A Web Based Integrated Global Project Management System for Large Scale A/E/C Projects

by

Ajit Kumar Sutar

B.Tech. (Hons), Ocean Engineering and Naval Architecture, IIT Kharagpur, 1998

Submitted to the Department of Civil and Environmental Engineering in Partial Fulfillment of the Requirements for the Degree of

Master of Science

at the

Massachusetts Institute of Technology

January 2000

© 2000 Massachusetts Institute of Technology
All rights reserved

Signature of Author

Department of Civil and Environmental Engineering
January 14, 2000

Certified by

Feniosky Peña-Mora
Associate Professor of Civil and Environmental Engineering
Thesis Supervisor

Accepted by

Daniele Veneziano
Chairman, Departmental Committee on Graduate Studies
A Web Based Integrated Global Project Management System for Large Scale A/E/C Projects

by

Ajit Kumar Sutar

Submitted to the Department of Civil and Environmental Engineering on January 14, 2000, in Partial Fulfillment of the Requirements for the Degree of Master of Science

ABSTRACT

The globalization of architecture, engineering and construction (A/E/C) industry has added new dimensions to project management. A wide spectrum of tools and software are used by project managers to manage these geographically distributed projects. PC or LAN based project management software is already widespread and popular in A/E/C industry. However, the access to the software is limited to the LAN and the server where they reside, and to use the software, project managers have to undergo extensive training. Current practice in collecting latest progress data for updating schedule is time consuming, and it is difficult to communicate this information to all the participants involved, especially if they are in different companies and located in different countries.

The goal of this thesis is to develop an Integrated Web based Global Project Management System (IGPMS). IGPMS provides a web based graphical user interface that allows project managers to view and update the projects online. IGPMS provides project managers with web-based analysis and reporting tool for the project schedules. IGPMS also integrates the different information systems used for project management through software interoperability and provides a "common language" for product, process and resource information. IGPMS also introduces the concept of a web based project repository where project managers can submit a new project to the server and instantly view, update and analyze the project schedule through a web browser.

Thesis Supervisor: Feniosky Peña-Mora
Title: Associate Professor of Civil and Environmental Engineering
ACKNOWLEDGEMENTS

I am thankful to Professor Peña-Mora for his valuable time and inexhaustible advice on my research and thesis. Without his help and guidance this thesis would not have been a reality. It was a great experience working with him.

I am indebted to my friends in IESL for their help and support. I had a nice time in IESL with friends like Kiran, Paddu, JP, Vada, Petros, Tong, Abel, Jaime, Moonseo, Tamaki, Michael, Emma, Ben, Jingsong, Julio, Justin, Kathy and Adriana. I would especially like to thank JP for his help in my research. I would also like to thank my roommate Sahoo for his help and support. I am thankful to Joannie and Cynthia Stewart for helping me out in the administrative matters. I am also thankful to my non-IESL friends like Amit, Yogesh, Niranjan, Vinay, Ganti, Gangs and Deepak.

I am thankful to my family for their love, patience and understanding. Even though they live on the other side of the globe, they have been a constant source of inspiration and encouragement during my studies at MIT. It has been almost two years since I left my home in India and I cannot wait till I go back to meet my family.

Ajit
# TABLE OF CONTENTS

**LIST OF FIGURES** ................................................................. 6

**EXECUTIVE SUMMARY** ......................................................... 8

**CHAPTER 1** ....................................................................................... 11

**RESEARCH BACKGROUND** .......................................................... 11

1.1 INTRODUCTION .................................................................................. 11
1.2 OBJECTIVES OF THE RESEARCH ..................................................... 15
1.3 BENEFITS OF THE RESEARCH ......................................................... 15

**CHAPTER 2** ....................................................................................... 16

**WEB BASED PROJECT MANAGEMENT** ............................................... 16

2.1 INTRODUCTION .................................................................................. 16
2.2 CURRENT TECHNOLOGIES IN PROJECT MANAGEMENT .................... 16
2.2.1 PRIMAVERA PROJECT PLANNER ................................................... 16
2.2.2 RA AUTOMATION ENGINE ............................................................ 18
2.2.3 VISUAL C++ .................................................................................. 19
2.2.4 COMMON GATEWAY INTERFACE (CGI) PROGRAMMING .............. 20
2.2.5 PERL ............................................................................................ 21
2.2.6 MICROSOFT INTERNET INFORMATION SERVER (IIS) ................. 22
2.2.7 ACTIVE SERVER PAGES (ASP) ....................................................... 23
2.2.8 OPEN DATABASE CONNECTIVITY (ODBC) .................................... 24

**CHAPTER 3** ....................................................................................... 26

**ENGINE MODEL** .............................................................................. 26

3.1 ENGINE COMPONENT ......................................................................... 26
3.2 THE ENGINE MODEL .......................................................................... 26
3.3 THE ENGINE CORE MODEL .............................................................. 27
3.4 ENGINE COMPONENT ARCHITECTURE .......................................... 31
3.5 SUMMARY OF THE ENGINE COMPONENT ...................................... 32

**CHAPTER 4** ....................................................................................... 34

**INTERFACE MODEL** ........................................................................ 34

4.1 INTERFACE COMPONENT .............................................................. 34
4.2 COMPONENTS OF INTERFACE ....................................................... 36
4.3 SUMMARY OF THE INTERFACE COMPONENT ............................... 36

**CHAPTER 5** ....................................................................................... 38

**ANALYZER MODEL** ......................................................................... 38

5.1 ANALYZER COMPONENT .............................................................. 38
5.2 SYSTEM ARCHITECTURE OF ANALYZER COMPONENT ................. 39
LIST OF FIGURES

FIGURE 1: PRIMAVERA PROJECT PLANNER 3.0 INTERFACE [5] ................................................................. 17
FIGURE 2: RA CLASS DIAGRAM [3] ........................................................................................................ 18
FIGURE 3: VISUAL C++ 5.0 INTERFACE [37] .......................................................................................... 19
FIGURE 4: COMMON GATEWAY INTERFACE ARCHITECTURE ............................................................ 20
FIGURE 5: MICROSOFT MANAGEMENT CONSOLE [38] ........................................................................ 23
FIGURE 6: ACTIVE SERVER PAGES ARCHITECTURE ............................................................................... 24
FIGURE 7: ODBC ARCHITECTURE ............................................................................................................ 25
FIGURE 8: ENGINE CORE MODEL IN EXPRESS-G [3] ............................................................................ 28
FIGURE 9: OBJECT REQUEST BROKER ARCHITECTURE [3] .................................................................. 29
FIGURE 10: COM-BASED AUTOMATION [3] ............................................................................................ 31
FIGURE 11: DISTRIBUTED COMPUTING ARCHITECTURE [3] .................................................................. 32
FIGURE 12: THE MODEL FOR THE INTERFACE COMPONENT .................................................................... 35
FIGURE 13: SYSTEM ARCHITECTURE OF ANALYZER ........................................................................... 39
FIGURE 14: SYSTEM ARCHITECTURE OF IGPMS .................................................................................. 43
FIGURE 15: ENGINE COMPONENT MAIN SCREEN [3] ............................................................................. 47
FIGURE 16: PROJECT TEMPLATE SELECTION [3] ................................................................................... 48
FIGURE 17: ENGINE COMPONENT MODEL IN AUTOCAD [3] .................................................................. 48
FIGURE 18: ENGINE OBJECT BROWSER [3] ............................................................................................ 49
FIGURE 19: ENGINE OBJECT EDITOR [3] ............................................................................................... 49
FIGURE 20: ENGINE ACTIVITY BROWSER [3] ....................................................................................... 50
FIGURE 21: ENGINE ACTIVITY EDITOR [3] ............................................................................................ 51
FIGURE 24: ENGINE PROJECT MONITOR [3] ....................................................................................... 54
FIGURE 25: ENGINE CONSTRUCTION SEQUENCE SIMULATOR [3] ....................................................... 54
FIGURE 29: DAILY RESOURCE ALLOCATION [3] .................................................................................... 56
FIGURE 30: RESOURCE ASSIGNMENT [3] .............................................................................................. 57
FIGURE 32: SCHEDULE RECOVERY II [3] ............................................................................................... 59
FIGURE 33: NEW DAILY CASH FLOW [3] ............................................................................................... 60
FIGURE 34: NEW CUMULATIVE CASH FLOW [3] ................................................................................... 60
FIGURE 35: GPMS HOME PAGE ............................................................................................................. 61
FIGURE 36: INTERFACE COMPONENT HOME PAGE ............................................................................ 62
FIGURE 37: DISPLAYING A SCHEDULE IN INTERFACE COMPONENT .................................................. 63
FIGURE 38: DISPLAYING FILTERS FOR A PROJECT ............................................................................... 64
FIGURE 39: INTERFACE COMPONENT DISPLAYING PROJECT INFORMATION .................................. 65
FIGURE 40: UPDATING A PROJECT SCHEDULE WITH INTERFACE COMPONENT ........................... 66
FIGURE 41: INTERFACE COMPONENT SHOWING THE UPDATED SCHEDULE ...................................... 67
FIGURE 42: ANALYZER HOME PAGE .................................................................................................... 68
FIGURE 43: EVOLUTION REPORT OF A PROJECT .................................................................................. 69
FIGURE 44: ACTIVITIES FINISHED SINCE LAST REPORT ....................................................................... 70
FIGURE 45: SELECTION OF THE FIRST AND THE SECOND TIME FRAMES AND A REPORT ............... 71
FIGURE 46: ANALYZE TWO DIFFERENT TIME FRAMES OF A PROJECT ............................................. 72
FIGURE 47: SECURITY FEATURE OF PROJECT SUBMISSION ............................................................... 73
FIGURE 48: PROJECT SUBMISSION HOME PAGE .................................................................................. 73

6
FIGURE 49: BROWSE PROJECT REPOSITORY BEFORE SUBMITTING A PROJECT .......................... 74
FIGURE 50: SUBMIT A ZIPPED PROJECT (E.G. FPS1) ................................................................. 74
FIGURE 51: FPS1.zip UPLOADED TO THE PROJECT REPOSITORY .............................................. 75
FIGURE 52: UPLOADED PROJECT FPS1.................................................................................... 75
FIGURE 53: PERSONAL GANTT VIEW IN MICROSOFT PROJECT CENTRAL [33] ......................... 77
FIGURE 54: WEBSTER FOR PRIMAVERA [34]............................................................................. 78
EXECUTIVE SUMMARY

Project management systems are computer programs designed to help project managers successfully manage a variety of project activities. The beginning of these systems are usually identified with the introduction of Program Evaluation and Review Technique (PERT) [1] and Critical Path Method (CPM) [1] in the 1950s. With the personal computer revolution starting from the eighties, now there are many mature commercial packages of such project management systems. These PC or LAN based project management systems such as Primavera™ [5], SureTrak™ [35], Microsoft Project™ [30], Expedition™ [36] and Timberline™ [18] have made life easier for project managers. However, the access to these software is limited and these software are not easy to use for ordinary people without extensive training. Generally, once the project has started with an original schedule, the actual field data about schedule and cost have to be recorded on paper before they are passed through one office to another before reaching the scheduler for updating the project schedule weekly or monthly. As modern construction projects are dynamic and complex in nature, the changes that occurs during the construction process are inevitable. It is not easy to notify all the participants who will be affected by these changes in a timely manner, especially if they are in different companies and located in different countries. Current practice in communicating and resolving these schedule and cost changes is far from satisfactory therefore causing completion delays, cost increases as well as a lot of controversies despite the help of computer project management systems. Thus, in the current practice the information flow is not dynamic. Therefore there exists an information flow problem in project management that needs to be solved.

The Information Super Highway offers a solution to this information flow problem. The Internet, especially its most organized section: the World Wide Web, is one of the fastest growing, most exciting technologies in today’s world. With an easy to use graphical Web browser, a user can access a wealth of information free or almost free (except for the fee for connection). One
The other major problem with large and global projects is that a large number of information systems are involved. So an information-sharing model for the integration of design and construction product and process information is necessary. For example, a construction project information may consist of drawings e.g. AutoCAD™ [4], schedules e.g. Primavera Project Planner™ [5], reports e.g. Crystal Report™ [6] and databases e.g. MS Access™ [7] or MS SQL Server™ [8]. Project planning tools must incorporate the product and process data, which may involve a lot of these information systems. The systems deployed for different aspects of the same project may be based on different algorithms and different data structures. Collaboration of a wide array of information systems is needed for the management of large-scale engineering projects. The thesis describes the attempt to solve this problem by developing a system that provides a common language of communication between these information systems.

The other problem that project managers encounter while planning large-scale A/E/C projects is that these projects may involve as many as 300 different multiple organizations [10]. The complexity of these projects is likely to keep growing. The project information and the amount of data transaction among project participants tend to expand substantially. Integration of project information becomes a significant issue for the management of large-scale engineering projects. The communications across the various sub-disciplines of an A/E/C project is inefficient and ineffective due to the inflexibility of the current data exchange. The misinterpretation of documents and drawings also leads contractors to employ inappropriate construction methods, set up infeasible schedules, waste resources and misestimate project cost. An interoperable integrated
project management system for A/E/C product and process information, based on the system developed by Yee-Sue Chiou [3], is also proposed in this thesis. This information-sharing model augments the Industry Foundation Classes (IFC) [11] and STEP [12] with object oriented methodologies and software interoperability concepts, establishing a common language for the product, process and resource information through the entire life cycle of a A/E/C project.

This thesis describes an Integrated Global Project Management System (IGPMS®). IGPMS has an interface component that allows users to access the project schedule residing on a web server, update the schedule, recover the schedule by reallocating resources and view the updated schedule through a web browser at any time and from anywhere. IGPMS has an engine component that allows the user to view the product, process, resource and cost information for a project from a web browser. IGPMS also has a component called analyzer, which can analyze two time frames or the whole evolution of a project allowing for the generation of reports by the users and gives the user an insight on the actual performance of a project.
CHAPTER 1

RESEARCH BACKGROUND

1.1 INTRODUCTION

Project management is the planning, organizing, and managing of tasks and resources to accomplish a defined objective, usually with constraints on time and cost. A simple project can be as simple as a list of tasks and their start and finish dates written on a piece of paper. A complex schedule, on the other hand, might deal with thousands of tasks and resources and a project budget of billions of dollars. As the project becomes more complex, so are the requirements of project management system.

The process of computerizing project management began in the 1950s. Dupont Corporation and Remington Rand, in an effort to improve project-scheduling techniques, developed a scheduling system called the Critical Path Method (CPM) [1]. CPM is a mathematical model that calculates the total duration of a project based on individual task duration and logical relationships, and identifies which tasks are critical to the on time completion of the project. This model is the fundamental scheduling method used in project management software system today. During the 1950s, the United State Navy developed the Polaris Missile Project; a submarine based weapons system. Lockheed, the primary contractor on the project, created the Project Evaluation and Review Technique (PERT) [1], which uses statistical probabilities to calculate expected duration. Three time estimates are required with PERT; most likely, pessimistic, and optimistic. However, multiple time estimates required for PERT are cumbersome to use and estimates produced are more difficult to manage, since project duration can be shown to be correct only 50 percent of the
time. Due to this difficulty, PERT is not as widely used as CPM.

In a separate development of project management system, the Gantt chart schedule was developed by Henry L. Gantt about 1900 [13]. It is used to graphically represent activities across a time scale. Initially called bar chart, these graphic representations have since been renamed Gantt Charts in honor of the inventor of the system. Gantt chart has been widely used as a visual aid in planning and scheduling. Project activities or tasks are shown graphically as bars against a horizontal time scale. As progress is made, the bars can be updated to show the status of each task and a vertical line can be drawn to show the current point in time. By comparing the progress indicated on the horizontal bars to the line showing the current point in time, users can readily visualize whether tasks are on schedule, ahead of schedule, or behind schedule. Most project management system software packages on the market today integrate at least CPM method and Gantt chart. The Primavera Project Planner 3.0 version also integrates PERT into scheduling calculation. Most of these systems are designed for desktop computers, some have Local Area Network (LAN) version. Since they are designed for personal computers, these systems do not work on UNIX or Macintosh machines. This feature has limited the availability of the system to certain machines, or certain office buildings that are not accessible from other geographical locations. Each separate PC has to have a copy of the software, which occupies a sizable hard disk space on each machine. For updating a project schedule, all the copies on separate machines have to be changed. Even if the system runs on a LAN system. The objective of this thesis is to break this barrier, reduce the hardware and software requirements for project management system and let more people have access to the latest project schedule and update it through Internet.

Design-build is a unique trend in the delivery strategy of civil engineering projects. It reduces burden on the owner as in this case only one party is wholly responsible for the entire delivery process of the project. But the delivery process, planning and coordinating activities become more complex and difficult. Therefore a design evaluation criterion has to be established and the available resource for construction is to be monitored constantly. Li and Peña-Mora [14] proposed a methodology to achieve robust planning and control where unforeseen changes can be absorbed without major disruption in the plan. To develop this integration method, they have combined the application of the concepts of axiomatic design, concurrent engineering, GERT and system dynamics. While these techniques can achieve robust control on design-build projects, the
The problem related to the vastness and the global nature of these projects still stands. When a design-build project is large and globally based, it becomes very difficult to manage the different installations. So in addition to robust control in planning, global management of such project becomes highly necessary.

The Internet is a space, where you can gather, disseminate, and exchange all types of information. WWW stands for "World Wide Web." The WWW project, started by Tim Berners-Lee while at CERN (the European Laboratory for Particle Physics), seeks to build a "distributed hypermedia system." In practice, the web is a vast collection of interconnected documents, spanning the world. The web was conceived so that, computer users could have a single, unified means of accessing hypermedia documents from anywhere on the Internet. The WWW uses the Hyper Text Transfer Protocol (HTTP). This protocol is the primary protocol for distributing information on the web. It is a relatively simple, flexible, higher level protocol that runs on top of Transmission Control Protocol/Internet Protocol (TCP/IP). The browsers display documents that are typically in Hyper Text Markup Language (HTML) format. HTML is similar to many other markup languages such as LATEX, RTF, and Frame. Markup languages define areas of text by special "tagging" the specific formatting associated with those areas. When viewing an HTML document there are highlighted text (Hyper Text) that will reroute the user to another section on the same document or another document that can either residing in the same server or a different web server thousands of miles away. The web server can be anywhere as long as there is a connection to the Internet. The Hyper Text that the user clicks on to trigger the browser to jump from one web location to another is called the Universal Resource Locator (URL). The web supports hypermedia and multimedia. Hypermedia is a combination of hypertext and multimedia, where hypertext refers to the linking of related information and multimedia refers to using different types of data to represent information. In the early stages of web most of the contents are static HTML documents for publishing information purpose, that is why newspapers, magazines are among the first commercial users of the web. With the introduction of the Common Gateway Interface (CGI), computer users can fire up an application with resides in a remote web server and receive dynamic results based on user input. This feature has enabled a new class of specialized web applications that exploit the Internet's dynamic, interactive nature. Dynamic web pages are not built from ordinary HTML text files; instead, small programs or scripts create the HTML source text, which a web browser then displays to the user. Because scripts generate
HTML documents on the fly, they are capable of incorporating information which changes or which cannot be determined in advance. They can also be used to solicit and interpret user-supplied data, retrieve requested information, or produce content, which has been customized for a particular user. After new web browsers incorporate security features to protect transmitted information and the introduction of firewall to block unauthorized access, banks and financial institutions are moving onto Internet at an unprecedented rate.

In the present scenario, the globalization of construction industry has revolutionized the project management system. When a project is large and various facilities and installations are located at different parts of the world, it becomes difficult to notify all the participants about the change in schedule or any other decision made on the project. The proliferation of World Wide Web offers a solution in this context as through the internet a user can access a wealth of information any time, any place with no third party involvement. One problem with such large and global projects is that a large number of information systems are involved. So an information-sharing model for the integration of design and construction product and process information is necessary. For example, a construction project information may consist of drawings (e.g. AutoCAD), schedules (e.g. P3) and databases (e.g. MS Access). Project planning tools must incorporate the product and process data, which may involve a lot of information systems.

So, collaboration of a wide array of information systems is needed for the management of large-scale engineering projects. The systems deployed for different aspects of the same project may be based on different algorithms and different data structures. Large-scale A/E/C projects may involve hundreds of multiple organizations. The complexity of these projects is likely to keep growing. So the project information and the amount of data transaction among project participants tend to expand substantially. So integration of project information becomes a significant issue for the management of large-scale engineering projects. The communications across the various sub-disciplines of an A/E/C project is inefficient and ineffective due to the inflexibility of the current data exchange. The misinterpretation of documents and drawings also leads contractors to employ inappropriate construction methods, set up infeasible schedules, waste resources and misestimate project cost.
1.2 OBJECTIVES OF THE RESEARCH

This research attempts to achieve robust planning of A/E/C projects by introducing web based project management tools and components to the existing project management technology. The research also hopes to achieve software interoperability through object oriented approach amongst different project management software. The research also aims at providing project managers an analyzing and reporting tool for their projects.

1.3 BENEFITS OF THE RESEARCH

This research will help project managers overcome the handicap of not being able to plan their projects online. Project managers can submit the project schedules, view and update a schedule from a web browser. They can also analyze the different schedules and view reports of the desired category. Project managers can also view the product, process, resource and cost information of their projects in a user friendly Graphical User Interface (GUI). Project managers can also monitor their projects running a simulation and generating reports and charts based on the simulation.
CHAPTER 2

WEB BASED PROJECT MANAGEMENT

2.1  INTRODUCTION

Project Management software has improved over the past decade and but the most important of all problems is the communications of project plans and keeping the projects up to date. Various solutions have been emerging over the years including client/server applications, email, fax but nothing has shown a higher potential than the Internet. A great part of this research was devoted to study these existing methodologies and different tools available for project management.

2.2  CURRENT TECHNOLOGIES IN PROJECT MANAGEMENT

The different existing technologies in Project Management were studied and evaluated before designing the Integrated Global Project Management System (IGPMS) and its components. Each component of IGPMS is described in detail in the following chapters. The different technologies considered for this research are:

2.2.1  PRIMAVERA PROJECT PLANNER

Primavera Project Planner (P3) is the high-end market leader of professional project management software product for more than 15 years. Primavera’s Concentric Project Management products
address the specific project management requirements of project-driven industries such as construction, engineering, telecommunications, utilities and petrochemical processing, representing approximately $650 billion annually in U.S. construction project spending and $21 billion annually in U.S. research and development spending [15]. Primavera Project Planner helps you manage resources, tasks and priorities across multiple-projects. P3 is ideal for managing important, large, or complex projects in industries requiring significant planning and control. It is designed specifically for project managers in environments that require multi-user and multi-project capabilities, in-depth analysis, cost control, electronic mail communications and sophisticated reporting. To organize projects up to 100,000 activities, P3 provides unlimited resources and an unlimited number of target plans. Massive data requires sophisticated, yet highly flexible organization tools: P3 supports 24 activity codes, 16 custom data items, 19 levels of sort, 28 levels of selection criteria and 31 activity calendars. The Primavera Project Planner for Windows Version 3.0 (P3 3.0) is used in this thesis. A typical project schedule in Primavera Project Planner is shown in Figure 1.

Figure 1: Primavera Project Planner 3.0 Interface [5]
RA is An OLE Automation Server and Scheduling Engine. RA is a significant step forward in the development of interfaces to project management functionality, processes, and project data for use in a wide range of business applications. As a server, RA enables access to Primavera Project Planner's advanced scheduling, leveling, resource and activity data, and cost calculations through an object-oriented automation server based on OLE 2.0 [16].

By using the RA Software Developer Kit (SDK), programmers can access, manipulate, and/or create project data with industry-standard programming tools, such as Microsoft Visual Basic, Microsoft Visual Basic for Applications, Microsoft Visual C++, Microsoft Test, Borland Delphi, Oracle Power Objects, Powersoft PowerBuilder and SoftBridge Basic Language. RA offers the flexibility and power to customize existing tasks, link existing applications, and build entirely

![RA Class Diagram](image)
new applications around Primavera Project Planner's functionality, data integrity, and business rule set. Corporate systems throughout the enterprise can utilize RA as a background process, feed data to be run through the scheduling engine, and then use the subsequent results. Figure 2 shows the RA class diagram.

2.2.3 VISUAL C++

Visual C++ is a Windows-based programming language based on ANSI C++. It provides an environment for creating Windows applications, a framework to begin with, and tools which allow you to custom-build your application. At the heart of Visual C++ is a library of Microsoft Foundation Classes (MFC) which are general-purpose classes of all types. Thanks to data isolation, we can use code, which has been written by someone else to support our own applications. The Integrated Development Environment (IDE) for Visual C++ is the Developer Studio where the programmer can access all the templates, tools, libraries, and even on-line help. In this thesis, Visual C++ 5.0 is used to write the code for accessing RA engine OLE functions. Figure 3 is a typical Visual C++ environment for developing C++ program.

![Figure 3: Visual C++ 5.0 Interface [37]]
The CGI (Common Gateway Interface) is the standard by which external programs (often called gateways or gateway programs) can interface with an HTTP (Web) server. Thus CGI programs act as gateways between the HTTP server and databases, or between the server and local programs or document generators.

Figure 4: Common Gateway Interface Architecture

Figure 4 illustrates the flow of data when a user accesses a CGI program. The solid line shows data flow using HTTP and CGI. HTTP transfers data from the client to the HTTP server, and back again. The CGI mechanism controls the flow of data from the server to the gateway programs (shown as the prism) and back again. These are called gateway programs because they generally act as gateways between the World Wide Web and server-side resources such as databases, feedback forms, click-able image-maps, and so on. CGI specifies how data are sent to the gateway program, and how data can be returned by the gateway program, through the server and back to the browser that originally sent the data. The basic mechanisms are:

- **Sending Data to the Gateway Program**

The CGI specifies how data are sent to the gateway program (as environment variables or as data read, by the gateway program, from standard input) and what data are sent (in general, all the data
sent by the client to the server, plus extra environment variables describing the status of the server).

- **Returning Data to the Client**

To return data back to the client program (the user's web browsers) the gateway program just writes data to its regular (standard) output. These data are sent back the client, after some processing by the server to ensure they have the correct message headers describing the data and the state of the transaction. The gateway program must write to standard output, before anything else, header messages explaining the MIME type of the data being returned by the program, this is the only way that the server can know what type of data are being returned by the gateway program.

### 2.2.5 PERL

Perl is the "Practical Extraction and Reporting Language." It was created by Larry Wall as a tool for writing programs in the UNIX environment and is continually being updated and maintained by him [21]. Perl combines many features of shell, awk, sed, C, and Basic programming in an easy-to-use scripting language that is ideal for text-file manipulation and general system management tasks. It is designed to assist the UNIX users with common tasks that are probably too heavy or too portability-sensitive for the shell, and yet too weird or short-lived or complicated to code in C or some other UNIX tool language. Unlike C, Perl is an interpreter-based (and not a compiler-based) language. Because of this, a Perl program does not need to be recompiled after a modification. The Perl "script" contains the actual commands (as written by the programmer), while the Perl "program" interprets these commands and executes the script. While the use of an interpreter avoids the need for recompilation, it does result in slower execution times compared with a similar program written in C. Since Perl is in the public domain, it may be obtained free-of-charge from the Internet such as ftp://prep.ai.mit.edu. Perl is available for UNIX, Mac, DOS, Amiga, and several other platforms. Perl is an excellent tool for text and file processing, this is why Perl has become the most popular programming tool for CGI scripts used on web servers.
The version used in this thesis is Perl 5.0 for DOS, installed in C:\PERL5.

2.2.6 MICROSOFT INTERNET INFORMATION SERVER (IIS)

Microsoft Internet Information Server (IIS) is an Internet file and application server included with the Microsoft Windows NT® Server operating system. IIS can be used alone as a Web server, or in conjunction with compatible technologies to set up Internet commerce, to access and manipulate data from a variety of data sources, and to build Web applications that take advantage of server script and component code to deliver client-server functionality. Because of its tight integration with Windows NT Server, IIS guarantees the network administrator and application developer the same security, networking, and administration functionality as Windows NT Server. Above and beyond its use of familiar Windows NT Server tools and functionality, IIS also has built-in capabilities to help administer secure Web sites, and to develop and deploy server-intensive Web applications.

The Microsoft Management Console (MMC) (see Figure 5) provides one utility for administrators to manage the network environment. With the Management console, network administration tasks can be customized for each administrator. Snap-ins for the MMC provide the ability to administer specific server components. One of the snap-ins for the MMC is IIS's Internet Service Manager, which gives administrators the ability to create Web and FTP sites, change default settings of a site, assign operators to tasks, start and stop sites, manage transactions, view statistics, manage tasks locally and remotely, and perform all other administrative tasks for the Web server or servers.
IIS also includes a Web-based administration tool that makes remote administration of Web servers possible through use of a Web browser. One of the most exciting features of IIS 4.0 is its powerful application development platform. Active Server Pages (ASP) technology, server components, search and index features, and new transaction-processing capabilities are making development of server-intensive Web applications one of the fastest growing components of Web development. Web developers need access to features that enable commerce, database access, personalization, and dynamic content generation on the Web. Intranet developers also need these features along with the ability to full-text search across all documents on their networks.

2.2.7 ACTIVE SERVER PAGES (ASP)

Active Server Pages is an open, compile-free application environment in which you can combine HTML, scripts, and reusable ActiveX server components to create dynamic and powerful Web-based business solutions. Active Server Pages enables server side scripting for IIS with native
An ASP script begins to run when a browser requests a .asp file from a Web server. The Web server then calls ASP, which reads through the requested file from top to bottom, executes any script commands, and sends a Web page to the browser. As the scripts run on the server rather than on the client, the Web server does all the work involved in generating the Web pages that is sent to browsers. The Web server does all the script processing, transmitting standard HTML to the browser. Server-side scripts cannot be readily copied because only the result of the script is returned to the browser. The users cannot view the script commands that created the page they are viewing. Figure 6 displays the ASP architecture.

![ASP Architecture Diagram](image)

**Figure 6: Active Server Pages Architecture**

### 2.2.8 OPEN DATABASE CONNECTIVITY (ODBC)

Open Database Connectivity (ODBC) is a widely accepted application programming interface (API) for database access. It is based on the Call-Level Interface (CLI) specifications from X/Open and ISO/IEC for database APIs and uses Structured Query Language (SQL) as its database access language. The ODBC interface makes it possible for applications to access data from a variety of database management systems (DBMSs). The ODBC interface permits maximum interoperability—an application can access data in diverse DBMSs through a single interface. Furthermore, that application will be independent of any DBMS from which it accesses data. Database applications call functions in the ODBC interface, which are implemented in database-specific modules called drivers. The use of drivers isolates applications from database-specific calls in the same way that printer drivers isolate word processing programs from printer-
specific commands. Because drivers are loaded at run time, a user only has to add a new driver to access a new DBMS. Figure 7 illustrates the ODBC architecture.

The ODBC architecture has four components:

- **Application.** Performs processing and calls ODBC functions to submit SQL statements and retrieve results.

- **Driver Manager.** Loads and unloads drivers on behalf of an application. Processes ODBC function calls or passes them to a driver.

- **Driver.** Processes ODBC function calls, submits SQL requests to a specific data source, and returns results to the application. If necessary, the driver modifies an application’s request so that the request conforms to syntax supported by the associated DBMS.

- **Data source.** Consists of the data the user wants to access and its associated operating system, DBMS, and network platform (if any) used to access the DBMS.

![Figure 7: ODBC Architecture](image-url)
CHAPTER 3

ENGINE MODEL

3.1 ENGINE COMPONENT

Engine component (originally called "SCHERE") was developed by Yee-sue Chiou and Feniosky Peña-Mora [3] with aim to achieve interoperability among software components and establish a common language for information exchange among project management systems. In this research the Engine component was studied and integrated with the Interface and Analyzer model, which will be discussed in the following chapters. Some of the modules in Engine component (SCHERE) were also modified to make SCHERE run on any computer where the software components like P3, RA, MS Access, MS Excel and AutoCAD are installed. Initially SCHERE could be only run on a specific server and this research removed this handicap and integrated the Engine component with the other components of Integrated Global Project Management System (IGPMS), namely Interface and Analyzer.

3.2 THE ENGINE MODEL

The Engine model aims to integrate information used in the design and construction processes for better management of large scale engineering systems (LSES). Different disciplines use individual software tools to create, manipulate and maintain their product and process information. For example, architects use Autodesk AutoCAD [4] for representing the physical characteristics of a project. Schedulers use Primavera Project Planner [5] for capturing the logic
and time needed to build the project. Estimators use Timberline [18] for developing the cost associated with the project. Accountants use special accounting systems, such as J.D. Edwards' Financial Suite [19], or customized database systems to track the expenditures on a project. As a result, diverse LSES models and software tools are being used by each individual specialty in the current community.

Object-oriented modeling technology can be used to encapsulate and manage these different types of entities in the product and process models. Thus the object-oriented modeling processes, which follow the standards of ISO STEP [12], ISO Building Construction Core Model (BCCM) [20], IAI [11] and Industry Foundation Classes [11], are used to develop the Engine model for integrating these diverse product and process models in infrastructure projects.

### 3.3 THE ENGINE CORE MODEL

The major role of the Engine core model is to integrate project information into the product, process and resource models. The foregoing section defines an applicable process for the integration among the product, process and resource models. Figure 8, in EXPRESS-G [22], gives an illustration of the interconnectivity among them.

First, top-level abstract entities, such as product, process, and resource are identified. In the second level, the entity groups, which are aggregated from individual domain elements, are derived from these abstract entities. Meanwhile, the entangled nature of A/E/C abstract entities are represented by the relationship between entity groups. Furthermore, three subtypes of resource usage object keep track of the resources consumed by entities of the product and process models; the cost account, in connection with the resource model, will perform the model-driven takeoff. After the creation of the Engine Core Model, any change to the original design will generate new construction schedules and estimate the equitable cost.
Implementation the Engine Component uses AutoCAD14™ [4] as the graphical engine, Primavera RA™ [5] as the scheduling engine, MS Access™ [7] and MS Excel 97™ [23] for cost and resources. Since the Engine Model does not deliberately overhaul the original design cycles to accommodate the computer-based information flow, A/E/C project participants need to adjust their discipline as little as possible. For instance, as the design team, which is composed of architects and engineers, finishes the schematic design in AutoCAD, all of the components within the design work will be systematically organized into the building component hierarchy of the
product model. As the construction team, following the scheme, completes the construction plan in Primavera Project Planner, the construction task hierarchy of the process model is formed at the same time. Then, the resource objects are created from the resource database according to the resource class template. Afterwards, all mappings among the Engine product, process and resource entities are generated in accord with the predefined object-oriented diagram in Figure 9.

The concept of the Engine model sheds new light on the integration of data and knowledge of an infrastructure project through the entire project life cycle. The three Engine sub-models manage the product, process and resource information for different aspects of project management. The Engine core model presents a systematic way, which conforms to the IAI IFC [11] development process, to integrate these three sub-models. As a result, the integration of project information among project team members and their computer systems can be achieved through the implementation of the Engine model.

![Figure 9: Object Request Broker Architecture [3]](image)

The attempt to integrate different A/E/C systems has been proposed since the 1980s [24], but the main integration mechanisms were limited to file exchange and shared software libraries. Due to
the lack of a generic A/E/C information representation mechanism, their information-sharing levels have been confined to geometrical and text-based information. Furthermore, the consistency of A/E/C information is hard to maintain and the synchronization of these systems needs a great deal of labor.

The evolution of object-oriented technology has promoted the evolution of software interoperability. The Object Management Architecture (OMA), defined by Object Management Group (OMG), provides a high level of abstraction [25] to allow the integration of a wide variety of objects. The core of the OMA is the Object Request Broker (ORB), which delivers requests to server objects and returns the result to the client objects. The software interoperability enables an object-oriented system to take advantage of a variety of software products through the standard software interfaces. A number of software developing tools, such as IONA Orbix [26], IBM DSOM [27] and Microsoft COM [28], allow the interoperability of different systems. Due to the current availability of interoperable software components, the Engine component has been implemented in Microsoft Component Object Model (COM). Unlike traditional application programming interfaces (API), a COM-based application, which is also known as Automation Engine, exposes its interfaces precisely and completely through COM. That is, COM defines a common way for software components to access software services [28].

The responsibility of an Automation Engine, which is deployed in a COM-based object model, is to describe its methods grouped by its COM interfaces and to maintain data of activated COM objects. As a result, other COM-based objects can navigate the internal COM model, retrieve data, and perform any internal methods of an Automation Engine. Figure 10 shows two kinds of automation methods:

- **In-Process Automation**, in which service objects are in the same process as the client application. Usually an In-Process server is implemented in a dynamic-link library (DLL). For example, the service of RA Engine to the Engine component is launched through this kind of automation [5].

- **Local Automation**, in which COM services are provided in a separate process running on the same machine. For example, AutoCAD provides its function to the Engine component.
through such a mechanism.

Figure 10: COM-based Automation [3]

3.4 ENGINE COMPONENT ARCHITECTURE

The Distributed Computing Architecture is illustrated at Figure 11. The user interface components and specific engineering components can be deployed on the client machine. Meanwhile, the middle-tier machines construct the application servers and the database servers are running on the back-end machines. Thus, it can be estimated that the interoperability will evolve into a fundamental underpinning for the integration of a full array of engineering software components. As a result, the collaboration of project team members will also be supported in a distributed computing environment. The implementation of the Engine Component demonstrates the cooperation of engineering design systems and project management systems. The interoperability among AutoCAD™, Primavera Project Planner™, RA™ Automation Engine, Microsoft Access 97™ and Microsoft Excel 97™ promotes the functionality of integrated project management system. Moreover, the deployment of the product, process and resource entities of Shimizu Smart Tower proves the feasibility of the interoperable integrated project management
3.5 SUMMARY OF THE ENGINE COMPONENT

Engine component integrates the product, process, cost and resource information and provides a common language of communication between the respective information systems. Engine allows project managers to visualize the project from a user friendly GUI. A user can look at the different drawings essential to a project. The user can also edit any part of the drawing by clicking on the object. Engine also allows users to access the schedule information of the project. The user can either update the schedule or take help of the in-built recovery strategies of Engine for example, crashing of activity duration, hiring of new resources or assigning overtime. Engine enables users to view the resource information of a project and allows the user to update this information. The most interesting feature of Engine Component is that it also allows the user to run a simulation of the project and view different kind of project reports and sequence of construction. Engine thus integrates physical, schedule and resource information of a project while planning and simulating a project.

Engine will be used mostly by project managers who make important decision with respect to
product, choice of process, cost and resource allocation. Project managers also can simulate the progress of a project and view the product, process, cost and resource information as a function of time and take decisions accordingly. Also, project managers who wish to recover a schedule can do so with the help of the Engine's different schedule recovery strategies. Thus Engine Component caters mostly to high-level project managers. A project schedule is maintained and updated by on-site contractors and schedule managers who do not take the high-level decisions regarding process, cost and resource but play an important role in the schedule of a project. They require that the schedule information is available to them in an interface where they can visualize the schedule and also make changes instantly to the schedule from the visual interface. The Interface Component caters to this need by providing schedule information in a visual interface through the web and also tools to update the schedule from the web based interface. The Interface Component is described in detail in the next section.
CHAPTER 4

INTERFACE MODEL

4.1 INTERFACE COMPONENT

Interface component (originally called "GPMS") was developed by Hong Chen and Feniosky Peña-Mora [2] with aim to develop a web based interface component by combining the power of existing project management system with the World Wide Web, making the project schedule available to anyone from anywhere. With the GPMS system, a very small management team will able to manage many projects in one office, running a single copy of this system on a single computer with the live field data coming in through Internet. The schedule will be updated instantaneously instead of weekly or monthly. In this research the Interface component (originally called "GPMS") was studied and integrated with the Engine and the Analyzer model. Analyzer model will be discussed in the following chapter. Initially GPMS could only open a specific Primavera project (e.g. "APEX") and show only 18 activities of the project. This research removed this handicap and now the Interface component can open any Primavera project and show all the activities for the project. Now with the new Interface component, project managers will be able to view filters defined in the Primavera project as well.

The Interface model is based on client/server architecture on the Internet. At the client side, the users can access the project schedule by going to the web page of the project through a web browser, update the duration of the activities, and recover the delayed activity by making existing resource work over time or hiring new resource. After the changes are made, this information is sent in HTML form input to CGI gateway programs on the server side. Then CGI scripts call
Visual C++ program to access the project schedule in Primavera Project Planner™ through the RA Engine™ to update the project schedule and reschedule all activities based on activity relationships and the user input. The latest schedule progress in HTML format is sent through C++ interface and CGI gateway program to be shown on the web by Java applets. In other words, the project schedule can be updated from anywhere at anytime by any authorized people from any machine that has a web browser running. The latest schedule is immediately available to all those who has a web browser such as Netscape Navigator™ or Internet Explorer™ by a single click, whether the machine he or she uses is a UNIX, or Macintosh or Windows PC. Since Primavera Project Planner™ only runs on PC, this alone is a big step forward for those who don’t use PC. Better communication is achieved with less direct contact with all the participants in the project. The cost savings in fewer management personnel, few computers and software purchases and fewer offices to accommodate them are enormous let alone there is no long distance communication charge.

![Diagram](image)

**Figure 12: The model for the Interface Component**
4.2 COMPONENTS OF INTERFACE

The Interface component is a web based project management system by combining the power of existing project management system with the World Wide Web. The Interface component will allow users to access the project schedule resided in a web server, update the schedule, recover the delay of the schedule by reallocating resources and view the updated schedule through a Web browser at any time from anywhere. The following section is a description of all the components involved in developing the Interface Component, including both server side and client side software.

Primavera Project Planner™ (P3) is the high-end market leader of professional project management software product for more than 12 years with an installed user base of more than 50,000. The Primavera Project Planner™ for Windows Version (P3) is used in the Interface component. RA™ is An OLE Automation Server and Scheduling Engine from Primavera Inc. As a server, RA™ enables access to Primavera's advanced scheduling, leveling, resource and activity data, and cost calculations through an object-oriented automation server based on OLE. On the server side, Perl scripts and Java applets are also components of the Interface. One of the most obvious advantages of the Interface component over other project management system is that no special proprietary software is required at the client side. All the user needs is a web browser running on any UNIX, Macintosh or Intel machine. The web page acts as the user interface to pass user input to the web server for processing.

4.3 SUMMARY OF THE INTERFACE COMPONENT

The Interface Component helps on-site project managers and contractors to view and update the schedule information from the web. The information is instantly updated and made available to all the participants of the project irrespective of their location. Interface also helps users to update a schedule in order to recover a schedule by adopting different strategies such as hiring new resources or increasing overtime. The Interface Component is user friendly and does not require extensive training on computers especially Primavera Project Planner™. An on-site contractor
need not know Primavera tools to update a schedule through the interface component. Interface thus helps in faster and efficient schedule access to the people who are more concerned with the scheduling part of a project.

While the current schedule is very important while updating or recovering a schedule by hiring new resource or increasing overtime, project managers also take into account the past performance, trends, anomalies, patterns and evolution of the schedule. While Interface component helps project managers and on-site contractors to view and update the schedule in real time, it does not have the functionality of displaying the evolution of a schedule over time and analyses of different time frames. Analyzer component caters to this need by providing a schedule analysis engine integrated with a web interface. The analyzer component is discussed in detail in the next section.
CHAPTER 5

ANALYZER MODEL

5.1 ANALYZER COMPONENT

Schedules of a project can be analyzed from a browser with the help of the Analyzer component. The Analyzer Component allows users to access schedules for which they have permissions. The Analyzer requires users to log on and a list of projects is then displayed to the user. The user can select any project for analysis. The user can view the reports based on the changes made to the schedule from one time period to the other. For example, the activities finished after the first time period. The user also has an option of viewing all the changes made to project on a category which he can select at run time.

The basic building blocks of Analyzer are Active Server Pages (ASP). An Active Server Page is basically an html file along with VB script and/or Java script codes. An active server page is used for server side database interactions from the web. A client can interact with the server database by sending requests to the active server pages, which reside on the server. The user sends the requests through a web browser. The active server page processes the request and then the active server page interacts with the database on the server side. The active server page sends the user security information and the query to be processed to the database. The results are then fetched by another active server page, which displays the results to the user through the web browser.

The Analyzer can be used for comparing different schedules of a project. Based on the comparison several reports can be generated by the Analyzer. The following types of reports
5.2 SYSTEM ARCHITECTURE OF ANALYZER COMPONENT

The basic components of Analyzer are Primavera Project Planner™, RA Engine™, ODBC™ and ASP™.

![System Architecture of Analyzer](image)

**Figure 13: System Architecture of Analyzer**

Primavera projects are incorporated to the Btrieve database by updating the DDF files (File.ddf, Index.ddf and Field.ddf) in the primavera project directory (typically C:\P3WIN\PROJECTS). After the DDF files are setup for the projects, an ODBC data source is created. For the System DSN, a suitable name for the data source, an appropriate driver (Pervasive ODBC-32) is selected. In the Pervasive driver menu the location of the DDF files and the Primavera projects are specified. After the ODBC System DSN is setup, SQL queries can be run from the web through Active Server Pages.
The building block of the reports that the Analyzer Component generates is a SQL query. When the user selects a particular type of report, a corresponding query is run in the server. The first and second time frames or versions of the same project schedule are passed as arguments. For example, the corresponding query for "Activities Added since Last Report" looks something like this:

```
"SELECT DISTINCT " + Second + "ACT.ActId, " + Second + "ACT.ActivityTitle, " + Second + 
"ACT.ES" + Second + "ACT.EF," + Second + "ACT.TotalFloat," + Second + "ACT.OrigDur
FROM" + Second + "ACT WHERE " + Second + "ACT.ActId not in (SELECT DISTINCT " + 
First + "ACT.ActId FROM" + First + "ACT)"
```

"First" and "Second" are two arguments specifying the first and second version of the schedule respectively. A typical report request may look something like this:

```
```

Here 12.asp corresponds to the ASP page that contains the query for the report category "Activities Added since Last Report". The first time frame is the schedule "ASPN" and the second time frame is the schedule "CONT". Both these schedules are versions of the same project (Edge Hill Continuing Care Community Project). For the queries to run successfully, it is absolutely necessary that these projects are incorporated into the Btrieve database. The projects can be incorporated into the Btrieve database by updating the DDF files from Primavera Project Planner.

This query is run on the server by the corresponding ASP script when the user requests a particular category of report and specifies the first and the second time frames. After the query is run on the Primavera ODBC data source, the result of the query is displayed in the form of an HTML table.
The Analyzer component helps users to analyze the schedules of a project and generate reports that help plan the subsequent modifications to the project schedule. The user also gets an idea about the trends and patterns of changes made to a schedule in a given category. The user can also study the activities and recognize the activities which might influence the project schedule more than other insignificant activities. The Analyzer component generates reports, which can be used for administrative purposes and also could be used by individual contractors to estimate the duration of the activities in the next revision. The Analyzer component is thus a necessary tool in the decision making process while scheduling a project.
CHAPTER 6

INTEGRATED GPMS MODEL

6.1 IGPMS

The integration of the three Components is achieved through the design of an architecture, which integrates the three without any major changes in their existing architecture. The architecture uses CGI technology for integrating all the three components into a single web based system. The product information for example, AutoCAD™ drawings is incorporated by ActiveX™ technology. The simulation results in the form of Excel™ charts and graphs are supported Internet Explorer™ and can also be viewed in Netscape™ after downloading required plug-ins. The system architecture of Integrated Global Project Management System is discussed in detail below.

6.2 SYSTEM ARCHITECTURE

The basic components of IGPMS are Primavera Project Planner™, RA™, AutoCAD™, MS Excel™, MS Access™, Visual C++™ programs, CGI scripts, ODBC™ and ASP™.

IGPMS integrates all the three components discussed before in a common framework rather than treating each of the architecture as different units. Figure 14 shows the integration framework for IGPMS. The Engine component is a Visual Basic program, which integrated the product, process, cost and resource information. The Visual Basic program interacts with Primavera the same way as the Interface component. In addition, it also interacts with AutoCAD, MS Excel and MS
Access using the COM technology.

Figure 14: System Architecture of IGPMS

The Interface component interacts with RA Engine through CGI and publishes project information to a web browser. The project information is displayed on a JAVA applet, so any JAVA enabled browser can access the project information and also can update projects through the web using the Interface component.

The Analyzer component interacts with Primavera Project Planner through Pervasive ODBC driver for Btrieve and publishes HTML reports to the web by analyzing different time frames of
the same project and also the analysis of the evolution of a project. The user can view the reports on a web browser and does not need to install any special software for this purpose.

6.3 SUMMARY OF THE INTEGRATED SYSTEM

The Engine Component presented in this thesis has demonstrated a conceptualized framework to improve the integration of project information. The underlying Engine model brings a new level of openness to A/E/C domain practitioners through object-orientation and software interoperability. The software interoperability of an integrated project management system allows the estimation of the project cost, the creation of new schedules and the analysis of design alternatives beyond the capability of a single A/E/C application. The project management practice across specific engineering or managerial areas reinforces the evaluation of the construction plan and supports the decision making process. Hence, the quality and efficiency of current project management are improved. Besides the concept of integrated project management, this thesis also sheds a new light on the development of collaborative computing systems. As distributed computing and interoperability technologies become mature, more dedicated A/E/C software components are likely to burgeon. Thus, a new computing system can easily be assembled, that allows the collaboration of architectural, engineering and construction information system, by a variety of software components through their standard object-oriented interfaces in a distributed computing environment [25].

The Interface component presented in this thesis has demonstrated that it is possible to manage a project schedule through Internet without disregarding legacy schedules, with a scheduling engine running in a web server, and Java applets within web browser as user interface. This would eliminate the need to install multiple copies of scheduling software on many computers while making the project schedule accessible to all people who have a web browser in their computer, whether it is a UNIX, Macintosh or Intel machine. The interactive nature of the web makes it possible for the users to update the schedule in the web server from any where at any time, thus greatly facilitate the collaboration among participants involved in a project who are located at different geographical locations. Interface allows users to access the project schedule resided in a Web server, update the schedule, recover the delay of the schedule by making the exiting resource
to work over time or hiring new resource, view the updated schedule and review the cost changes through a Web browser at any time from anywhere. With this functionality, schedule will be updated instantaneously instead of weekly or monthly.

The Analyzer Component presented in this thesis has demonstrated that it is possible for a project manager to analyze project schedules and view evolution of a project schedule through the web and take decisions based on the analysis. The analysis reports are in Industry Standard form and can help in the decision process. The Integrated Project Management System (IGPMS) presented in this thesis integrates the three components in a seamless manner. To the user IGPMS appears as a single system with three powerful project management tools.
CHAPTER 7

IMPLEMENTATION

7.1 ENGINE COMPONENT

The engine component can be downloaded from http://gpms.mit.edu/scherec/scherec.zip and should be unzipped to C:\SCHERECDirectory. The Engine program can be launched by running the executable file C:\SCHERECD\SCHERECD0114.exe. The Engine program requires that Visual Basic, AutoCAD, Primavera Project Planner, RA Engine, MS Excel and MS Access are properly installed.

7.1.1 PROJECT MANAGEMENT USING THE ENGINE COMPONENT

Once the project starts, project managers will launch Engine component to perform the project management tasks. Figure 15 shows the main screen of the system, and users are asked to select the geometrical engine, scheduling engine, accounting engine or dynamically loading according to their specific needs.
After the initialization of the selected automation engines, the Engine component will ask the user to select the project template for different types of project as Figure 16 shows. In current implementation, there are three different project templates available. The project manager may choose a 4F, 6F or 8F project template, whose construction plan and architecture design conform to the Shimizu Smart Tower project. The constructions of Engine classes follow the IAI IFC standard in the current version of the Engine component. In future versions, the Engine component may allow users to create ISO STEP and BCCM based classes. As a result, different implementations, which follow the same modeling format, will be able to share project data.
Then, project managers may click the object browser tab to create instances of objects in the Engine component model as demonstrated in Figure 17. The Engine component uses different colors to denote different types of class in AutoCAD™.
Thereafter, the two arrow buttons are employed to navigate the Engine's building component hierarchy and the object editor helps users to modify the attributes of the located entity as Figure 18 and Figure 19 show.

Figure 18: Engine Object Browser [3]

Figure 19: Engine Object Editor [3]
After the establishment of the Engine product model in AutoCAD, project manager can click the activity browser to load the project level construction schedule, which is in Primavera Project Planner format as Figure 20 shows. Meanwhile, the Engine Component will organize the activities in the schedule into the Engine construction task hierarchy as discussed before. Since the Engine building component hierarchy has been created in AutoCAD, the process entities in the Engine construction task hierarchy are mapped to the corresponding product entities in the same position in the Engine building component hierarchy.

![Figure 20: Engine Activity Browser [3]](image)

Project managers may highlight a specific activity and press the “Edit Activity” button to launch the activity editor. As Figure 21 illustrates, the activity editor allows users to redesign properties of activities, such as CPM/PERT, resource assignment and schedule recovery parameters. Afterwards, user may recalculate the updated schedule by pressing the “Reschedule” button as illustrated in Figure 22.
Figure 21: Engine Activity Editor [3]

Once the Engine product and process model has been created, the resource browser instantiates cost account, resource and resource use objects according to the product and process model, which have been created by the object and activity browser. Then, project managers can monitor resource and cost account information as shown in Figure 23.
Moreover, after the establishment of the product, process and resource models, the Engine Object Editor enables users to monitor the corresponding activity in the Engine Activity Editor by pressing the “Edit Activity” button in Figure 22.

Thus, this functionality allows project managers to specify, create, monitor and modify the connections between the product entity and its corresponding activities.
At last, the project monitor tab enables project managers to select the types of progress reports that he/she may want as shown in Figure 24. When project proceeds, the construction sequence simulator help user control the speed of progress and monitor the project at any specified time as Figure 25 illustrates. The Gantt chart of the project is displayed in Figure 26 to present the graphical schedule information. The daily cash flow diagram estimates the project expenditure on a daily basis as Figure 28 shows. Similarly, the cumulative cash flow diagram, as Figure 27 shows, demonstrates the spent budget so far. The daily resource allocation diagram is illustrated in Figure 29, displaying the percentage of each resource in a specific project time. The resource assignment diagram is helpful in the analysis of resource availability and resource leveling as Figure 30 shows.
Figure 24: Engine Project Monitor [3]

Figure 25: Engine Construction Sequence Simulator [3]
Figure 26: Gantt Chart [3]

Figure 27: Cumulative Cash Flow [3]
Figure 28: Daily Cash Flow [3]

Figure 29: Daily Resource Allocation [3]
7.1.2 SCHEDULE RECOVERY

The Critical Path Method (CPM) was designed to control both the time and cost aspect of a project [29]. However, the current CPM tools do not define a clear mechanism to update the change in the costs. The trade-off between time and cost is usually established by the experience of project managers.

Project delay, caused by differing site conditions, unknown subsurface condition, insufficient resource or the breach of contracts by subcontractors, is not uncommon in an A/E/C project. Thus, the estimation of cost to recover the schedule becomes one of the most important project management tasks. The Engine component, which is built on an integrated model, defines two functions to expedite a construction activity to recover the schedule and help project managers to
monitor the increased project cost.

The first method is to make existing resources work overtime, which allow project managers to specify the normal working hours per day, the overtime percentage of existing resource, the overtime salary ratio and overtime productivity.

![Activity Editor](image)

**Figure 31: Schedule Recovery I [3]**

The other method is to hire more new resources. These parameters of new hire and overtime are outlined by the attributes of an activity object. Thereafter, the project monitor of the Engine Component can present the new daily cash flow diagram and cumulative cash flow diagram after
the product, process and resource model incorporate these schedule recovery attributes.

Here the following scenario is provided to demonstrate the schedule recovery functionality of the Engine Component. During the excavation of the foundation, the underground conditions at the test borings are materially different from that shown in the construction documents. As Figure 31 shows, both the free float and total float of the “Excavation” activity are zero. That is, this activity is in the critical path and the differing site conditions may delay the entire project.
As Figure 31 illustrates, the Engine activity editor allow project managers to hire 5% more new resource with salary ratio 0.9 and make existing resource 10% overtime with salary ratio 1.20 for the activity, “excavation”. After pressing committing, the duration of the activity will be shortened to 4.45 days as Figure 32 shows. Figure 33 and Figure 34 illustrate that the additional $45,000 is in need to crash the activity for 0.55 day.
7.2 INTERFACE COMPONENT

The Interface component (GPMS) resides at the “gpms.mit.edu” web server. A user can go to the GPMS web site by typing the URL http://gpms.mit.edu/ in a web browser. Figure 35 is the home page of GPMS, which functions as a global project management system on the web.

7.2.1 SCHEDULING WITH INTERFACE COMPONENT

Figure 35 shows the homepage of GPMS. The user is prompted for a login and password. These information are the same as that of Primavera Project Planner.

![Figure 35: GPMS Home Page](image)

The user provides the login and password information and submits this information to the server and this information is validated by Primavera Project Planner through the RA automation engine and then valid users can log in to the GPMS. The following page loads after the login and
passwords are verified.

Figure 36: Interface Component Home Page

The user has two options to choose from (see Figure 36). First option is to select any project from the drop down menu and view the schedule. The second option allows the user to view the different filters according to the different activity codes defined for the project.

When the user selects the first option, the following page loads, which shows the information for the first 18 activities of the project. The information is shown in tabular as well as graphical form. The table in Figure 37 shows the details of the different activities such as "Activity ID", "Description", "Early Start", "Early Finish", "Duration", "Percent Complete", "Total Float", "Resource", "Budgeted Quantity", "Quantity to Date", "Units per Time Period", "Budgeted Cost" and "Cost to Date".

A Gantt chart shows in a graphical form the start and finish dates of each activity. The user can
view the next 18 activities or previous 18 activities by clicking on the appropriate button at the bottom of the page.

![Global Project Management System - Microsoft Internet Explorer](image)

<table>
<thead>
<tr>
<th>Description</th>
<th>Earliest Start</th>
<th>Earliest Finish</th>
<th>ACT ID</th>
<th>June</th>
<th>July</th>
<th>August</th>
<th>September</th>
<th>October</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Subcontracts</td>
<td>MAY 12 1997</td>
<td>MAY 12 1997</td>
<td>0003</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. General Requirements</td>
<td>JUL 22 1997</td>
<td>SEP 1 1997</td>
<td>0004</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Stowage</td>
<td>JUN 2 1997</td>
<td>JUL 11 1997</td>
<td>0005</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7. Concrete &amp; Floors</td>
<td>JUN 2 1997</td>
<td>JUL 25 1997</td>
<td>0008</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8. Masonry</td>
<td>AUG 1 1997</td>
<td>SEP 11 1997</td>
<td>0009</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10. Stone</td>
<td>JUL 1 1997</td>
<td>AUG 11 1997</td>
<td>0011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15. Damp. &amp; Waterproof</td>
<td>JUN 2 1997</td>
<td>JUL 11 1997</td>
<td>0016</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18. Roofing &amp; Sheet Metal</td>
<td>JUN 2 1997</td>
<td>JUL 11 1997</td>
<td>0019</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 37: Displaying a Schedule in Interface Component**

When the user selects the second option the filters defined for the project are shown in the form of a drop down menu. The user can select any of the filters for the project and view the activities in that filter category by clicking on the "Display Schedule" button.
Figure 38: Displaying Filters for a Project

Figure 38 shows the different filters defined for project AG21. The user can select any of the filters and click on the "Display Schedule" button to view the activities that belong to this category. Figure 39 shows that "Building E" is selected from the drop down menu for the project "AG21".
7.2.2 UPDATE ORIGINAL DURATION

The duration of each activity can be changed by typing directly into the Original Duration column in the schedule table shown on the web page, followed by hitting the return key on the keyboard after each entry. Users can make multiple changes. Every time when return key is hit, Java applet Schedule.java catches the Activity ID and the new Original Duration on the row where return key has been hit, passes the information to the system for updating.

Figure 40 displays the first 18 activities for the project "APEX". The duration for the activity
before update was 30 days. The user can change the duration to a new value (e.g. 10 days) and view the updated schedule by clicking on the "Submit" button at the bottom of the page.

![Global Project Management System - Microsoft Internet Explorer](image)

**Figure 40: Updating a Project Schedule with Interface Component**

Figure 41 shows the updated schedule after the schedule is recalculated according to the changed duration(s).

The Interface component does not allow the duration of completed activities and activities with constraints to be changed arbitrarily by the update option. Also, the Interface component does not allow the modification of the original duration of the activities, which have already started.
7.3 ANALYZER COMPONENT

The analyzer component resides at the “gpms.mit.edu” web server. A user can go to the GPMS web site by either typing the URL http://gpms.mit.edu/analyzer.html in his web browser or clicking on the link "Click Here to Launch Analyzer" from the Interface Component.

The Analyzer component loads in the same window as the Interface component. Figure 42 shows the Analyzer component home page.
Figure 42: Analyzer Home Page

Figure 42 shows the main menu of the Analyzer Component. The user has two options to choose from. The first option enables the user to view the evolution of a project and the second option allows the user to analyze two different versions of the same project.

7.3.1 VIEW THE EVOLUTION OF A PROJECT

The user can select any of the project group listed in the drop down menu and any type of report category and view the evolution of the project in that category. For example when you select the Project Group "Edgehill Continuing Care Community", a report category "Activity Finished Since Last Report" and click on the "Submit" button, an project evolution report page loads.
Figure 43: Evolution Report of a Project.

Figure 43 shows the project evolution report. The report is in HTML format and can be printed or saved for future reference.

The user can view the different data dates, descriptions for the various versions of the project. By clicking on the appropriate link the user can also view the report that shows the analysis of the two time periods chosen by the user.
### Figure 44: Activities Finished since Last Report

Figure 44 shows the activities that were finished between the first version (e.g. ASPN) and the second version (e.g. CONT) of the schedule for the project "Edgehill Continuing Care Community".

#### 7.3.2 ANALYZE TWO VERSIONS OF THE SAME PROJECT

The second option enables the user to select two different versions of the schedule and view reports. The following page loads when the user selects a project group and hits the "Submit" button.
Please Select Two Different Time Frames

Select the First Time Frame PN - ASPN - DD - 6/1/1997 - DESC - Edgehill Continuing Care Retirement


Please Select a Report Activities Finished Since Last Report

Submit  Reset

Figure 45 shows the drop down list boxes for First Time Frame, Second Time Frame and Report Category. The user can select the appropriate first and second time frame and a report category and hit "submit" button to view the report.

Figure 46 shows the report generated by the Analyzer. The report is in HTML format and can be printed or saved for future reference.
### 7.3.3 SUBMIT PROJECTS ONLINE

The user can submit zipped P3 Projects to the server using the Project Submission Page ([http://gpms.mit.edu/upload.asp](http://gpms.mit.edu/upload.asp)). Figure 47 shows the login and password window that opens up when a user sends a request for the project submission page. The login and password are verified by the Windows NT authentication program. Figure 48 shows the Project Submission Home Page after the login and password entered by the user are verified. After the user uploads the zipped projects to the server, he/she can unzip the projects by clicking on the appropriate link ([http://gpms.mit.edu/cgi-bin/runperl.exe/unzip.pl](http://gpms.mit.edu/cgi-bin/runperl.exe/unzip.pl)) on the Project Submission Page. The user can also view the existing projects listing by clicking on the appropriate link ([http://gpms.mit.edu/p3win/](http://gpms.mit.edu/p3win/)) on the Project Submission Page. Figure 49 shows the P3WIN directory content before project submission.
Figure 47: Security Feature of Project Submission

Figure 48: Project Submission Home Page
Figure 49: Browse Project Repository before Submitting a Project

Figure 50: Submit a Zipped Project (e.g. FPS1)
Figure 50 shows a zipped Primavera project (e.g. FPS1.ZIP) being submitted to the Project Repository through a web browser. Figure 51 shows the upload status of the project. Figure 52 shows the recently uploaded project FPS1 in the Interface component’s "Display Schedule" option list box. Interface component dynamically loads the projects from the P3WIN directory.

Welcome
Here is your upload status:

Destination: http://gpmsmit.edu/p3win/

Uploaded File: FPS1.zip
Physical Path: F:\P3WIN\PROJECTS\FPS1.zip
Size: 100228

Figure 51: FPS1.zip Uploaded to the Project Repository

Figure 52: Uploaded Project FPS1
CHAPTER 8

CONCLUSIONS

The Integrated Global Project Management System described in this thesis has demonstrated that Project Management is poised to take a leap in the Internet Age. Robust Project Planning can be achieved through a medium such as the Internet and an interface such as the web browser. IGPMS achieves the goal of planning projects through the web. IGPMS achieves software interoperability through its object-oriented approach. IGPMS provides users with a powerful web based project analyzer and reporting tool. IGPMS also achieves online viewing and updating projects thus facilitating Robust Project Planning.

8.1 FUTURE RESEARCH

There is a plenty of scope for further research as the technology in web based project management is evolving at a faster rate. Software vendors are announcing new web based project management tools and with Microsoft launching its Project 2000 [30] and Project Central software [9] with the new Windows 2000 Operating System [31], we will see a revolution in the area of web based project management.

8.1.1 WEB BASED ENGINE COMPONENT

Engine component can be made web based through ActiveX™ [32] technology. AutoCAD™, MS Access™ and MS Excel™ can be integrated to the web through ActiveX™.
8.1.2 INTEGRATION OF IGPMS WITH MS PROJECT 2000 AND MS PROJECT CENTRAL

Recently Microsoft announced its web based project planning tools Project 2000™ and Project Central™ [9] with the release of its new operating system Windows 2000™. Since Project 2000 and Project Central are Windows based and use Internet Information Server as the prime web server, it will be possible to integrate Project 2000 with IGPMS.

![Microsoft Project Central](image)

Figure 53: Personal Gantt view in Microsoft Project Central [33]

Figure 53 shows a personal Gantt chart in a web browser through Microsoft Project Central.

8.1.3 INTEGRATION OF IGPMS WITH WEBSTER FOR PRIMAVERA

Primavera Inc. has launched a new web based project management tool Webster for Primavera™ [34] designed specifically for team members and resources to see their assignments across multiple projects using only a Web browser. Figure 54 shows the GUI of Webster for
Primavera™ 2.5. Webster is the web-based interface to Primavera Project Planner. Individuals without formal project management training working on projects can report their work hours and progress on assignments directly to the project through Webster.

Webster and IGPMS both use Primavera Project Planner and so it would be possible to integrate Webster to IGPMS. Webster does not support online submission of projects and online analysis of project schedules. Webster integrated with IGMPS will have a robust interface to Primavera Project Planner and also tools like Engine, Analyzer and Online Project Submission. Webster will replace the Interface component of IGPMS in this new scheme of integration.
REFERENCES

19. JD Edward's Enterprise Software Inc. (December, 1999), http://www.jdedwards.com/
26. IONA (December 1999), "IONA Orbix", http://www.iona.com
27. IBM (December, 1999), "IBM DSOM", http://www.ibm.com
34. Primavera Inc. (December, 1999), "Webster for Primavera™ 2.5", http://www.primavera.com/products/webster.html
APPENDIX

SOURCE CODES OF IGPMS

For all the code used in this thesis please contact Prof. Feniosky Peña-Mora for further information. You can also download a zipped file containing all the source codes from the following URL.

http://gpms.mit.edu/igpms.zip

Visual C++ Projects

  analyzer
  compare
  evolproc
  evolution
  expand
  extract
  filter
  filtertargetfirst
  filtertargetnext
  filtertargetprev
  filterraup
  raup
  unzip
  updateddf

CGI/Perl scripts

  login.pl
  updateddf.pl
  unzip.pl
  ragetfilter.pl
  filter.pl
  expand.pl
  analyzer.pl
  evolproc.pl
Java Applets

Schedule.java
Schedule2.java

HTML files

index.html
raout
unzip
anaylze.html
analyzer.html
table.html
chart.html
but.html
but1.html
but2.html
but3.html

ASP Scripts

login.asp
process.asp
upload.asp
1.asp
2.asp
3.asp
4.asp
5.asp
6.asp
7.asp
8.asp
9.asp

83
Visual Basic codes

BcActivity.cls
BcEntity.cls
frmActivityEditor.frm
frmCashFlow.frm
frmConstructionSequenceSimulator.frm
frmCumulativeCashFlow.frm
frmDailyCashFlow.frm
frmDataLoader.frm
frmLoad.frm
frmObjectEditor.frm
frmObjectLoad.frm
frmSchedule.frm
frmShowSchedule.frm
frmSplash.frm
Main.frm
UserForm1.frm
HierarchyOperation.bas
Frm4Dmain.dsr

AutoCAD Drawings

1F.dwg
ansi.dwg
bea.dwg
col.dwg
elev.dwg
flo.dwg
Object_Building.dwg
Object_Building2.dwg
Object_Building3.dwg
Object_Building4.dwg
Site.dwg
STEEL_FRAME.dwg
StructureFrame.dwg

Microsoft Office Files

Resource.mdb