many software providers started focusing on providing FM solutions. These solutions usually consist of modules that interrelate information from diverse sources such as personnel lists, facilities diagrams and as-built drawings. These solutions can be complex and used to incorporate and integrate information from different buildings and companies all over the world. As a result, the facility lifecycle cost of typical building often exceeds the cost of the design and construction phases combined. By monitoring these costs through the facility's lifecycle, the FM solutions can be beneficially used to minimize these costs.

The FM solutions are usually integrated systems developed mostly by Computer-Aided Design (CAD) vendors. Some of these solutions are designed to work with certain CAD applications and read certain formats, and some read information from several CAD formats and have their own CAD editors to change model data. Most of these applications integrate with existing databases such as Oracle and Microsoft SQL.

Many FMSs have been developed to satisfy the FM industry needs. Because the implementation of FMSs can be complex and expensive, many major CAD manufacturers, such as Graphisoft, Autodesk, and Bentley Systems, offer products and consulting services for their clients. The rest of this section summarizes the available systems for FM and evaluates their main features and limitations.

2.2.2.1 Bentley Facilities Planner

Bentley Facilities Planner has been developed by Bentley Systems. The system allows users to access information about the state of space or assets. In addition, the system uses 2D or 3D displays and combines MicroStation application to represent the drawings with the robust engineering document management capabilities of ProjectWise to create a unified collection of plans, documents and databases (Bentley, 2005).

The system has many interesting features such as using hierarchy relationships navigation, automatic *Object Linking*, *Automatic Area Calculations* and the general document management functionality. From there, a facility inspector of maintenance employee can find the attached attributes or information about this specific facility. Figure 2.2 shows the main interface of the system, where the facility CAD drawing (e.g., in Microstation native format) can be shown to interactively locate people, spaces or furniture (e.g., computers and lab equipment). In addition, the inspector can read and write the values of the attached condition attributes. Bentley is committed to support interoperability and exchange of digital building lifecycle information. Microstation V8 allows users to work with DGN, DXF and DWG as native formats.

As IAI board member, Bentley actively supports interoperability throughout the phases of the building lifecycle. Bentley systems have been certified as IFC 2x compliant and are currently awaiting approval on the IFC2x2 certification (Bentley, 2005).

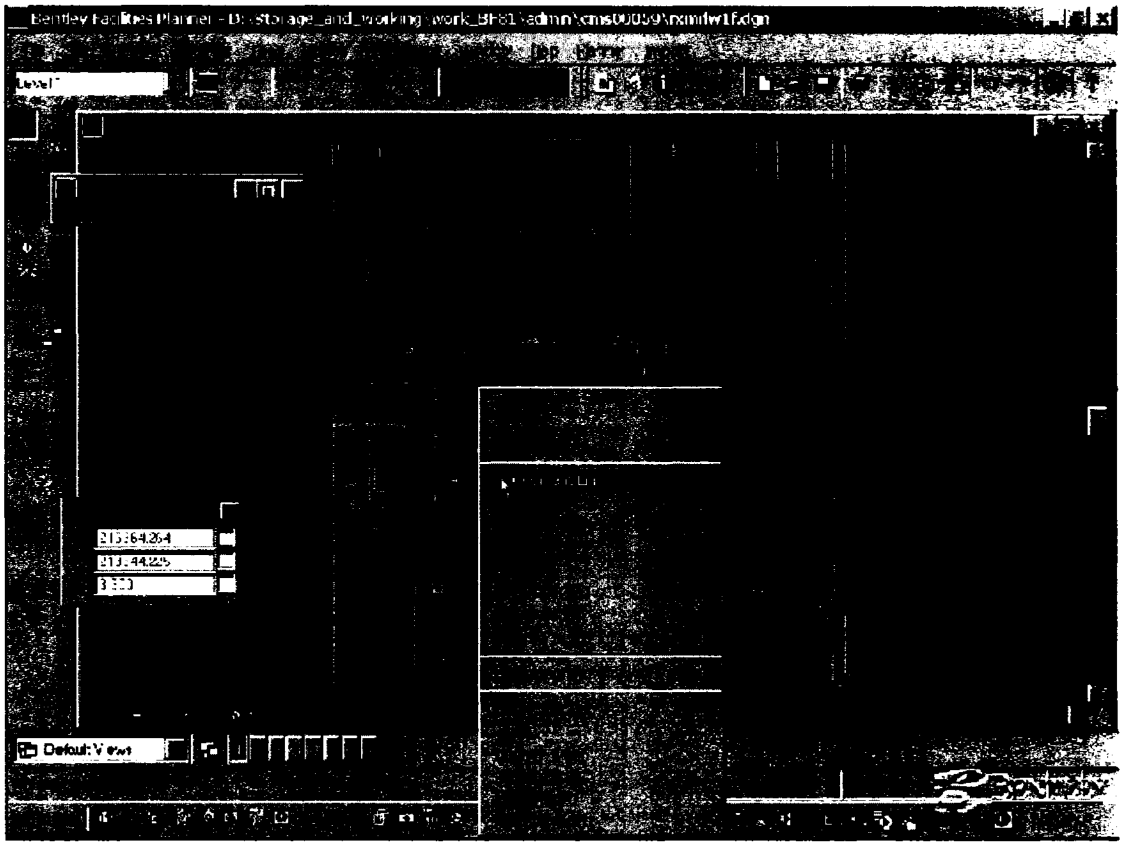


Figure 2.2 Bentley facilities planner main user interface (Bentley, 2005)

2.2.2.2 ArchiFM

The provider of ArchiCAD software has contributed to the FM domain by providing ArchiFM which is an integrated system built on ArchiCAD software (Graphisoft, 2005a). The system automatically coordinates information among facilities databases, floor plans and other external data sources. In ArchiFM, plan objects that represent rooms, furniture, people, and other assets are linked to a database. In this case, any changes on the plans are automatically updated in the database, and vice versa. Figure 2.3 shows the main user interface of the system, where the user can navigate through the tree list to find facility components (e.g., room, person, furniture, etc.) and attach information to it, which will be automatically saved in the database.

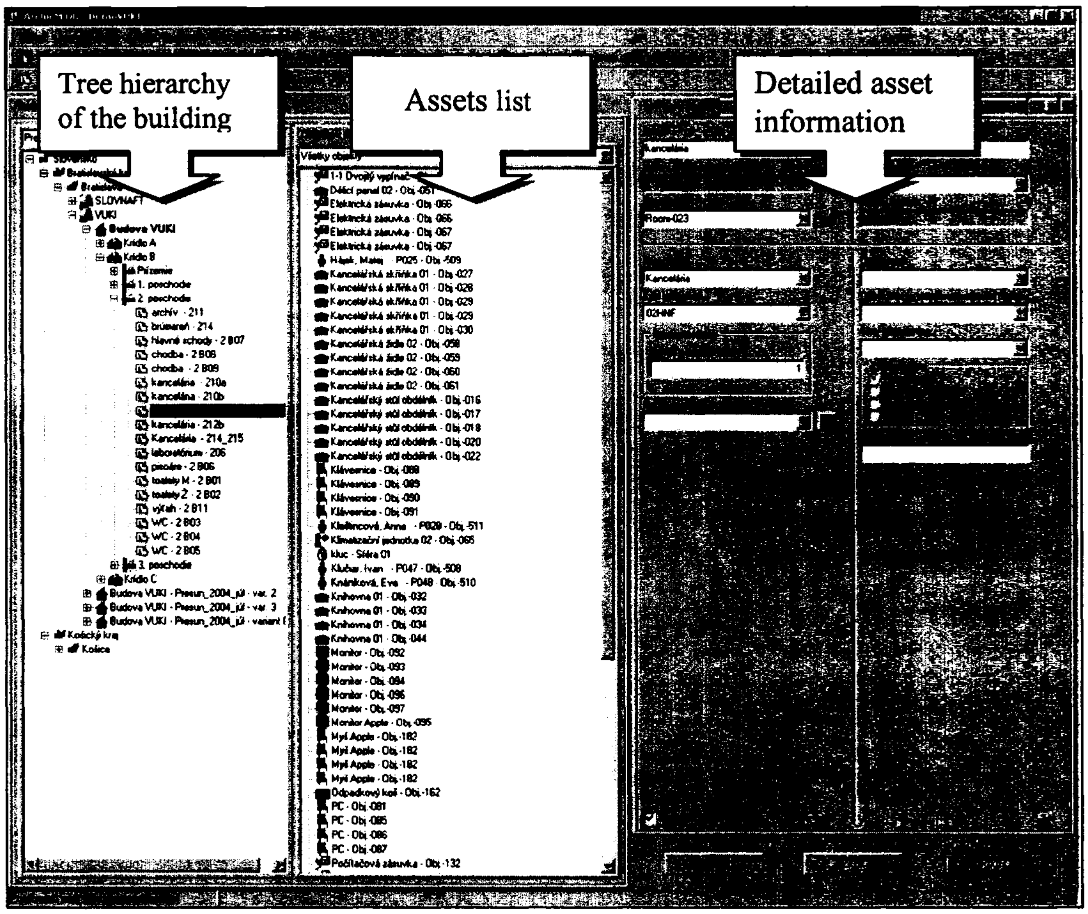


Figure 2.3 Graphisoft's ArchiFM integrated FM system user interface (Graphisoft, 2005)

ArchiFM works as a standalone application as well as a web-based application. ArchiFM WebServer lets the user distribute the facilities information, documents and drawings. It is scalable and can work with small or enterprise-wide applications. ArchiFM consists on the following modules: (1) Space management, (2) occupancy management; (3) move management (people, departmental data, voice and telecommunications needs, space standards, or personal requirements), (4) lease management, (5) asset management (computer equipment, furniture, or other fixed or moveable equipment); (6) cost calculation and chargeback, (7) data management, and (8) drawing management. For integration with other computer applications, ArchiFM supports a variety of data file formats including ASCII, text, RTF, and Microsoft Office formats. ArchiFM has an Application Programming Interface (API), which supports C, C++, Visual Basic, Java, and XML (Graphisoft, 2005). ArchiFM also supports variant 2D and 3D formats and Table 2.1 summarizes these formats.

Table 2.1 Supported formats by ArchiFM (Graphisoft, 2005a)

Formats	Description
DWF	Drawing Web Format
DXF	Drawing eXchange Format
DWG	Native format for Autodesk CAD systems
DGN	Native format for Microstation system
HPGL	Hewlett-Packard Graphics Language
PLN	Native format for ArchiCAD
IFC	Industry Foundation Classes

2.2.2.3 MicroROOFER

MicroROOFER is a specific Engineered Management System (EMS) designed to evaluate built-up and single-ply roofs (TAC, 2005). MicroROOFER is a member of the EMS software family which includes MicroPAVER and RAILER. The system was developed by the U.S. Army Construction Engineering Research Laboratory (USACERL), with assistance of U.S. Army Cold Regions Research and Engineering Laboratory and the U.S. Army Center for Public Works. This collaboration produced a decision support tool which provides systematic methods of assessing roof condition, selecting repair or replacement needs, and deciding work priorities based on maximizing benefits and minimizing costs. MicroROOFER has been widely used by variant sectors and users such as universities, military installations, governmental agencies, and private buildings. Figure 2.4 summarizes the MicroROOFER's main components.

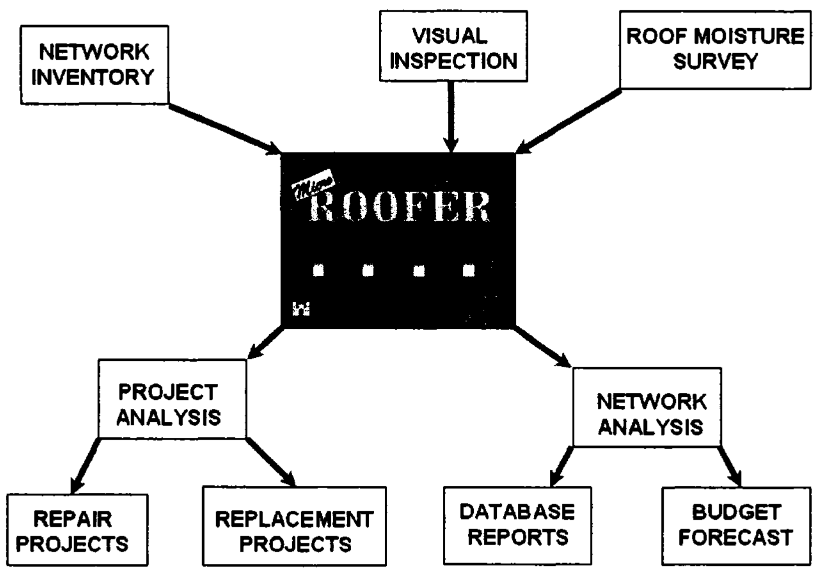


Figure 2.4 MicroROOFER components (TAC, 2005)

MicroROOFER allows the user to easily see the condition of one section or the entire roof system and to manage the information in a database. The system consists of network inventory, condition inspections, and network and project-level management. The system stores inventory and inspection data, performs calculations, analyzes data and generates customized reports. The following items summarize the main advantages of implementing MicroROOFER in roof inspection: (1) Providing a roof inventory that is organized and easily accessible; (2) Providing reliable condition evaluation of built-up and single-ply roofs; (3) Systematic engineering basis for determining needs and deciding priorities; (4) Identification of budgeting requirements for maintaining roofs; (5) Cost analysis for selecting repair/replacement alternatives; (6) Quick access to the stored information; and (7) Small roof problems are detected and corrected before they become costly replacement projects.

The user or the agency is able to collect physical and historical inspection information about each roof section, including the physical dimensions and material types of each roof component (e.g., deck, membrane, surface, insulation and flashing). The most important part of the system is the condition inspection component, where the user can evaluate the condition of built-up and single-ply roofs after the inspection, the severity and quantity of the roof distresses are entered into MicroROOFER database. This system also has the capability to provide reports on roof condition and maintenance cost and to allocate budget for maintaining a roof for the next ten years.

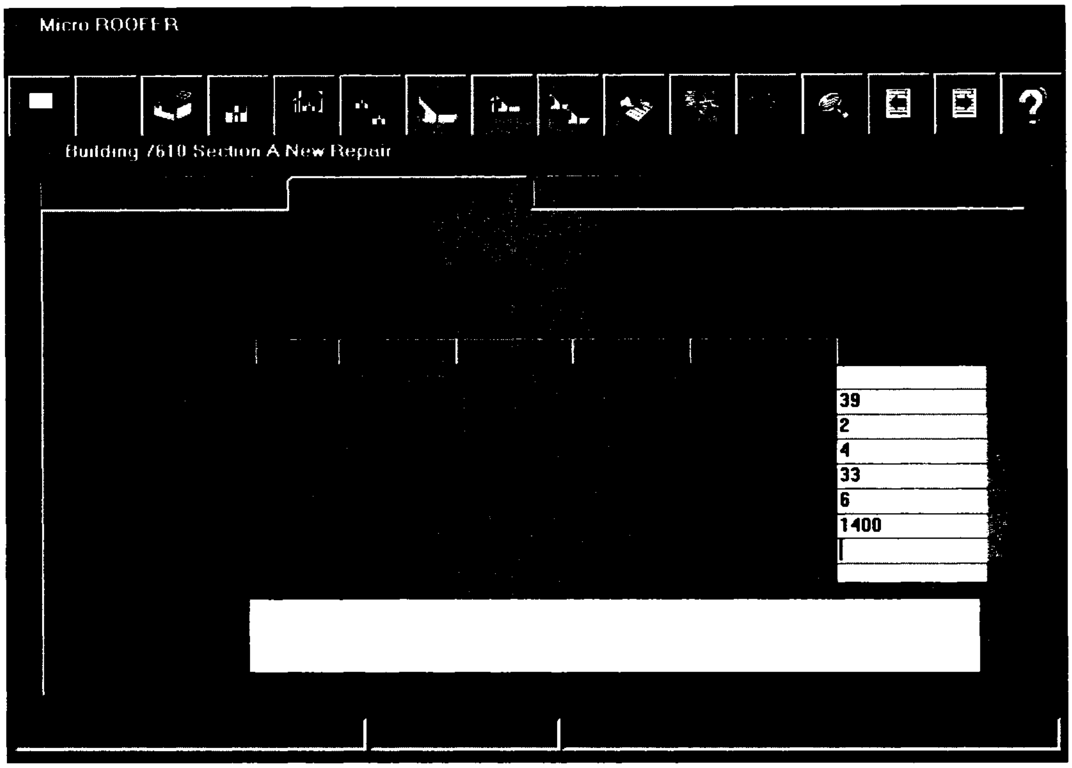


Figure 2.5 MicroROOFER's visual inspection user interface (TAC, 2005)

Figure 2.5 shows the condition inspection user interface. The user can assign inspection records to define the severity level of each defect record and add attributes such as dates, quantities and repairing notes.

2.2.2.4 ARCHIBUS/FM

ARCHIBUS/FM is an AutoCAD-based system for facilities and infrastructure management (ArchiBus, Inc., 2005). It consists of several integrated applications (e.g., Autodesk's Architecture desktop, Buzzsaw and MapGuide), each designed to improve the operational efficiency of the system and to help users make better management decisions. ARCHIBUS/FM applications include: (1) real property, space and lease management, (2) strategic master planning, (3) overlay for AutoCAD with design management, (4) furniture, equipment, telecommunications and cable management, (5) building operations management, (6) condition assessment, (7) environmental sustainability assessment, and (8) fleet management.

ARCHIBUS/FM integrates information sources and different applications supporting places, people, processes, as well as physical assets into a comprehensive integrated solution. The system has the capability to read external formats such as XML for human resources, Excel sheets and Visio native format. It can also integrate with other enterprise applications using databases such as Oracle and Microsoft SQL, which makes data sharing and exchange with other organizations easier. Figure 2.6 shows the user interface of the ARCHIBUS/FM Enterprise system. Facilities information is visually presented in a browser format, available to FM staff anywhere, anytime using the Internet.

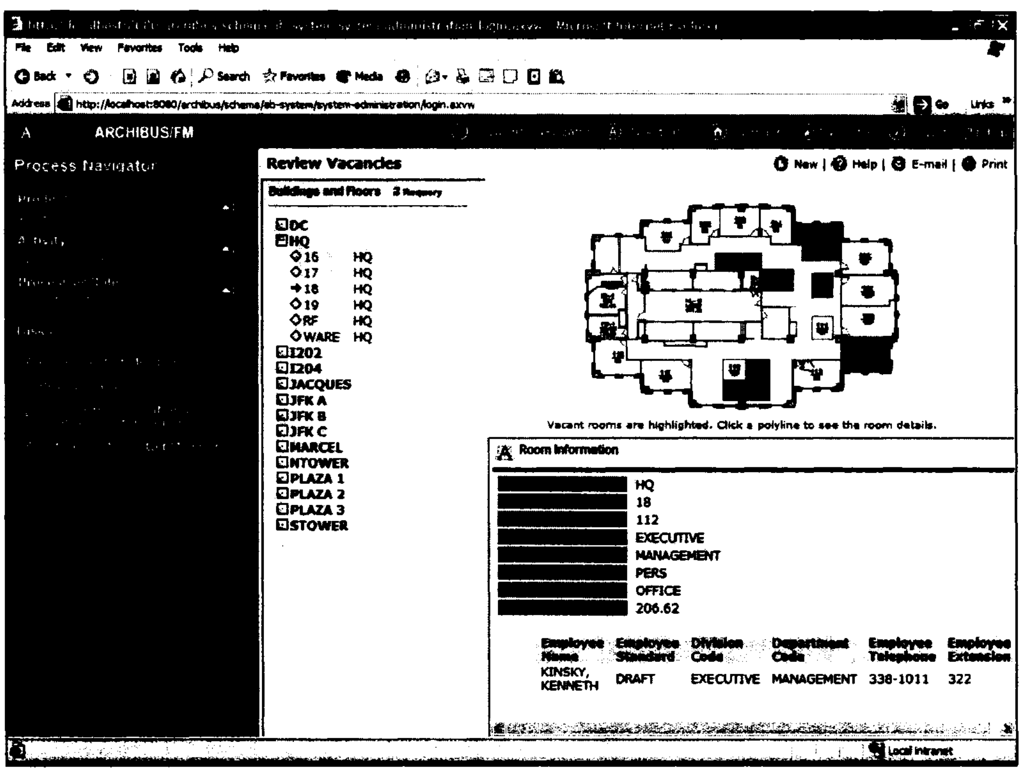


Figure 2.6 ARCHIBUS/FM application user interface (ArchiBus, 2005)

2.2.2.5 MAXIMO

MAXIMO is an entry level asset management software (MRO, 2006). It is a solution for small to medium sized companies which require robust asset management capabilities. This software package helps improve efficiency, reduce costs and optimize asset performance. MAXIMO's capabilities maximize the lifetime value of complex assets and closely align them with the overall business strategy, allowing users to increase return on assets, decrease costs and increase productivity. In addition, this solution can improve asset-related decision-making and improve regulatory compliance. A case study of a computerized maintenance management system in an automotive plant is investigated (Noveloso, 2005). The research focuses on the implementation of MAXIMO in several General Motors plants, specifically, the implementation at General Motors Truck Product Center (GMTPC) which proved problematic because of data sharing problems.

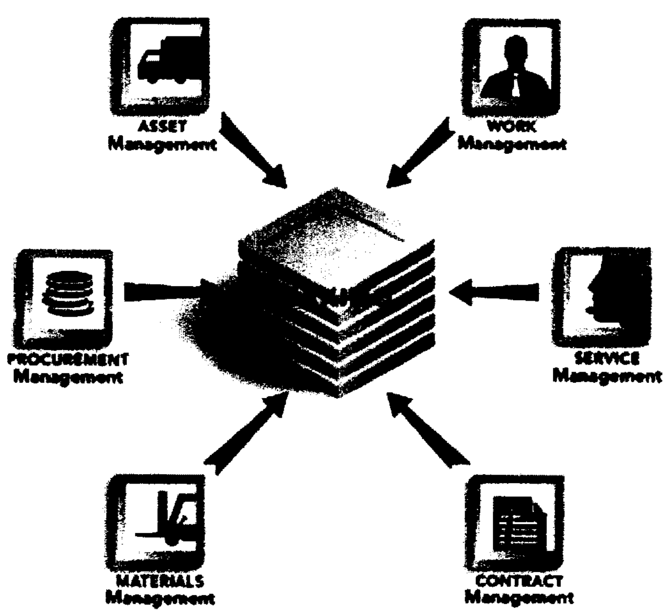


Figure 2.7 MAXIMO's six key systems (MRO, 2006)

MAXIMO package consists of six key systems that enable companies to manage assets, including production equipment, facilities and transportation assets, in alignment with their business objectives (Figure 2.7).

2.2.2.6 BELCAM

The research group from the National Research Council Canada (NRCC), in collaboration with Public Works and Government Service Canada (PWGSC), has accomplished a remarkable IT-based project called BELCAM (Building Envelope Life Cycle Asset Management). The project effort essentially focused on assisting asset managers to predict the remaining service life of building envelope components and to minimize the maintenance expenses. In addition, BELCAM develops a service-life prediction and management tool for low-slope roofs that incorporates risk-assessment techniques as well as an interactive Graphical User Interface (GUI).

The BELCAM project centers around six enabling technologies: (1) Life cycle economics, (2) service life prediction, (3) user requirement models, (4) risk analysis, (5) product modeling, and (6) maintenance management (NRCC, 2005). BELCAM system produces and delivers to its partners the following: (1) An instruction manual on how to conduct a condition-assessment survey for low-slope roofing; (2) Preventive maintenance checklists; (3) An evaluation of available roofing software, (4) A Canada-wide roofing survey; (5) Inspection protocols; and (6) A guideline on how to improve roofing service-life and asset management. The objectives of the BELCAM project mainly were developing techniques to predict the remaining service life of building envelope components and developing procedure to optimize the maintenance management of those components (Kyle et al., 2002).

2.3 INTEROPERABILITY ISSUES IN BUILDING ENGINEERING

2.3.1 Model interoperability and the neutral layer concept

The computerized applications in AEC/FM have provided great benefit, compared with paper-based drawing, regarding accuracy, modifiability and its easiness to exchange among users. On the other hand, building engineering software users, working in different fields or different domains, apply different applications separately without direct communication. Therefore, they may not be able to share and exchange data with each other. To make interoperability possible, it is essential to have a neutral layer connecting different applications. Figure 2.8 summarizes the two different approaches for data sharing and exchange; the direct communication approach (e.g., the CAD model) and the neutral layer approach (e.g., the IFC Model).

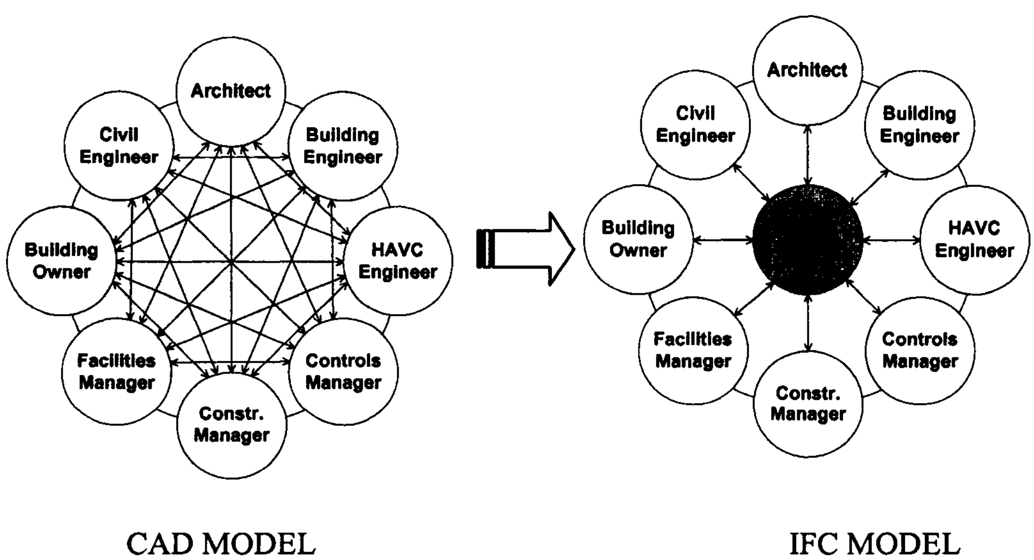


Figure 2.8 Data sharing and exchange in the CAD model and IFC model (VTT, 2005)

Over years of working on improving interoperability, many solutions have been proposed. The most successful have been proposed as national standards for geometric data exchange such as VDAFS (standard used for the transfer of freeform shapes) in Germany, Secure Electronic Transaction (SET) in France and the Initial Graphics Exchange Specification (IGES) in the USA. Later these efforts started under the International Standards Organization (ISO) to work on developing one international standard to represent all aspects of product model data named *STandard for the Exchange of Product* model data (STEP) (STEP Tools, 2005). Figure 2.9 shows some systems that use STEP for data exchange

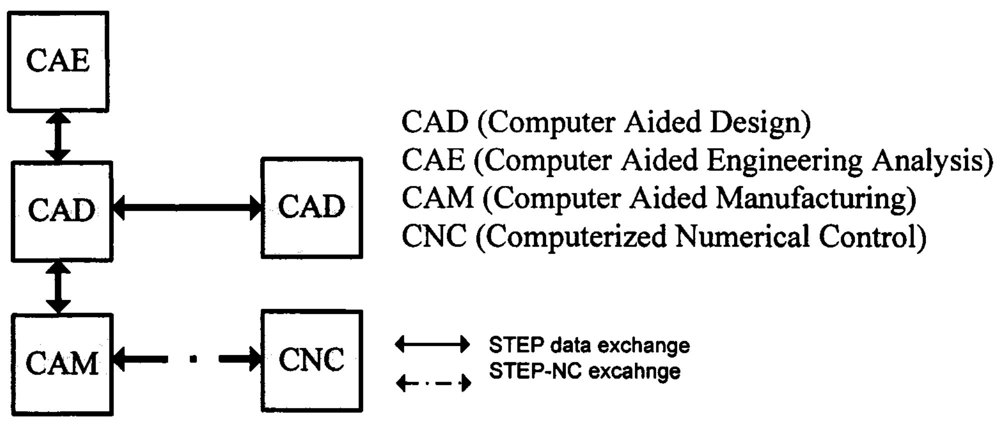


Figure 2.9 Systems use STEP standard for data exchange (STEP Tools, 2005)

Almost all major CAD and Computer-Aided Manufacturing (CAM) systems now contain modules that allow them to read and write data defined by STEP Application Protocols (APs). For example, in the U.S. the most commonly implemented protocol is called AP-203. As shown in Figure 2.10, the AP-203 protocol is used to exchange design data represented as solid models as well as assemblies of solid models. A similar protocol called AP-214 is used in Europe that performs the same function (ISO 10303-1, 1994).

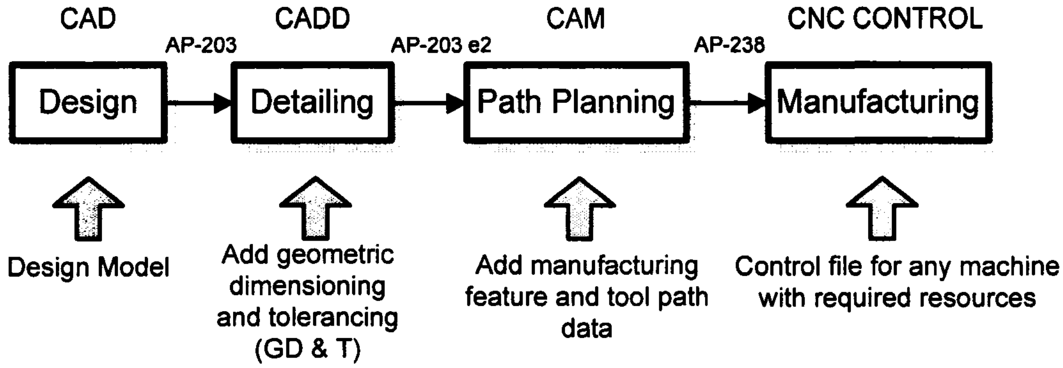


Figure 2.10 Application protocols development in STEP (STEP Tools, 2005)

In Figure 2.10, the AP-238 file reduces the requirement for drawings on the shop floor and it allows manufacturing to send requests for changes back to design by annotating the original full fidelity design information (STEP Tools, 2005).

This research reviews most of the available domain-specific standards that support interoperability in AEC/FM industry. The following are some of these standards or what called neutral layers:

(1) Initial Graphics Exchange Specification (IGES)

(2) Drawing eXchange Format (DXF)

(3) STandard for the Exchange of Product model data (STEP)

(4) Industry Foundation Classes (IFC)

(5) XML-based standards (aecXML, LandXML and IfcXML)

The development efforts of neutral layers have started first in 1979 with the IGES. Then ISO TC184/SC4 committee introduced STEP as an international standard (VTT, 2005). In 1997, IAI published IFC and introduced it as a rich international standard for AEC industry. IFC has been developed based on the schema language EXPRESS, ISO 10303 Part 11 (ISO, 1994). IFC is also based on the clear text format, ISO 10303 Part 21 (ISO, 2002) as described for STEP. In addition, STEP and IFC standards are affected by the XML standard for document representation on the World Wide Web. Some of these XML-based standards, such as aecXML, LandXML and IfcXML, will be discussed later in this chapter.

2.3.2 Initial Graphics Exchange Specification (IGES)

The CAD vendor community in U.S. has announced in 1979 the first national standard for CAD data sharing and exchange called Initial Graphics Exchange Specification or IGES (Goldstein et al., 1998). IGES is a neutral exchange format for 2D or 3D CAD product models, drawing and graphic. IGES is a set of protocols or a standard for the display and transfer of design drawings from one CAD application to another. An IGES file is basically an ASCII clear text format file and consists of sections (e.g., a directory entry and parameter data) (IGES, 2005). The IGES entity consists of a directory entry and a parameter data entry. All directory data are organized in fixed fields and includes an index for each entry for accessing the description about the data, where the parameter data shows the specific entity definition (Smith, 1988). IGES files can be imported into AutoCAD applications using a plug-in ARX tool which adds a new command to AutoCAD called "IGESIN" (IGES, 2005a).

2.3.3 Drawing eXchange Format (DXF)

After the growth of Personal Computers (PCs) use as the dominant platform for architectural design and building-oriented CAD systems, many CAD systems venders vied for leadership. As a result of this competition, Autodesk Inc., developed AutoCAD which became widely known and the largest selling CAD system in the AEC industry, followed by Bentley Systems' Microstation in the U.S. In the late 1980s, more powerful workstation-based CAD systems became common and the AutoCAD users' community became larger. Many firms have migrated to AutoCAD and they maintained files developed on another system. They were also using multiple systems because of the different functionality provided in each system. In the late 1980s, Autodesk has responded to the high demands for data exchange in the AEC industry by defining a public and external file format called Data eXchange Format (DXF). This file format is open to all interested developers. DXF was defined in both a textual and a binary format. However, DXF was limited to the basic kinds of graphic entities (Eastman, 1999). DXF was initially based on a limited set of assumptions, particularly regarding associatively, which makes translating to and from other formats easy to achieve. Today, DXF has become a de-facto standard for exchanging 2D information. However, DXF is still limited in 3D modeling and the necessity of using DXF translators is still essential in many CAD companies.

2.3.4 STandard for the Exchange of Product model data (STEP)

STEP is a comprehensive ISO standard (ISO 10303), which describes digitally a product information for sharing and exchange between dissimilar CAD applications. The development of STEP allows data exchange including geometry data, relationships, attributes, assemblies, topology, configuration, etc. This makes STEP a good data representation to cover more phases in a product lifecycle. STEP has been developed and constructed as a multi-part ISO standard. The basic parts of STEP have been completed while other parts are still under development. The parts that are still under development cover general areas, such as file formats, testing procedures, programming interfaces, and other industry specific information.

In STEP, the APs developed for different industries are the main part of this standard. STEP is a standard built on EXPRESS language (ISO 10303-11) and the power of this language makes STEP an extensible standard, which allows it to describe the structure and correctness of engineering information. The EXPRESS language can also document constraints and data structures. These constraints are an explicit correctness standard for the digital product data. Figure 2.11 shows the structure of the STEP standard. Infrastructure parts have been separated from the industry specific information models (STEP Tools, 2005).

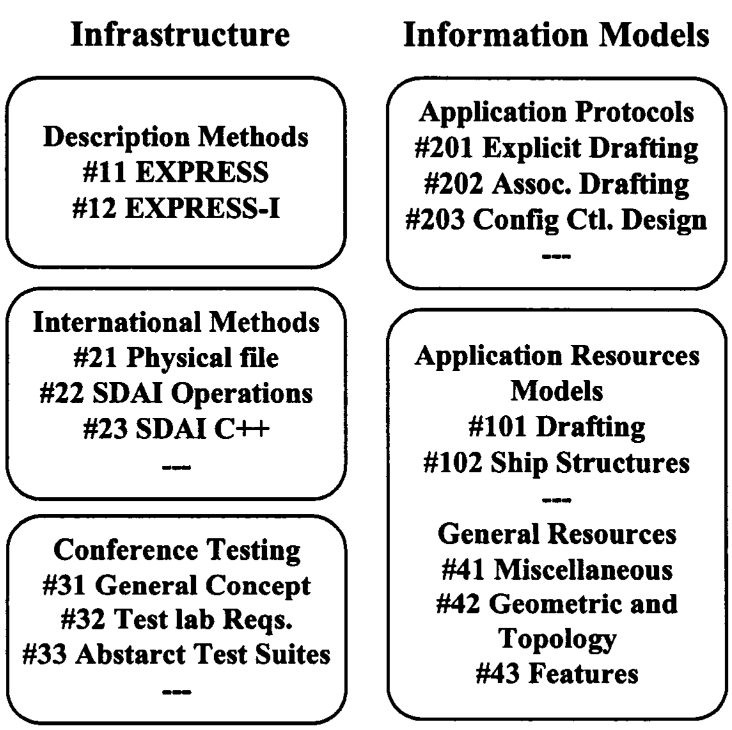


Figure 2.11 High level structure of STEP (Loffredo, 2003)

2.3.5 Industry Foundation Classes (IFC)

After the development and the success of ARX (Autocad **R**untime e**X**tensions) system in AutoCAD Release 13, a group of companies in the U.S. decided to invest more effort in interoperability development. This development was meant to be beyond the product geometric information and to cover more lifecycle information for building projects. This effort leaded to the development of a vendor neutral standard for software interoperability. The need of organizing these different groups has resulted in the establishment of the **I**ndustry **A**lliance of **I**nteroperability (IAI) in North America in October 1995. By October 2005, IAI has 447 members, 11 chapters, 29 tool developers and 24 involved countries (IAI, 2005a).

The fast growth of the organization changes its focus which consequently changes its name to **I**nternational **A**lliance for **I**nteroperability (IAI) to reflect its international wide-ranging reach. The IAI created a text-based, comprehensive and multidisciplinary data model of buildings that defines data throughout a building's lifecycle. The first full release of the **I**ndustry **F**oundation **C**lasses (IFC) was issued in January 1997. Several further releases have been issued since, such as IFC1.5, IFC1.5.1, IFC2.0, IFC2x2 and recently IFC2x3.

IFC is developed as an open and extensible international standard for building data sharing and exchange. Physical objects such as walls, columns, and windows are presented in IFC. In addition, abstract objects such as projects, sites, etc. are also presented in IFC. IFC data are based on the EXPRESS language, the same language that has been used to develop ISO STEP project (ISO, 1999), and organized into a hierarchy structure following the object-oriented method. In addition, IFC is freely available to all AEC/FM software developers. Consequently, interoperability among AEC/FM software applications is feasible and possible based on IFC.

The first appearance of IFC in the industry encourages many software developers to develop software applications compatible with IFC. Examples are: ArchiCAD from Graphisoft and Architecture Desktop (ADT) from Autodesk in CAD drawing and modeling (IAI, 2005); Solibri Model Checker – SMC in design spell checking (Kam et al., 2003); SAP2000 (Wan et al., 2004) in structural design; RIUSKA - BSPro (Granlund, 2005) and EnergyPlus (DOE, 2005) in building energy simulation; and Precision Estimating CAD suite (Timberline, 2005) in quantity takeoff using CAD integrator and cost estimation. In the meantime, there are many research efforts focusing on the implementation of IFC in the AEC/FM industry. For example, Romberg et al. (2004) applied IFC-based models to extract geometric information for the finite element analysis in structural design. Another example is the proposed framework for building envelope evaluation (He et al., 2005).

To satisfy the needs of these applications, the IAI has committed, as their mission, to specifying domain-specific components such as physical components (e.g., doors, walls, windows, etc.) and abstract concepts (e.g., spaces, organizations, processes, etc.). These specifications deliver a data structure that can be electronically represented in a customized computer interface to make management mission easier. In IAI's schema definition, these specifications are called "Classes" and each class describes components that have the same type and characteristics. The classes, as defined by the IAI, are: (1) Specifically defined for the AEC and FM domains; (2) The foundation of the model for the exchange of data; and (3) Defining agreed on object classes to develop a common language for the AEC/FM industry (IAI, 2005).

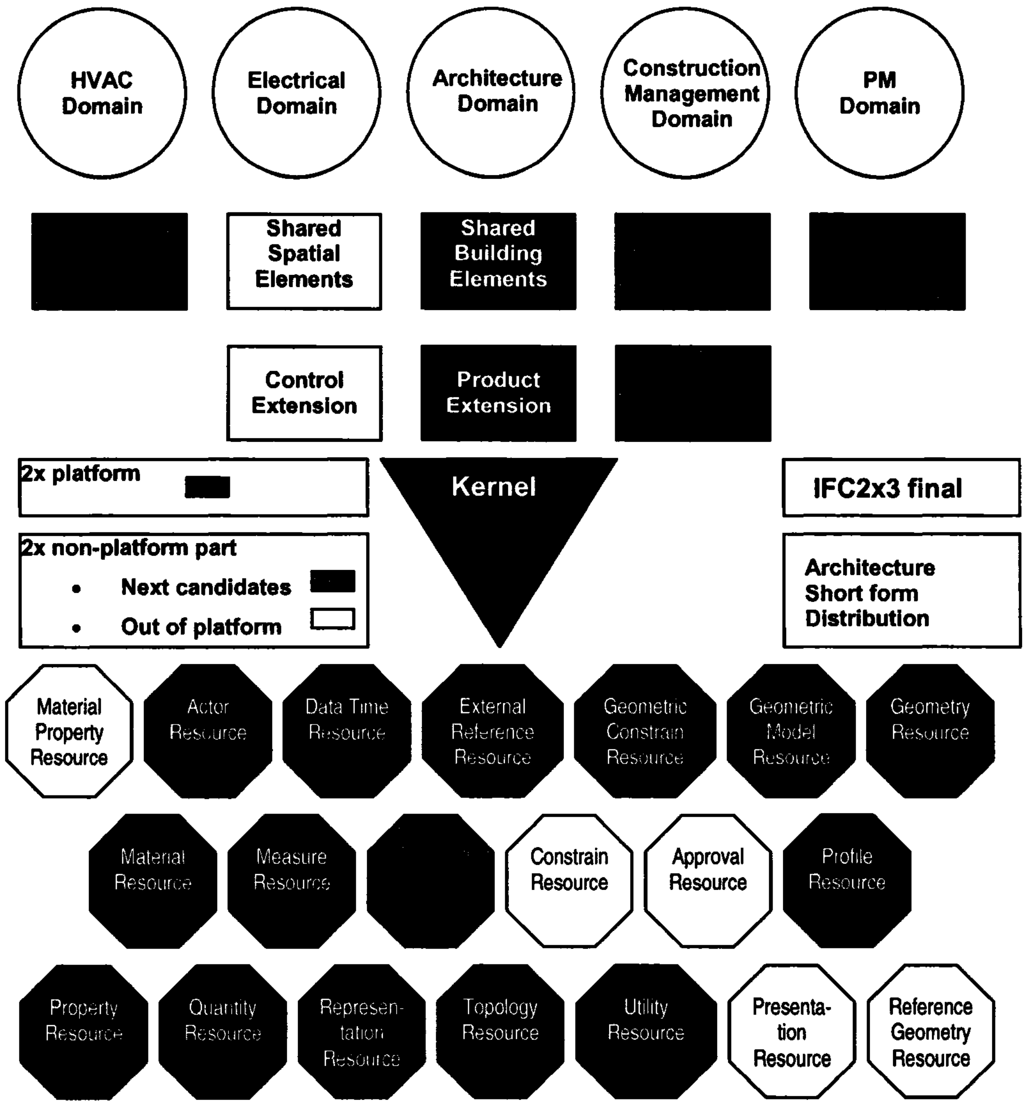


Figure 2.12 Schema overview of IFC2x3 (IAI, 2005e)

In addition, IFC is developed in an extensible structure to allow partners, AEC applications' venders and even researchers to contribute and extend the use of IFC. In February 2006, the latest release of IFC has been published by IAI. The IFC2x Edition 3 (IFC2x3) is the latest official IFC release recommended for implementation and contains some improvements over the IFC2x2 Addendum 1 (IAI, 2005). Figure 2.12 shows the schema architecture diagram of IFC2x3. The schema is divided into five categories of layers: (1) Domain Layers, (2) shared elements, (3) extension layers, (4) kernel layer, and (5) resources layers. In addition, the schema can be categorized as two parts: 2x platform (equal to ISO/PAS 16739) and 2x non-platform. The 2x platform is the stable part and covers many layers such as Shared Building Elements, Product Extension, Kernel, and most of the Resources. However, the 2x non-platform is not stable and can be changed with any future releases of IFC. The 2x non-platform part covers all *Domains*, some *Shared Elements*, some *Extensions* and some *Resources*.

IFC is a text-based data model that can present 3D product models and goes beyond the graphical representation of these models. IFC is unlike the graphical data formats such as DWG or DXF, which are drawing native formats for Autodesk's CAD applications. These CAD-based native formats are limited to graphical and geometric data of a product. IFC however extends this information to cover more information about the non-product data or what is called the process data. On the other hand, IFC is not considered as a native format for most CAD applications and these applications still use add-on tools to translate from/to IFC.

In addition, the IFC 3D model includes information about locations, relationships and the properties of each element (e.g., Wall), such as its serial ID number, and material layers description (Graphisoft, 2005). In topical 2D CAD formats (e.g., DXF), the CAD application deals with entities such as lines, circles or polygons that are not automatically composing an element such as a wall or a column. IFC-compatible software, however, deals with 3D elements directly where lines, circles and polygons can define an element such as IfcCoumn, IfcWall, and IfcRoof (IAI, 2005b).

The interoperability is the most essential feature in IFC. For instance, IFC allows a user using a CAD application to exchange information with another user (e.g., a cost estimator using Timberline Office). Although these users have different backgrounds and they are working in different environments, IFC allows both of them to access and exchange the relevant project data for each user using his/her own application. Without such a standard, accessing information (e.g., cost, construction scheduling, etc.) could be impossible for non-CAD users. To achieve this goal, AEC-based software developers should work on developing IFC-compatible applications that extend and internationalize the use of IFC. Furthermore, developing such applications allow members of a large-scale project to share the project lifecycle data with even international partners in a consistent, coordinated and electronic format.

The IAI has included industry leaders from all parties of the AEC/FM industry. According to the development structure of IFC in Figure 2.13, the IAI members have participated in defining the general requirements (IFC DEFINITION) for data exchange throughout the building or the facility lifecycle. A number of IFC data models have been released and a large number of software vendors (IMPLEMENTATION) have already adopted and used these standards in their software products. On the other hand, the IAI is not a software development company. The IAI community works independently from the software development companies that serve the AEC/FM industry, but supports developing and providing the IFC specifications. In addition, the IAI enables these software vendors to develop IFC-compatible software applications (IFC MODEL EXCHANGE) that apply the potential of IT in the AEC/FM industry. This common project model serves as the foundation upon which software authors can develop applications that can read and write physical file formats used to communicate with the IFC (IAI, 2005d).

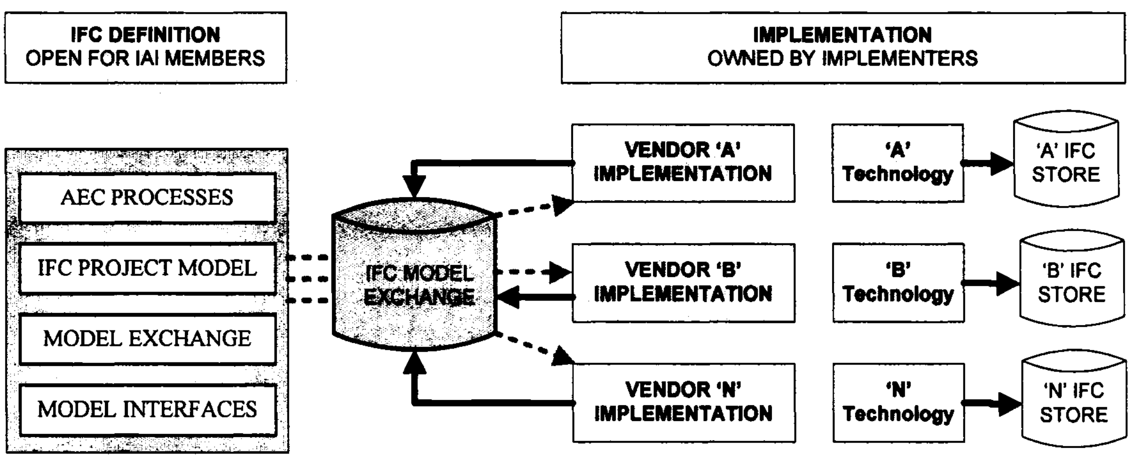


Figure 2.13 The setup for IFC development (IAI, 2005d)

The characteristic an IFC standard identified it as an object-oriented data model. The IAI has made specifications for elements that may be included in a constructed buildings or facilities. These specifications represent clear data structures, called classes, which describe a number of components that have similar characteristics. For example, beams and columns have the characteristic of being structural elements. Thus, *Beams* and *Columns* are names of classes. Then, IAI named them "*Industry Foundation Classes*" to relate them with the industry (IAI, 2005d).

2.3.5.1 The EXPRESS language

EXPRESS is schema language used to develop standards. Therefore, IFC was developed based on STEP and applies the STEP's schema language EXPRESS (ISO 10303-11) as its description method. One of the main objectives of developing STEP is to provide a neutral mechanism that can describe product data throughout the lifecycle of a product. The neutral mechanism is independent of any hardware or software (ISO 10303-11, 1994). In addition to the neutral file exchange, this nature makes it suitable as a basis for implementing and sharing product databases and archiving data. EXPRESS is a description method that can help to achieve this purpose. EXPRESS is used to specify the information requirements of other parts of ISO-10303. It is a conceptual schema language which provides for the specification of classes belonging to a defined domain or attributes pertaining to those classes (e.g., shape, size, color, etc.), and also the constraints on these classes (e.g., unique, exclusive, etc.). EXPRESS can be used to define relations between those classes and the constraints applied on such relations (O' Sullivan et al., 2002).

One of the most important functions of EXPRESS in IFC is to define classes, which is based on the same concept as in STEP. All the attributes and behaviors within the class definition are declared, which eventually characterize the class. As shown in Figure 2.14, a class is declared by the keyword ENTITY and terminated by the keyword

END_ENTITY. The entity declaration creates a class and gives it a name. Attributes, such as Identifier, OwningUser, and OwningApp are the characteristics which are required to support the use of the class. The Attributes can be represented by simple data types, such as IfcString and IfcActor or it can be represented by other classes (IAI, 2005b).

```
ENTITY IfcOwnerID;
Identifier: IfcString;
OwningApp : IfcString;
OwningUser : IfcActor;
END_ENTITY;
```

Figure 2.14 The EXPRESS language definition for a class (IAI, 2005b)

For graphic purposes, EXPRESS-G is another method to represent EXPRESS language. EXPRESS-G is a graphical modeling notation developed within STEP. Similar to EXPRESS, EXPRESS-G is also used to identify classes, data attributes of classes, and the relationships between those classes. Figure 2.15 shows the EXPRESS-G language definition for a supertype/subtype relationship. For the layered element, slab, floor, and wall have already been indicated as subtypes. Each subtype has all the characteristics of the layered element acquired by inheritance. However, each subtype may have additional attributes. To make the subtypes exclusive, that is an IfcLayeredElement may be an IfcWall or an IfcFloor or an IfcRoofSlab, the number "1" is written at the branch of the relation. The term ABS is used with the IfcLayeredElement to show that it is an abstract supertype. This means that it can not exist by itself and only by virtue of its subtypes (ISO 10303-11, 1994; IAI, 2005c).

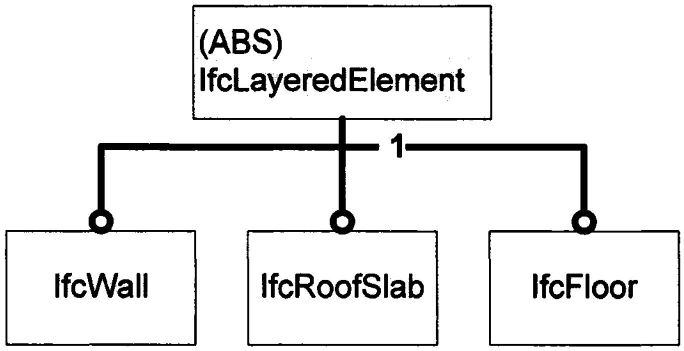


Figure 2.15 The EXPRESS-G language definition for supertype/subtype relationship (IAI, 2005c)

2.3.5.2 IFC-compatible middleware to access IFC data

Interoperability in AEC/FM has been addressed in the early stages of building lifecycle data sharing and exchange, starting from the geometric data of the building design. Therefore, design CAD tools such as Architectural Desktop (ADT) from Autodesk, ArchiCAD from Graphisoft and Microstation Tirforma from Bentley Systems, are IFC-compatible software applications (IAI, 2005). These applications support both import and export CAD geometric drawings from and into IFC files. However, when it comes to tools integration, it is quite difficult for a third-party application to read the huge amount of IFC text data directly. Therefore, developing middleware tools, called toolboxes, is necessary to allow software developers accessing and working with IFC data with less effort in programming. Consequently, some of IAI members have developed and offered their own toolboxes for accessing IFC data, for example, IFC Toolbox from Eurostep, IfcObjectCounter from FZK, EXPRESS Data Manager from EPM Technology, Olof Granlund Oy-BSPro COM Server, and IFCsvr Active Component from Secom Ltd (Ifc-mBomb, 2006). Figure 2.16 shows the three layer relationship between the application, IFC Toolbox and IFC product data and the responsibility of a toolbox in IFC implementation, where the IFC pre/post processor reads and writes the IFC product data by the IFC toolbox.

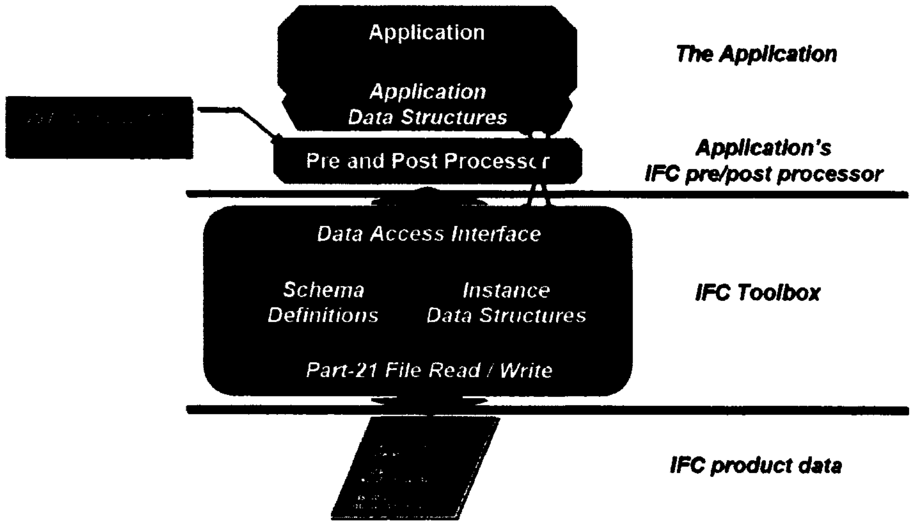


Figure 2.16 Toolbox roles in IFC implementation (Eurostep, 2005)

2.3.5.3 IFC extension schema development

The IAI is involving many international partners in the IFC development processes, which can lead to many extensions of IFC schema in the future. By developing such extensions, IFC will be able to include objects in AEC/FM, which are not included in the current IFC versions. Basically, IFC extensions are developed as extension projects with guidance and quality assurance from IAI technical teams. After the extension implementers finish the development work, the extension models will feed into the next IFC platform release in no less than two to three years (IAI, 2005a).

The COnstruction and Real Estate NETwork (CORENET) project (Hua, 2005) is an example of IFC extensions. This project aims on supporting: (1) collaborative design systems, (2) procurement systems, (3) construction systems, and (4) FMSs. In the overall project, each clause of the building regulation is analyzed for its information input requirements. The required input is then compared with the information provided by the IFC2x product model. If the information requirements are not directly reflected by IFC2x, they will be dealt within the IFC extension project code and standards and will be added to the next addition of IFC. This extension is fully coordinated with other parts of the IFC 2x2, such as IAI BS8 project, which is related to building services. This guarantees that the effort in developing a data model in CORENET project could be accepted internationally (Liebich et al., 2002).

2.3.5.4 Summary of IFC and its applications

The necessity for using a standard data representation to allow data sharing leaded to the development of international standards such as IFC. Applying such standards is the new trend in building engineering. IFC is a well-developed Building Information Model (BIM) and it is an open source standard. It has been also widely accepted by researchers, developers, and engineering software users in the building industry. Based on the understanding of the latest IFC release schema and its extension projects, it was found that it is good approach to develop an IFC-based framework to serve model-based FMSs.

2.4 RESEARCH EFFORTS IN IFC-BASED FACILITIES MANAGEMENT

An FM department, as in all management departments, needs some tools to facilitate its mission and help in making the right decisions. Decision-support tools such as those suggested in the *Building Envelope Life Cycle Asset Management* project (BELCAM at NRCC 1997), which has been discussed earlier in Sub-section 2.2.2.6, can provide standardized interfaces for asset managers (Lounis, 1999). Complex and sophisticated calculations, integrated with numerous computer applications, could provide better understanding and answer many maintenance, repair and renewal questions. Proper user interface provides the facility manager with the relevant data needed to prioritize projects using multi-objective optimization, which eventually help in decision making. An IFC-based research project lunched at the University of British Colombia aimed to implementing standards and defining transaction for data sharing and exchange within the AEC/FM industry. Information sharing and management are major problems in this industry. That is due to the fact that this industry is widely fragmented. Information exchange can generally be described in terms of individual information transactions (Pouria and Froese, 2001). There are several ongoing efforts to standardize the content of data exchange transactions without standardizing the transactions themselves. Standardizing the transactions will potentially provide better communication between users and increased productivity which can reduce costs and control delays.

Applying IFC to represent data model for integrated maintenance management was discussed for roofing systems to demonstrate the applicability of a proposed generic framework for integrating the maintenance management of built-assets. The model builds upon IFC (IFC2.0 and IFC2.x) to define object requirements and relationships for sharing the maintenance information between applications (Hassanain et al., 2000).

An example of interoperability of non-product information is provided by the recent research work on FM. Figure 2.17 provides an overview of the processes involved in maintenance management. It has been developed as an integrated maintenance management prototype (Hassanain et al., 2003) that demonstrates the opportunities of using IFC to enhance interoperability within this domain. The first step is identifying the elements within the inventory of buildings (e.g., the roofing sections) which need to be treated as assets for maintenance management. IFC project data can be used to import product models of the buildings, and identify their elements that require maintenance management. As a second step, these objects are treated by the system as *assets* and represented using the *IfcAsset* class. In addition, maintenance management requires identifying the conditions of these assets relative to prescribed performance requirements. The system represents these requirements using the IFC property set mechanism.

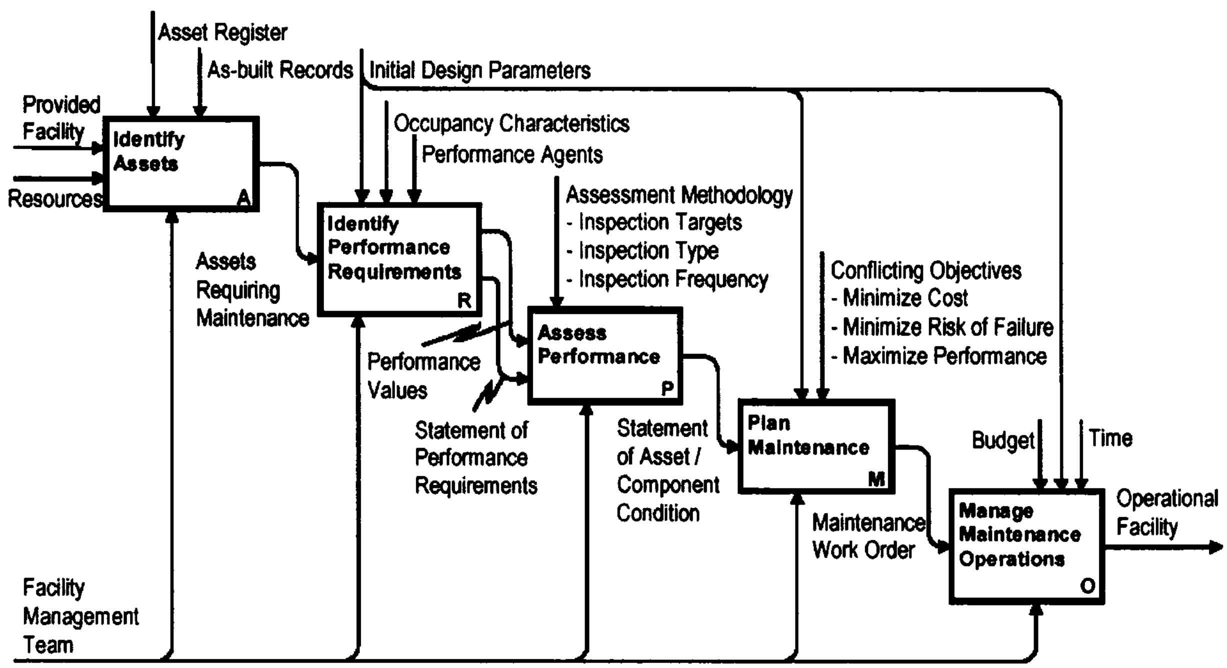


Figure 2.17 Processes involved in maintenance management (Hassanain et al., 2003)

As a third step, the actual condition of the assets relative to the performance requirements is assessed in the system. In the integrated prototype, IFC data exchange is used to load the inventory of assets into a condition assessment application, and then to return the measured asset condition information back into the integrated system.

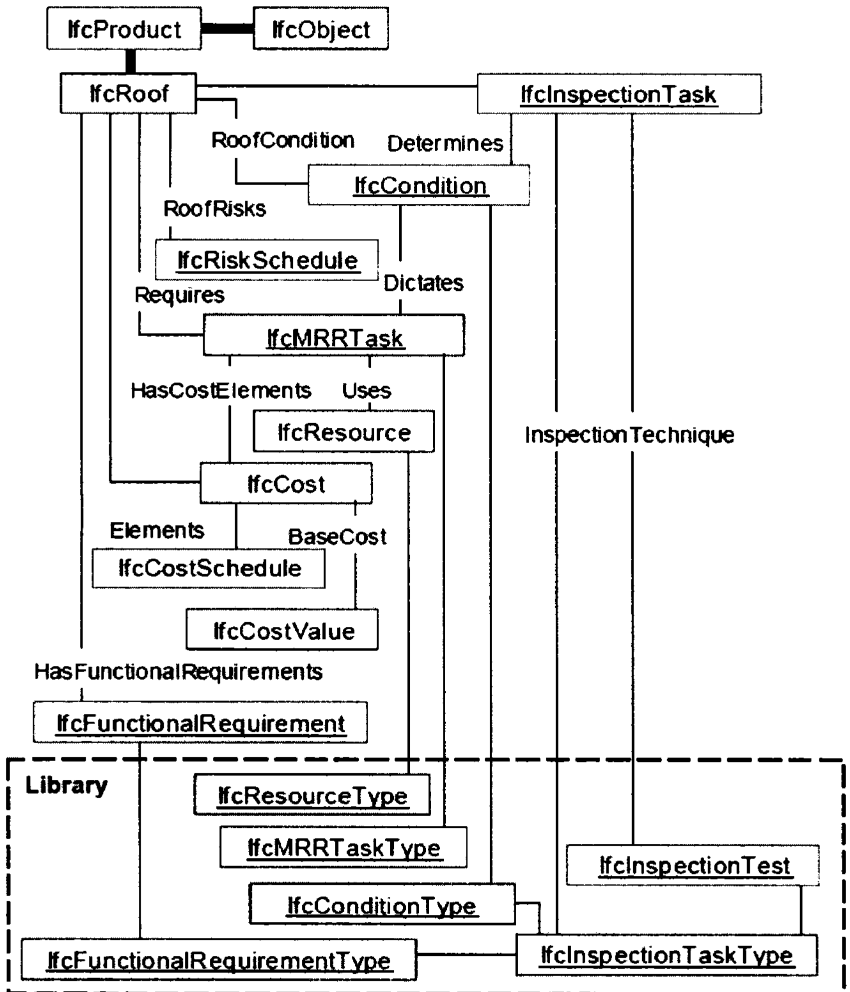


Figure 2.18 Overview of roofing maintenance management data model (Hassanain et al., 2003)

The fourth step uses strategic planning to decide what and when maintenance activities will take place. Therefore, the integrated prototype used IFC to export planned maintenance activities to construction scheduling software. Figure 2.18 shows the planning-related IFC information. This approach can be extended to interact with costing and resource planning systems.

Model-based data standards have been recognized as the main enabling technology for developing integrated AEC project systems (Halfawy, 2005). In the past decade, several efforts have been ongoing to develop standard data models to support data exchange among AEC/FM applications (Froese et al., 1999). Object-based data models normally define schemas that represent the structure of project data in the form of a class hierarchy of objects. The use of an object model has significantly enhanced the consistency of project information, and served to integrate different project aspects and facilitate the exchange of information.

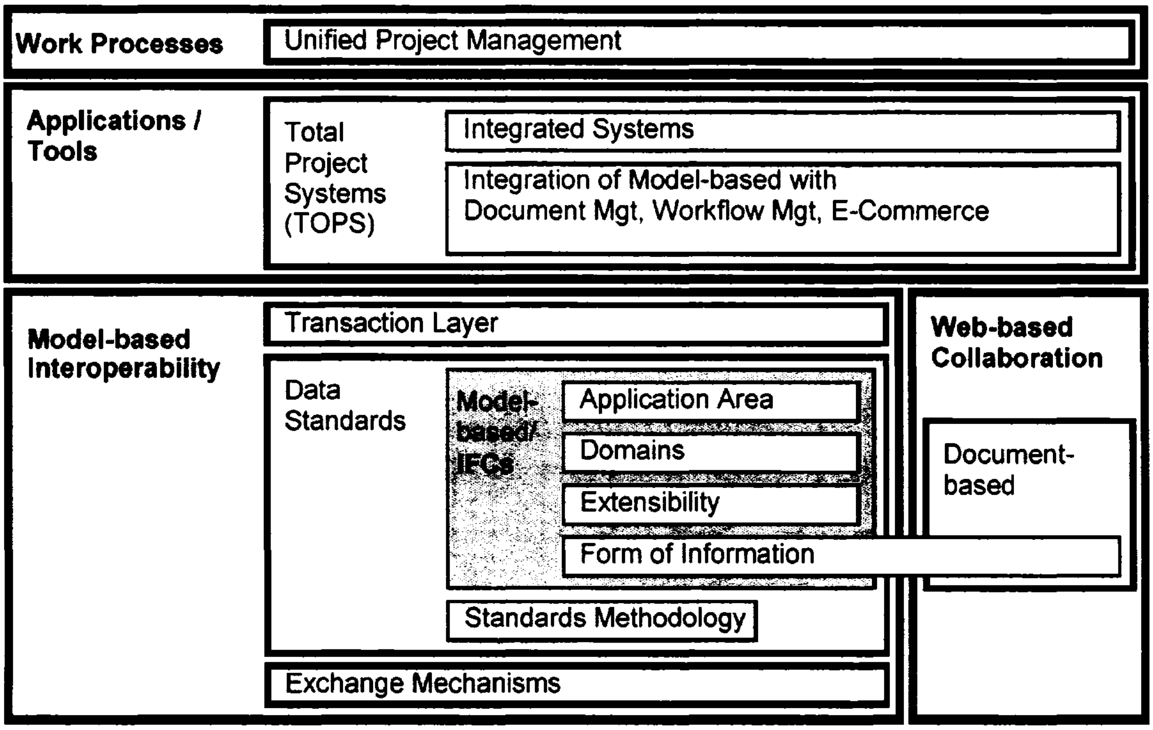


Figure 2.19 The topic of IFC within an overall IT framework (Froese, 2003)

As shown in Figure 2.19, the basic IFC approach to interoperability requires further development and extension. These extension areas are drawn from a synthesis of several

IFC-related research efforts and development projects ongoing at the University of British Colombia.

2.5 XML-BASED STANDARDS

As BIM implementation gains momentum in the AEC/FM industry, the issue of interoperability as means to integrate a variety of model-based applications into an efficient workflow is emerging to the forefront of professional attention. For the broader AEC/FM industry groups, the word interoperability has become synonymous with the IFC effort. However, IFC is not the only open standard for interoperability in the AEC/FM industry. This is primarily because it is not yet complete in its representation of all the disciplines. It serves the needs of most early-stages design, bidding, general contracting, and some FM tasks, but still lacks a sufficiently complete representation of structural engineering. However, several software developing companies such as Autodesk and Graphisoft have put a lot of effort to apply IFC and adopt it as a data format in their applications. Fortunately, new CAD-based application (e.g., ArchiCAD 9.0) has applied and used XML technology, as a main stream, for representing data using IFC2x2. The output of this effort brought the new XML-based data representation (i.e., IfcXML2x2). This new data format allows data sharing and exchange over the Internet.

2.5.1 XML Technology

The eXtensible Markup Language (XML) is a World Wide Web Consortium (W3C) initiative that allows information and services to be encoded with semantically meaningful structure that computers and humans can understand (w3.org, 2006). XML is written in SGML (Standard Generalized Markup Language) (ISO 8879:1986). XML technology plays a critical role in all software development projects and it is an efficient language for information exchange (not a programming language), and can easily be extended to include user-specified and industry-specified tags (e.g., books listing, e-business pricing, etc.). XML is a language for describing other languages (metalanguage), which lets users design their own markup languages for limitless different types of documents (XML, 2005). XML technology has received widespread support and adoption in the computer industry because of its simplicity, flexibility, extensibility, and interoperability. XML also improves the functionality of the web by letting users identify their information in a more accurate, flexible, and adaptable approach. For example, Google Earth uses XML-based format called KML (Keyhole Markup Language) to define locations and their description (Google Earth, 2006). It is extensible because it is not a fixed format like predefined markup languages such as Hypertext Markup Language (HTML).

The simplicity, interoperability and extensibility of XML encourages the AEC/FM industry to develop and use variant XML-based formats such as LandXML, aecXML and IfcXML.

2.5.2 LandXML

LandXML is an industry standard schema that allows exchange of data created during the surveying process by making the data more readily accessible to all participants in the project. This standard also provides interoperability between different applications. In addition, LandXML is a web compatible and will eventually allow engineering participants to share data over the Internet. LandXML is supported by many participants and interested groups and organizations in the AEC industry (Landxml, 2006). The first version of LandXML was ratified in July 2002 as an international industry standard. This provided assurance to software developers that LandXML is a stable, recognized standard that they can support with some confidence. LandXML has been developed by Autodesk in consultation with other software developers, vendors and users such as civil engineers and surveying groups. The leading software developers in the AEC industry have come together in partnership with federal, state, and local government agencies as well as larger engineering consulting firms to develop an industry standard data format for land development projects.

2.5.3 AecXML

AecXML was initially proposed by Bentley Systems in 1998 to satisfy the needs of the AEC industry, and is now being part of the IAI (Cheng et al., 2002). AecXML is an XML-based language used to represent AEC information and it is AEC-specific standard. This standard is capable of representing resources information such as projects, documents, organizations, materials, professionals, as well as processes or activities information, such as proposals, design, cost estimating, scheduling and construction. Figure 2.20 demonstrates the architecture of the aecXML framework.

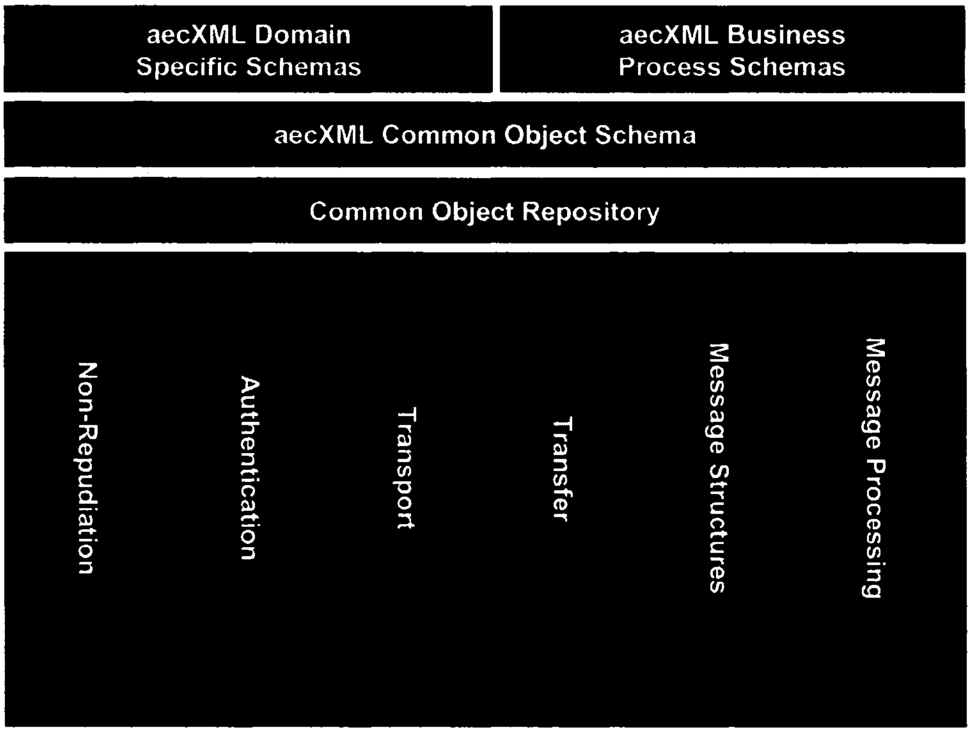


Figure 2.20 The architecture of the aecXML framework (AecXML TC, 2000)

Furthermore, aecXML specifically focuses on AEC transactions to avoid duplicating work already in established standards. Some of these AEC-specific needs can directly impact both the business documents and the elemental parts of the standard. In AEC environment, the aecXML is chosen based on what is specifically required to make a key business process (e.g., purchase orders and payment applications). For instance, invoicing in a project where the project contractor delivers based on phase and obviously the invoice will be based on phase. Consequently, she/he will not get paid if the invoices do not include AEC-specific information. AecXML standard can build on standard definitions such as the IAI's IfcXML) but must include distinct refinements before these transfer standards truly deliver in an AEC setting.

The aecXML standard can be also defined as a standardized language of communicating information in a business transaction and not only as a repository for storing these information. For this reason, aecXML is different from model-based standards such as IFC's, which can support a model server that keeps the entire project's design data. However, using aecXML within a defined business process, much of the information is understood from context, and the user can achieve remarkable process improvements with a much focused transfer. For example, in cost estimation, the architect would take out and send the estimator the necessary parameters and a subset of the model. The architect, on his/her side, specifies that this data is accurate. The estimator, on the other hand, would do his/her own checks on the data, and then send back an estimate based on double-checked accurate data.

2.5.4 IfcXML

The IfcXML representation is an implementation of ISO-10303 Part 28 Edition 2 standard. This standard provides an XML schema specification that is an automatic conversion from EXPRESS (ISO 10303 Part 1) representation of IFC schema. The mapping from EXPRESS to XML schema is guided by a configuration file that controls the specifics of the translation process. For IfcXML this configuration file is standardized and published for each version of the corresponding IFC schema (IAI, 2006).

There are many tools support read/write XML files. For example, XMLSpy is a general XML tool that supports editing, parsing and converting XML schema and documents (XMLSpy, 2005). The tool has no awareness of Part28 and IfcXML, but maybe used to validate and further process IfcXML documents. The latest version XMLSpy 2005 has a greatly improved and tighter validation against the schema. Eurostep Model Server is another tool that allows writing IfcXML files. This EXPRESS based server tool supports Part28 conversion of populated data models. The tool accepts the IfcXML configuration file to write IfcXML data (IAI, 2006). Table 2.2 shows the different representation between IFC and IfcXML. The relationships in IFC are defined using number (e.g., # 86). Each number refers to another IFC instance which links the related components in the IFC model. In IfcXML however, the relationship between nodes are defined as Parent-Child relationship in a hierarchy tree list. The representation of IfcXML is much easier to understand the related components.

Table 2.2 Comparison between IFC and IfcXML definitions

IFC format data sample	ifcXML format data sample
#84=IFCPROPERTYSINGLEVALUE ('**Red**',$,IFCINTEGER(**255**),$); #85=IFCPROPERTYSINGLEVALUE ('**Green**',$,IFCINTEGER(**0**),$); #86=IFCPROPERTYSINGLEVALUE ('**Blue**',$,IFCINTEGER(**0**),$); #87=IFCCOMPLEXPROPERTY ('**Color**',$,'**Color**',(#84,#85,#86));	<IfcComplexProperty id="i87"> <Name>**Color**</Name> <UsageName>**Color**</UsageName> <HasProperties> <IfcPropertySingleValue> <Name>**Red**</Name> <NominalValue> <IfcInteger>**255**</IfcInteger> </NominalValue> </IfcPropertySingleValue> <IfcPropertySingleValue> <Name>**Green**</Name> <NominalValue> <IfcInteger>0</IfcInteger> </NominalValue> </IfcPropertySingleValue> <IfcPropertySingleValue> <Name>**Blue**</Name>

2.6 VIRTUAL REALITY TECHNOLOGY

Virtual Reality (VR) is a technology that builds a virtual three-dimensional (3D) model in a computer to visually reproduce the shape, texture and movement of objects (Miyamoto et al., 2005). VR can be also defined as a computer-generated simulation of a real or imagined environment which a user can experience. Other virtual objects can be added in the virtual scene such as text objects, shapes and 3D sound effects. The environment where all these models consist is known as VE. The VE may represent a real-world object, such as a house; it might be an abstract model or detailed one that serves the engineering applications such as those introduced in AEC industry. In addition, EV can be also used to represent and model completely imaginary science fiction world such as interactive 3D game environments (e.g., SimCity).

The main key features in using VR are: (1) The user believes that he is actually in this different world; and (2) The interaction capabilities with the model so that if the human moves his head, arms or legs, the shift of visual cues must be those he/she would expect in a real world. In other words, besides immersion, there must be navigation and interaction.

An advanced application of using the virtual models is the Augmented Reality (AR) mode. This technology combines the viewing of the real-world or video-based environments with superimposed 3D VR models that can be manipulated by the application user. Thus, AR supplements rather than replaces the user's real world (Virtual Reality Laboratory website, 2006). The most recent development in AR is a wearable system in which users wear a backpack with a portable computer, see-through Head-Mounted Display (HMD), and headphones with motion trackers to place and manipulate VR objects as they move within their real world environment (Halden Virtual Reality

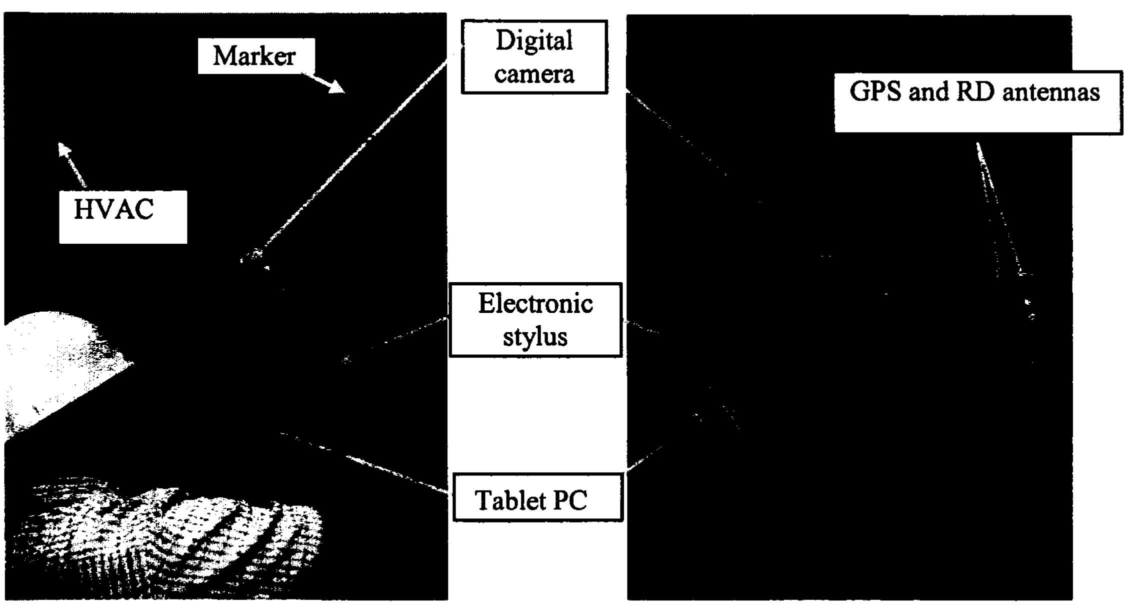
Center, 2006). Figure2.21 shows an example of using AR technology in urban planning. A round table is used as the base of the VR model and the users (urban planners) gather around it wearing HMDs and interact with the model using pointing devices (Moeslund et al., 2004). The Virtual Reality Modeling Language (VRML) is one popular language to represent VR models (Virtual Reality Laboratory website, 2006).

Figure 2.21 An application of AR in urban planning (Moeslund et al., 2004)

2.7 TRACKING METHODS FOR MOBILE LOCATION-BASED COMPUTING

There are three tracking methods used in Location-Based Computing (LBC) to satisfy the requirements of model-based FMS in mobile situations: (1) Tracking the location of the facility inspector in the VE; (2) Tracking the real-world location of the FM inspector for indoor applications; and (3) Tracking the real-world location of the inspector for outdoor applications. Garrett et al. (2002) discussed the issues in mobile and wearable computer-aided inspection systems for field tasks. Sunkpho et al. (2002, 2003) developed the mobile inspection assistant that runs on a wearable computer.

Tracking the location of the inspector in the VE can be done by tracking the viewpoint of the user. There are several sensor-based positioning systems that can be used for indoor tracking, such as video, electromagnetic, infrared or ultrasonic systems (Karimi and Hammad, 2004). Video tracking technique is used to track physical markers by means of a video camera (Kato et al., 2006). The GPS-based tracking can be used for outdoor tracking. Figures 2.22 (a) and (b) show the video-based and GPS tracking methods that can be used for indoor and outdoor tracking, respectively (El-Ammari et al., 2006; Mozaffari et al., 2005). In both indoor and outdoor scenarios, the FMS user is equipped with a tablet PC, an electronic stylus, and a small video camera. The camera is fixed on the hardhat for taking pictures and recording video clips of the inspected elements (Hammad et al., 2006). However, as shown in Figure 2.22(a), the video camera has an additional function of recognizing markers attached to a building component (ceiling) at known locations. This helps the system keep track of the user location.



(a) Indoor inspection using Video-based tracking

(b) Outdoor inspection using GPS-based tracking

Figure 2.22 Examples of location-based inspection for FMIS

Tracking technologies such as RTK-GPS can be used for exterior applications using Trimble 5700 RTK-GPS receiver, and video tracking using ARToolKit for interior applications (Kato et al., 2006). The ARToolKit has been developed allowing video tracking of markers using a video camera and computer vision algorithm to calculate the camera position and orientation relative to physical markers in real time (Hammad et al., 2006).

2.8 MOBILE AND LOCATION- BASED COMPUTING ISSUES

As discussed in the previous section, LBC is based on tracking the location of the user and providing information based on this location in a distributed mobile computing environment (Satyanarayanan, 2001; Davies et al., 2001). For instance, data collection for FMSs can be done using a tablet PC equipped with a tracking device that allows dispatchers of FM work orders view and monitor workers based on their real-time location and assign appropriate tasks based on this location. Besides, the virtual model acts as a guide for the FM inspectors to locate the facilities components in 3D representation. This approach enables inspectors to interact with the VE, which significantly enhances their understanding of the work environment.

The key component of LBC is the mobile computing platforms. These platforms, such as Tablet PCs, are being developed with wireless communications and integrated barcode technologies for real-time inventory and work order applications. Advanced data collection technologies will allow FM field workers to download, view, and update schedules as well as update work orders, issue and receive inventory items, track product or equipments movements, and communicate with FM personnel in inspection or maintenance operations. Dennis (2003) introduced a conceptual model of integrated FMIS. Figure 2.23 show the main parts of this integrated framework. Different resources (EDMS, CPMS, CMMS, CAFM, and SRS) are integrated through a Local Area Network (LAN) for FM. The SRS is used to allow external users to submit FM requests. This facilitates the facilities data sharing through the World Wide Web.

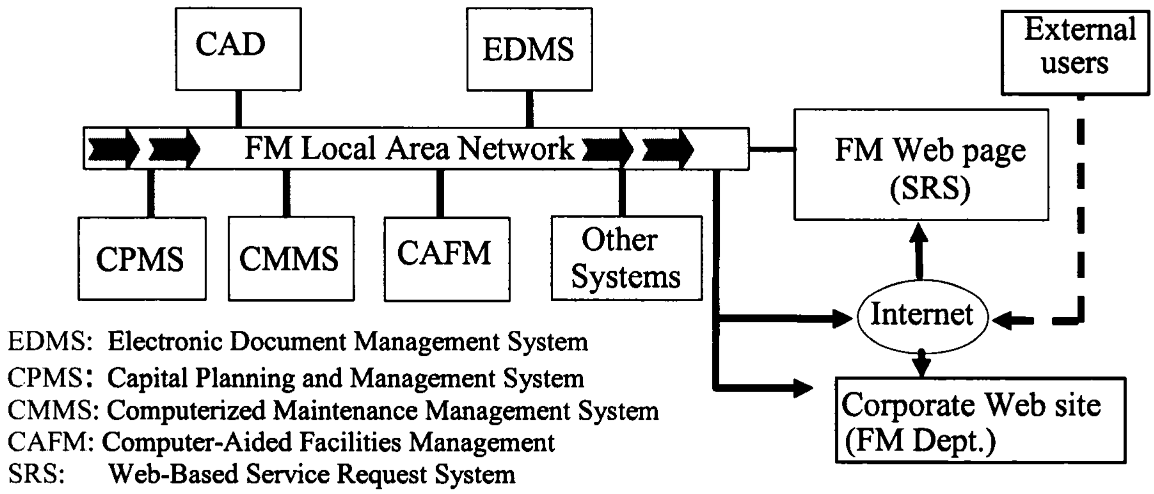


Figure 2.23 Conceptual Model of an Integrated FMIS (Dennis, 2004)

Hammad et al. (2004) discussed the concept and requirements of data collection system in mobile situation for engineering field tasks. Figure 2.24 shows the conceptual framework called LBC for Infrastructure field tasks (LBC-Infra).

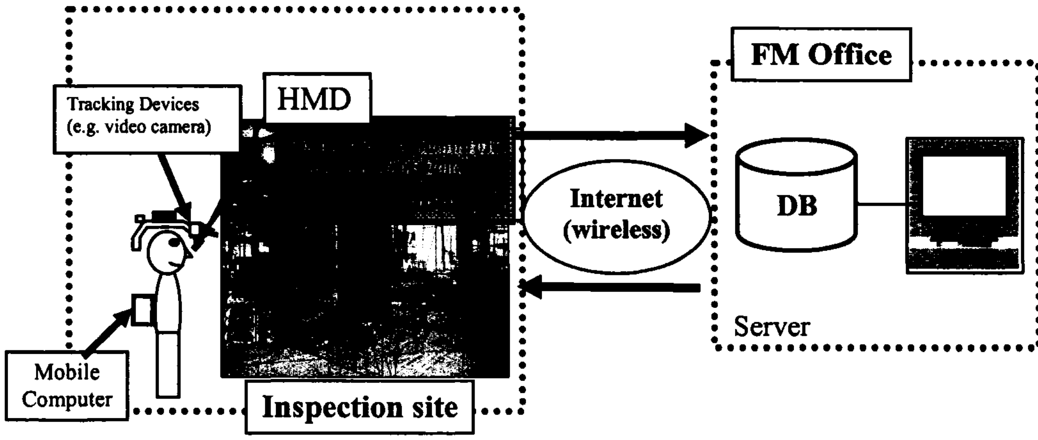


Figure 2.24 Concept of LBC-Infra (Hammad et al., 2004)

In this figure, an FM inspector, equipped with a mobile or wearable computer, is inspecting a building facility searching for damages (e.g., corrosions, cracks, etc.). The wearable computer has a wireless communications card and is connected to tracking devices. Based on the location and orientation of the inspector and the task required, the system may display information about the parts of interest within the inspector focus. To guide the inspector in the VE, the system provides a navigation arrow pointing at the location where damages are most likely to be found. The inspector compares the changes in conditions by wirelessly accessing and viewing any of the previous inspection reports stored in the office database using spatial queries based on his/her current location and orientation. The spatial database of the facilities and the surrounding environment, and the tracking devices attached to the inspector, make it possible to locate facilities structural components and detected problems and provide navigation guidance to these objects. In addition, all these collected information about damages and structure problems are tagged in the VE.

2.9 SUMMARY AND CONCLUSIONS

In this chapter, the literature about the current FM applications has been reviewed. There is significant development in those applications in terms of satisfying the FM requirements such as space management, condition assessment, equipment tracking records, etc. Data collection in these applications is still done in the way of paper-based and semi-automated level, which makes the collected data inaccurate and inefficient. These inspection data can directly affect building condition evaluation and maintenance decisions.

In addition, there are still interoperability limitations regarding standardizing data representation for these assets or facilities. This chapter provides a comprehensive review about AEC/FM standards and the available modeling tools that allow the creation of these standards. This review and comparison result in recommending the use of IFC and IfcXML representations provided by IAI in FMSs.

Visualization is another issue in FM applications and providing interactive VE environments is essential in such applications to allow FM personnel to interact with the facility model and to record their inspection information on the model of the facility instead of using the conventional paper-based methods. VR modeling is a great solution to satisfy these interactive FM systems.

To overcome the limitations of present FM applications and to satisfy future FMSs' requirements, the objectives of this research are to develop a model-based FMS with integration of various visualization and interaction models to facilitate on-site data collection, and investigating various standards that satisfy this prototype system requirement.

CHAPTER 3 MODELING AND VISUALIZATION IN MODEL-BASED FACILITIES MANAGEMENT SYSTEMS

3.1 INTRODUCTION

Model-based applications in FM domain require visualization as well as suitable modeling methods. The user of such applications (e.g., facility inspector) needs to visually recognize the 3D model of a building or a facility and interact with it to retrieve information associated with a particular building element. This research proposes using an IFC-based model of a building (i.e., geometric and process information) and the VRML for visualization. Furthermore, images are used for representing the details of the exterior of the buildings (i.e., texture mapping). Standardizing data representation is a key issue in such applications to enhance interoperability and achieve better results of minimizing cost and improving productivity.

Building non-graphical attributes are extracted from the IFC model and stored in a relational database that allows the FM inspector to update the model information. The creation of this database will be discussed in this chapter. The updates of these stored building information require model interaction and database organization relationship. Therefore, VRML-IFC data linkage is essential in such applications to instantly connect the visual version of the model (VRML model) with the related information (IFC model).

As introduced in Chapter 2, FM applications have been useful for many services such as condition assessment. However, interoperability is not considered in those applications. A facility data comes from different sources of information, and each source has its own data format. For example, the facility model can be in a CAD format, such as DWG or 3DS, whereas the process data can be represented in different formats such as cost spreadsheets or scheduling Gantt charts. Considering automation for data collection and other data processing, working with different data formats may cause redundancy or data conflicts.

Data sharing among applications is a hard task to achieve with the conventional methods (e.g., paper-based data sharing). In order to create a large-scale 3D virtual model, huge amounts of time, effort and money have to be invested to build this model and integrate it with the necessary data.

This chapter introduces a framework for data integration method to create large-scale virtual environments. This model can be created by automatically integrating GIS maps, CAD models, images of buildings' facades, and database for the cost, scheduling, and other lifecycle data of buildings. In addition, this chapter discusses a framework for using these resulting models to satisfy the requirements of the MBFMS. The system can be used by facilities' mangers and inspectors and instantly allows them to interact with geo-referenced facilities models. In addition, the system allows automatic retrieving of the necessary information based on the user location and orientation, and the task context (Hammad et al., 2004). Further discussion will be introduced about the creation of the virtual detailed models of the building interior as well the IFC models.

3.2 IFC-BASED BUILDING MODELING

The product model used in any industry (e.g., AEC/FM) has geometric data and other information related to this product, which may be called non-product or process data. The geometric data of a building provide the graphical representation of the building model, while the process data represent information about how to build, construct and maintain this building. This research proposes using IFC to represent both building geometric model and the process model. This section discusses the main steps of constructing IFC-based model using IFC-compatible CAD systems. Figure 3.1 shows the main steps used to create multi-model IFC-based database.

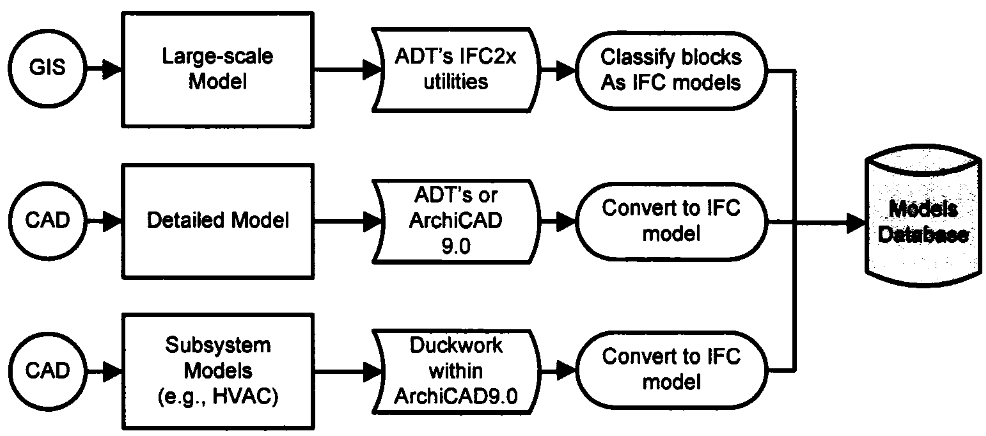


Figure 3.1 IFC-based building model databases

3.2.1 Data transfer from CAD drawings to IFC files

The IFC2x is physically complied with the regulations prescribed in STEP ISO-10303-21 (ISO, 2002). There is a huge amount of information related to a building; the data of a building are usually expressed in thousands of lines in an IFC text file. Therefore, it is almost impossible to manually generate an IFC text file from CAD drawings. As an alternative, this is normally automatically implemented by computer software. Building model information can be transferred to IFC model through IFC-compatible CAD applications. The Autodesk's Architectural Desktop (ADT) with its IFC2x Utility (Inopso GmbH, 2005) and Graphisoft's ArchiCAD with its add-on interface (Graphisoft, 2005) are two typical CAD applications that can generate an IFC file from the CAD drawings. These two particular applications, with other add-on tools, are capable of creating 3D models of the building's subsystems such as HVAC, mechanical and electrical system models. The add-on tools for IFC translation can be downloaded freely from the Web. ArchiCAD 9.0 is also capable of creating IfcXML2x. The XML-based representation of the IFC information makes data retrieving much easier.

3.2.2 Creating the IFC model

In most 3D CAD drawings, objects are described by there shapes. For example, a building element such as a round column is descried as a cylinder that has a diameter, height and other geometric and location attributes. However, this representation does not differentiate between a round column and a vertical pipeline. This limitation of CAD drawings explains the necessity of using an interoperability standard such as IFC. To describe large-scale urban environment in IFC representation, especially when considering using location-based building engineering applications, the following information needs to be attached to the 3D objects: (1) the geometry of the bounding box that represents the building, (2) the location attributes of this bounding box, and (3) the building information (e.g., name: library building, address: 1010 Maisonneuve blvd., etc.).

This research project recommends using some CAD applications (e.g., ArchiCAD 7.0 and above and Architecture Desktop) to assure compatibility with IFC releases. ArchiCAD 9.0 student version has been used to create the IFC2.x and IfcXML2x models. There are two different ways to create these models; one method is to import the original 3D CAD model and classify each object manually, for instance, a round column in the CAD model can be classified as IfcColumn entity which specifies child entities such as IfcsiUnit, IfcDirection, IfcShapeRepresentation, IfcMaterial, etc. The following is a description of the content of an IFC model:

Schema:

```
ISO-10303-21;
HEADER;
```

File information:

```
FILE_DESCRIPTION(('ArchiCAD 9.00 Release 1 generated IFC file.','Build
Number of the Ifc 2x3 interface: 63005 (05-01-2006)\X\0D\X\0A'),'2;1');
FILE_NAME('C:\\columnifc2x3.ifc','2005-08-
20T20:16:25',('Architect'),('Building Designer Office'),'PreProc - EDM
4.5.0033','Windows System','Infra');
FILE_SCHEMA(('IFC2X3'));
ENDSEC;
```

Model data:

```
DATA;
#1= IFCORGANIZATION('GS','Graphisoft','Infra',$,$);
#5= IFCAPPLICATION(#1,'9.0','ArchiCAD 9.0','ArchiCAD');
#6= IFCPERSON($,'Khaled El-Ammari',$,$,$,$,$,$);
#8= IFCORGANIZATION($,'Concordia - Infra Group',$,$,$);
.
.
#103= IFCCOLUMN('COL(H_7)',#13,'C26',$,$,#188,#177,$);
.
.
ENDSEC;
```

Closing:

```
END-ISO-10303-21;
```

In addition to the method explained in Figure 3.1, another method to create the IFC model is by directly using an IFC-compatible application, which allows creating the building elements as IFC entities (e.g., *IfcColumn*, *IfcWindow*, *IfcDoor*, etc.). This method assures that the UID number is the same as in the VRML model created with the same CAD application. The identical IDs are used to link both the IFC and the VRML model to allow interaction with the virtual model, which will be explained in more detail in Chapter 4.

Using an add-on tool for ADT 2005 helps creating and classifying CAD objects as IFC objects. Figure 3.2 explains the steps of creating and classifying each building element as an IFC entity to create a large-scale urban environment.

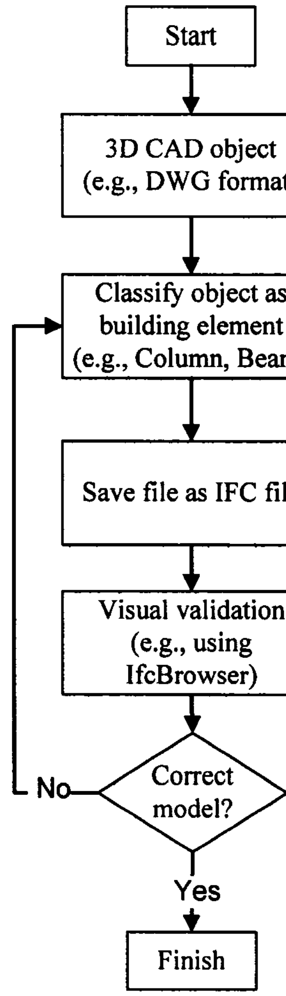


Figure 3.2 The process of creating an IFC model for large-scale urban environment

To validate the IFC model created with this method, an IFC visualization tool (e.g., IFC Engine Viewer) can be used to visually inspect the model (IFC Browser, 2005). This IFC viewer allows inspecting the geometry of the IFC model and the hierarchy relationships between the model components. Figure 3.3 shows an IFC model using IFC Engine Viewer (TNO, 2005).

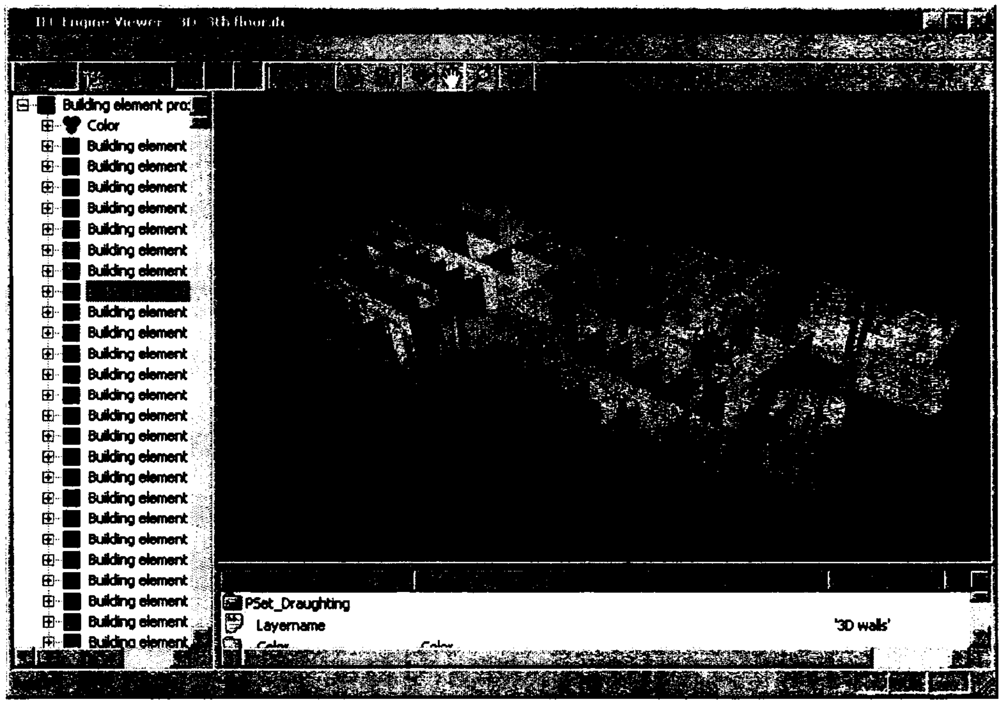


Figure 3.3 IFC model visualization tool

3.3 DATA INTEGRATION FOR CREATING THE VIRTUAL MODEL

To develop VR applications for large-scale FM, it is important to use a systematic technique in creating the virtual model. A large-scale facility (e.g., a university campus) can be represented by maps using commercial GIS software. However, this technique focuses only on creating models of the footprints of buildings and do not represent the structural elements which are necessary in FM applications. These elements are represented using CAD techniques. This research proposes an innovative framework to integrate different data sources to systematically create a virtual model of a large-scale Virtual Environment (VE) including building details. The proposed data integration method for creating the virtual building models is based on synthesizing information from GIS maps, images of building facades, CAD models, and IFC-based databases of the cost, scheduling, and other data about the buildings lifecycle (e.g., inspection and maintenance). Taking into consideration the Level of Details (LoD) concept for the VE, two representations are created for each building with different LoD: one is for the exterior of a building (LoD_e) and the other one is for the detailed 3D model of the interior of the same building (LoD_i). Buildings' façade images can be used as texture mapping for the 3D models to make the buildings more realistic and recognizable by the user.

The attributes of these buildings which are generated throughout their lifecycle will be added to the building graphical model by the user of the system at different stages (e.g., design, construction, and inspection). In a more advanced scenario, the building model can be used as an interface for a web-based collaboration system. The building model may include a number of subsystem models such as the HVAC model to fulfill the requirements of FM applications. Figure 3.4 explains the main steps of the integrated method for creating the building model and consists of three phases (Phase: A, B, and C): (A) preparing data such as CAD drawing, GIS maps, buildings façade images, and other process data, (B) creating building models with LoDs, and (C) interact with these models based on the user location to retrieve and update the facilities information from the database.

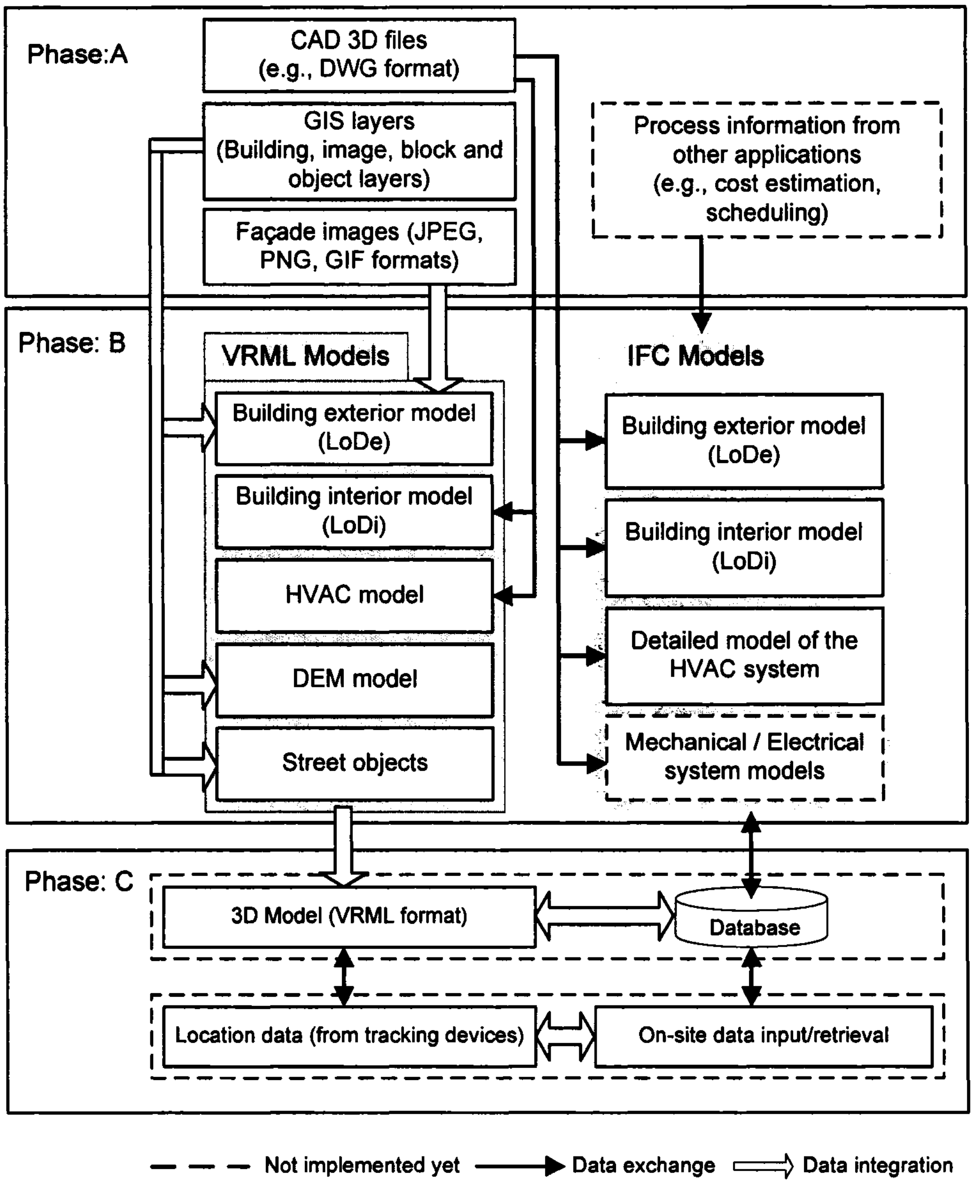
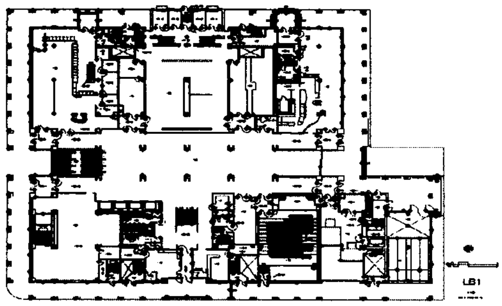


Figure 3.4 Data exchange and integration method

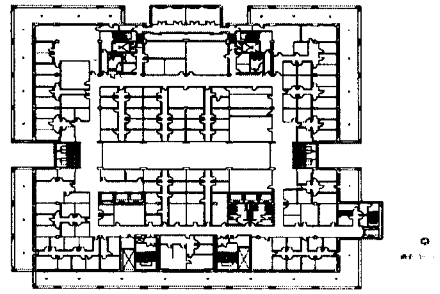
This integration method aims to create the virtual model of the buildings, which is used for visualization and interaction purposes in the VE. On the other hand, the non-graphical attributes of buildings and building elements are extracted from the IFC representation and stored in the database. Similar to the VR building model, the IFC model has also two levels of details (Figure 3.4 (Phase B)): one is for the exterior of a building, which is the bounding box representation, and the other is for the detailed 3D model of the interior of the same building. The attributes of the subsystem models, such as the HVAC model, are also extracted from IFC files and stored in the database.

3.3.1 Creating GIS layers

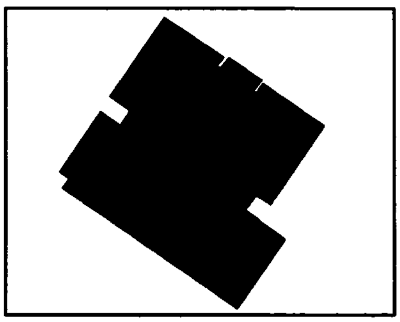
The model of a large-scale urban area can be created using a GIS map that represents the footprint of building blocks. To construct the exterior representation of a building (LoD_e), a GIS-based polygon layer containing the footprints of buildings is prepared. CAD data are used to create the detailed footprint for a building as shown in Figure 3.5 (a) and (b). After that, attributes are added to each polygon to indicate the base level and the height of the polygon. For visualization purposes, texture mapping techniques can be applied (Figure 3.4 (Phase A)). The images of the buildings' facades can be collected and processed to create orthogonal images representation. These modifications can be done using any image processing tool that has suitable editing features (e.g., Photoshop 9.0). The modified facades images are pasted to the surfaces of the models of the buildings. The locations of the facades images corresponding to these surfaces are defined by an image layer in the GIS. This layer contains lines and each line represents a facade image' location. Attributes are added to the image layer such as the base level, height, and file name of each image.



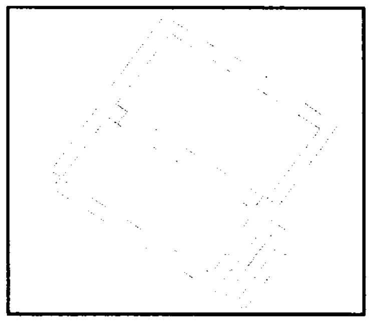
(a) 1st floor plan
(for extrusion from 1st to 4th floor)



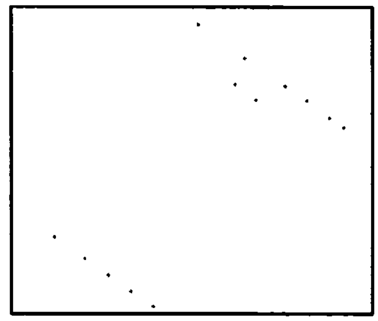
(b) 5th floor Plan
(for extrusion from 5th floor)



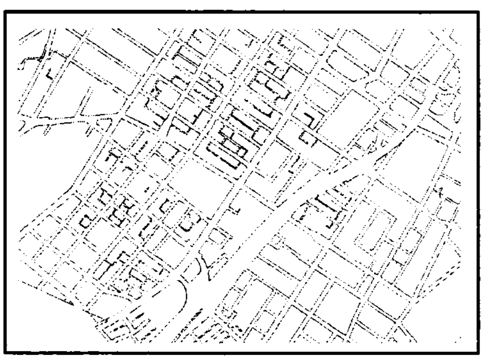
(c) Building layer (polygons)



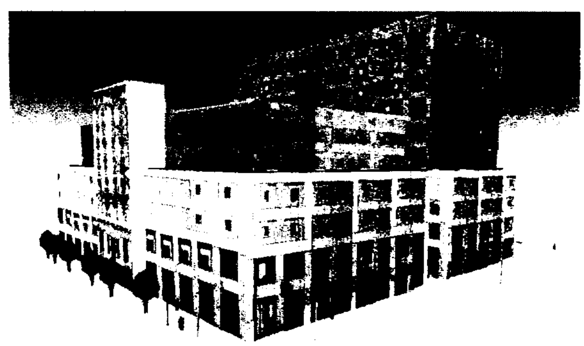
(d) Image layer (lines)



(e) Object layer (points)



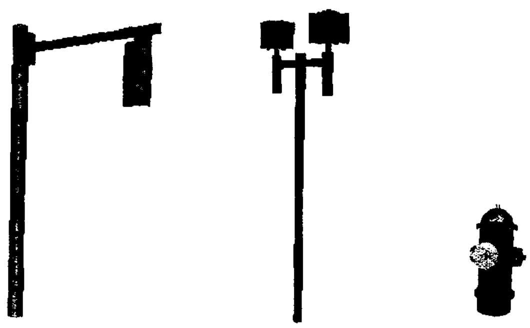
(f) Block layer (polygons)



(g) Rendering example of a building exterior

Figure 3.5 Example of data used in creating 3D models

Figure 3.5(d) shows an example of the image layer for a building. Figure 3.5(f) shows an example of the block layer that has been added in the polygon layer to represent pedestrian sidewalk areas surrounding buildings. In addition, other street objects are added to the 3D VE such as traffic lights, street lights and street furniture. These objects are located in the VE using one point for each object. An object layer of these points is created (Figure 3.5(e)). A standard library of 3D street objects is used to simplify the modeling process (Figure 3.6).



(a) Traffic light (b) Street light (c) Fire hydrant

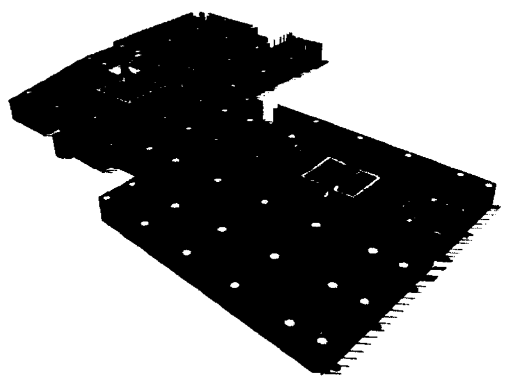
Figure 3.6 Examples of objects from the 3D object library

3.3.2 Translating 3D shapes to virtual reality model

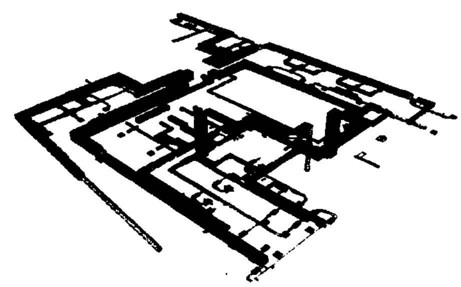
The above mentioned layers (building, image, block and object layers) are used to automatically extrude the 2D shapes into 3D shapes. The resulting shapes are texture mapped and inserted into the virtual 3D model. An example of the result of this synthesis is shown in Figure 3.5(g). The interior and the subsystem models are created automatically using VRML-compatible CAD application (e.g., ArchiCAD 9.0).

3.3.3 Creating interior models of buildings

The interior detailed model is essential in FM. The individual interaction with each building or structure element is the basic concept in these applications. The 3D CAD model of the interior representation of a building (LoD_i) can be prepared using IFC/VRML-compatible CAD application (e.g., ArchiCAD 9.0). The resulting CAD model is translated to both VRML and IfcXML2x. Figure 3.7(a) shows an example of a floor in a building, where Figure 3.7(b) shows a model of the HVAC system for the same floor.



(a) Model of one floor



(b) Model of HVAC system

Figure 3.7 Examples of detailed 3D CAD models

To spatially locate the model of the building in the 3D virtual model, two points on one edge of the building are identified in both models. These points' coordinates are used to calculate and apply model transformation. This transformation includes rotation, scaling and translation from the local coordinate system of the CAD model to the global coordinate system of the virtual model. This transformation is necessary because of the differences between the orientations of the coordinate axes used in different modeling and visualization applications. For instance, the axis in the height direction is considered as the Z-axis in 3D Studio Max whereas it is considered the Y-axis in Java 3D (the 3D

API of Java language). These differences need to be considered to apply correct transformation and precisely locate the interior models.

3.3.4 Transferring IFC data into a database

Building data represented in IFC2x or IfcXML2x formats can be transformed into database tables considering hierarchy structure and elements relationships. This research project uses two methods of reading IFC data: (1) Using an application developed using a middleware (Eurostep, 2005) for accessing, reading and writing IFC instances; and (2) Using an IfcXML parser. Figure 3.8 explains the process of parsing an IfcXML model.

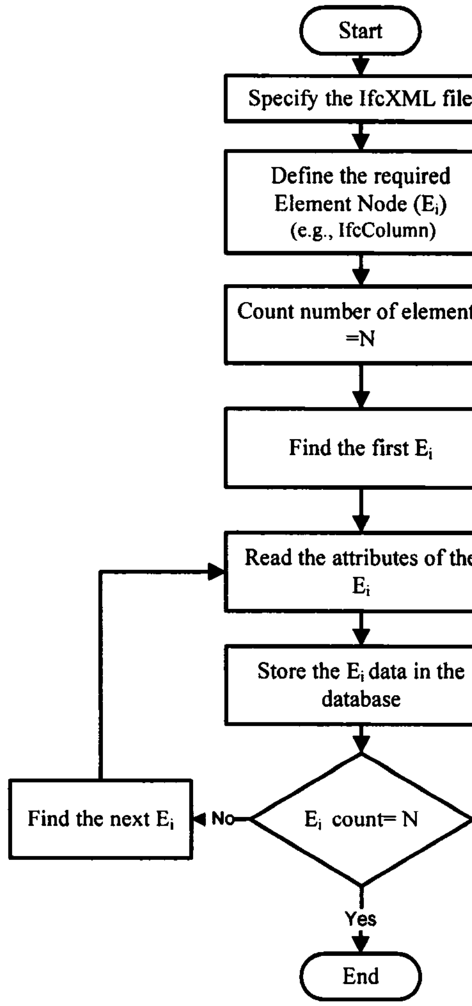


Figure 3.8 Main steps of parsing an IfcXML file

The IfcXML parser starts reading by specifying the IfcXML file. After defining the *Element Node* (E_n), the parser starts looking for this node and reading its attributes and save them in the element table in the database. This parsing is applied on all building components such as *IfcColumn*, *IfcWall*, *IfcWindow*, etc.

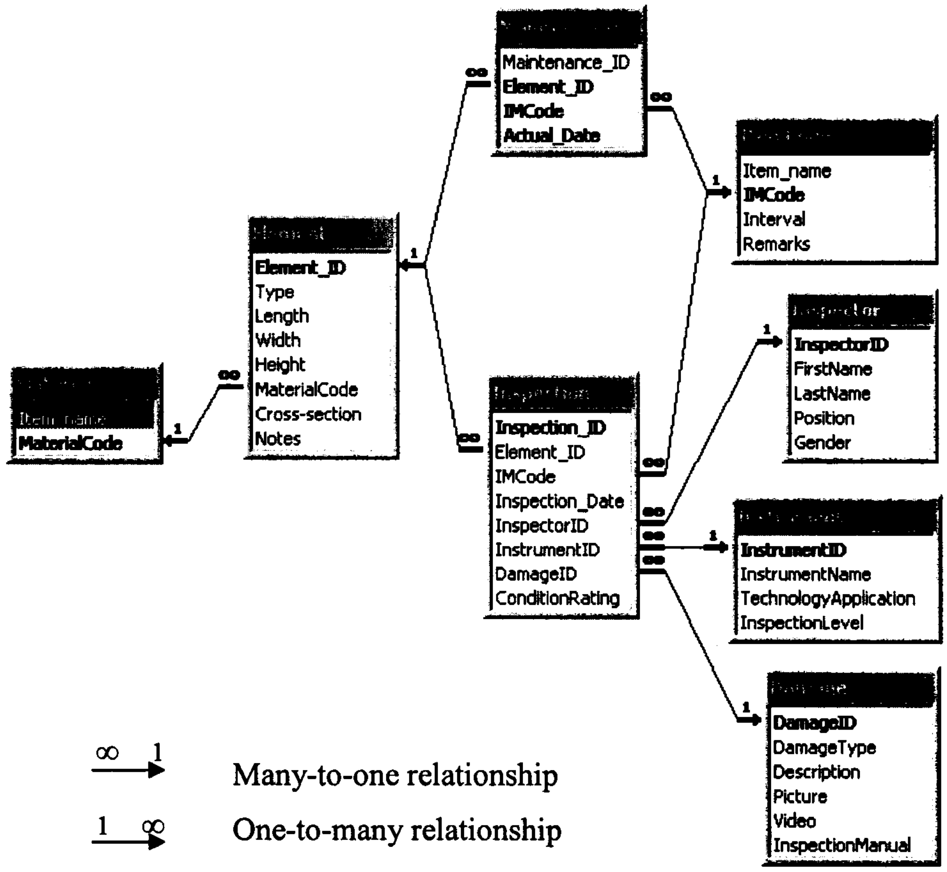


Figure 3.9 Simplified example of a relational database for building FMS

Figure 3.9 shows a simplified example of a relational database for building FMS. Each box in the figure represents a table in the building database which contains some attributes related to this particular table. For example, *Element* table contains attributes such as *Element_ID*, *Type*, *Material*, etc. This table can be linked with other tables in the database by sharing one key attribute. For instance, this table will have one-to-many relationships (1-∞) with other tables such as *Inspection* and *Maintenance* tables by sharing the *Element_ID* attribute and many-to-one relationships (∞-1) with tables such as *Material* table by sharing the *MaterialCode* attribute.

3.3.5 Linking the virtual model to the database

In addition to the geometry, each building (or building element) in the VR model is linked through a unique identification number (UID) to a database where the attributes related to that building (or building element) are stored. This database contains data of cost, activities' scheduling and other process information such as maintenance and inspection activities. The IFC, as an interoperability standard, is capable of exporting/importing data to/from engineering applications (Seren and Karstila, 2001). For mobile-based applications, users can collect data onsite using tracking devices to keep track of the user location and allow collecting spatially located field data. Then, the collected data can be related to the corresponding building element in the virtual model and saved in the database as shown in Figure 3.4 (Phase C). In addition to the data of the building model, an FMS requires information about damages, inspectors and instruments used to perform FM tasks.

3.4 SHARING BUILDING INFORMATION

Sharing information in FM industry is essential and needs integrating building lifecycle applications. Integration between building lifecycle stakeholders is considered essential and standardized practices and methods of storing, viewing and sharing building information are fundamental to work efficiently. The long list of CAD systems' users which includes architects, cost estimators, engineers, manufacturers, constructors, and facility managers is just an indication of the significant impact of these systems on the industry. During a facility lifecycle, architects start sharing CAD data with cost estimators to produce a detailed estimate for constructing and maintaining the building facilities. This data sharing requires integrating a CAD system (e.g., AutoCAD) with a cost estimation application (e.g., Timberline Office).

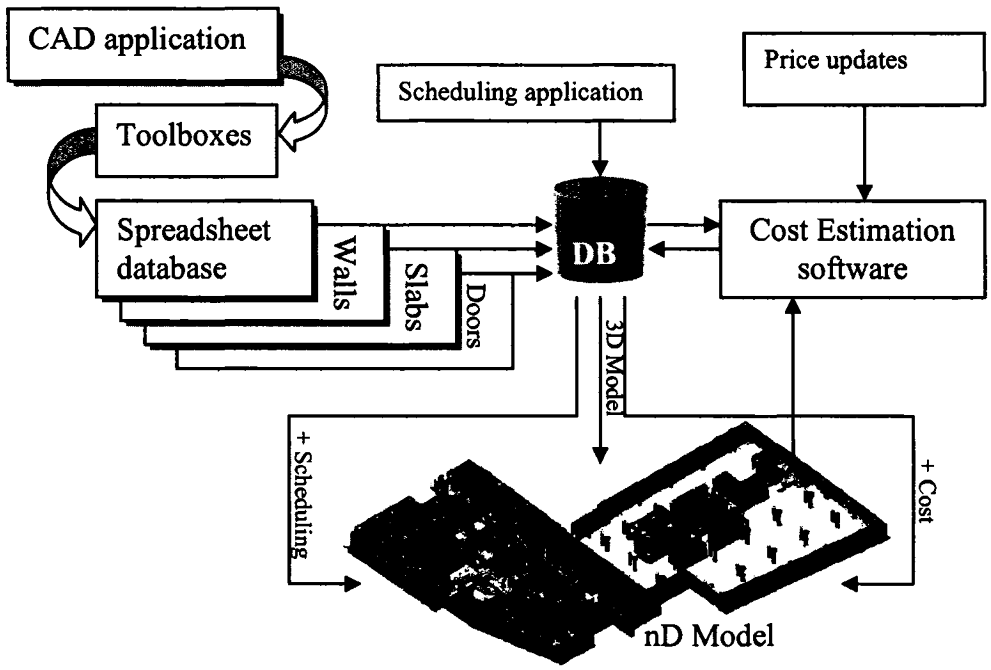


Figure 3.10 Programming toolboxes approach

Figure 3.10 shows a programming toolboxes approach for sharing data of building model, cost and scheduling information. The geometric CAD data can be read and saved in a database using ActiveX Automation. To make these ideas clear, an experimental work is discussed in Chapter 5.

3.5 SUMMARY AND CONCLUSIONS

In this chapter, several innovative and practical methods have been proposed for data integration for creating large-scale virtual models. These methods are summarized as the following: (1) A new data integration method for creating the VE by synthesizing information from GIS maps, CAD models, images of building facades, and databases of the cost, scheduling, and other data generated during the lifecycle of buildings. Furthermore, the creation of these models with different level of details was presented for both building exterior and building interior; (2) A new method for extracting the building model data from an IFC file. This standard-based information is stored in a relational database that contains the large-scale facility model information and the extended building lifecycle information; (3) Visualizing the facilities with different LoDs is proposed to identify the proper format for representing these models within the FMS; and (4) Sharing the facilities data between different applications using tools integration approaches.

CHAPTER 4 MODEL-BASED NAVIGATION AND INTERACTION

4.1 INTRODUCTION

Location-based applications in FM require supporting mobility to perform field tasks. In building facilities inspection scenarios, the inspector has to move most of the time in order to perform her/his task. Therefore, identifying the exact location of the inspector with respect to the inspected building or structure elements can greatly facilitate the data collection. This can be achieved by automatically identifying the elements and specifying the locations of defects on these elements. Approximate location information can still help in focusing the scope of data collection by identifying the building elements that are of interest. Current techniques of capturing location information using paper or digital maps, pictures and drawings can lead to ambiguity and errors in interpreting the collected data. Using LBC combined with Geographic Information Systems (GISs), 3D models of buildings and their elements can be located using suitable tracking devices. In addition, defects on specific elements can be recorded more efficiently and accurately.

In this chapter, location-based model interaction is discussed. The integrated inspection system based on this approach guides and navigates the system user (FM inspector) to the locations he/she needs to inspect, displays information about the historical inspection information as augmentations of the 3D model, and identifies the inspection sequence and the defect locations with navigation guidance. The design of the MBFMS's user interface, which allows such location-based interaction with the virtual facilities, is discussed. Furthermore, the new approach allows the inspector to add inspection information, including the location of defects, simply by picking a building element in the virtual model at the location where the defect has been found. Then, by selecting the type and the severity level of the defect using the inspection GUI, the user can add and modify inspection information.

The proposed approach for MBFMS requires re-engineering the inspection process and using the location of the inspector to facilitate inspection tasks. The features of the proposed approach are discussed in the following sections.

4.2 INSPECTION PROCESS THROUGH LOCATION-BASED TECHNOLOGY

A building facility inspection is a multi-task process that is usually preformed at three stages: (1) inspection management, (2) inspection and condition assessment, and (3) reporting (Figure 4.1). In the first stage, the inspection plan, which consists of gathering information about elements locations, schedule, inspection staff, and equipment, is prepared based on inventory data before starting the field inspection. In the second stage, the field inspection and condition assessment occurs following the sequence of validating targeted building elements, visual inspection, and condition evaluation. In the third stage, all field inspection data are consolidated with relevant CAD-based drawings to generate a final inspection report. On the other hand, the conventional method for building inspection relies on paper-based records that have to be marked on site and then the inspector returns to the office to make semi-manual reports using a word processor or/and spreadsheet application. The disadvantages of using this conventional method for data collection are mainly the inaccuracy of data input and low efficiency, which may cause re-work of the inspection tasks.

To overcome these disadvantages, location-based computing technology can assist MBFMS in automating the process and can eventually become an integral part of the entire inspection process to improve building inspection (Figure 4.1).

In Figure 4.1, the inspection management preparations are done with the help of computers (stage 1). At stage 2, on-site inspection and condition assessment are processed based on LBC technologies. At stage 3, the inspection results can be automatically saved into a database and FM reports can be generated.

The potential advantages of using mobile computing are: (1) facilitating acquisition, transfer, and querying of FM data, (2) improving the efficiency and accuracy during building inspection by considering location information based on integrating GIS, Global Positioning System (GPS) and 3D models to facilitate updating inspection data, and (3) supporting collaboration between inspectors in large-scale work fields.

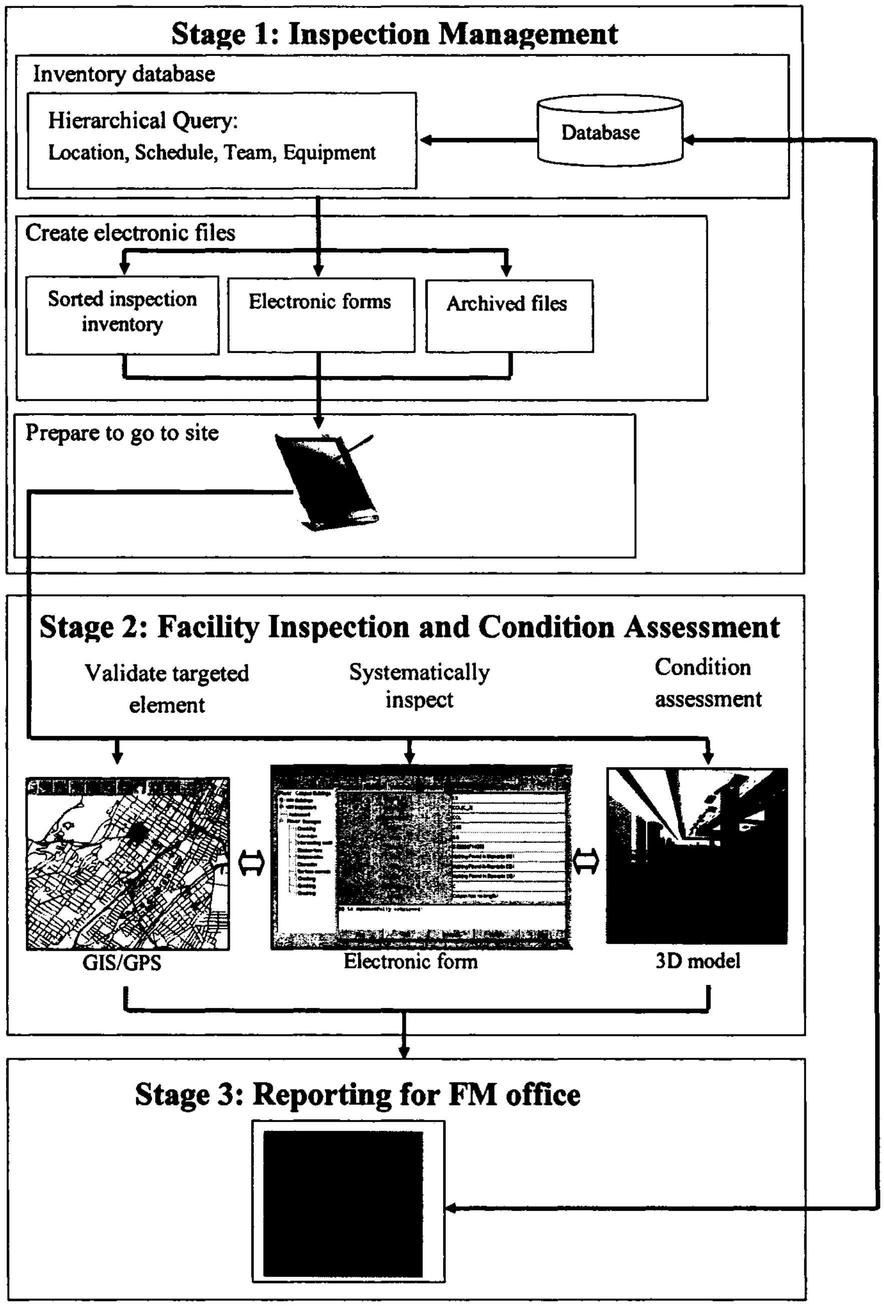


Figure 4.1 Visual FM inspection re-engineering process

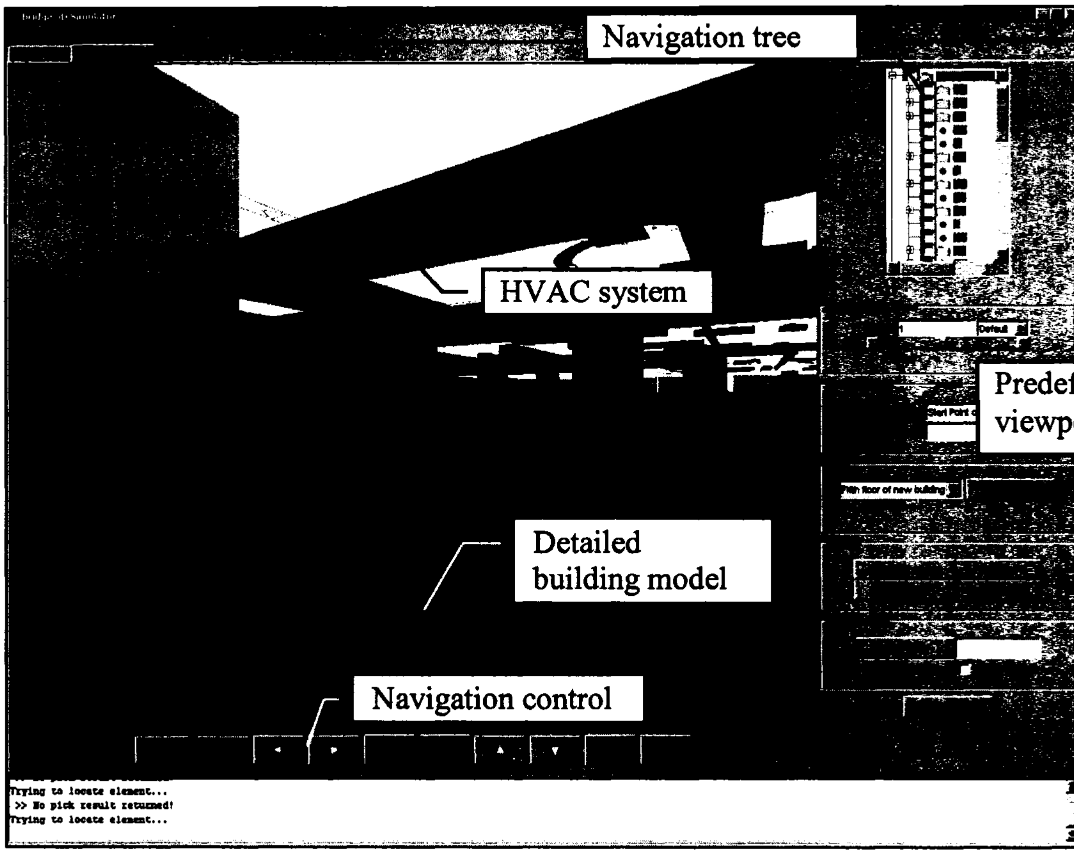
4.3 MAIN USER INTERFACE DESIGN

In order to provide easy-to-use FM applications, developers must consider the human-computer interaction patterns that follow the logical task performance for each domain. For example, a FM inspector uses protocols for conditions assessment depending on the function and the structure type of the facility (IRCC, 1993). These inspection protocols are used to develop the sequence that the system user will follow while performing his/her inspection tasks. To achieve this task, a well-designed GUI is designed to satisfy the requirements of the location-based and model-based data collection.

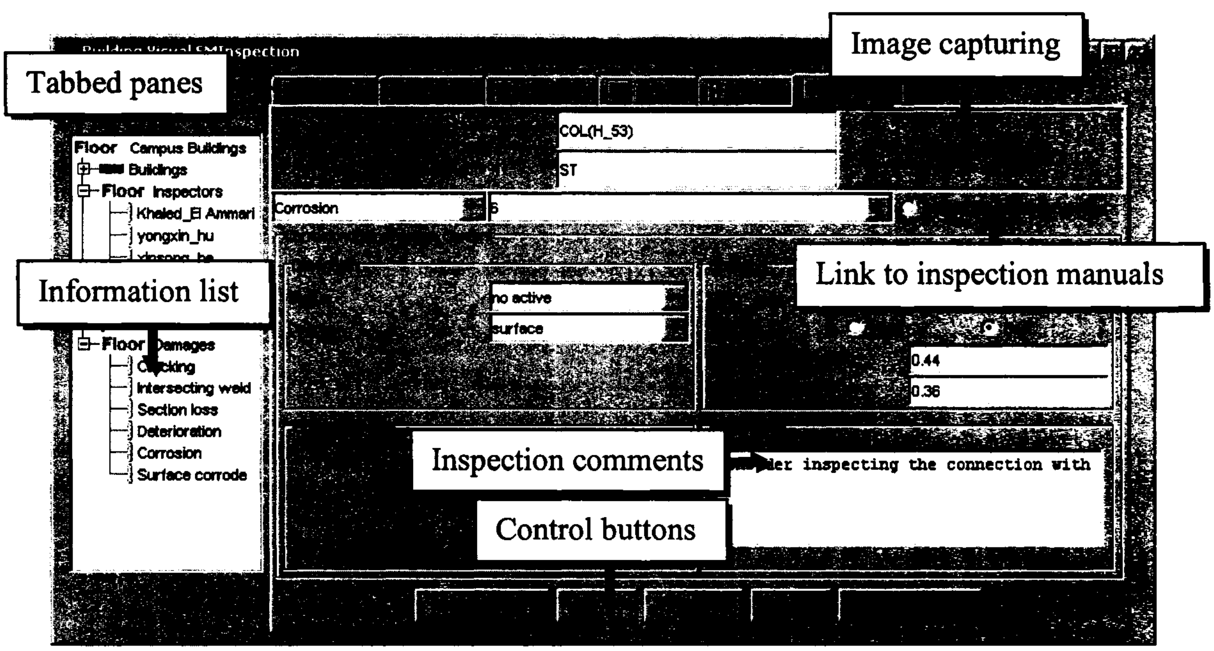
Navigation and model interaction functions are essential in the development of the GUI, especially when considering a mobile situation where the facility inspector uses a small-screen tablet PC. The main part of the GUI should be used for the virtual environment, where the user can navigate and interact with the facility's virtual model. Another navigation method can be used to select directly a building or an element from a hierarchy tree list. This selection instantly relocates the inspector based on his/her current location and the view will be updated. This interactive list, with the hierarchy relationships, allows the system users to effectively find buildings, structure elements and other objects in the large-scale VE. This tree list is automatically generated from the buildings' database following the logical hierarchy relationships of these buildings and their structural elements.

The root of the tree is found and the root node is created by querying the buildings database. After that, other queries are applied to find elements nodes based on the data stored in the elements database table. Each node in the tree has a check box for showing or not showing a specific element in the VE. In addition, different viewpoints are created to provide guidance and relocating the facility inspector to specific locations. Specific viewpoints such as front, side and the top view of the building can be predefined in the system. More advanced techniques can be used to generate a viewpoint when the user picks an element to assign a defect. These viewpoints locations will be saved in the defects' database and displayed as a tree list of defects. As a result, when the user picks a listed defect in the tree list, the viewpoint location of the user will be updated based on the stored defect location. Thereby the user will be able to locate that specific defect in the VE. Figure 4.2 (a) shows the design of the main GUI of MBFMS.

After picking the approximate position of the defect based on visual observation, the inspector can complete the data input and save these data with the help of the defect inspection input pane (Figure 4.2 (b)). In this interface, an inspector can apply inspection procedures by using a number of ordered tabbed panes. The panes are *Inspector, Schedule, Instrument, Process, Element, Damage,* and *Task.* In the first two tabbed panes, some general inspection information needs to be input about the inspector and the scheduled inspection task. The user can find, add, and update the facility inspection data by querying the database. In the *Instrument* pane, a suitable inspection tool can be selected depending on the type of the defect.



(a) Main user interface design



(b) Facility inspection user interface

Figure 4.2 Model-based interaction user interface of the prototype system

The *Process* pane is used to guide the inspector to find the elements that need to be inspected based on the scheduled routine inspection or the previous defect locations. In the *Element* pane, the inspector can choose the element to inspect according to a customized inspection scheme by picking the element in the VE. In this pane, abstract information about the selected element, such as type, material, and dimensions, will be retrieved from the database and added to the pane's text fields. The *Damage* pane is the main part of the inspection GUI where the inspector can input defects information. The *Task* pane summarizes the previous inspection information for future assessment.

4.4 FRAMEWORK FOR A GENERIC GUI

The GUI design needs to improve and organize inspection data entry, reduce input errors, and automate accessing information that can support inspectors. For instance, the GUI text fields for locations, dates and other inspectors and building elements' attributes can be automatically retrieved and displayed on the GUI which eventually minimizes the development effort to update the GUI. This GUI is part of the main FM GUI which is generated based on the structure type (e.g., building) or building element type (e.g., a roof). Different elements have different attributes. Therefore, selecting the necessary fields of element attributes is important to help inspectors and save display area on the portable computer (e.g., tablet PC). Figure 4.3 shows the main steps of an auto-generated inspection GUI.

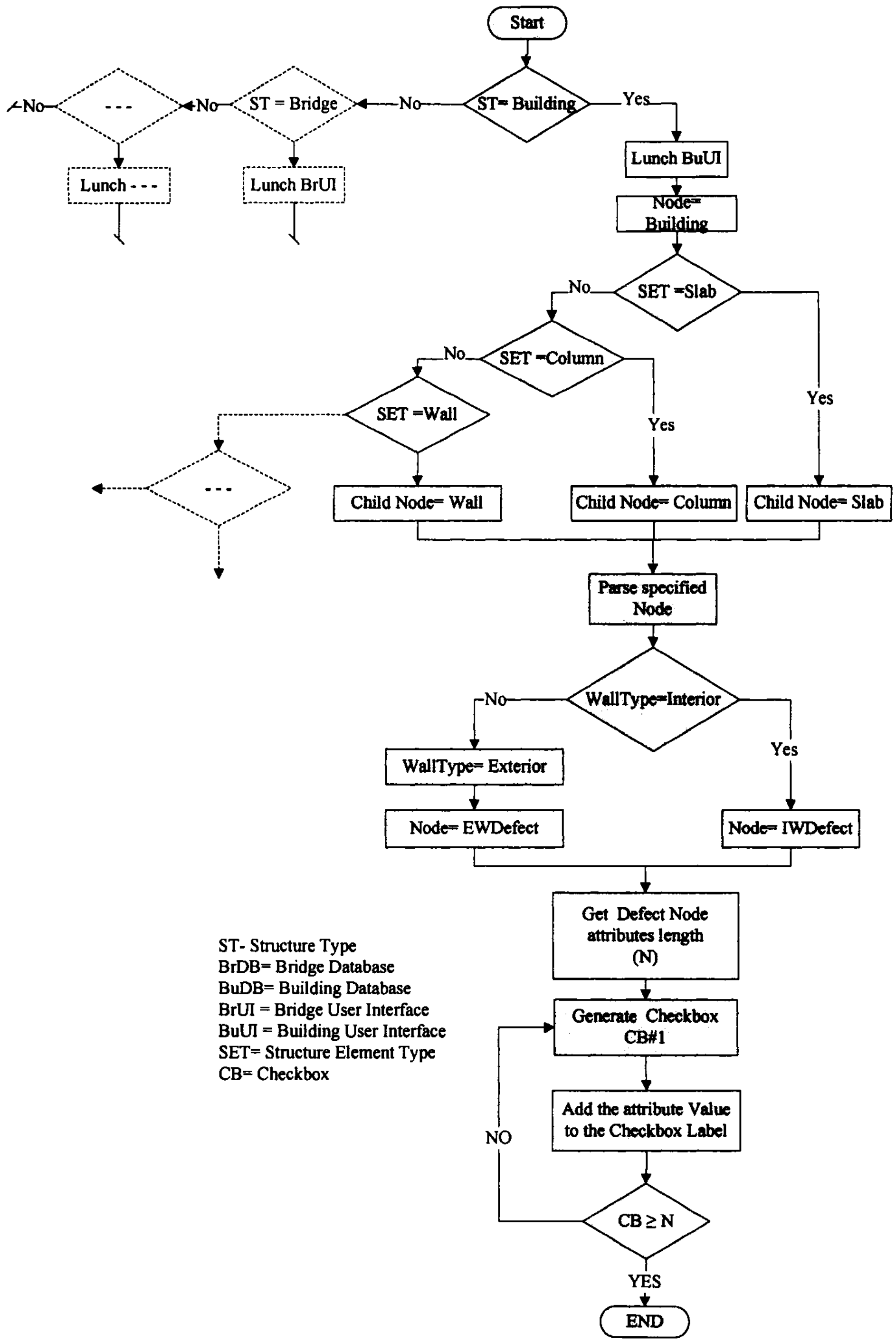
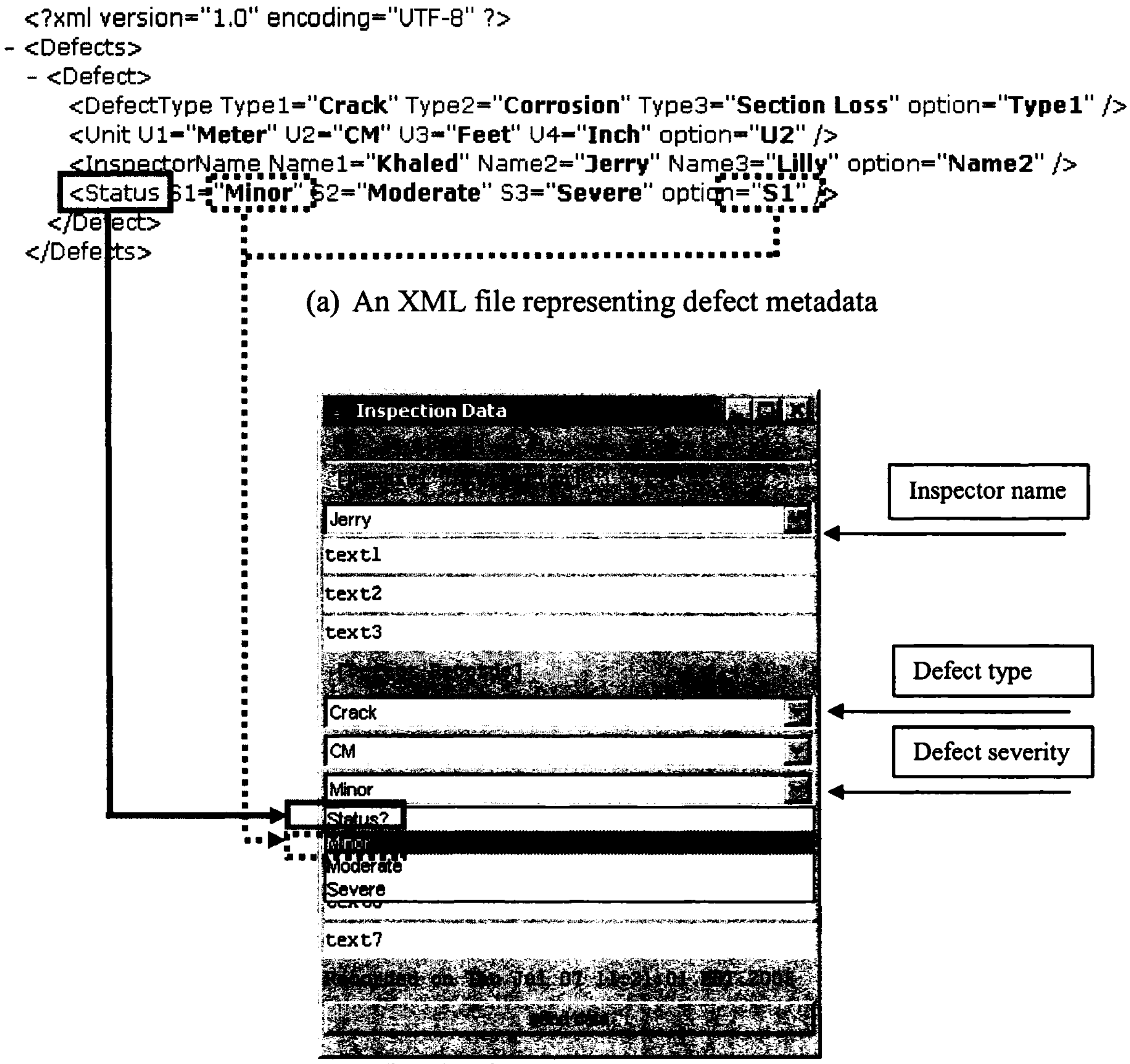


Figure 4.3 Flowchart of the automated GUI generation using XML metadata

Upon selection of an element, the facility type can be identified based on IFC information (e.g., a building). This identification allows realizing which information needs to be retrieved from the database to update the GUI. After identifying the facility type, a suitable XML-based metadata will be used to update the GUI. In a case of a building type facility (ST= Building), the building GUI is updated and lunched. The building elements are described as *Child Nodes* in the XML metadata file. For example, if the structural element is a slab (SET= Slab), slab node attributes (Child Node= Slab) will be retrieved. If not, checking the element type is automatically applied again to identify the relevant element information. Based on the inspection practice and protocols, building elements that are located inside the building (e.g., interior wall) have different common defects from those attached to the envelope elements (e. g., exterior wall). This is because the exterior elements are exposed to environmental conditions. These differences affect the generation of the GUI design. For example, there is no need for the inspector to check a long list of defects that may cover all types of interior and exterior walls, as well as materials such as steel, wood and concrete. This may disturb the inspection process, which will eventually affect productivity of the inspector. Sufficient and correct data input is an essential concept for using this model-based FMS.

The metadata for these GUIs can be presented in XML format which make future updates more possible. The well-designed GUI and auto-generation technique will guarantee the reliability of the data collection. Figure 4.4 shows a simple example of an auto-generated GUI based on an XML file that represents the metadata. This metadata written in XML is read using java XML parser.



(a) An XML file representing defect metadata

(b) Simple FM inspection GUI

Figure 4.4 GUI metadata represented in XML format

The metadata represented in XML allows future updating of the GUI by simply updating the XML file. For example, as shown in Figure 4.4(a), there are three statuses (S1= Minor, S2= Moderate, and S3= Severe) described as attributes of the status (a *Child Node)* which is inherited from a defect record (a *Parent Node*). The option attribute in status node allows the inspector to define the status of the defect (e.g., S1 = Minor). As shown in Figure 4.4 (b), whatever attributes data added to the *Defect* Child node in the

XML file will be listed in the FM inspection GUI as options for the inspector to choose from.

Separating the metadata in an XML file can help in customizing or updating the GUI anytime in the future depending on the facility type (e.g., Building, Bridge, etc.). Furthermore, this technique helps in translating the GUI to different languages (e.g., Arabic, French, Spanish, etc.) by translating the XML file.

4.5 DEFECTS MARKING ON 3D MODEL

The interaction with the 3D model is essentially facilitated by selecting an element of the 3D building model using a picking device such as electronic stylus or a mouse. This allows the facility inspector to interact with the VE by picking an element. The picking behavior, developed in the MBFMS, allows data retrieving for the selected element and instantaneously displays these data on the inspection GUI. In addition, inspection data can be added to the element inspection attributes and stored in the model database. Information about defect location, inspection date, etc. is automatically added by the system. This eliminates the need of extra effort and time to draw and write notes about this defect information, which consequently minimizes cost and improves productivity.

Picking is the process of selecting shapes in the 3D VE using the 2D coordinates of the picking device (e.g. electronic stylus). The pick object could be a ray, segment, cone, or cylinder. The pick object extends from the viewpoint location, through the picking device location and into the VE. When a pick is requested, pickable objects that intersect with the pick object (e.g., pick ray) are computed. The pick returns a list of objects, from which the nearest object can be computed. Figure 4.5 shows an example of the picking behavior.

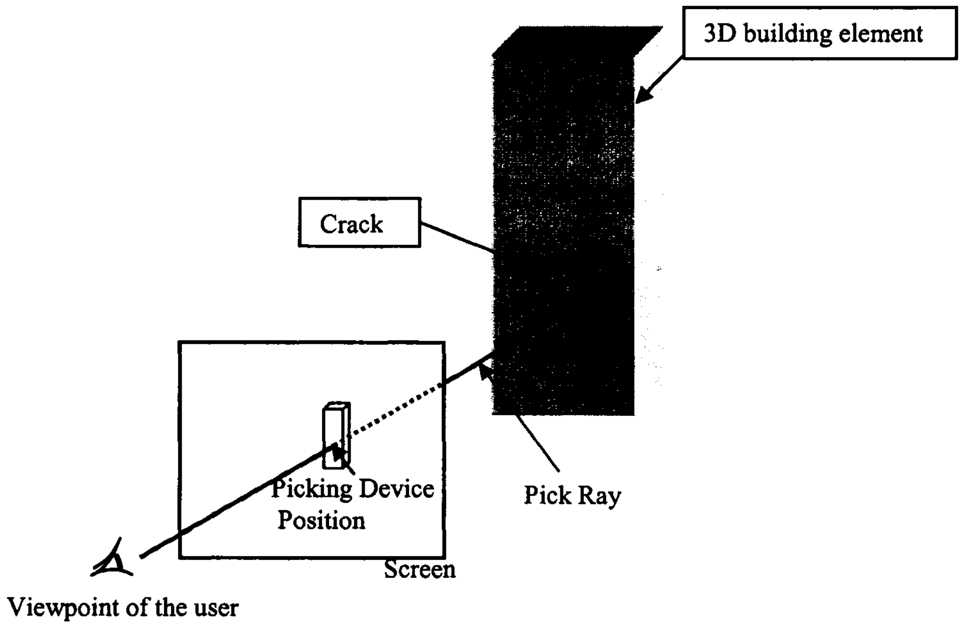


Figure 4.5 Example of picking a 3D virtual building element

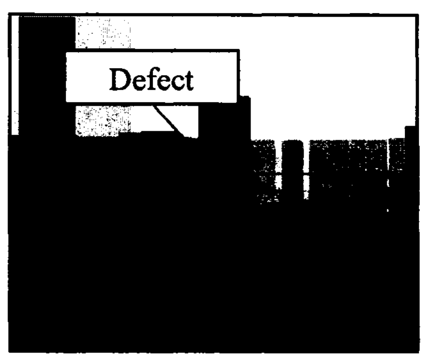


Figure 4.6 Example of adding defect on a column

An application that is facilitated by picking in the proposed framework is adding an object to the VE, such as adding a defect to a building element (e.g., a column). In the case of structural element inspection, inspector can mark defects by picking a virtual element in the EV. These defects are represented by 3D objects (e.g., a sphere), on the surface of the inspected element (Figure 4.6). The shape and the color of the defect object represent the type and the severity of the defect, respectively.

Another application of the picking is adding a workspace around building element (e.g., column). Upon the selection of a column, a workspace around this column is created (Figure 4.7) and abstract relevant information is displayed such as the UID and the dimensions of the column. This function is useful for workspace analysis. The purpose of workspace analysis in FM is to investigate the workspace representation and conflict detection in FM inspection and maintenance activities. Defining these workspaces for FM activities is essential for satisfying the workers' safety requirements as well as planning these activities in a 3D environment.

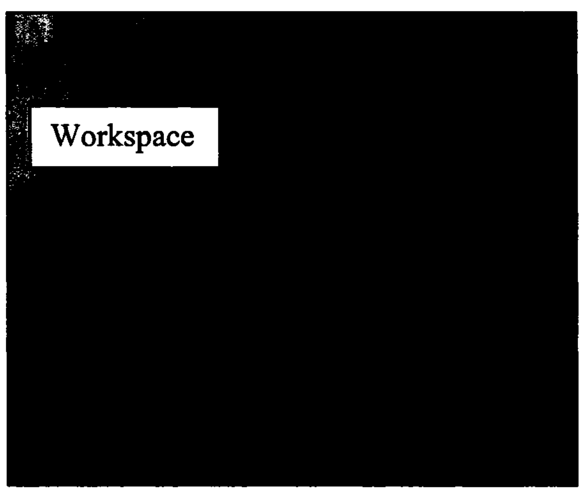


Figure 4.7 Example of adding a workspace around a column

Figure 4.8 shows a flowchart of picking and highlighting a 3D element. Each object in the VE (e.g. buildings or building elements) has an ID, which is linked with the data stored in the database.

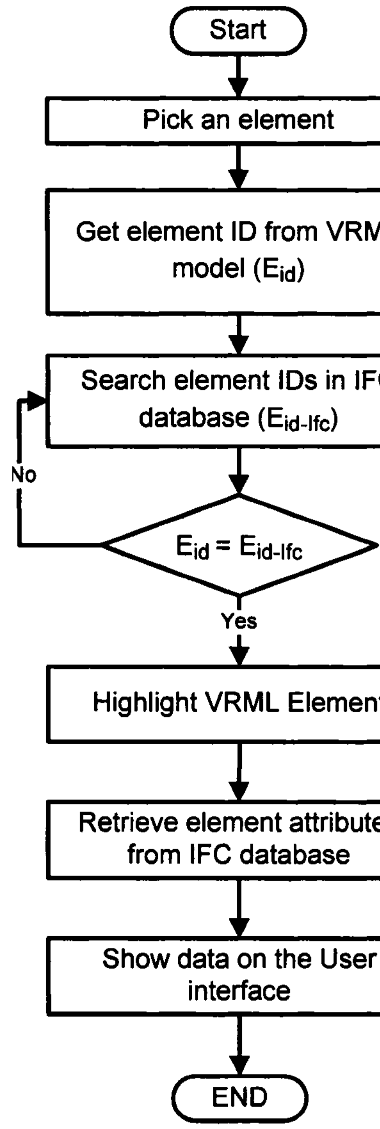


Figure 4.8 Flowchart of picking and highlighting a 3D element

VRML Model

```
DEF COL(H_01) Transform {
  translation 18.78 1.638
33.33
  children [
          geometry DEF
COL(H_01)-FACES
IndexedFaceSet {
        ccw TRUE
        solid FALSE
        coord DEF
COL(H_01)-COORD Coordinate
{ point [
          0.3557 -1.337 -
0.0218, 0.3557 1.374 -
0.0218, 0.353 -1.337 -
0.05254,
          0.353 1.374 -
0.05254, 0.3477 -1.337 -
0.08293, 0.3477 1.374 -
0.08293,
          .
          .
                -0.1686
1.374 0.4632]
          .
          .
```

IfcXML Model

```
<IfcColumn id="i70486">
 <GlobalId>2aRizw0xv7UAX$4pVqnSAy</GlobalId>
 <Location>
  <IfcCartesianPoint id="i70525">
    <Coordinates ex:cType="list">
<IfcLengthMeasure
ex:pos="0">350</IfcLengthMeasure>
<IfcLengthMeasure ex:pos="1">-
846530.095</IfcLengthMeasure>
<IfcLengthMeasure
ex:pos="2">7650</IfcLengthMeasure>
    </Coordinates>
       </Position>
         <Radius>80</Radius>
 <IfcShapeRepresentation ex:pos="1"
id="i70518">
      <XDim>17951.6</XDim>
      <YDim>17951.6</YDim>
      <ZDim>68850</ZDim>

<IfcDescriptiveMeasure>[redacted]</IfcDescrip
tiveMeasure>
```

Figure 4.9 VRML and IfcXML data representations

As shown in Figure 4.9, the unique ID for a building element (COL(H_01)) in VRML model is the same as the one in the IFC-based data stored in the database. Upon selection, the building element will be highlighted and a query is activated to retrieve the element information.

4.5 SUMMARY AND CONCLUSIONS

This chapter includes discussions about the following innovative approaches and methods: (1) A new model-based computing approach is proposed to support the MBFMS requirements for data collection activities during building inspection. This approach was designed based on re-engineering data collection of building inspection; (2) A well-designed GUI for MBFMS that can be used in mobile situations to facilitate on-site model-based data collection was developed; and (3) The proposed interaction methods with the VR model of the facilities allow the inspector to read, modify, and update the non-graphical information about the facilities' conditions. These interaction techniques for navigation, picking, and marking defects on the 3D building model were investigated.

CHAPTER 5 IMPLEMENTATION AND CASE STUDIES

5.1 INTRODUCTION

Based on the discussion in Chapters 3 and 4, a prototype system is developed following the framework architecture of data integration and visualization to investigate the feasibility and usability of using international standards for data exchange to support FMS. In order to proof the efficiency of the proposed framework, a case study of Concordia University's downtown campus in Montreal is used to demonstrate the prototype system. In addition, two scenarios for using the developed system are discussed, using VR and AR for facilities inspection tasks.

5.2 DEVELOPMENT TOOLS

Java is a general purpose programming language with a number of features that make the language suitable for use on the internet and extend the benefit of data sharing world wide. Java, as a development environment, is chosen because of its features such as object orientation, safety, simplicity, and breadth of the standard library (Horstmann, 2004). Object orientation enables systems developers to spend more time on the design of their system and less time debugging and coding. In addition, the standard library of java consists of many important components for development such as graphics, GUI development, database access, multithreading, and network programming. The Java 3D API allows describing the 3D scene of the VE using graphical objects and defining these objects using transforms, scaling, and appearances that include materials, lights, etc.

Compared with Open GL, the Java 3D code is more readable, maintainable, reusable, and easier to write (Selman, 2002).

MapObjects-Java Edition is available to build custom applications that incorporate GIS mapping capabilities (MapObjects-Java, 2006). MapObjects-Java Edition allows developing applications that perform a variety of GIS-based displays, queries, and data retrieval (ESRI, 2005).

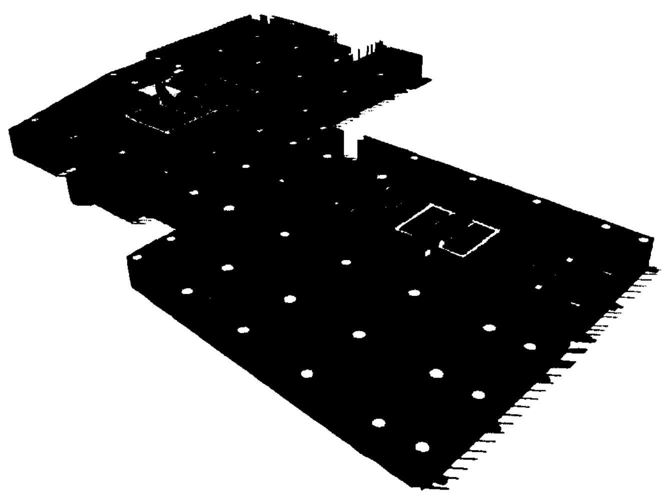
The open standard used for virtual reality models is VRML. VRML is the most popular interoperability standard for describing interactive 3D objects and virtual worlds delivered across the internet (Nadeau and Moreland, 1996). Furthermore, there are many VRML-compatible CAD applications (e.g., ArchiCAD 9.0) that support also IFC, which makes VRML better choice for modeling the building facilities.

The database of the facilities, cost, and FM task scheduling is designed with Microsoft Access to represent the information of all the facilities such as structural elements, spaces and subsystems such as HVAC information. Java Database Connectivity (JDBC) is used to access information stored in databases. IfcXML is the main data format for representing the IFC data of the building facilities in the developed system. XML parser is used to pull out the necessary data for the MBFMS. More details about software requirements and installation guide of the prototype system are included in Appendix B.

5.3 BACKGROUND OF THE CASE STUDY

Concordia downtown campus (Sir George Williams campus - SGW) in Montreal is chosen as the subject of the case study. The campus is located in Montreal downtown area and it covers a number of buildings and areas, which makes the case study ideal for representing a large-scale facility for the prototype system. Recently, two new buildings are added to the plan of the SGW campus: (1) the first building consists of 17-storey block for Engineering, Computer Science and 12-storey block for Visual Arts school; it is called the *Integrated Engineering, Computer Science and Visual Arts Complex* (Figure 5.1), (2) the second building is also multi-storey building for the new John Molson School of Business. In addition, the campus is largely expanded by the recent acquisition of the Grey Nuns' Mother House. This growth results in a new vision for the SGW campus and the surrounding area, which is called *Cartier Concordia* (Concordia, 2006).

(a) Multi-story building



(b) Virtual detailed model for one floor

Figure 5.1 Multi-story building facilities

5.4 IMPLEMENTING THE FRAMEWORK

5.4.1 Prototype system development

A prototype system has been developed to demonstrate the feasibility of the proposed methodology. In order to allow collaboration and data exchange that can be shared on the Internet, Java programming language is used to develop the system. Java is a versatile language that enables developers to construct web applications and facilitate the client-server web services. As discussed in Section 5.2, Java 3D is used to implement the 3D graphics of the system (Walesh and Gehinger, 2001). Java 3D is an 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 MBFMS requirements and to realize the following major functions: (1) representing the multidimensional model of buildings facilities with different Level of details (LoDs), (2) designing a user-friendly interface with user setting control, and (3) developing a comprehensive databases for the facilities that include construction, inspection, and maintenance records based on a standard data model.

The developed system has the following main GUI components (Barrilleaux, 2001): visualization and feedback, control, access, navigation, manipulation, and collaboration. The system combines these components with the virtual model and IFC-based database for the facilities lifecycle information. In addition, other components are used to provide tracking and navigation capabilities such as video tracking and GIS maps as shown in Figure 5.2.

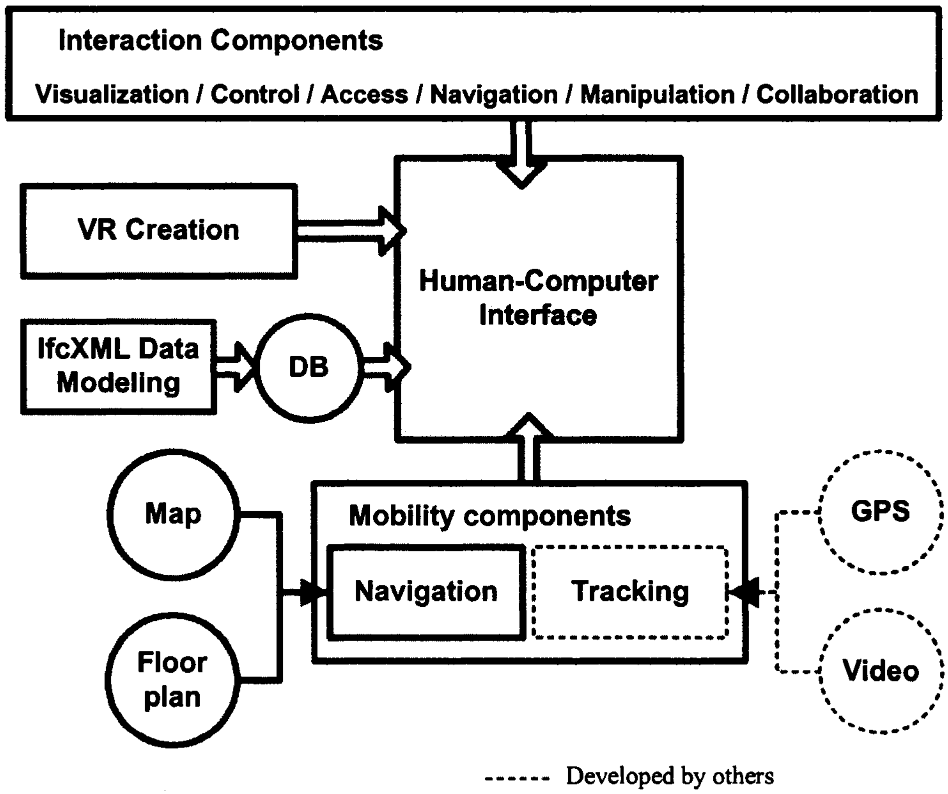


Figure 5.2 The main components of MBFMS architecture

The following interaction patterns for each component are typical examples that have been identified based on common tasks that FM inspectors usually perform and the type of information they collect.

As visualization and feedback, the system can display: (1) Graphical details to the facility inspector about structural details retrieved from previous inspection reports. This can happen in a proactive way based on spatial events. For example, once a defected element is within an inspector field-of-view, the system displays the defect on that element discovered during previous inspections. This will help focus the inspector's attention on specific locations. The user of the system can control the LoDs of representing objects depending on his or her needs; and (2) Non-graphical information and instructions: The user interface can provide links to documents related to the project, such as reports, regulations and specifications. In addition, The system allows for displaying context sensitive instructions on the steps involved in a specific task, such as instructions about the method of checking new cracks and measuring crack size and crack propagation.

Within a specific field task, the system can guide the inspector by providing him/her with navigation information and focusing his or her attention on the next element to be inspected. Once an element is found in the VE, the inspector can add defect information by directly editing its color and tag it with a colored 3D object (e.g., sphere) to represent location, type and the severity of the defect. The system facilitates wireless communications among a team of facilities inspectors, geographically separated in the project site, by establishing a common spatial reference about the site of the project. In some cases, the facility inspector may collaborate and consult with a senior engineer stationed at the FM office who monitors the same scene generated by the mobile unit in the field.

5.4.2 Object-relational data model

The data model used in the framework is essentially an object-relational data model. This is realized by combining the relational data model with object-oriented development tools. It is still common to use relational database management systems. On the other hand, object-oriented programming tools are commonly used in software development and can significantly improve the quality of the software due to their flexible data structure. Therefore, a combination of the two approaches is the object-relational approach for database development, which can arrange and connect information in the relational database with the data structure of the building elements as explained in object-oriented programs (Object-Relational Mapping, 2006). For instance, the data stored in the database about the buildings and their structural elements are read automatically and a hierarchy tree that describes buildings and elements relations is created based on the structure hierarchy relationships. Figures 5.3 shows an example of a hierarchy tree representing Concordia campus buildings. By querying the database, the root of the tree is found and the root node is created. Then, queries are applied recursively to get other nodes based on the stored data in the database table.

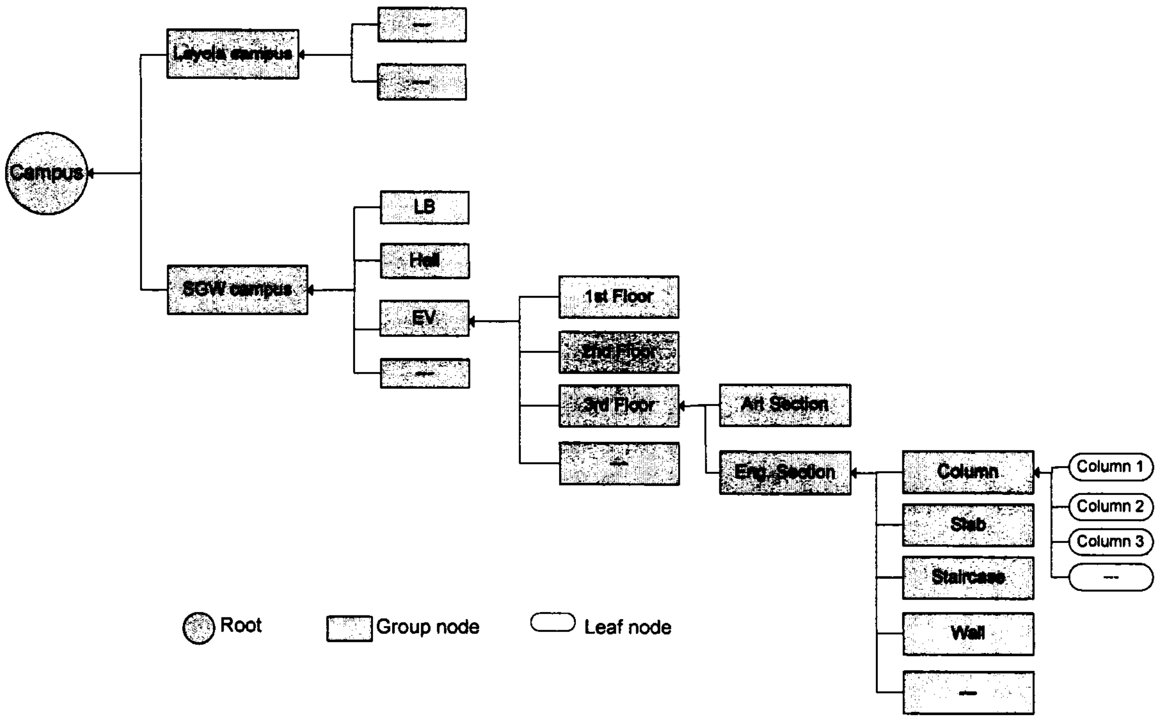


Figure 5.3 Example of the object tree

5.4.3 Building the VR model of the campus

Based on the data integration for creating the virtual model discussed in Section 3.3, Large-scale 3D virtual model of Concordia downtown campus is created using various data: (1) 2D CAD drawings of the buildings obtained from the Facilities Management Department of the University; (2) A digital map of the city of Montreal obtained from the municipality of Montreal; (3) A Digital Elevation Model (DEM) of the city obtained from USGS website; (4) Orthogonal digital images of the buildings facades; and (5) A library for small VRML objects to be used in the 3D VE (e.g., street furniture).

The 2D CAD drawings of the campus area are used as outline footprints of the buildings. This polygon-based map is imported as a layer in ArcView (ArcView, 1996) to create the building layer. Other layers are created in ArcView such as the object layer, image layer for texture mapping, and tree objects layer. Attribute tables of the layers are created to input the required information. Many issues were considered such as the coordinate system of these CAD drawings and apply a proper importing setting. In addition, drawing units are set to meters for all CAD and GIS data.

The collected images of the buildings facades are processed as in the following: (1) Trim the images to fit the building facade in the virtual model; and (2) Edit the parts that need to be transparent using proper tools (e.g., Adobe Photoshop). The images and VRML files of street furniture are linked to the image and object layers, respectively. The GIS layers, images and 3D objects are translated into VRML format. An application developed in Visual Basic is used for VRML translation. The application uses a GIS library to extrude the polygons in the GIS shape files and create the 3D model in VRML format (Figure 5.4).

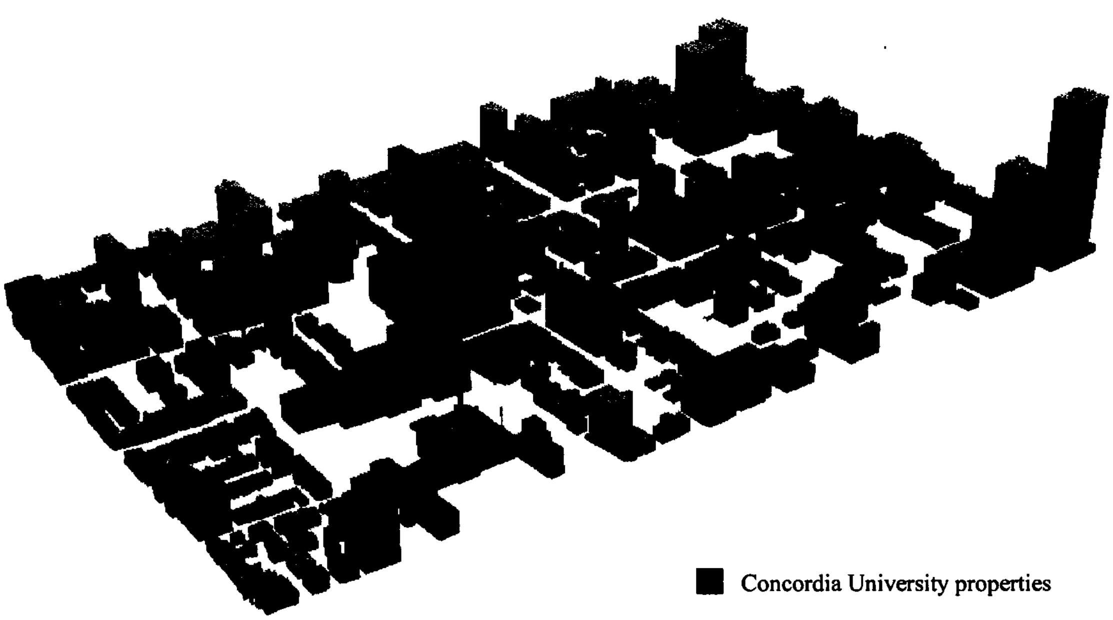


Figure 5.4 Virtual model of the university area

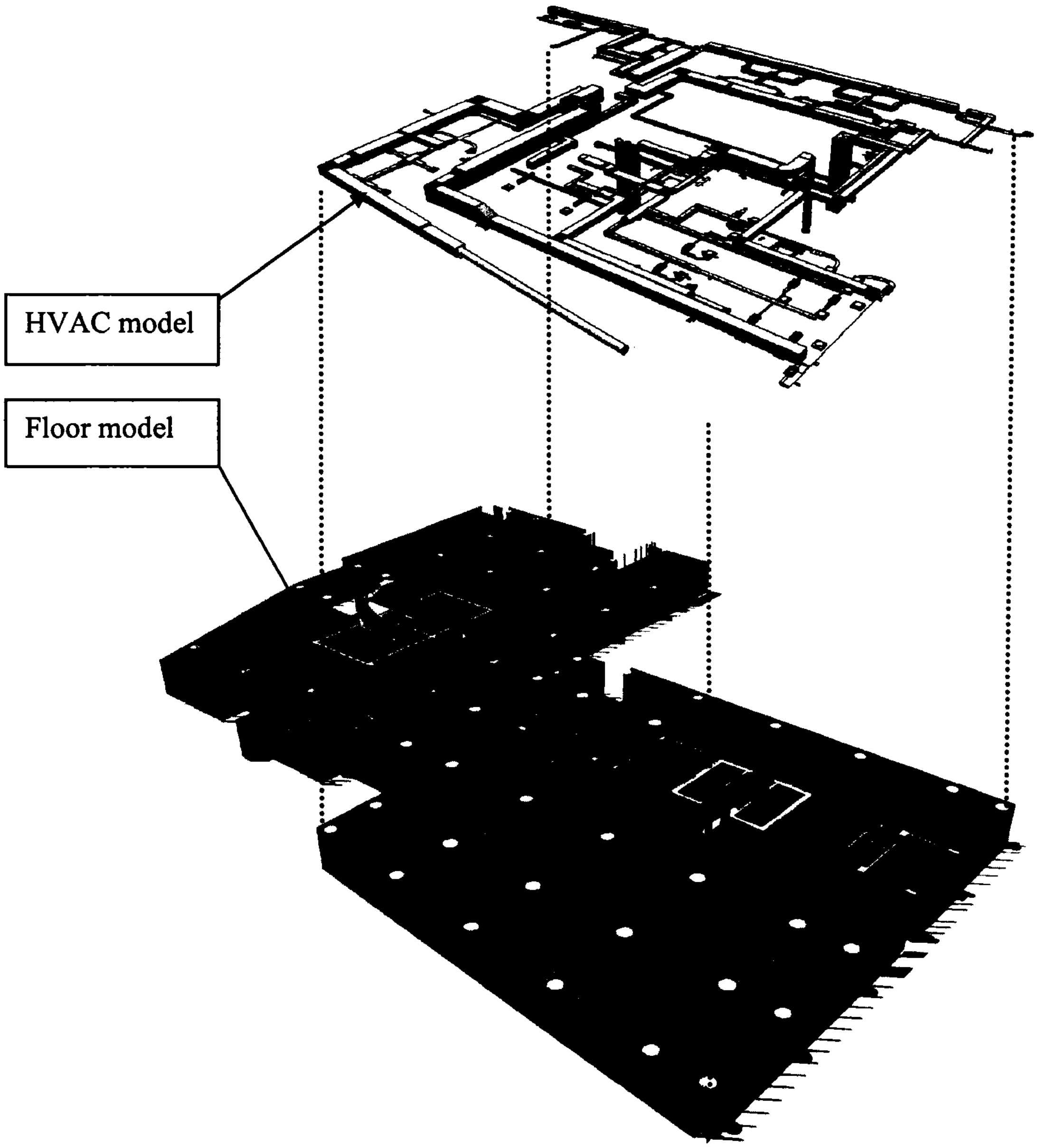


Figure 5.5 Detailed model of one floor

The virtual model of one floor consists of the structural elements such as slabs, columns and walls as shown is Figure 5.5. In addition, it includes spaces' partitions and the glassy envelop. The model has been created by a VRML and IFC-compatible CAD application

(ArchiCAD9.0). Each element in this model has a unique ID. These IDs are also used for interacting with the virtual model.

More details can be added to the virtual facility about building subsystems such as the mechanical, electrical, plumbing and drainage systems. In this study, the 3D virtual model of the HVAC system has been added to the VR facility model as an example of building subsystems.

The digital map and the DEM data of Montreal are used to generate 2D and 3D maps using the Modified Transverse Mercator (MTM) projection. The user of the system (FM inspector) can access any required information by picking and interacting with the 3D model through a customized GUI. The virtual model can link to all necessary specifications, drawings, procedures, and inspection and maintenance records. At this stage, the VE includes the 3D model of the structural elements and the HVAC system of one floor of a building. Future work will consider adding other information about other subsystems such as mechanical/electrical equipment, emergency evacuation plans, etc. This integrative approach can help the user to easily understand the entire facilities configuration and access the task-related information.

5.4.4 Visualization of the virtual environment

Java 3D allows creating virtual universes using *scene graphs,* which are created by assembling objects that define geometry, orientation, location and appearance that includes lights and materials. Java 3D scene graph is constructed from node objects using *BranchGroups* to form a tree structure based on parent-child relationships (Figure 5.6).

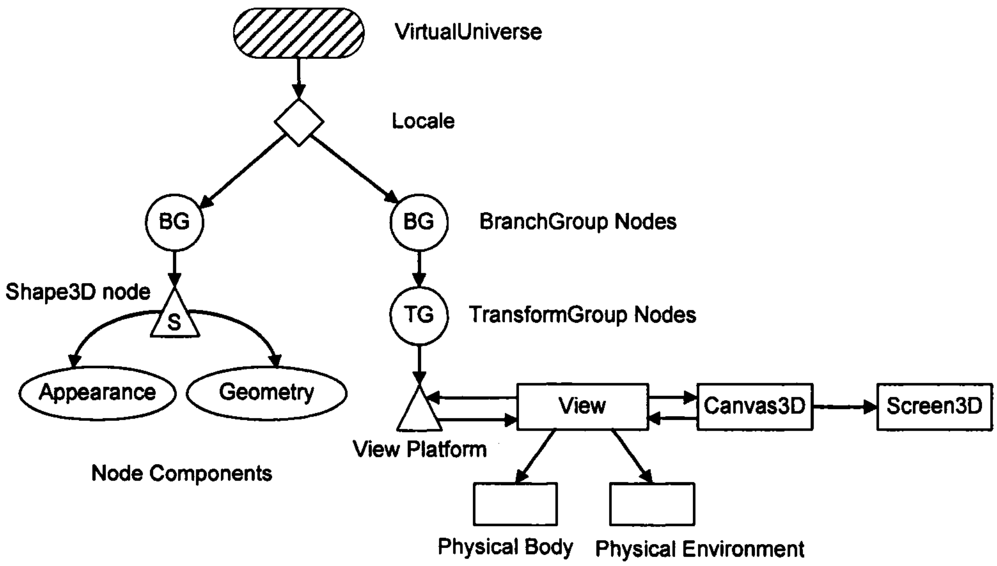


Figure 5.6 Scene graph organization (Walesh and Gehringer, 2001)

The *TransformGroup* objects are constructed using *Transform3D* objects, which represent transformations of 3D geometries such as translations and rotations (Walesh and Gehringer, 2001). To visualize the VE, CAD-based drawings and VRML models have been loaded. The visualization starts by importing all VRML files into the Java 3D scene graph of the prototype system using VRML 97 API (J3d.org, 2006).

The names of the buildings elements are read, using element IDs, from the VRML file while loading and registered in a HashTable. Each building element is assigned a unique ID equal to the name of the building element in the database and registered in the HashTable, which has a key and an associated element ID. The HashTable allows instant retrieving of the existing element based on a key value. The VRML files are spatially located in the 3D VE due to the GIS data in these files. In addition to the campus model, which is representing an abstract model for these building facilities, a 3D CAD model of one floor of a building (a topical floor of the EV building) is created and converted to VRML format. To ensure the quality and the reliability of this model, the creation of this model is done by following some quality assurance instructions: (1) Checking the original CAD file and identifying common drafting errors; (2) Fixing these errors to facilitate the modeling processes; (3) Creating the 3D model based on correct CAD data; and (4) Using IFC-VRML-compatible CAD application to avoid compatibility problems.

In order to locate the 3D detailed floor model in the 3D VE, coordinates of two points are identified in the model and in the 3D VE. Then, these coordinates are used to calculate the transformation matrix, including scaling, rotation and translation from the local coordinate system of the CAD-based virtual model to the global coordinate system in VE. The HVAC system for one section of a floor is loaded and located in the correct position in the VE using two points for the alignment with the 3D floor plan.

To facilitate the data sharing aspects of the developed system, CAD files in DWG format are loaded using DWGLoader library (MapObjects Java edition, 2003). In addition, DXF files are loaded and visualized in the VE using DXFLoader library (j3d.org, 2006). This library supports loading various CAD files formats (DXF, 3DS and OBJ). The geometry of these drawings can be represented in solid, wireframe or point cloud mode in the scene graph. In addition, more 2D CAD drawings can be loaded such as sewer pipelines, electrical system, etc. Furthermore, roads and buildings names can be shown as 3D text objects in the VE to provide navigation support and user guidance.

5.4.5 IfcXML parser class

As discussed in Chapter 3, the IfcXML model has been created using ArchiCAD 9.0. The hierarchy relationships of the model components can be viewed using an XML viewing tools (e.g., XMLSpy). Figure 5.7 shows an IfcXML file for one floor using XMLSpy 2005.

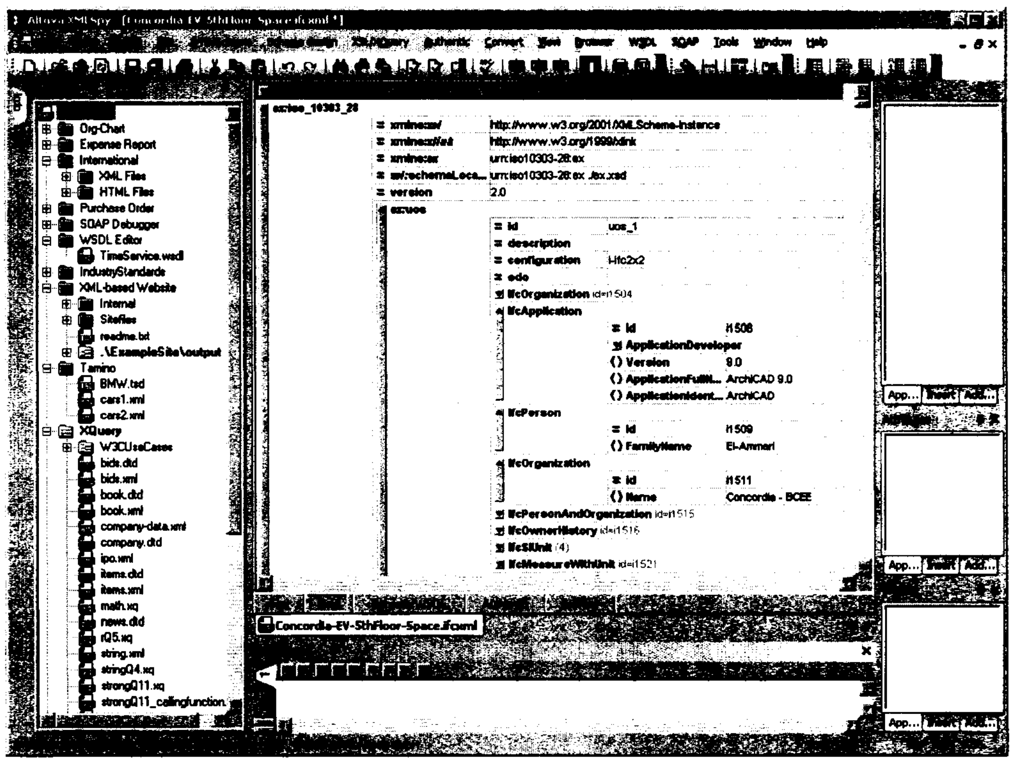


Figure 5.7 Checking IfcXML model of one floor using XMLSpy tool

Reading these IfcXML data is implemented using a package called "IfcXML Parser". This package was developed to allow users to parse and update the model database from any IfcXML2x2 model. This package contains several classes for loading IFC files as well as reading and writing IFC instances in IfcXML representation. Figure 5.8 shows an example of parsing an IfcXML model with the IfcXML parser class.

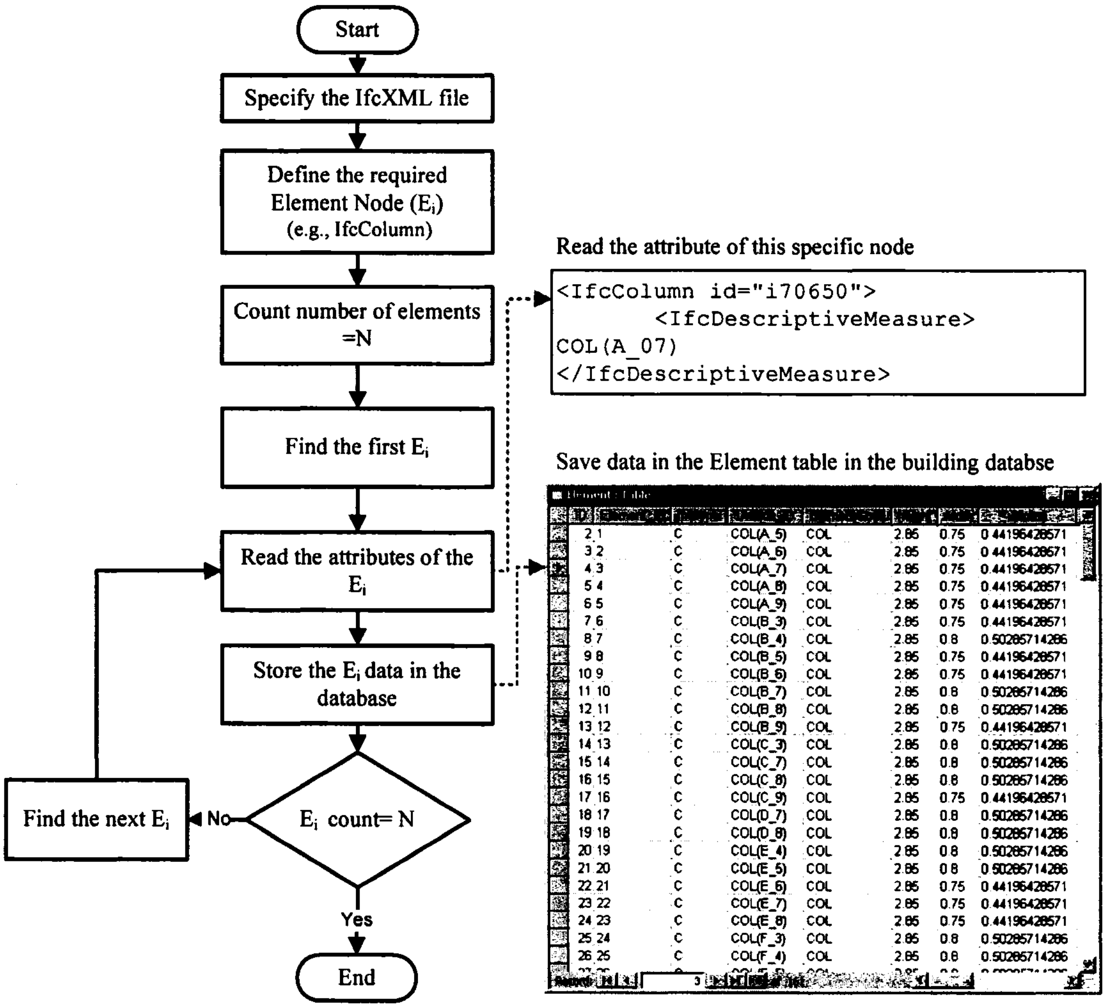


Figure 5.8 The main steps in the IfcXML parser class

The IfcXML parser package uses Xerces Java Parser (apache.org, 2005). Xerces Java Parser provides XML parsing and generation, and implements the W3C XML and DOM (Document Object Model) standards, as well as the SAX (Simple API for XML) standard. The current usage of this parser is to read IFC data of the building geometry and save it in the building database. This building information is used interactively in MBFMS database.

5.4.6 Programming approach for sharing and visualizing building information

To share data about facilities lifecycle, integrating different AEC/FM applications is important to build an nD model of the facilities that can be shared among AEC/FM personnel. Toolbox-based programming can be a powerful approach to integrate and share different sources of information that vary from CAD geometry, cost and maintenance information. This research has experienced a programming approach using VBA to share data between different applications. This approach can be used to share data between applications such as AutoCAD, MS-EXCEL, MS-ACCESS, Timberline Office, and Primavera P3.

Figure 5.9 shows a programming toolboxes approach for sharing data of a building model, cost and inspection scheduling information. The geometric data can be read and saved in a database using VB and VBA. The following sections give an experimental work of this approach.

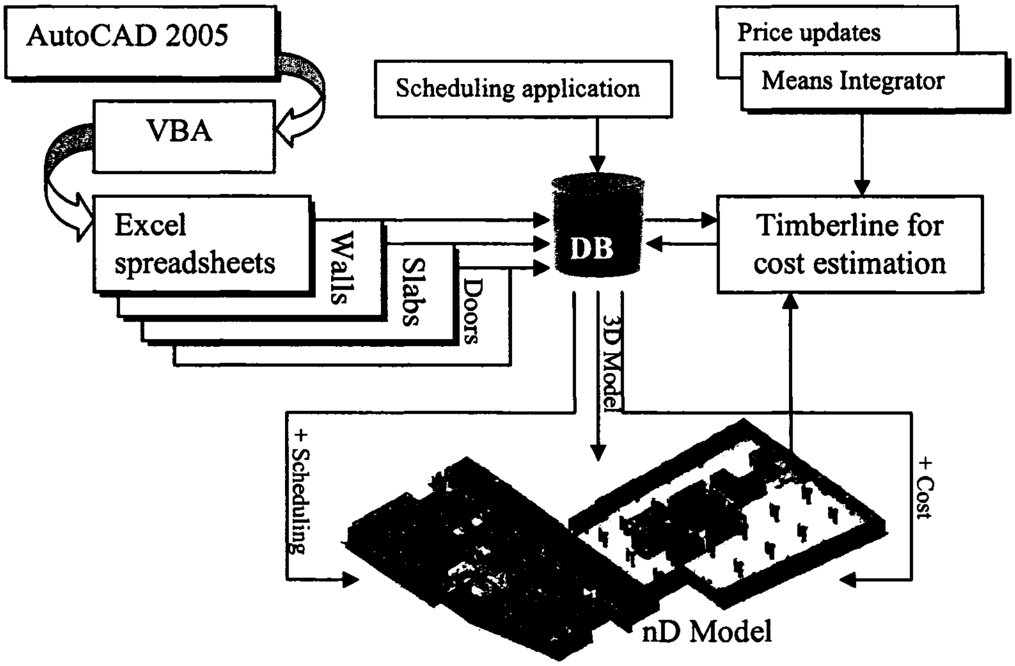


Figure 5.9 Integration of building lifecycle

5.4.6.1 Visual Basic example

This example is mainly about the VB integration approach and emphases integrating the programming power of AutoCAD's VBA tool with the functionality found in other Windows applications, specifically, Microsoft Excel (Autodesk, 2005a) and Microsoft Access (Autodesk, 2005b). Through VB programming within AutoCAD, we can launch Excel and bring the full power of a spreadsheet application into an AutoCAD drawing session. The information in this section will demonstrate how to generate, organize, query, and extract line during an AutoCAD session without spending time on additional compilers or documentation.

There are several benefits of integrating AutoCAD and MS-EXCEL (Autodesk, 2005a):

(1) To use blocks with attributes: To count blocks (or maybe even select them) based on the values in their attributes. Excel is an easy way to calculate the sum or average of that information.

(2) For reporting, querying, and attachment control: These functionalities can be offered very well by a linked spreadsheet application.

(3) Naming a block using its attributes: To specify the name of a block with an attribute value that falls within a certain numeric range and have those entities meeting that criterion reported with full text formatting in a real spreadsheet template.

(4) Portability and data reporting: We can view, report, and analyze drawings with all analytical tools in a spreadsheet (flow charts, pie diagrams, trend charts, and so on).

(5) Externalized data: Data externalization provides some benefits such as external editing and manipulation of data and drawing-size reduction, which increases drawing performance. It is possible to open a spreadsheet and change the value of a cell without opening AutoCAD.

(6) Formulas: A spreadsheet can contain a formula to generate cell content based on other cells.

5.4.6.2 Visual Basic for Application example

This following programming approach explains the idea of how to integrate AutoCAD with Excel using VBA, as a step of quantity takeoffs for the purpose of cost estimation.

This begins with a brief description of the AutoCAD VBA Integrated Development Environment (IDE) (Autodesk, 2005a). Figure 5.10 shows a program developed within AutoCAD environment for quantity takeoffs.

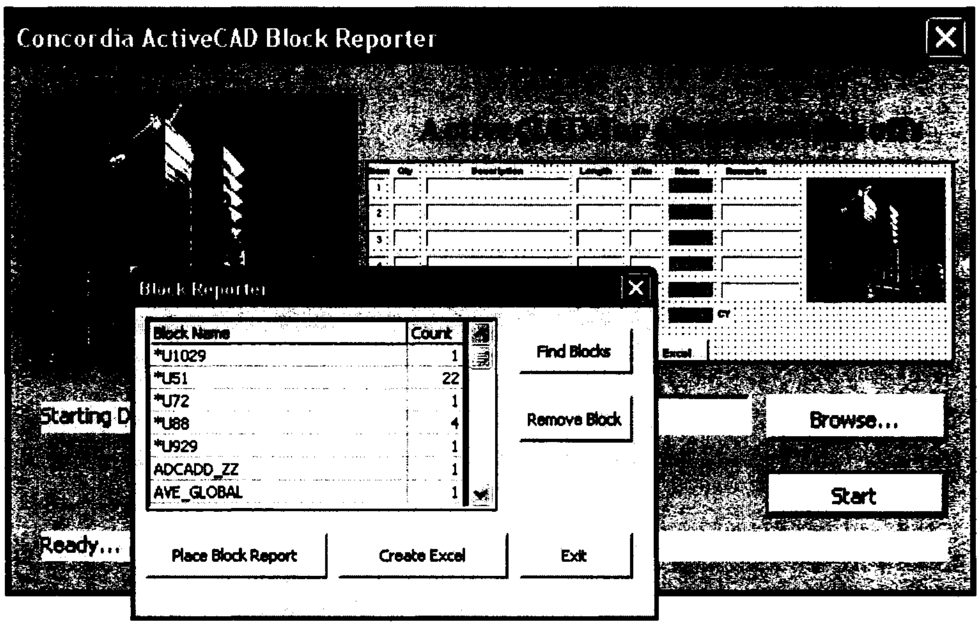


Figure 5.10 VBA application for quantity takeoff

The following experience covers the procedure of CAD data transformation.

(1) Learning to create Excel spreadsheet from AutoCAD. This Program was saved as "Launch Excel".

(2) To Place values in an Excel sheet which has been opened and then closing that sheet after asking for saving it or not.

(3) Finally creating an interface to perform the following tasks:

- Creating a spreadsheet;
- Reporting all the blocks in the AutoCAD drawing into the IDE;
- Connecting this blocks information to the opened spreadsheet; and
- Quitting the Excel application after saving.

For construction or repair progress monitoring, VBA is capable of visualizing and monitoring a construction progress within AutoCAD environment. There could be two ways to show the progress, either graphically (i.e., by graphs, charts, bar charts, etc.) or geometrically (i.e., by some shapes or drawings). In this example, the former type of progress monitoring is done through PRIMAVERA (version 3.0) and the latter through AutoCAD (version 2004). In order to interface PRIMAVERA to AutoCAD, VBA has been used within AutoCAD environment to extract and transfer data. As VB pulls out data easily from MS-EXCEL so we provided a transition, i.e., data from PRIMAVERA is first send to MS-EXCEL and then it is pulled out by VBA, processed and send to the AutoCAD drawing for visualization (El-Ammari et al., 2004).

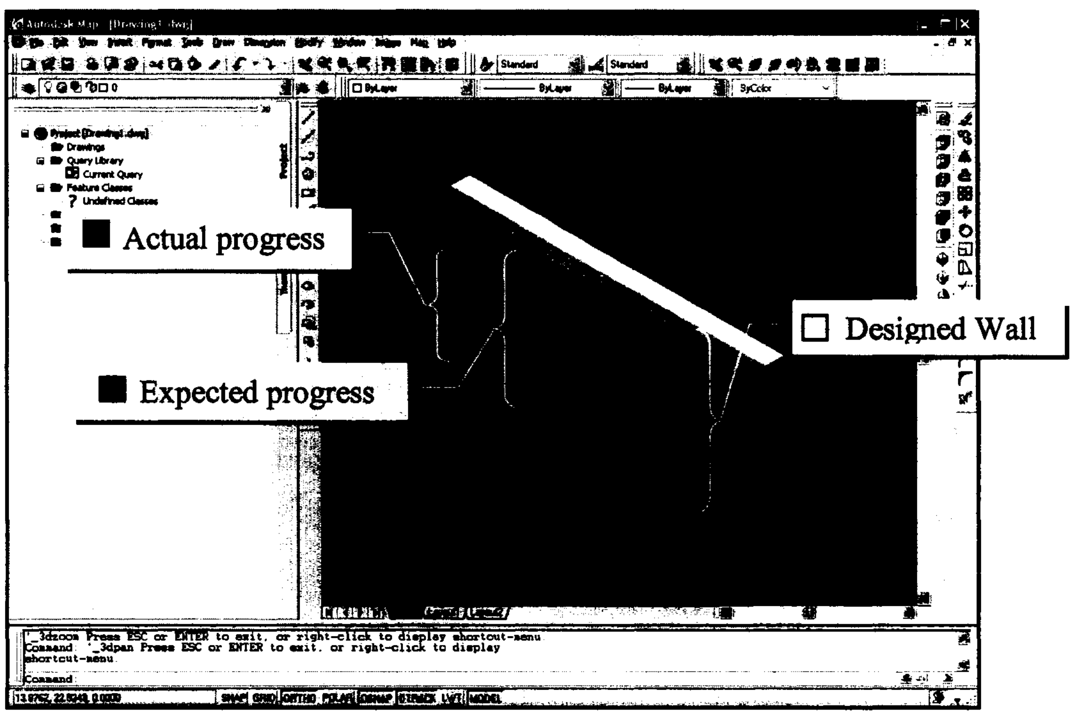


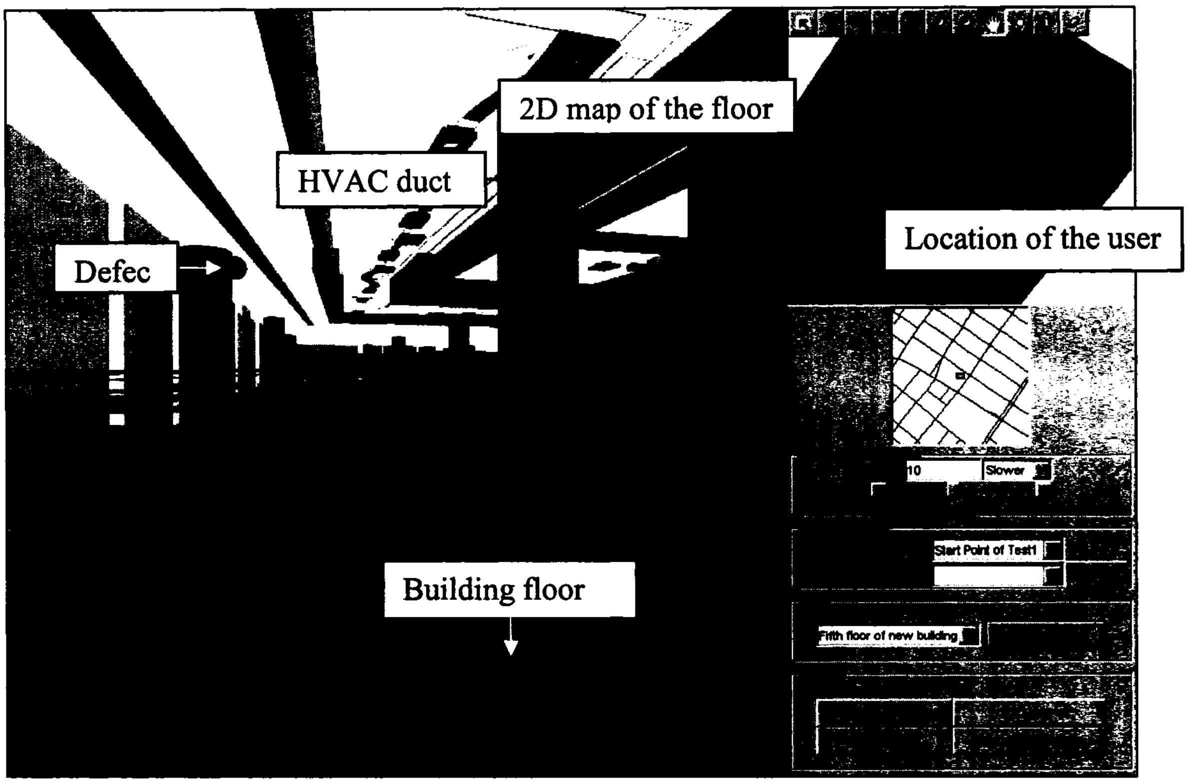
Figure 5.11 Visualization of a wall construction progress

In terms of visualization and graphical representation for CAD data, working within AutoCAD's VBA environment to visualize a construction progress has been tested. As it will be the graphical representation part of a GUI that manages and monitors the whole building lifecycle starting from 3D shape and ending with the cost and scheduling processes. Figure 5.11 shows a visualization of construction progress of a wall based on designed, actual and expected progress values. The visualization of the construction progress has been accomplished using VBA with AutoCAD 2005. Three values are used to represent the construction progress: (1) The model as designed, (2) The model at actual progress, and (3) The model at expected progress based on the schedule.

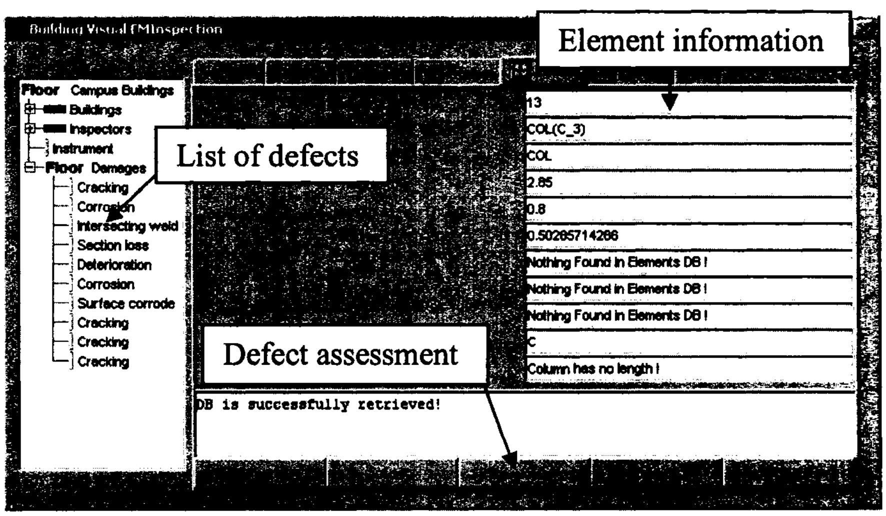
5.5 CASE STUDIES

5.5.1 Non-immersive model-based application for facilities management

In the first case study, the prototype system is used with a non-immersive VR user interface that can be used to collect inspection data in MBFMS. As a basic example, an inspection routine of a building was linked to the 3D model. In addition to the building structure model, more details have been considered such as the HVAC system model. Using the main FM navigation user interface (Figure 5.12(a)) allows inspectors to interact with the 3D model of the facility and retrieve related data instantly by picking an element. The information related to the defects on that element will be displayed in the inspection window (Figure 5.12(b)).



(a) The 3D navigation user interface



(b) Inspection window

Figure 5.12 The MBFMS main user interface

Defected elements can be tagged with colored objects (e.g., spheres, cubes and cones) to represent the location and the severity of the defects. The visualization of this information improves data sharing between the inspector and the FM office.

At the beginning of the inspection activities, navigation support features are used to provide guidance to the inspector, such as a 3D virtual arrow pointing at the defined destination of the inspector, where he/she is supposed to perform his/her inspection task. Another feature is using a hierarchy tree that describes all defects stored in the inspection database. By picking on the tree using predefined camera stored in the previous inspection, the inspector can automatically find specific defects and adjust his/her relative location in the VE.

As shown in Figure 5.12 (b), the multi-pane inspection GUI is designed to satisfy the logic of the inspection steps. After picking an element, the inspector is asked to select the specific defect type. The possible locations of the selected defect type are indicated on the inspected element using assisting feature (e.g., animated arrows). The arrow is dynamically created and added to the VE. The route of the arrow is computed based on the current position of the inspector and the location of the defect.

5.5.2 Augmented reality application for facility inspection

The same model of the campus is used as a test bed for indoor and outdoor mobile AR applications. Two tracking technologies are investigated: (1) RTK-GPS for exterior applications using a Trimble 5700 RTK-GPS receiver, and (2) video tracking using

ARToolKit for interior applications (Kato et al., 2004). The automatic picking algorithm (Hammad et al., 2006) was implemented and tested using the same virtual model discussed in the first case study.

The markers used for the ARToolKit have an edge length of 4 inches (about 100 mm). Different markers are loaded into the application and configured using a configuration file. This file specifies the real world coordinates and the direction of the view to which the marker is to be associated. The coordinates and the normal vectors of the surfaces are retrieved using the picking tool of the application. The normal vectors are used to compute the viewpoint location when the associated marker is detected. A reference 3D object from the VE is used to compute the real world location where to paste the paper markers. For example, in this case study, the locations of the markers have been measured with respect to the edge of the wall on which the makers are pasted.

Figure 5.13 simulates the view that the user sees when the real ceiling is augmented with the virtual model of the HVAC system. The inspector can assign defects on the virtual model and retrieve duct element information. The facility inspector equipped with the AR devices can perform a routine inspection task for the HVAC system (Figure 5.14). The inspector's position and orientation are tracked and used to update the 3D view.

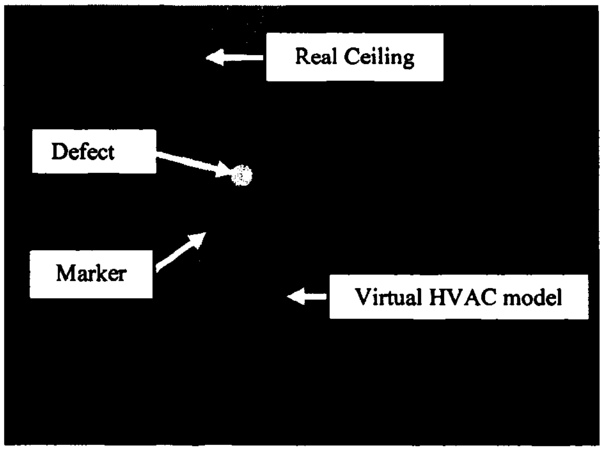


Figure 5.13 Augmentation of the HVAC model as seen by the inspector

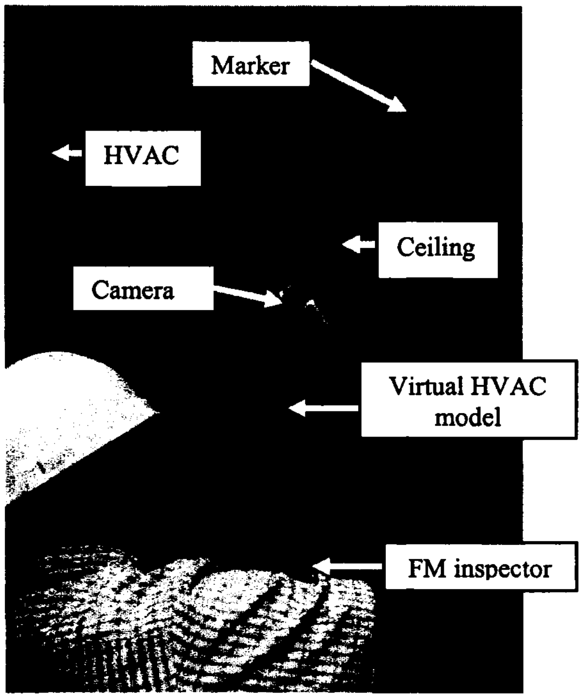


Figure 5.14 FM inspector equipped with HMD and camera for tracking

5.7 SUMMARY AND CONCLUSIONS

Compared with the conventional manual data input, using the prototype system improved the efficiency of FM data collection with the help of model-based navigation and directly marking defects on the 3D models of the facilities. In addition, visualizing the facilities in 3D helps the inspector to select elements, add defects, and update inspection information. The interaction with the VE helps in simulating the inspection process which can be also used for training purposes.

Based on the innovative approaches and methods discussed in Chapter 3 and 4, this chapter discusses the significance of the proposed framework and the developed prototype system by examining two cases studies: (1) Non-immersive model-based application for FM; and (2) AR application for facility inspection. Both case studies are used for a university campus, which demonstrates the validity of the proposed framework for large-scale facilities. Furthermore, this chapter has demonstrated the value of using 3D virtual models of the facilities linked with the IFC-based databases.

In summary, model-based FMS is useful in helping the facility inspector easily realize visualizing inspection scenarios and collect inspection data based on facilities models. This helps in eliminating input errors during inspection process and helps the inspector to focus on the inspection task.

CHAPTER 6 SUMMARY, CONCLUSIONS AND FUTURE WORK

6.1 SUMMARY

As introduced in the literature review, FMSs require very detailed information about each facility. This information can be a product model and a non-product model. Combining these models is necessary to build VE that can be used for FMSs. However, in order to create 3D models, enormous amounts of effort and money have to be consumed to obtain the data needed to build these detailed models and then to create them. Large-scale VE can be created using commercial GIS software. However, these models focus only on the exterior shapes of buildings, which are not satisfying the requirements of engineering applications such as the detailed information about design, cost and scheduling. Such information can be extracted from a building model represented in an open standard such as IFC and IfcXML.

This research introduced innovative data integration methods for automatically generating models of large-scale facilities by integrating GIS maps, CAD models, images of the buildings facades, and databases of the facilities lifecycle information. In addition, the research investigated the advantages of using IFC data for representing the facilities lifecycle information. Furthermore, the research proposed a new framework for using these models to suit the requirements of FMSs. This framework can be used by a FM inspector to interact facilities models and automatically retrieve the necessary information based on location and the task required. The research has also introduced different approaches for interacting with the 3D VE. The interaction with the facilities models can be a direct interaction by simply picking an element in the EV or indirect interaction using a hierarchy tree list.

To allow FM inspector to interact comfortably with the 3D model, a MBFMS has been designed with a well-designed user interface. This interface allows inspector to interact with the EV and to read, write and update the facilities information by marking defects on the building elements in the EV.

In addition to the design of the GUI, a new framework has been proposed to develop generic GUIs based on metadata. The necessary data for the GUIs such as defects types, inspectors' attributes, etc. can be stored in an XML file that can be updated anytime without re-developing the GUI. This eventually minimizes the development effort to update these GUIs.

A case study of a University campus is used to demonstrate the prototype system and the applications using the new proposed approaches and methods. Two scenarios have been demonstrated using VR and AR settings for MBFMS. In the first scenario, the MBFMS was used as a VR application where the inspector can perform inspection tasks in a non-immersive mode. In the second scenario, the MBFMS was used as an AR application for an HVAC inspection task. The two scenarios have shown how the MBFMS can be useful to facilitate FM requirements (e.g., marking defected building elements).

6.2 CONCLUSIONS AND CONTRIBUTIONS

The contributions of this research are:

(1) A new data integration method was developed for creating a system to automatically generate VR models of a large-scale facility by integrating GIS maps, CAD models, images of building facades, and databases of the facilities' lifecycle information.

(2) An innovative method is proposed for linking the facility model with the necessary information about buildings and building elements based on standards (IFC2x2 and IfcXML2x2). This method is based on parsing the IfcXML file to read the IFC-based building information and to save this information in a relational database.

(3) A framework was developed based on LBC for using the facility models to suit the requirements of the FMSs. Several issues necessary to realize the framework were discussed such as the GUI design based on inspection protocols and interaction methods to allow inspectors to perform their tasks easily with less data entry and less input errors.

(4) An innovative method for using metadata representation to automatically generate GUIs based on metadata XML file was discussed. This allows defining the GUI information in separate files that can be customized or updated in the future.

(5) A prototype system, implemented in Java, was developed and two scenarios of using the case study were used to demonstrate the feasibility of the above mentioned approaches and methods.

6.3 FUTURE WORK

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

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

(2) The interoperability of the system can be enhanced by extending the IFC representation in the system databases to cover more aspects of the building lifecycle (e.g., scheduling plans, cost, etc.)

(3) Creating 4D model of the large-scale facilities including the time dimension to show the changes of the building model during its lifecycle is another topic for future research.

(4) The usage of the proposed framework as a collaborative environment for field workers needs to be investigated through real-world scenarios.

REFERENCES

AecXML TC (2000) [online]. AecXML Technical Committee. aecXML Framework. Available from http://iai-france.org/telechargement/aecXML_framework(v90a).doc. [cited on January 18, 2006].

Autodesk (2005) [online]. Available from http://usa.autodesk.com. [cited on July 23, 2005].

Autodesk (2005a) [online]. *VBA: Integrating with Microsoft Excel.* Available from http://usa.autodesk.com/adsk/servlet/item?siteID=123112&id=2767040&linkID=2475912. [cited on January 3, 2005].

Autodesk (2005b) [online]. *VBA: Integrating with Microsoft Access.* Available from http://usa.autodesk.com/adsk/servlet/item?siteID=123112&id=3027392&linkID=4198946. [cited on January 3, 2005].

ArchiBus, Inc., (2005) [online]. Available from http://www.archibus.com. [cited on July 20, 2005].

ArcView GIS (1996), The Geographic Information System for Everyone, Environmental System Reasearch Inistitute.

Bentley (2005) [online]. Available from http://www.bentley.com/en-us/ [cited on January 18, 2005].

BIFM (2005) [online]. British Institute of Facilities Management. Available from http://bifm.org.uk [cited on January 24, 2006].

BSPRO (2005) [online]. *BSPro COM-Server for IFC-Files.* Available from http://www.bspro.net/ . [cited on October 8, 2005].

Cheng, J., Trivedi, P., and Law, K.H. (2002), *Ontology Mapping Between PSL and XMLBased Standards For Project Scheduling*. 3rd International Conference on Concurrent Engineering in Construction, Berkeley, CA, pp. 143-156. http://citeseer.ist.psu.edu/cheng02ontology.html

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

Concordia (2006) [online]. Concordia University website. Available from http://www.concordia.ca/. [cited on March 10, 2005].

Dennis, D. (2003). *Facility Management Information Systems*, <http://www.automatedbuildings.com/news /jan03/articles/dennis/dennis.htm>

DOE (2005) [online]. US Department of Energy. Available from http://www.eere.energy.gov/buildings/energyplus/. [cited on May 13, 2005].

Eastman, C. (1999). *Building Product Models: Computer environments supporting design and construction.*

El-Ammari, K., Hammad, A., Khabeer. B., and Mozaffari, E. (2006). *Integration and Visualization Issues in Large-Scale Location-Based Facilities Management Systems*, 6th International Conference on Construction Applications of Virtual Reality, Orlando, Florida, USA.

ESRI (2005), Getting Started with MapObjects-Java [online]. Available from http://www.esri.com/software/mojava/download.html. [cited on July 20, 2005].

Eurostep Group (2005) [online]. *Eurostep IFC toolbox products*. Available from http://www.eurostep.com/ [cited on May 23, 2005].

Froese, T. Fischer, M., Grobler, F., Ritzenthaler, J., Yu, K., Sutherland, S., Staub, S., Akinci, B., Akbas, R., Koo, B., Barron, A., and Kunz, J. (1999). *Industry Foundation Classes for Project Management - A Trial Implementation*, ITcon Vol. 4, pp. 17-36, http://www.itcon.org/1999/2.

Froese, T. (2000). *Industry Foundation Classes: A Brief Tutorial*. 8th International Conference on Computing in Civil and Building Engineering (ICCCBE-VIII), Stanford University, California, USA, August 14-17, 2000.

Froese, T. (2001). *Data Exchange Standards for AEC/FM*, Presentation given to various groups in Japan, August, 2001.

Froese, T. (2003). *Future directions for IFC-based interoperability*, ITcon Vol. 8, Special Issue IFC - Product models for the AEC arena, pp. 231-246.

Froese, T. and Yu, K. (2000). *Architecture Issues For Distributed AEC/FM Systems*. 8th International Conference on Computing in Civil and Building Engineering (ICCCBE-VIII), Stanford University, California, USA, August 14-17, 2000.

Froese, T., Yu, K., Liston, K., and Fischer, M. (2000). *System Architectures For AEC Interoperability*, Construction Information Technology 2000, Proceedings of CIT 2000 - The CIB-W78, IABSE, EG-SEA-AI International Conference on Construction Information Technology, Reykjavik, Iceland, 28-30 June, 2000, G. Gudnason (Ed.), Icelandic Building Research Institute. Vol. 1, pp. 362-373.

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.

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

Goldstein, B., Kemmerer, S., and Parks, C. (1998). *A Brief History of Early Product Data Exchange Standard-NISTIR 6221.* NIST, U.S.A. Available from http://www.mel.nist.gov/msidlibrary/doc/proddata.pdf [cited on October 20, 2005].

Google Earth (2006). Google Earth KML Documentation. Available from http://earth.google.com/kml/. [cited on February 24, 2006].

Granlund (2005) [online]. *RIUSKA software.* Available from http://www.granlund.fi/granlund_eng/ohjelmistomyynti/frameset.htm [cited on February 24, 2005].

Graphisoft (2005) [online]. *Archicad9.0* . Available from http://www.graphisoft.com/products/archicad/?source=google_GSUS [cited on October 26, 2005].

Graphisoft (2005a) [online]. ArchiFM introduction- *A better solution to the fundamental requirements of facility management.* Available from http://www.graphisoft.co.uk/products/archifm/ [cited on October 6, 2005].

Hammad, A., Garrett, J.H. and Karimi, H. (2004). *Location-Based computing for infrastructure field tasks*, In Karimi, H. and Hammad, A. (editors), "Telegeoinformatics: Location-Based Computing and Services," CRC Press.

Hammad, A., Mozaffari, E., Khabeer. B., and EL-Ammari, K. (2006). *Framework for Virtual and Mixed Reality Applications in Civil Engineering*, Joint International Conference on Computing and Decision Making in Civil and Building Engineering, Montreal, Canada.

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, 2000.

Hassanain, M., Froese, T., Vanier, D. (2003). *Implementation of a distributed, model-based integrated asset management system*, ITcon Vol. 8, pp. 119-134, http://www.itcon.org/2003/10.

Halfawy, M., Froese, T. (2005). *Integration of data and processes of aec projects using the industry foundation classes*. Proceedings of the 6th Construction specialty Conference. Toronto Canada. CT-175.

He, H., Hammad, A. and Fazio, P. (2005). *Application of IT and international standards to improve building envelope performance*. The Ninth International Building Performance Simulation Association Conference, Montreal, Canada.

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

Hua, G. (2005). IT barometer 2003: Survey of the Singapore construction industry and a comparison of results, ITcon Vol. 10, pp. 1-13, http://www.itcon.org/2005/1.

IAI (2005a) [online]. A short history of the IAI and the IFC information model. Available from http://www.iai-international.org/iai_international/ [cited on August 7, 2005].

IAI (2005b) [online]. *The EXPRESS Definition Language for IFC Development*. Available from http://www.iai-international.org/iai_international/Technical_Documents/documentation/The_EXPRESS_Definition_Language_for_IFC_Development.pdf. [cited on August 22, 2005].

IAI (2005c) [online]. *Data Modelling Using EXPRESS-G for IFC Development.* Available from http://www.iai-international.org/iai_international/Technical_Documents/documentation/Data_Modelling_Using_EXPRESS-G_for_IFC_Development.pdf. [cited on August 25, 2005].

IAI (2005d) [online]. *Introduction to IAI and Its IFCs*. Available from http://cig.bre.co.uk/iai_uk/iai/ . [cited on October 27, 2005].

IAI (2005e) [online]. *IFC2x edition2 model implementation guide version 1.7.* Available from http://www.iai-international.org/Model/files/20040318_Ifc2x_ModelImplGuide_V1-7.pdf [cited on October 29, 2005].

IAI (2005) [online]. Available from http://www.iai-international.org/iai_international [cited on February 14, 2005].

Ifc-mBomb (2006). *Ifc Model Based Operation and Maintenance of Buildings* (Ifc-mBomb). Available from http://cig.bre.co.uk/iai_uk/iai_projects/ifc-mbomb/. [cited on January 13, 2006].

IFG (2005) [online]. Available from http://www.iai.no/ifg/ [cited on March 13, 2005].

IGES (2005) [online]. *About IGES.* Available from http://www.nist.gov/iges/ . [cited on November 8, 2005].

IGES (2005a) [online]. IGES Import for AutoCAD - IGES file import plug-in for AutoCAD. Available from http://fpsols.com/iges_import_ac.html [cited on March 3, 2006].

Inopso GmbH (2005) [online]. *IFC2x Utility.* Available from http://www.inopso.com/ [cited on November 12, 2005].

IRCC, Institute for Research in Construction, Canada (1993). *Protocols for building condition assessment.*

ISO (1999). *International Standard ISO10303-225.* Geneva, Switzerland: International Organization for Standardization.

ISO (2005) [online]. *Industrial automation systems and integration - product data representation and exchange -- Part 11.* Available from http://www.iso.org/iso/en/aboutiso/introduction [cited on October 20, 2005].

ISO 10303-1 (1994). *Industrial Automation Systems and Integration Product Data Representation and Exchange - Overview and Fundamental Principles, International Standard, ISO TC184/SC4.*

ISO 10303-11(1994). *Industrial Automation Systems and Integration – Product Data Representation and Exchange – Part 11: Description Methods: The Express Language Reference Manual.* Geneva, Switzerland: International Organization for Standardization.

ISO 10303-21 (2002). *Industrial automation systems and integration -- Product data representation and exchange -- Part 21: Implementation methods: Clear text encoding of the exchange structure.* Geneva, Switzerland: International Organization for Standardization.

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

Kato, H., Billinghurst, M. and Poupyrev, I. (2006). ARToolKit version 2.33: A Software Library for Augmented Reality Application. http://hitl.washington.edu/artoolkit.

Kam C., Fischer M., Hänninen, R., Karjalainen, A., and Laitinen, J. (2003). *The product model and Fourth Dimension project*, ITcon Vol. 8, Special Issue IFC - Product models for the AEC arena , pp. 137-166, http://www.itcon.org/2003/12.

Karimi H A, Hammad A. 2004. Telegeoinformatics: Location-Based Computing and Services, CRC Press: London.

Kyle, B., Vanier, D. and Lounis, Z. (2002). *The BELCAM Project: a summary of three years of research in service life prediction and information technology*, 9th International Conference on the Durability of Building Materials and Components (Brisbane, Australia, 3/17/2002), Paper 138, pp. 1-10, April 01, 2002 (NRCC-45189).

LandXML (2006) [online]. LandXML schema development. Available from http://www.landxml.org/. [cited on January 18, 2006].

Loffredo, D. (2003). *Fundamentals of STEP Implementation*, STEP Tools, Inc., Rensselaer Technology Park, Troy, New York 12180.

Lounis, Z., D.J. Vanier, M.A. Lacasse, and B.R. Kyle (1999). *Decision-Support System For Service Life Asset Management: The BELCAM Project.* Durability of Building Materials and Components, National Research Council Canada, Ottawa, 4:2338-2347.

MapObjects-Java web site (2006). Avalible from http://www.esri.com/software/mojava [cited on January 23, 2006].

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

Moeslund, T., Störring, M., Broll, W., Aish, F., Liu, Y. and Granum, E. (2004). *The ARTHUR System: An Augmented Round Table*. The 8th World Multi-Conference on Systemics, Cybernetics and Informatics, Orlando, Florida.

Mozaffari, E., Hammad, A., and El-Ammari, K. (2005). *Virtual Reality Models for Location-Based Facilities Management systems*, 1st CSCE Specialty Conference on Infrastructure Technologies, Management and Policy, Toronto, Ontario, Canada.

MRO (2006) [online]. MRO Software. Available from http://www.mro.com/ [cited on January 15, 2006].

Nadeau, D. and Moreland, J. (1996). The VRML Sourcebook, John Wiley & Sons, Inc., publication, ISBN: 0471141593.

Noveloso, C. O. (2005). *Maximo: A case study of a computerized maintenance management system in an automotive plant*, Ph.D. dissertation, Capella University, AAT 3166789.

NRCC (2006) [online]. National Research Council Canada. Available from http://nrc-cnrc.gc.ca/. [cited on January 27, 2006].

NRCC (1997). National Research Council Canada National. BELCAM project. Available from http://irc.nrc-cnrc.gc.ca/pubs/ci/v3no1/v3no1_6_e.html. [cited on December 11, 2005].

Object-Relational mapping website (2006). http://www.service-architecture.com/java-databases/index.html.

O' Sullivan, D.T.J., Keane, M.M., and Kelliher, D. (2002). *Specification of a STEP compliant integrated environment to support performance based assessment and control of building energy system.*

Pouria, A. and Froese, T. (2001). *Transaction and Implementation Standards in the AEC/FM Industry*, Annual Conference of the Canadian Society for Civil Engineers, Victoria, BC. May 30-Jun 2, 2001. Paper C43.

Romberg, R., Niggl, A., van Treeck, C., and Rank, E. (2004). *Structural Analysis based on the Product Model Standard IFC.* The 10th International Conference on Computing in Civil and Building Engineering (ICCCBE), Weimar, Germany.

Satyanarayanan, M. (2001). *Pervasive computing: vision and challenges*, IEEE Personal Communications, pp. 10-17.

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

Seren K., and Karstila K. (2001). *MS Project – IFC Mapping Specification.* Available from http://www.eurostep.fi/public/4D/4D_Mapping_V1_0.pdf [cited on Augest 13, 2005].

Smith, B. (1988) [online]. *The Initial Graphics Exchange Specifications.* Available from http://nvl.nist.gov/pub/nistpubs/sp958-lide/246-249.pdf. [cited on January 18, 2005].

Springer, T. (2001). *Facility Management – An Introduction.* 1st chapter of the Facility. Design and Management Handbook.

STEP (2003). *SC4 Open: Developers of International Industrial Data Standards.* Available from http://www.tc184-sc4.org/SC4_Open/. [cited on January 18, 2005].

STEP Tools (2005) [online]. *What is STEP?.* Available from http://www.steptools.com/library/standard/. [cited on January 25, 2005].

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

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.

TAC (2005) [online]. University of Illinois, Technical Assistance Center - Roofer Software. Available from http://www.tac.uiuc.edu/software/roofer/roofer.html. [cited on January 17, 2006].

Timberline (2005) [online]. Timberline Office. Available from http://www.timberline.com/software/estimating/applications/cad_integrator.aspx . [cited on January 27, 2005].

TNO (2005) [online]. IFC Engine Series. Available from http://www.ifcbrowser.com/ [cited on August 20, 2005].

Virtual Reality Laboratory website (2006). Virtual Reality: a Short Introduction, Virtual Reality Laboratory, University of Michigan, College of Engineering. Available from http://www-VRL.umich.edu. [cited on January 5, 2006].

VTT (2005) [online]. *IFC Model Server Research and Implementation Project*, Available from http://cic.vtt.fi/projects/ifcsvr/ppt/IFCproject_en_files/frame.htm [cited on October 19, 2005].

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

Wan C, Chen P - H, Tiong R. L. K. (2004). *Assessment of IFCs for structural analysis domain* , ITcon Vol. 9, pp. 75-95, http://www.itcon.org/2004/5.

XML (2005) [online]. Available from http://www.xml.org [cited on November 5, 2005].

XMLSpy (2005) [online]. Available from www.altova.com/XMLSpy/ [cited on August 3, 2005].

Yu, K., Froese, T., and Grobler, F. (2000). *A Development Framework for Data Models for Computer-Integrated Facilities Management.* Automation in Construction, special issue on facilities management, Vol. 9, No. 2, 2000, pp.145-167.

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

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

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

APPENDIX A: IFC Tools and Specifications

A.1 Architectural Desktop with its IFC2x Utility

Autodesk Architectural Desktop includes a utility to import and export IFC 1.5.1 format files. Currently, support for the IFC2x format is provided through the Ifc2x Utility. This add-on tool is developed by Autodesk with the cooperation of Inopso GmbH. This collaboration leaded to the development of the IFC2x Utility and a CAD-IFC-translator, which allows the building industry professionals to freely share domain-specific data. This tool allows professionals to convert their CAD application to an IFC-compatible CAD application which enhances the interoperability and data sharing with other interested parties. These data could be created in Autodesk Architectural Desktop 3.3, 2004, and 2005 software. Accordingly, with the IFC2x Utility, Autodesk can support industry standards for integration and interoperability (Autodesk, 2005).Figure 3.2 shows the ductwork add-on tool for AutoCAD for creating the HVAC model.

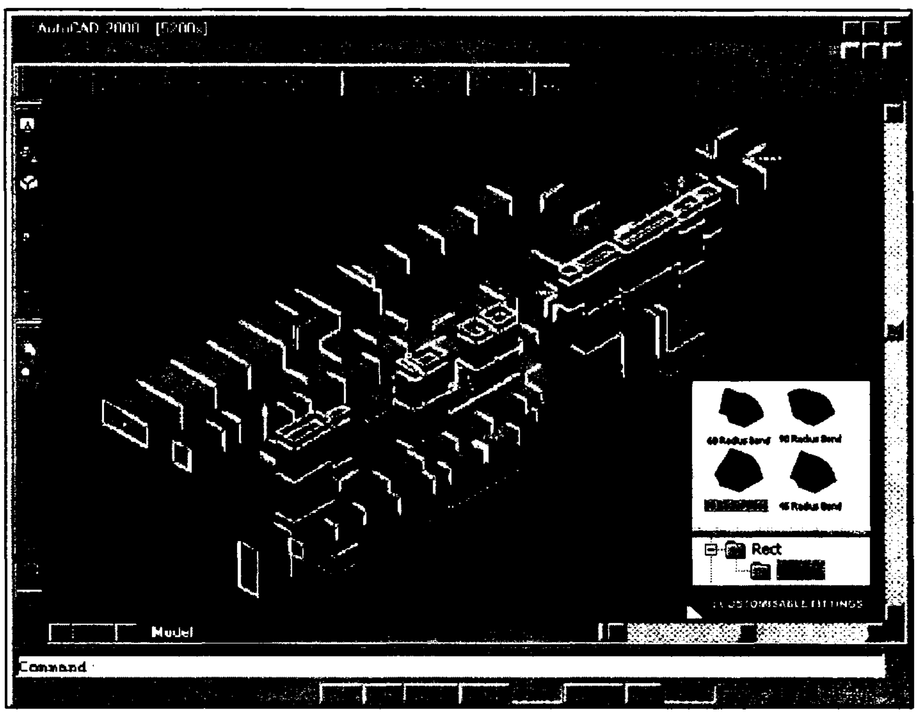


Figure A.1 HVAC modeling using ductwork add-on tool

There are variant add-on tools for integrating with CAD applications such as ADT and ArchiCAD. Several commercial add-on tools for IFC creation have been developed such as CAD-duct for AutoCAD and ADT and Ductwork for ArchiCAD to allow modeling of the HVAC systems. These tools allow CAD applications' users to create IFC-based HVAC models.

A.2 ArchiCAD's Add-on Tool for IFC

Graphisoft's ArchiCAD 9.0 with its add-on tool is an application to automatically generate IFC files from the CAD drawings. Graphisoft has been a member of the IAI since 1996 and has contributed in supporting IFC by developing IFC-compatible CAD applications. The ArchiCAD's add-on interface provides the users with a tool which is compatible with the releases IFC 1.5.1 and IFC 2.0. Moreover, Graphisoft is currently implementing and supporting IFC releases such as IFC2x, IFC2x2 and Xml-based IFC (IfcXML2x).

Table A.2 IFC Model Specifications (Ifc-mBomb, 2006)

Information	**Description**
IFC 2x (addendum 1) online specification	Complete HTML documentation of the IFC 2x (first addendum) model.
IFC 2x Model Implementation Guide	IFC 2x Model Implementation Guide, version 1.4, 5-July-2002
IFC 2x2 Final online specification	Complete HTML documentation of the IFC 2x2 Final model. (May 2003)
IFC 2x2 (addendum 1) online specification	Complete HTML documentation of the IFC 2x2 (first addendum) - a technical corrigendum that addresses a few implementation issues, like the profile property assignment that came up in this group. (July 2004)
IFC 2x Edition 2 Model Implementation Guide	IFC 2x Edition 2 Model Implementation Guide, version 1.7, 18-Mar-2004
more IFC documentation	Further IAI IFC documentation

Table A.3 IFC Browsers and Tools (Ifc-mBomb, 2006)

To install the programs save the files and follow any readme information that is provided.

Application	**Description**
IfcEngine Viewer	A viewer to display the Ifc models, developed by TNO. Supports Ifc 1.5.1, 2.0, 2x and 2x2 models E-mail Peter Bonsma at TNO for more information http://www.ifcbrowser.com
IfcEngine Viewer beta version	A viewer to display the Ifc models, developed by TNO.

IfcEngine Basic	A viewer to browse IFC model data and to display the model, developed by TNO. Supports Ifc 2x models
IfcEngine Basic beta version	A viewer to browse IFC model data and to display the model, developed by TNO.
IFC Quick Browser	You can use IFC Quick Browser to text-browse even very large IFC files. The IFC file is displayed in a tree structure.
DDS Ifc Viewer	This "viewer" is not primarily made to be fast and efficient as a 3D browser, but this is simply a tool used internally in DDS to check IFC files. For more information.
Ifc Tree	Utility tool for IFC files. Supports small Ifc151 and 2x models that are less than 1 mBytes in size.
IfcObject Counter	This program has been designed for checking files that claim to conform to the IFC (Industry Formation Classes) specification of IAI (International Alliance for Interoperability). Supported IFC versions are IFC 1.5.1, IFC 2.0, IFC 2.x, and IFC 2x2 (final). E-mail Karl-Heinz Haefele for more information.
IfcStorey Viewer	A program to view stories in an Ifc model. Supported IFC versions are IFC 2.x, and IFC 2x2 (final). E-mail Karl-Heinz Haefele for more information.
Programmer's file editor	Programmer's File Editor, is a programmer-oriented text editor for Windows that has wide application in all areas needing the ability to edit files.

Table A.4 IFC2x Import/Export Modules (Ifc-mBomb, 2006)

Implementation	**Description**
IFC-Utility 2x for ADT (select IFC products)	IFC-Utility 2x for ADT is the IFC interface (IFC 2x) for the Autodesk® Architectural Desktop. IFC-Utility 2x for ADT runs as an ARX application together with Autodesk® ADT 3.3, ADT 2004 and ADT 2005.
ArchiCAD IFC addon final versions	Download final versions of the ArchiCAD IFC 151 and 2.0 Add-ons.
ArchiCAD IFC addon beta versions	Download beta versions of the ArchiCAD IFC 151, 2.0, 2x and 2x2 Add-ons.

Triforma v8 IFC utility	E-mail Volker Thein at Bentley for more information regarding the Triforma IFC 2x utility.
ALLPLAN 2003	E-mail Rasso Steinmann at Nemetschek for more information regarding the ALLPLAN 2003 IFC 2x utility.

APPENDIX B: Installation guide of the prototype system

B.1 Software requirements

(1) Borland JBuilder 2005 Enterprise: used to develop the prototype system of BMS;
(2) MS Access (MS Access XP): used to store the lifecycle data of the buildings;
(3) ArcGIS (ESRI 2004): used to develop GIS application;
(4) Netica: used to develop the prototype of learning-based BN model;
(5) Windows XP: used as the operation system.

B.2 Installation guide

1. Copy four folders to corresponding driver and change the associated code in the project to match the driver path. The contents in these folders include:
   - *Currproject* or *infra_project* (The folder includes all codes of our project)
   - *Javasoft* (The folder includes all libraries which are required in our project)
   - *Bridgere* (The folder includes all 2d information)
   - *BridgeResources* (The folder includes all 3d information and models)

2. Click Start->Control Panel-> Administrative Tools->Database Source.
   And add ODBC data source as below:
   - Microsoft Access Driver: Name: bridge, Location: C:\ BridgeResources\db1.mdb

   (No password for data source is required, so just leave the password as blank.)

3. Launch Jbuilder, open the project.jpx file. Then click the menu of Jbuilder: Project->Project Properties. Click the tab "Required Libraries". Then edit or add the path of libraries as below:

VRML97 (VRML File Loader API)
- Download the library from: https://j3d-vrml97.dev.java.net/ and install it. Or you can get it from the path: D:/javasoft/loaders/vrml97.jar
- Edit or add the vrml97.jar file to the path of VRML97 library.

Jess61p4 (Java Expert System Shell API) - Version 6.1
- Download the library from: http://herzberg.ca.sandia.gov/jess/ and install it. Or you can get it from the path: D:/javasoft/Jess61p4/jess.jar.
- Edit or add the jess.jar file to the path of Jess61p4 library.

Jdk3D (Java development Kit 3D) - Version 1.3.1
- Download the library from: http://java.sun.com/products/java-media/3D/downloads/index.html and install it. Or you can get it from the path: D:/javasoft/ JRE/1.3.1_09/lib/ext

- Edit or add all jar files under the Jdk3d directory to the path of Jdk3D library.

Javacomm (Java Communications API) - Version 3.0

- Download the library from: http://www.sun.com/download/products.xml?id=43208d3d and install it. Or you can get it from the path: D:/Javasoft/commapi/
- Edit or add the comm.jar file under the commapi directory to the path of Javacomm library .

DXFLoader (DXF file loader API) - Version 1.0

- Download the library from: http://www.johannes-raida.de/index.htm?cadviewer and install it. Or you can get it from the path: D:/Javasoft/DxFloader/
- Edit or add the dxfloader.jar file under the dxfloader directory to the path of DXFLoader library.

SAX (Simple API for XML) Version 2.0.2 (sax2r3) final

- Download the library from: http://www.saxproject.org and unzip the package and relocate the folder in D:\JavaSoft\
- Edit or add the path of the API in the IfcXMLParser main class (InspectionXmlReader1)

MOJ (MapObject Java API) - Version 2.1

- Download the library from: http://www.esri.com/software/mojava/ and install it.
- Edit or add all jar files under the directory C:/ESRI/MOJ21/lib to the path of MOJ library. Please also add the tutsource.jar and tutorial.jar files that are available at the directory MOJ21\Samples\Tutorial.

Netica (Netica Java API) - Version 2.17

- Download the library from: http://www.norsys.com/netica-j.html#download and install it. Or you can get it from the path: D:/Javasoft/ NeticaJ_Win/bin
- Edit or add the path of the directory NeticaJ_Win/bin to the path of Netica library.

JMF (Java Media Framework API) - Version 2.1.1

- Download the library from: http://java.sun.com/products/java-media/jmf/2.1.1/download.html
  and install it. Or you can get it from the path: D:/Javasoft/ jmf211e_scst/build/win32/lib
- Edit or add the path of the directory jmf211e_scst/build/win32/lib to the path of JMF library.

JARToolkit (Java ARToolkit API) - Version 2.0

- Download the library from: http://jerry.c-lab.de/jartoolkit/ and install it. Or you can get it from the path: D:\JavaSoft\JARToolkit Dlls
- JARToolkit need the dll files as below:
  - JARFrameGrabber.dll
  - JARToolkit.dll

- JARVideo.dll
- libARvideo.dll
- libARvideod.dll
- msvcr70.dll

Make sure you put these dll files in your JARToolkit Dlls directory, then click start-> Control Panel->System->Advanced->Environment Variables, please add the path of JARToolkit Dlls directory to the option"path" of User Variables.

Notes:

* When you use a different account in the same computer, you have to separately set the path of libraries for every account. It means you cannot just set the libraries for all accounts at the same time.

* If your code cannot be compiled following the above instructions, please carefully check the installation instruction. If the error information is about Java 3D, that means your computer does not have Java 3D. You can get the Java 3D package from the Java 3D folder.

Table B.1 Summary of libraries used in the prototype system

Library	Description	Source	Version
DXFLoader	DXF File Loader API	http://www.johannes-raida.de/index.htm?cadviewer D:/Javasoft/DxFloader/	1.0
JARToolKit	Java ARToolKit API	http://jerry.c-lab.de/jartoolkit/ D:\JavaSoft\JARToolkit Dlls	2.0
Javacomm	Java Communication API	http://www.sun.com/download/products.xml?id=43208d3d D:/Javasoft/commapi/	3.0
JDK 3D	Java 3D API	http://java.sun.com/products/javamedia/3D/downloads/index.html D:/javasoft/ JRE/1.3.1_09/lib/ext	1.3.1
Jess61p4	Java Expert System Shell API	http://herzberg.ca.sandia.gov/jess/ D:/javasoft/Jess61p4/jess.jar	6.1
JMF	Java Media Framework API	http://java.sun.com/products/java-media/jmf/2.1.1/download.html D:/Javasoft/jmf211e_scst/build/win32/lib	2.1.1
MOJ	MapObject Java API	http://www.esri.com/software/mojava/ C:/ESRI/MOJ21/lib	2.1
Netica	Netica Java API	http://www.norsys.com/netica-j.html#download D:/Javasoft/ NeticaJ_Win/bin	2.17
SAX	SAX (Simple API for XML)	http://www.saxproject.org	2.0.2
VRML97	VRML File Loader API	https://j3d-vrml97.dev.java.net/ D:/javasoft/loaders/vrml97.jar	2.0

APPENDIX C: XML-based defects and building element data representations

C.1 List of defects

This XML file lists defects types based on protocols for building condition assessment (IRCC, 1993).


```
<?xml version="1.0" encoding="UTF-8" ?>
- <!--
edited with XMLSpy v2006 U (http://www.altova.com) by Khaled El Ammari
(Concordia Infra)
-->
- <dataroot xmlns:od="urn:schemas-microsoft-com:officedata" generated="2005-
11-01T15:58:05">
- <EWDefect>
<DefectType id="1" att="Observable differential movement" />
<DefectType id="2" att="Deformation" />
<DefectType id="3" att="Bulging" />
<DefectType id="4" att="Cracks in cladding, mortar" />
<DefectType id="5" att="Loose bricks, blocks, paneis, ties, etc." />
<DefectType id="6" att="Spalled material" />
<DefectType id="7" att="Exposed structure crushed" />
<DefectType id="8" att="Corroded ties or anchors" />
<DefectType id="9" att="Corroded reinforcing" />
<DefectType id="10" att="Dampness, wet spots" />
<DefectType id="11" att="Efflorescence" />
<DefectType id="12" att="Cavity wall weep holes plugged" />
<DefectType id="13" att="Waterproofing incomplete" />
<DefectType id="14" att="Flashing deficiencies" />
<DefectType id="15" att="Expansion joints" />
<DefectType id="16" att="Control joints" />
<DefectType id="17" att="Parapets" />
<DefectType id="18" att="Bearing for brickwork" />
<DefectType id="19" att="Width of gap below shelf angle" />
<DefectType id="20" att="Mortared-in gap below shelf angle" />
<DefectType id="21" att="Symptoms of chemical attack" />
<DefectType id="22" att="Visible repairs" />
</EWDefect>
- <IWDefect>
<DefectType id="1" att="Excessive differential movement" />
<DefectType id="2" att="Deformations" />
<DefectType id="3" att="Crack in interior finishes" />
<DefectType id="4" att="Loose material" />
<DefectType id="5" att="Dampness, wet spots, condensation" />
<DefectType id="6" att="Staining, discoloration" />
<DefectType id="7" att="Flaking paint" />
<DefectType id="8" att="Patches" />
<DefectType id="9" att="Visible repair" />
```

```
<DefectType id="10" att="Expansion joints" />
<DefectType id="11" att="Control joints" />
<DefectType id="12" att="Rotting, bad smell" />
  </IWDefect>
  </dataroot>
```


C.2 Building Element

This XML-based list of building elements is based on Protocols for building condition assessment (IRCC, 1993).


```
  <?xml version="1.0" encoding="UTF-8" ?>
- <!--
 FM Elements for metadata based on (Yu, Froese, Grobler 2000)
  -->
- <ElementOfFM>
- <BuildingSystemElements>
- <BuildingShell>
<Roofs />
<Walls />
<Windows />
<Doors />
<Columns />
<Beams />
<Ceilings />
<Floors />
<Stairs />
<Spaces />
<Signs />
- <Interiors>
<Carpentry />
<Carpent />
<Painting />
  </Interiors>
  </BuildingShell>
- <Environment>
- <Utilities>
<Electricity />
<Gas />
<HVAC />
<Water />
  </Utilities>
- <Connectivity>
<Cables />
<Data />
<Voice />
  </Connectivity>
- <Waste-Recycle>
<HazardousWaste />
```

<RecycleObjects />
</Waste-Recycle>
- <Security>
<Accesses />
<Alarms />
<Keys />
<Locks />
</Security>
</Environment>
- <Mechanical>
<Conduits />
<HVACEquipment />
<Motors />
<Piping-Ducts />
<Plumbing />
<Pumps />
<FireDetection-Suppression />
</Mechanical>
- <Electrical>
<Lights />
<Outlets />
<Panels />
<Switches />
<SwitchBoards />
<Wires />
</Electrical>
</BuildingSystemElements>
- <NonBuildingSystemElements>
<Site-Ground />
</NonBuildingSystemElements>
</ElementOfFM>

APPENDIX D: List of publications

(1) **El-Ammari, K.**, Hammad, A., Khabeer. B., and Mozaffari, E. (2006). Integration and Visualization Issues in Large-Scale Location-Based Facilities Management Systems, 6th International Conference on Construction Applications of Virtual Reality, Orlando, Florida, USA.

(2) Hammad, A., Mozaffari, E., Khabeer. B., and **EL-Ammari, K.** (2006). Framework for Virtual and Mixed Reality Applications in Civil Engineering, Joint International Conference on Computing and Decision Making in Civil and Building Engineering, Montreal, Canada.

(3) Mozaffari, E., Hammad, A., and **El-Ammari, K.** (2006). Navigation Methods for Engineering Applications in Urban Virtual Environments, 1st International Construction Specialty Conference, Management and Policy, Calgary, Canada.

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

(5) Mozaffari, E., Hammad, A., and **El-Ammari, K.** (2005). Virtual Reality Models for Location-Based Facilities Management systems, 1st CSCE Specialty Conference on Infrastructure Technologies, Management and Policy, Toronto, Ontario, Canada.