An Interoperable Integrated Project Management System for Architecture, Engineering and Construction Product and Process Information

by

Yee-Sue Chiou

Bachelor of Science

Taiwan University

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 1998

Signature of Author..
Department of Civil and Environmental Engineering
January 16, 1998

Certified by...
Feniosky Peña-Mora
Assistant Professor of Civil and Environmental Engineering
Thesis Supervisor

Accepted by...
Joseph M. Sussman
Chairman, Departmental Committee on Graduate Studies

An Interoperable Integrated Project Management System for Architecture, Engineering and Construction Product and Process Information

By

Yee-Sue Chiou

Submitted to the Department of Civil and Environmental Engineering on January 16, 1998, in partial fulfillment of the requirements for the degree of Master of Science

Abstract

This thesis presents an information-sharing model for the integration of design and construction product and process information. This information-sharing model augments the Industry Foundation Classes (IFC) and Standards for the Exchange of Product Model Data (STEP) 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 an engineering project.

The goal of this thesis is to provide a solution to achieve seamless information integration and sharing implemented through software, syntactic and semantic interoperability among engineering systems. Thus, a new integrated project management system is created on the basis of the proposed model. The interoperability features of this integrated project management system enhance the data exchange and information sharing across different software tools. As a result, it can be expected that the efficiency and quality of Architecture, Engineering and Construction project management can be improved.

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

Table of Contents

List of Figures

Chapter 1

Introduction

The management of large-scale engineering projects requires the collaboration of a wide array of information systems.[1] Each system, based on different algorithms and data structures, is deployed for separate aspects of the same project. The big challenge is how to incorporate interrelated project data from a broad range of implementations and how to manage the project data across the limitation of each single system.

This thesis aims to provide an information-sharing model to alleviate the current integration problems. This information-sharing model augments the Industry Foundation Classes (IFC) and Standards for the Exchange of Product Model Data (STEP) with object-oriented technology, establishing a common language for the product, process and resource information through the entire life cycle of an engineering project.

A new object-oriented system for the integration of product, process and resource information has been undertaken. The underlying object-oriented model represents design and construction information in different phases of project-delivery processes. By also incorporating the concepts of software interoperability, the system brings a new level of openness of software integration by exposing its interfaces to multiple engineering and

[1] Such as accounting systems, enterprise resource planning (ERP) systems, scheduling systems and financial systems.

project management systems. Consequently, the product, process and resource information managed by different software tools is integrated with the support of the interoperable infrastructure. Therefore, the quality and efficiency of integrated project management are promised to improve.

The reminder of this chapter is organized as follows: Section 1.1 outlines the current problems in information exchange within the A/E/C community. Section 1.2 documents the objective of this thesis, which defines an information-sharing model to alleviate the problems, discussed in Section 1.1. The organization of this thesis to develop the proposed information-sharing model is presented in Section 1.3.

1.1 Current Information Exchange within the A/E/C community

Large-scale A/E/C projects may involve as many as 300 different multiple organizations [see Peña-Mora, Sriram and Logcher (1995)]. Moreover, the complexity of A/E/C projects is likely to keep growing into the next century [see Frese and Waugh (1991)]. As the complexity increases, the elaboration of project information and the amount of data transaction among project participants tend to expand substantially. As a result, the integration of project information becomes a significant issue for the management of large-scale engineering projects.

In addition, every sub-discipline of an A/E/C project is constantly requested to perform his/her professional services on increasingly tight budgets and schedules.

However, design professionals are not used to take account of financial concerns or the constructiability of the design work. Therefore, numerous orders to change are often carried out to optimize the original design. Thus, the interaction and communication between the design and construction teams become the key to the success of an A/E/C project.

However, the communication across disciplines is inefficient and ineffective due to the inflexibility of the current data exchange. Today, almost all of the A/E/C related information is limited to paper-based documents, two-dimensional drawings or individual electronic files in different formats. These varied formats allow only the exchange of separate project information, such as basic geometrical data and project schedules.

Moreover, an identical project document or drawing is open to manifold interpretations. Due to the variety of different interpretations by multiple project participants, the current project management practice is suffering the loss of information. The misinterpretation often leads contractors to employ inappropriate construction methods, set up infeasible schedules, waste resources and misestimate the project cost.

Another significant issue in the current state of the industry is that any change to the original design expedites data inconsistency (historical) and horizontally (cross-disciplines) of an A/E/C project. For example, a typical order to change demands the correction of a handful of old shop drawings. The version management of the shop drawings emphasizes the historical consistence with existing design documents. All existing shop drawing should be consistent with the change order. Meanwhile, the new estimation for project cost and the new assignment of resources, caused by the change order, require the involvement of both design and construction teams. This cross-discipline consistence to project information forms the basis of the collaboration. The new estimation and new resource assignment should be consistent with the change order. Without a generic information-sharing model and a systematic mechanism, the updating of each corresponding data is uneconomical and error-bound. A fully integrated system to synchronize the transaction of the product, process and resource information is strongly needed.

1.2 Objective

This thesis attempts to define a new information-sharing model in support of the development of an integrated project management system. The proposed model organizes the product, process, resource and all the other project-related information into its corresponding entities. Meanwhile, the visualization of project information is achieved through the three-dimensional simulation in a computer-aided design system. Thus, the vision of project manager will not be limited to two-dimensional project data in specific areas of project management.

Using object-orientation and software interoperability, different types of project information, managed by different electronic systems, can be incorporated into the proposed model through the standardized software interfaces. As a result, the data exchange among the product, process and resource information become smooth and systematic. Thereafter, this conceptualized framework can be extended to different phases of project, from feasibility analysis to the delivery of project, to improve the efficiency and quality of project management through the entire project life cycle.

1.3 Presentation

This thesis is divided into six chapters. Chapter 2 examines the current modeling methodologies, especially object-oriented design and analysis modeling methods, needed for the development of the proposed information-sharing model. Chapter 3 discusses the current industrial information-sharing standards, including ISO STEP and IAI IFC, to examine these available object-oriented frameworks as they apply to the management of A/E/C project information. Based on the discourse in Chapter 2 and Chapter 3, Chapter 4 addresses the entities and their relationships of the proposed information-sharing model. Chapter 5 is a full discussion of the implementation strategies with an application of the system to the Shimizu Smart Tower project in western Tokyo, Japan.[2] Finally, Chapter 6 draws the conclusion and outlines the further directions of this research.

[2] All project data is contributed by the courtesy of Shimizu Corporation.

Chapter 2

Object-Oriented Information Representation

In the course of a project evolution, a large number of physical and logical entities are developed.[1] However, the mapping from these entities onto a conventional procedural modeling language can be extremely different and cumbersome. The complex real-world interlocked relationships consists of a great number of many-to-many relationships as well as integrated structure and behavior which make it impractical for non-object-oriented conventional methods to model the nature of an engineering project. On the contrary, the object-oriented methodologies present a systematic way to encapsulate these entities and their relationships, which will be discussed in later parts of this chapter.

This chapter covers the reasons why the object-oriented information representation is utilized by this research in Section 2.1. Therefore, the object-oriented methods, including the object-oriented design and analysis, are documented in Section 2.2 to demonstrate the modeling techniques for an infrastructure project. At last, the summary of this chapter, presented in Section 2.3, addresses the development of the object-oriented technology and software interoperability for an integrated project management system.

[1] For example, the building, different actors, resources and the mechanism of payment between the owner, general contractor and subcontractors are different entities documented in the project contracts.

2.1 Why Object-Oriented?

A variety of studies show that object-oriented modeling techniques present the following relevant characteristics to support the information representation of an A/E/C project management information system: (1) Project information can be organized into structured classifications [see Stumpf, Ganeshan, Gin and Liu (1996)].[2] (2) Explicit mapping between real world entities and the class in the object-oriented model can be developed [see N.K. Hong and S. Hong (1998)].[3] (3) Object-oriented model can be easily implemented in the high-level programming language, such as C++, SmallTalk and Java, requiring less effort from programmers [see Rumbaugh (1991)]. Moreover, software interoperability can be further deployed through standard object-oriented software interfaces [see http://www.omg.org (1997)].[4]

The reminder of this section presents what the object-oriented information representation is in Section 2.1.1 and the benefits of the object-oriented information representation in Section 2.1.2. At last, in Section 2.1.3, the object-orientation concepts are extended to the area of software interoperability.

2.1.1 What is Object-Oriented Information Representation?

Object-oriented technology has become the mainstream in the information technology industry [see http://www.rational.com (1997)]. Besides, the maturation of

[2] This structured classification provides a sound basis for the design of program data structure, and the structured programming techniques improve the quality and extendibility of the implementations.
[3] As a result, the clear mapping make the deployment of real world models effortless and more intuitive.
[4] See Chapter 5.

object-oriented theories has also made steady contributions to system analysis, programming, database design and a variety of related problem domains.

Object-oriented technologies are characterized by the following key concepts:

- Object

  The basic units of construction, be it for conceptualization, design or programming, are particular instances of an entity with individual identity [see Graham (1993)]. For example, a steel beam, a RC column, a worker, a construction activity and a project schedule can be considered objects. However, some basic characteristics of A/E/C projects, such as the color of an external wall or the dimension of excavation, are pure data values held by the objects. These characteristics do not have identity and are usually viewed as attributes, instead of objects.

- Class

  A class is a set of objects that share a common structure and a common behavior [see Booch (1993)]. For example, construction activities have duration, early start date, early finish date and other CPM/PERT parameters, and thus activities can be derived from an identical class, avoiding repetition in terms of shared characteristics.

- Attribute

An attribute defines a particular state of an object. Typically only the method of the owner object, which hold the attribute, has right to modify the data value of its attributes. For example, a red wall is described by its attribute, "red", while a steel frame can be described as "plate welded." Only the painting methods for a wall or the welding methods for steel frame can change the corresponding attributes.

- Method

A method demonstrates a behavior of an object. For example, marking, bolting, welding, re-bar fixing, scaffold forming and concrete pouring are common methods demonstrating the behaviors of RC beams and RC columns. Thus, the behaviors may be the action performed on the objects or done by the objects.

- Message

A message is the encoded information passed between objects and outside events. For example, the installation of a construction task can't be activated without receiving the message, "preceding activities finished," allowing the sequence of the construction activities to keep to the schedule.

2.1.2 The Benefits of Object-Oriented Information Representation

After the development of these concepts, discussed in Section 2.1.1, the major benefits of object-oriented information representation can be documented in the following areas:

- Encapsulation

  This term implies "information hiding." All detailed data structures of objects, except their specifications (interfaces), are hidden from the other objects in the object-oriented system. For instance, the lack of labor often causes the delay of project and increases the project cost. In an object-oriented system, the labor object in resource class may propagate the message "insufficient labor" to activity objects and cost account objects. The "reschedule" methods of activity objects and "estimate" methods of cost account objects are automatically executed. The labor object only need to take care of the format of its messages in conformance with the interfaces of corresponding objects, but do not need to know how the new schedule or the new project cost are calculated. This characteristics allows each object or class to develop its own efficient way to implement a concept without worrying about other objects accessing data if the interfaces are agreed it upon.

- Inheritance

  Inheritance permits a class to have the structures and behaviors of other classes within the upper side of its hierarchy. For example, because of inheritance, all beams and columns of a RC (Concrete-Reinforcement) building are implied to be made of concrete and reinforced by re-bars. The superclass "RC building" defines the basic implementation methods and attributes of both RC beam and RC column class. Therefore, RC beams and RC columns share the information and behavior defined by their upper class – "RC building."

- Polymorphism

  When one operation may take on different expressions in different classes, this is called polymorphism. For example, "Re-bar Fixing" is demanded for the formation of both steel and RC beams. Even though both classes utilize totally different underlying fixing techniques, the expression, "Re-bar Fixing," can still be used to denote different fixing procedures for both beams. On other words, an identical name is used for different beams to perform different operations.

- Reusability and extensibility

  Besides the robustness and stricter security coming along with the other benefits of object-oriented technology, the reusability and extensibility are also significant for an A/E/C project. As long as entities within an A/E/C system are defined in a systematic way, other similar projects can reuse the same entities with minimal effort. Furthermore, the relationship between existing entities and new ones can be obtained by applying the object-oriented mechanism, such as inheritance. As a result, the developing time and cost for A/E/C project management software systems will be greatly reduced.

These features, discussed above, of object-oriented information representation promote the understanding of the inborn A/E/C project structure. An object-oriented system would pass messages between different types of A/E/C entities, change the states of corresponding entities, trigger the internal methods pertaining to project management tasks through the object-oriented interfaces. That is, the intrinsic data structures of A/E/C software components are hidden from each other, avoiding unintentional interactions and unnecessary complexity. Thus, objects created by an A/E/C software system communicate only by message passing without understanding the complicated details of other software components.

2.1.3 Software Interoperability

Extending the reusability concept of object-oriented systems, the concept of "interoperability" has come of age. As object-oriented technology matures, emerging standards of the object-oriented methodology shed light on the interoperability among object-oriented applications. The interoperability, namely the collaboration among software components, promises to improve the current A/E/C practices by allowing software handling different types of data from the profession to be able to talk to each other through standardized software interfaces without redesigning the software structure.

Thus, many organizations, like the Object Management Group (OMG) [see http://www.omg.org (1997)], are devoted to integrating object-oriented technology among various object-oriented software standards, allowing interoperability of distributed objects. Nowadays two interoperability approaches, based on object-oriented notation, are widely regarded as the most promising standards to integrate current object-oriented methods [see http://www.omg.org (1997)]. These two approaches: the OMG's Common Object Request Broker Architecture (CORBA) [see http://www.omg.org (1997)] and Microsoft's Distributed Common Object Model (DCOM) [see http://www.microsoft.com/activeX (1997)], will be further discussed in Chapter 5.

In spite of the difference of CORBA and DCOM architecture, both of them provide a set of object-oriented software standards to achieve the interoperability among software components. The application of CORBA and DCOM to A/E/C systems provides a new way to integrate A/E/C software components and improve the data exchange between A/E/C computer applications through a communication standard. In order to take advantage of those features, the proposed object-oriented model follows those features.

2.2 Object-Oriented Methods

The purpose of object-oriented methods is to model the development of real-world systems. As a result, the design and analysis of the products and processes can take advantage of a variety of object-oriented modeling languages.

This section has three parts. Section 2.2.1 discusses the object-oriented modeling languages, which can be used for the modeling of an A/E/C project and examines the Unified Modeling Language [see http://www.rational.com (1997)], that was developed with the contributions of other leading modeling languages, as an example of object-oriented modeling languages. On the other hand, Section 2.2.2 presents EXPRESS and EXPRESS-G, which are the international modeling standards for engineering products [see http://www.iso.ch (1996)]. Chapter 3 will discuss the use of EXPRESS and EXPRESS-G by ISO STEP and IAI IFC, which are two most fundamental A/E/C modeling approaches [see Chapter 3].

2.2.1 Object-Oriented Modeling Languages

Object-oriented modeling languages are used to specify, document, create, test and visualize the object-oriented information representations of software system. Thus, the complicated software design process may exploit the functionality of object-oriented modeling language to construct the software architecture. Besides, high-level object-oriented programming languages, such as C++, Java and SmallTalk, simplify the implementation of the software blueprint defined by object-oriented modeling languages, allowing the mapping from information models onto executable software codes.

The reminders of this section presents one of the industry-standard modeling language, the Unified Modeling Language (UML). In an attempt to integrate existing modeling languages, Grady Booch (Booch's Model) and Jim Rumbaugh (Object

Modeling Technique) began their work on unifying the Booch and Object Modeling Technique in 1994. In the fall of 1995, Ivar Jacobson, the author of Object-Oriented Software Engineering (OOSE) method, joined this UML unification team. Their effort resulted in the creation of the Unified Modeling Language (UML version 0.9) in late 1996.[5] Although no leading object-oriented modeling languages dominate the software industry now, the UML 1.1 seems to be one of the most comprehensive object-oriented modeling tool currently [see Fowler and Scott (1997)].

The UML processes consist of the following diagrams to visualize the development of software architecture:

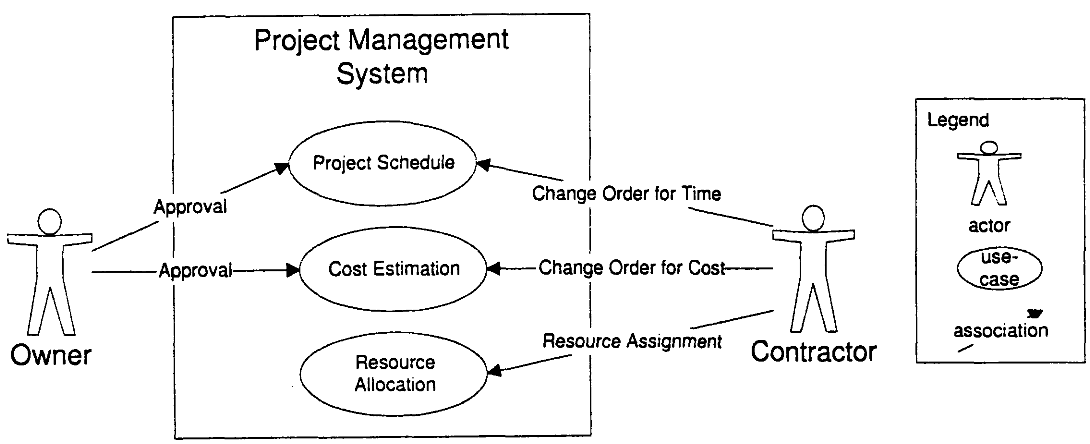


Figure 2-1 Use Case Diagram

- Use case diagram, which demonstrates the relationship between the functionality of a system and its potential users. Figure 2-1 shows the use case diagram for the construction of basic elements of a project management system.

[5] http:// www.rational.com/uml, UML Summary Version 1.1, 1997

- Class diagram, which show a class with the attributes and methods within the classes. Figure 2-2 shows a RC_beam in the UML notation.

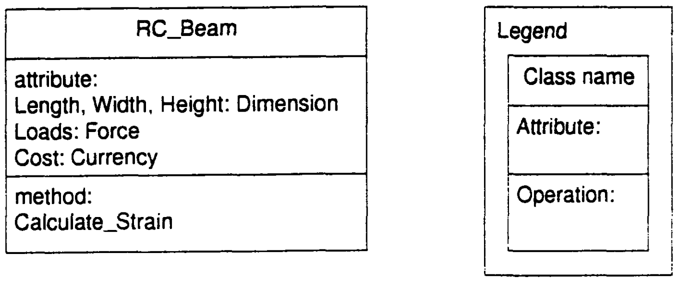


Figure 2-2 UML Class

- Behavior diagrams, which consist of the state chart diagram, the activity diagram, the sequence diagram and the collaboration diagram. These diagrams describe the dynamic parts of systems. The collaboration diagram is especially important to define the collaboration between different objects and their interaction. Figure 2-3 is a simplified collaboration diagram connecting a resource supplier, the project cost and the usage of that resource.

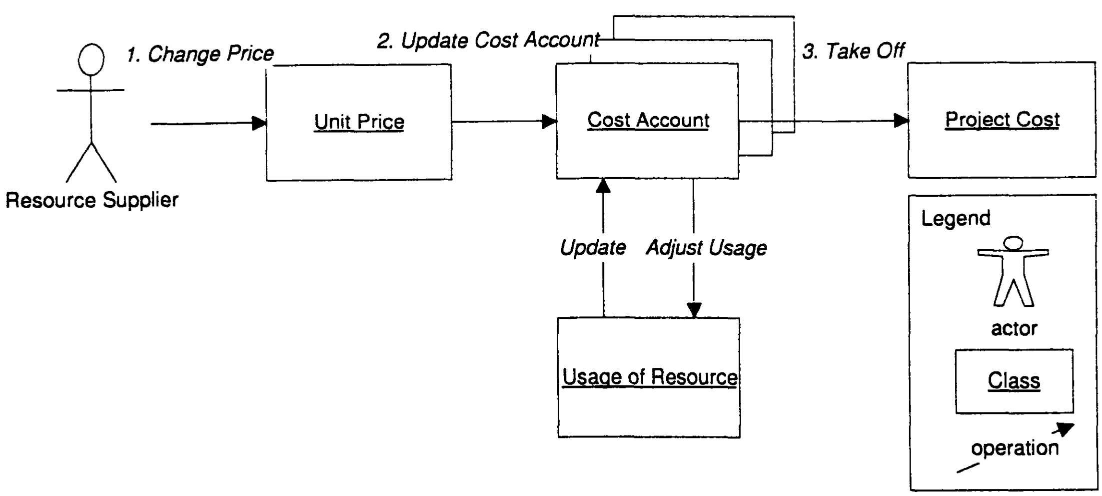


Figure 2-3 Collaboration Diagram

- Implementation Diagrams consist of the component diagram,[6] which augment the concept of interoperability of software components, and the deployment diagram, which outlines the runtime configuration of software processes and usually used for the final optimization of software performance [see http://www.rational.com (1997)].[7] Figure 2-4 illustrates the dependencies between software components of the management, planning and scheduling system and their interfaces.

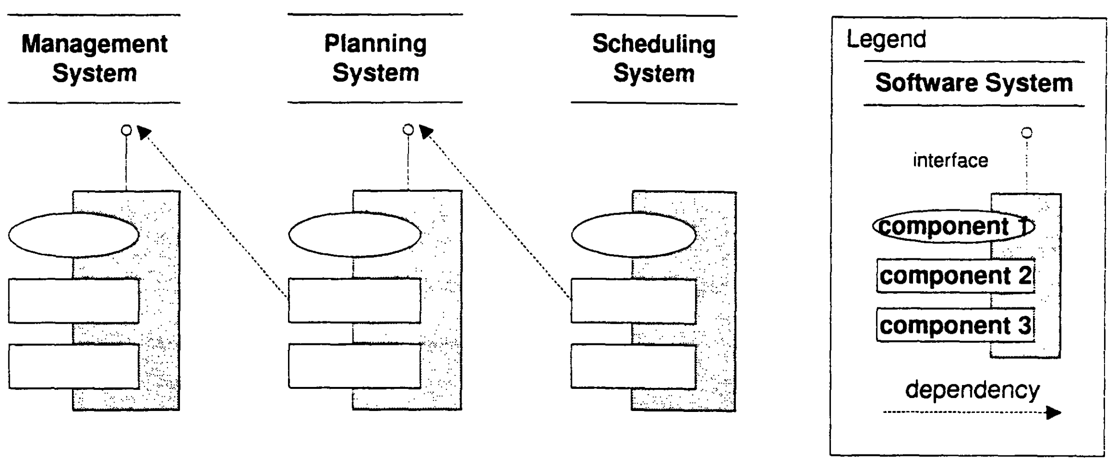


Figure 2-4 Component Diagram

In summary, the class diagrams deal with the static parts of systems, and the behavior diagrams describe the dynamic parts. Furthermore, the implementation diagrams include extensibility mechanisms for distribution, concurrency and software

[6] Chapter 5 will employ the component diagram to represent the relationship between software components. Please refer to Figure 5-1.
[7] The development of the proposed model does not adapt the UML deployment diagram. Please refer to http://www.rational.com for further detail.

interoperability.[8] The diagramming techniques of UML will be used extensively in this thesis to implement interoperability between the proposed system and existing computer-aided engineering programs.[9]

2.2.3 EXPRESS and EXPRESS-G

EXPRESS is a definition language used to specify the components of the engineering products in the Standard for the Exchange of Product Model (STEP) of the International Standard Organization. It is also part of the standard, ISO 10303 Part 11 [see Wix (1996)]. EXPRESS-G is the graphical notation of EXPRESS. Strictly speaking, both EXPRESS and EXPRESS-G should be regarded as definition languages rather than modeling languages. Unlike the conventional methods discussed in Section 2.2.2.1, EXPRESS and EXPRESS-G do not stipulate a standard development process and cannot represent any dynamic behavior of a system. The object-oriented modeling processes in EXPRESS and EXPRESS-G need to take advantage of other methodologies in order to overcome the deficiencies of their natures.[10]

However, all A/E/C physical and logical entities developed in this thesis will be described for the most part in EXPRESS and EXPRESS-G[11] rather than through the other conventional modeling methods, in conformance with international standards to enhance

---

[8] http://www.rational.com/uml, UML Quick Reference, 1997.
[9] Please refer to Chapter 5.
[10] For instance, Industry Foundation Class devises its own development process and the ISO Building Construction Core Model uses IDEF to model the dynamic behavior of A/E/C systems. Chapter 3 will discuss IDEF model and this topic further.
[11] Please refer to Chapter 4

communication while avoiding the overuse of proprietary technologies. As a result, all elaborate prototypes of A/E/C entities can share project information with other systems that comply with ISO STEP or IFC models. However, since neither EXPRESS nor EXPRESS-G does define any notation to describe the interoperability among software components, the UML method will be used to address the collaboration of software components.

The EXPRESS-G notation is utilized to present the major components of a building project in Figure 2-5 [see Appendix for the list of EXPRESS-G notation]. The Project entity is the supertype of Building and Site. The Building entity is the abstraction class of Building Components.[12] The Site supports the Foundation entity, and their relationship is one-to-one.

2.3 Summary

Current studies in information modeling provides the foundation to model the real world A/E/C project and to integrate separate A/E/C software components used by specific domains of engineering and managerial systems. The object-oriented design and analysis methods, especially the United Modeling Language and EXPRESS, allow for the development of a powerful information framework to enable the information sharing and interoperability among A/E/C software systems. Chapter 3 examines two existing A/E/C

[12] In EXPRESS-G, the (ABS) prior to the class' name implies that the type of this class is abstract. An abstract class is never instanced. All of its attributes can only be inherited by its inherited classes which are instanced.

information-sharing standards, which is derived from the object-oriented technology discussed in this chapter.

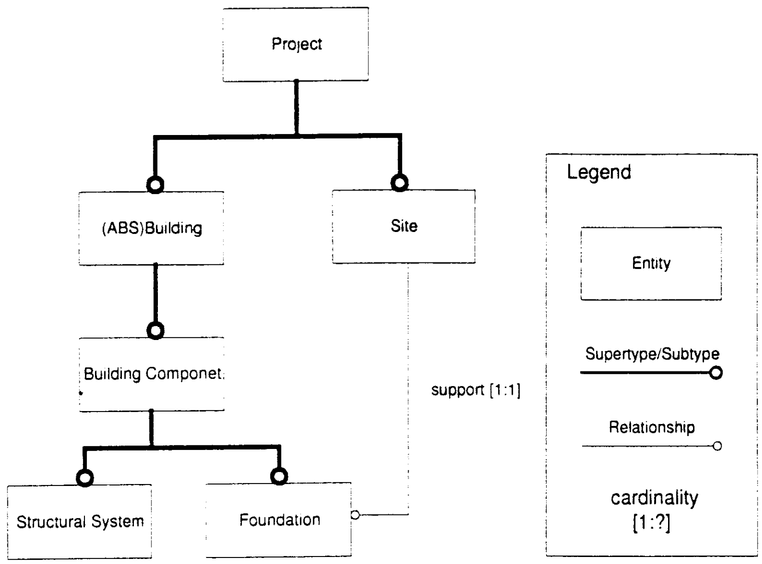


Figure 2-5 EXPRESS-G Diagram

Chapter 3

A/E/C Information-Sharing Standards

To date, no individual practice in separate disciplines can assert that it is persistently independent of industrial entities, commercial procedures, business rules, government statutes and electronic data processing. For example, on the technology side, structural analysis pertaining to automobile design and building design may be subject to similar principles; welding processes used in nuclear power plants and in the manufacture of satellites have several common properties; bolts in steel-frame buildings are likely be found in aircraft engines. Furthermore, computer-aided engineering has penetrated the design process and manufacturing process, and made the boundaries between industry sectors vague. Therefore, the information sharing among different industry sectors demands unified standards, allowing the interoperability of different computerized tools.

However, the lack of sophisticated information-sharing standards becomes the major barrier of data exchange within an industry sector and across industries. In the endeavor to alleviate the problem of inconsistent information-sharing between different domain practitioners, numerous large-scale international projects have been undertaken to accelerate the formation of information-sharing standards. This is to integrate industry-wide design and manufacturing information.[1] In the same manner, software vendors,

[1] For instance, a substantial standardization effort has been made by European projects, such as ESPRIT and JOULE.

research institutes and academic society have declared the need for standards in the emerging A/E/C information-sharing technology [see Froese (1996)].

The organization of this Chapter has three parts: Section 3.1 examines the international official standard - ISO STEP [see http://www.iso.ch (1996)]; Section 3.2 discusses the IAI IFC standard [see http://www.interoperability.com (1997)], which are created by industrial practitioners. Both of them are widely regarded as two most fundamental approaches for the A/E/C information modeling. At last, Section 3.3 documents the summary of this chapter and outlines the conformance of the proposed model to these standards.

3.1 ISO Standards

The International Organization for Standardization (ISO) is perhaps the most wide-ranging standardization federation in the world. More than 100 national standardization bodies organized the ISO, a non-governmental organization, in 1947. The formations of ISO's standards, which are international agreements, usually take years to reach the mutual consensus of ISO member countries due to their different agendas and objectives to support a particular national standard.[2]

The reminder of this section is organized as follows: Section 3.1.1 discusses the ISO STEP standards, which is a general-purpose modeling effort for engineering product. Furthermore, Section 3.1.2 outlines the ISO BCCM [see Wix (1997)], which augments the STEP modeling in a building project. Section 3.3.3 examines the IDEF0 method [see http://www.iso.ch (1997)], which is adopted as the process modeling tool for ISO STEP family.

3.1.1 ISO Standard for the Exchange of Product Model Data (STEP)

The well-known ISO Standard for the Exchange of Product Model Data (STEP) has been widely regarded as the most fundamental approach for describing the product design and manufacturing data throughout a product's life cycle [see Gann, Hansen, Bloomfield, Blundell, Grotty, Groak and Jarrett (1996)]. Not only does STEP give the definitions of product entities, but it also provides data exchange mechanisms and comes along with an official data definition language, "EXPRESS," for the modeling of engineering products.

STEP has gained success through some pilot projects in Europe[3], but the American counterpart of STEP[4], "the Product Data Exchanging using STEP," has not been widely accepted in the United States [see Gann, Hansen, Bloomfield, Blundell, Crotty, Groak and Jarrett (1996)].[5] The latest technology usually takes several years to get incorporated into an ISO standard. Thus, the U.S. industry tends to define their own

---

[2] http://www.iso.ch/infoe/intro.html

[3] For example, the Computer Models for Building Industry In Europe (COMBINE) project, which is conducted by Delft University of Technology (NL), Building Research Establishment (UK), Centre Scientifique et Technique du Bâtiment (FR), Technical Research Centre of Finland, VTT (FIN), has successfully developed an interface kit supporting STEP neutral file exchange

[4] Product Data Exchange using STEP is the American National Standard for STEP.

[5] Please refer to Information Technology Decision Support in the Construction Industry: Current Development and Use in the United States, by Science Policy Research Unit, University of Sussex, Brighton, UK

industrial standards based on the latest technology before they are officially accepted as ISO standards in order to satisfy demands of customers or overcome competitors .

STEP, still in development, is composed of individual published documents by a variety of ISO technical committee. The following list presents parts of the STEP documents at the International Standard (IS) level, which are closed related to the A/E/C industry and relevant to the proposed model in this thesis.

- Part 1 – Overview and Fundamental Principles: provides an overview for the STEP product representation and exchange [see TC 184/SC 4 ISO 10303-1 (1994)].
- Part 11 – EXPRESS Modeling Language [see TC 184/SC 4 ISO 10303-11 (1994)].[6]
- Part 21 – Clear Text Encoding of the Exchange Structure: defines the physical file formats for STEP data exchange in EXPRESS [see TC 184/SC 4 ISO 10303-21 (1994)].
- Parts 30 - 49 – Resource Schemata, which specify an application-independent general framework for the integrated resources that are documented in later parts of STEP [see TC 184/SC 4 ISO 10303-31 to 49 (1998)].
- Part 106 – Building Construction Core Model: will be further discussed in Section 3.1.2 [see TC 184/SC 4 ISO 10303-106 (1996)].
- Parts 200 - 299 – Application Protocols (APs): stipulate a variety of application protocols for the data contents defined by Resource Schemata, i.e. Parts 39-49, and

[6] Please refer to Chapter 2.

data exchange with other applications [see TC 184/SC 4 ISO 10303-200 to 299 (1998)].

3.1.2 ISO Building Construction Core Model (BCCM)

The concept of ISO BCCM is to form an A/E/C core model, which integrate the relevant parts of STEP, to model the products, processes and resources through the life cycle of a building project. Thus, the following tasks can be developed on the basis of ISO BCCM [see Wix (1997)]:

1. Interpretation of common requirements.
2. Specification of common data.
3. Development of a common framework.

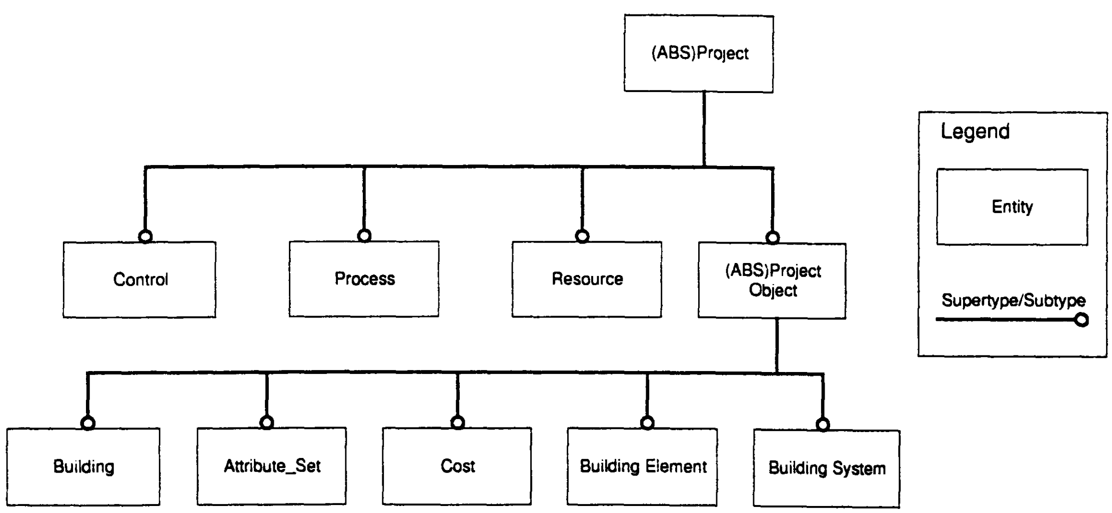


Figure 3-1 BCCM Core Model

As a result, STEP-compliant project data from different industrial sub-sectors is merged into ISO BCCM and the data exchange is achieved through STEP Application Protocols. Figure 3-1 shows the simplified schema of BCCM in EXPRESS-G.[7]

3.1.3 Integrated Computer-Aided Manufacturing Definition (IDEF)

ISO STEP adopts the EXPRESS language and the EXPRESS-G graphical notation to model entities. EXPRESS and EXPRESS-G provide a computer-interpretable unambiguous data definition. However, neither EXPRESS nor EXPRESS-G is designed to perform modeling of actions, activities and other dynamic behaviors within information systems. Consequently, the Integrated Computer-Aided Manufacturing Definition (IDEF0) [8] method is used to accomplish the dynamic modeling of ISO STEP [http://www.iso.ch (1997)].

The IDEF methods were originally developed by the U.S. Air Force Program[9] for Integrated Computer Aided Manufacturing in the 1970s. After receiving input from

---

[7] Presented at the ISO TC184/SC4 meeting in Chester, UK in March 1997.

[8] Federal Information Processing Standards Publication 183, issued by Computer Systems Laboratory, National Institute of Standards and Technology, U.S. Department of Commerce, Gaithersburg, Md. 20899, on December 21, 1993

[9] Hence, it is also called, "the U.S. Air Force IDEF Methods."

software vendors[10] and government agencies[11], IDEF has grown to a series of modeling methods.[12]

IDEF0 – Function Modeling Method uses the function box as shown in Figure 3-2 to simulate the input, output, control and mechanism of a dynamic system. IDEF0 techniques are widely used in government, industrial, and commercial sectors to model the activities, functions, mechanisms, underlying rules, and their relationships within information systems.

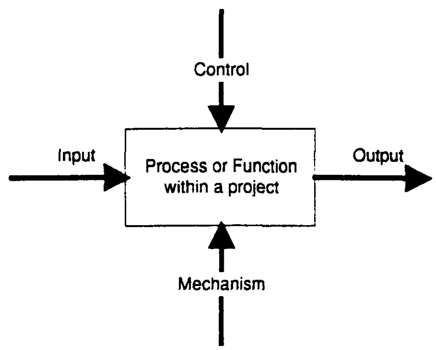


Figure 3-2 IDEF0 Notation

Figure 3-3 illustrates the Application Activity Model (AAM), which is used by STEP BCCM, describing the context of engineering process within an application domain. This IDEF0-based model presents the overall picture of the process model of

[10] For example, some software companies, like Knowledge Based Systems, Inc, have developed IDEF automated tools and refined the original IDEF.
[11] For example, the National Institute of Standards and Technology, an agency of the U.S. Department of Commerce's Technology Administration, is the major developer of IDEF standards.
[12] For example, IDEF0 (Functional Modeling Method), IDEF1 (Information Modeling Method), IDEF2 (Dynamics Modeling Method), IDEF3(Process Flow and Object State Description Capture Method) and IDEF 4 (Object-Oriented Design Modeling Method) merge into a systematic modeling family.

STEP BCCM[13] and demonstrates the control mechanism, the input and output data flow in different phase of an A/E/C project. Afterward, the system developer will be able to design the information flow and control algorithms for an ISO BCCM-based application.

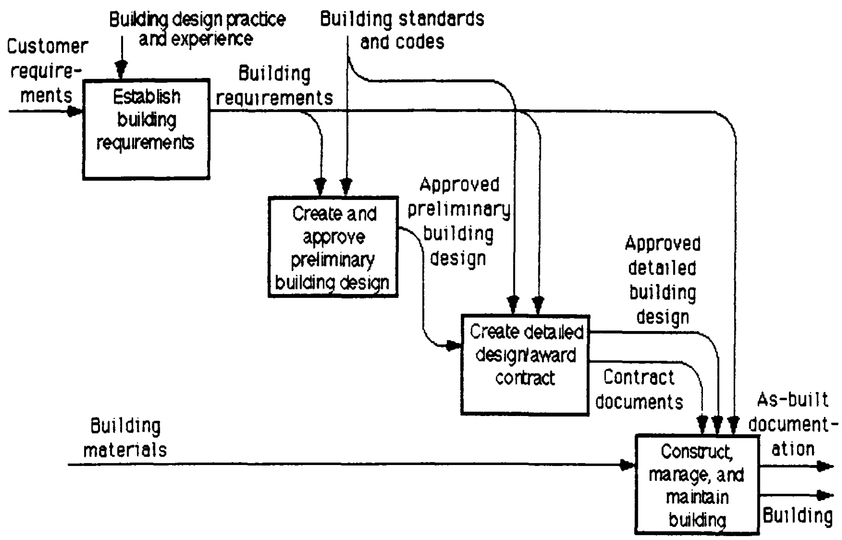


Figure 3-3 ISO STEP AP225

3.2 Industry Foundation Classes (IFC)

In addition to the efforts of ISO, another parallel industrial standard, "the Industry Foundation Classes (IFC)," has been undertaken by industry practitioners of the building industry and software vendors in support of the demand of architects, engineers, construction and facilities management (AEC/FM) community.

[13] Besides IDEF0, ISO STEP has not adopted other IDEF standards.

In 1995, the International Alliance for Interoperability was found by different industrial disciplines to define the IFC [see IAI IFC End User Guide (1997)]. Since a large portion of IAI members is from industry, the IFC standards are designed to accommodate the need of their daily practices. On the contrary, the ISO standards are created by multiple national standardization organizations, which are mainly government agencies. Thus, the neutral ISO STEP can not always adopt new technology to meet the demand of industry in a timely manner without identifying all the positive and negative inspects of the new technology.

In addition, a variety of IAI members are the developers of A/E/C tools.[14] These software vendors bring the latest software technology into IFC. For instance, the approach of IFC is not only concentrated on the design of data structures, but also the use of commercial software standards, such as Microsoft Distributed Common Object Model (DCOM) [see IFC Specifications Volume IV: IFC Model Software Interfaces (1997)].[15] Thus, the IFC has augmented the practicality of its standard for the demands of diverse A/E/C community.

The reminder of this section consists of three parts: Section 3.2.1 presents the architecture of the IFC Model. On the basis of the architecture, the IFC model allows the deployment of software interoperability, as discussed in Section 3.2.2. The joint research of IFC and ISO STEP, discussed in Section 3.2.3, implies the unification of both models.

---

[14] Such as Autodesk, Bently Systems, IBM and Visio Corporation.

[15] For example, The IFC release 1.0 volume IV: IFC Model Software Interfaces is presented using Microsoft Interface Definition Language, which can be implemented directly. In contrast, ISO STEP,

3.2.1 The Architecture of the IFC Model

The core of the Industry Foundation Class model is an object-oriented A/E/C shared information model. The development of IFC object modules is categorized into three layers as shown in Figure 3-4 [see IFC Specification Volume II: IFC Object Model for AEC Projects (1997)].

1. Independent Resources Layer, which groups resource objects for all domains of A/E/C projects. For example, any sub-model of architectural design, building construction and facility management can access the same geometrical information in

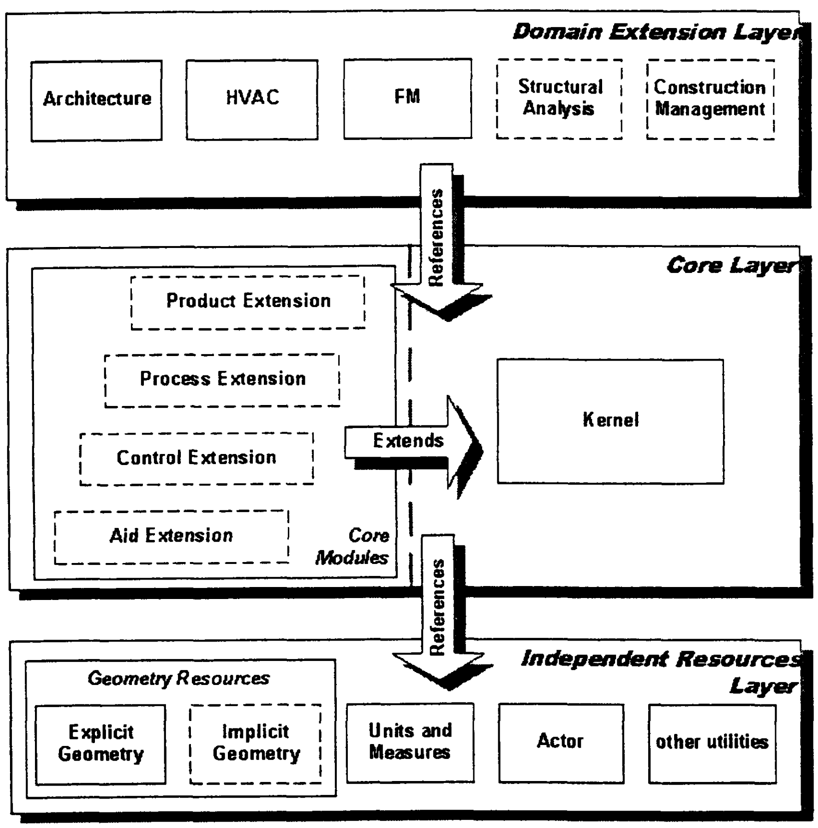


Figure 3-4 IFC Layered Structure [IFC Specification Volume II (1997)]

---

which doesn't emphasis any commercial tools, is more independent of implementing strategies. Therefore, the implementation of ISO STEP models is more flexible, but also more difficult than IAI IFC.

this layer. That is, general purpose objects should reside in this layer.

2. Core Layer, which connects both Independent Resources and Domain Extension Layer. The architecture of Core Layer will be discussed later.

3. Domain Extension Layer, which provides extension modules for individual A/E/C domains, such as construction management and architectural design.

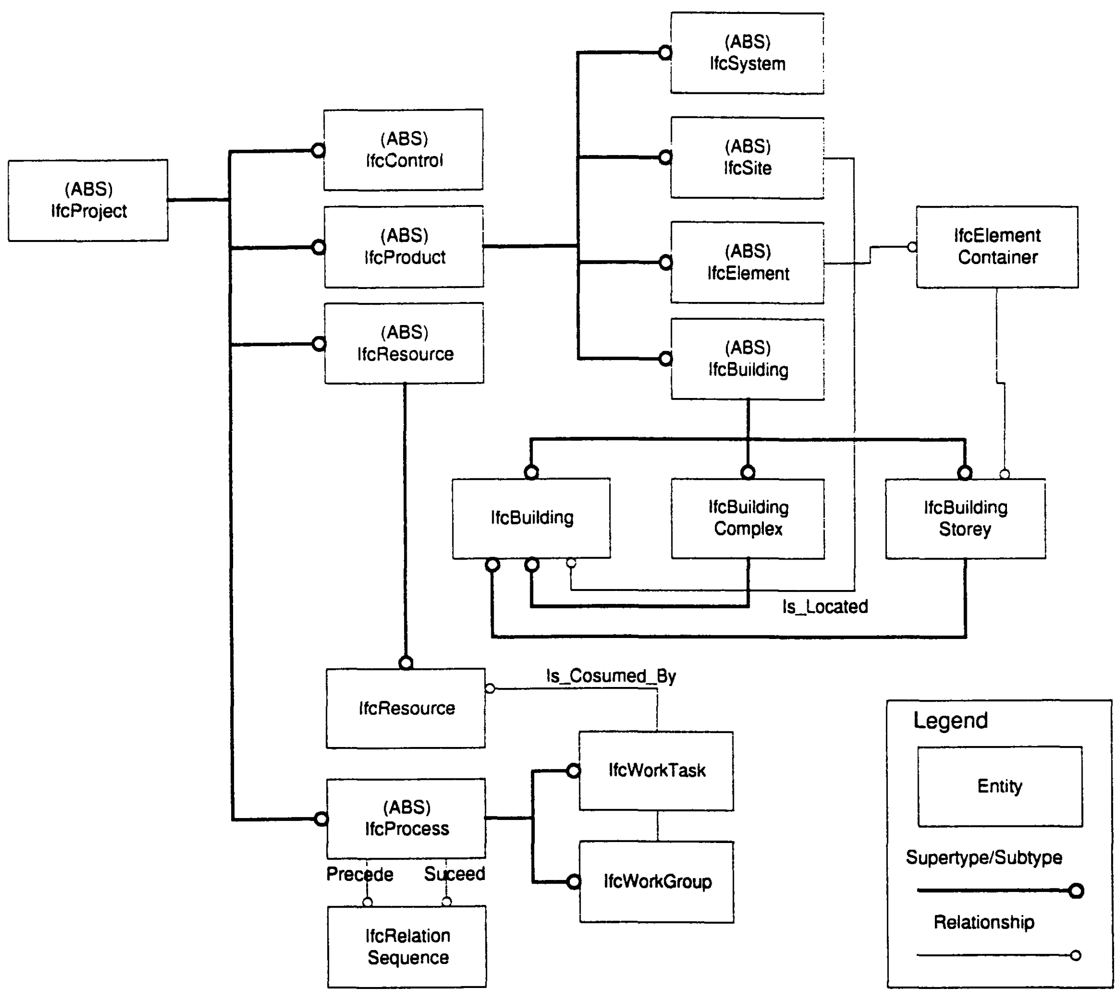


Figure 3-5 IFC Core Model in EXPRESS-G

Each layer groups objects with different scopes work together. In this referencing hierarchy, object modules of a higher level can access object modules in a lower level. However, the object modules of a lower level are independent from the implementation of its higher levels. Therefore, A/E/C information captured in the Core Layer and the Independent Resources Layer can be shared with all project team members in multiple disciplines.

In the Core Layer, IfcProduct, IfcProcess, IfcResource and IfcControl constitute the IFC Kernel Classes as shown in Figure 3-5 [see IAI IFC Specification Volume II: IFC Object Model Guide (1997)]. IfcProduct is the template for all physical elements of A/E/C projects, such as the building, walls and steel frames; IfcProcess defines the process of design, construction and management; IfcResource includes all resources that will be consumed by IfcProduct and IfcProcess; IfcControl imposes constraints of IfcProduct, IfcProcess and IfcResource. The relationship among IFC entities has also been developed, but the mechanisms to link them have not been well defined [see IAI

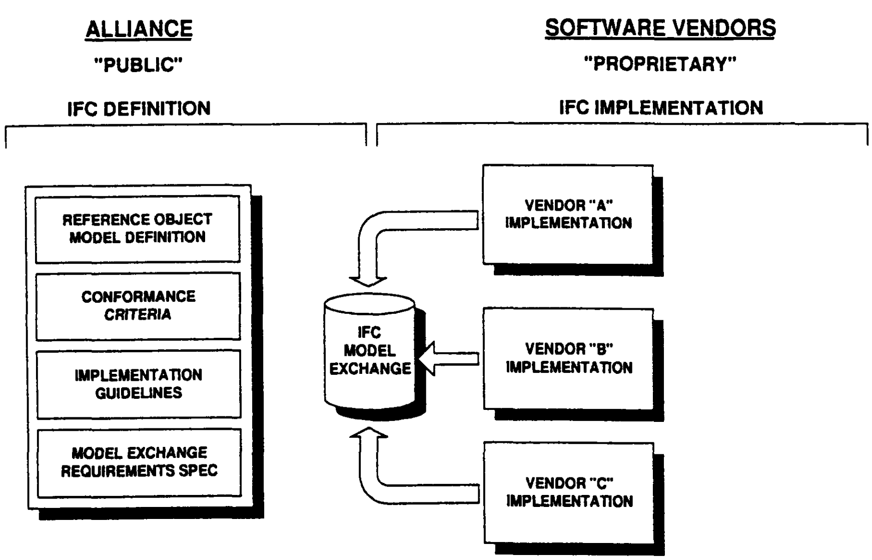


Figure 3-6 IFC Model Exchange [see IFC Specification III (1997)]

IFC Specification Volume II: IFC Object Model Guide (1997)]. However, the integration of IFC entities is an important step in the development of an integrated project management system.

3.2.2 IFC and Interoperability

Another significant characteristic of IFC is the IFC software interfaces specification, which provide a standard for IFC object sharing through the software interoperability. That is, the data exchange of IFC systems is achieved through IFC software interfaces first. Afterward, the dissemination of project information proceeds into IFC Core Model. Consequently, the information-sharing among A/E/C project participants is based on the integration of IFC systems as a foundation for interoperability as shown in Figure 3-6 [see IFC Specification Volume III: IFC Model Exchange (1997)].

3.2.3 The Joint Research of IFC and ISO STEP

Although the approach of the IAI and ISO to the formation of A/E/C information-sharing standards is different, the missions of IAI IFC and ISO STEP are basically identical. In June of 1997, IAI announced that IAI and ISO TC184/SC4 committee would work collaboratively for the development of the STEP standard. Their joint efforts will be release in 1998 [see http://www.interoperability.com (1997)].

3.3 Summary

Both ISO STEP and IAI IFC provide a solid basis for A/E/C information modeling. Their physical and logical entities attempt to describe geometrical design, scheduling, cost estimating, budgeting, resource allocation, change order and project management practice. Although their approaches are different, their A/E/C core models form unifying references for specific areas of project management. The SCHEREC model, developed in Chapter 4, will adopt the concept of the A/E/C core model to enable the interoperability among separate engineering systems.

Furthermore, the cooperation of the two most significant A/E/C information integration efforts promotes the development of A/E/C information-sharing standards. The ISO STEP and IFC standards are likely to grow to be the mainstream of standards and transform the current practice of the A/E/C industry. Hence, the proposed SCHEREC model will mainly comply with the deployment processes of ISO STEP and IAI IFC standard in order to be consistent with future implementations of IFC or STEP based systems.

Chapter 4

The SCHEREC Model

The SCHEREC 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 [see http://www.autodesk.com (1998)] for representing the physical characteristics of a project. Schedulers use Primavera Project Planner [see http://www.primavera.com (1998)] for capturing the logic and time needed to build the project. Estimators use Timberline [see http://www.timberline.com (1998)] for developing the cost associated with the project. Accountants use special accounting systems, such as J.D. Edwards' Financial Suite [see http://www.jdedwards.com (1998)], 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.

As discussed in the foregoing chapters, object-oriented modeling technology can be used to encapsulate and manage these different types of entities in the product and process models. Thus, in this chapter, the object-oriented modeling processes, which follow the standards of ISO STEP and Industry Foundation Classes, are used to develop the SCHEREC model for integrating these diverse product and process models in infrastructure projects.

Traditionally, project participants through a series of project documents share the product and process information of A/E/C projects. The project documents consist mainly of the agreement, the conditions of the contract (general, supplementary and other conditions), the drawings, the specifications and numerous different kinds of reports. These documents may be issued by different participants in different phases of project life cycle.

These documents are either circulated within individual domains or need to be communicated with other parties, and represent different types of information. As discussed in Section 3.1.2, ISO Building Construction Core Model (BCCM) categorizes the project information into project, process, resource and control entities. Similarly, the IAI IFC classified project information into product, process, resource and control entities. After the generalization and specialization processes, both ISO BCCM and IAI IFC create a small number of sub-models, such as product, process and resource, to manage these entities. Each sub-model takes into account the storage, exchange and maintenance of information in the specific domain of project management.

The project information related to product, process and resource are commonly identified by ISO STEP and IAI IFC core model. Hereafter, the propose SCHEREC model also makes the same classification in order to conform to these existing standards:

1. The Product Model consists of physical components described in the design and construction documents, such as the beams, the slabs and the columns. The

SHEREC product model will define a mechanism to encapsulate graphical representations and design parameters.

2. The Process Model consists of objects, which describe the evolution of the project through its project life cycle. The process model is used to manage the project schedules, PERT/GERT and CPM parameters [see Meredith and Mantel (1995)].

3. The Resource Model includes any resource entities required to implement the project, such as materials, labor force, equipment, and their associated cost accounts.

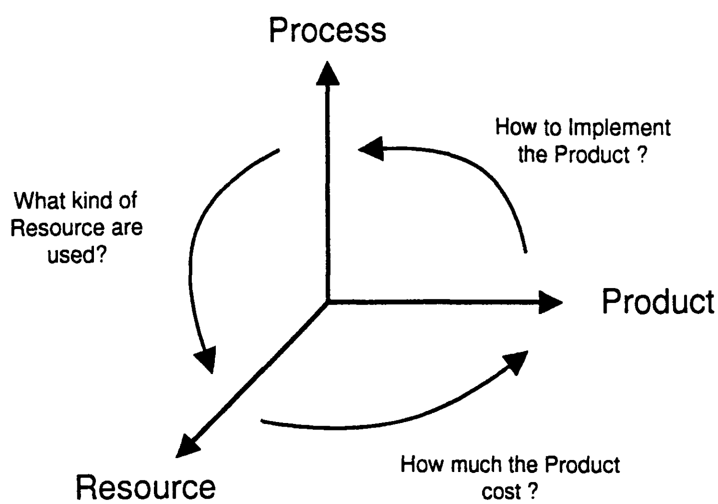


Figure 4-1 The Relationship of the SHEREC Sub-Models

As shown in Figure 4-1, the product, process and resource models are closely dependent on each other. The product model is implemented according to the process model. The resource model organizes any resource usage of the product and process model. However, the contexts of each model and their relationships have not been well established by ISO or IAI. The following sections will closely examine each model and the relationships between these models. Section 4.1 outlines the product model for A/E/C physical entities. The process model, describing the installation process of the product model, is documented in Section 4.2. The resource model, encapsulating consumed resources by entities in the product and process model, is discussed in Section 4.3. At last, Section 4.4 presents the proposed core model and the integration processes.

4.1 Product Model

The mission of a product model is to categorize any information concerning physical entities found in the life cycle of an infrastructure project. The elements of the product model should occupy a fixed physical space and exist for a long period of time. However, in the later sessions of this chapter, some movable items, such as temporary supports, will be introduced as part of the core of the proposed product model because those temporary elements also constitute a substantial portion of the project cost and construction duration [see Fisher (1996)].

Furthermore, if the concept of SCHEREC model is extended to the entire project life cycle, the entities in the product model will not only be utilized in the design and construction phases but also in the facility management. Since each entity in the proposed model has its own life cycles, the evaluation of design alternatives is not only limited to the schematic designs. Rather, the simulation of a product entity browses through its entire life cycle, even after the delivery of the project.

The reminder of this section is organized as two parts: Section 4.1.1 outlines the generic class template in the product model; Section 4.2.2 presents the creation of the building component hierarchy, utilizing the generic class template in Section 4.1.1.

4.1.1 The Generic Class Template in the Product Model

An entity in the product model should encapsulate its dimensions, location, material and other physical attributes. Moreover, it should be able to cooperate with activity entities in the process model to handle the project progress. In addition, the product entities also need to provide resource-related information, such as the quantity and type of resource, for the query of entities in the resource model. Furthermore, the interface templates to provide service for the process and resource model will be established as well.

As Table 4-1 shows, the attribute, "bc_state," of a SCHEREC physical component consists of six different pre-defined data value: PreliminaryDesign, Design, UnderConstruction, Implemented, Temporary and Demolished. Thus, after the activity entities in the process model propagate the message - "Activity_Finished" to corresponding product entities, the current state of a product entity will be updated simultaneously.

In the meanwhile, the resource model utilizes the attribute - "bc_material" of a product entity. The value of bc_material should be determined according to the product's construction scheme. The basic construction scheme used by the test case - Shimizu Smart Tower[1] can be roughly categorized as Concrete-Reinforcement (RC), Steel Concrete-Reinforcement (SCR) and Pre-cast Concrete (PC) [see Shimizu Corporation (1995)].

A proposed generic class template for the product model is developed in EXPRESS as Table 4-1 illustrates. In the template, the ENTITY and SUBTYPE OF statements define the name of the physical component and its relationships to the entire project. Afterwards, a series of attributes are adopted to describe its properties and special keywords (INVERSE, UNIQUE, TYPE and DERIVE) are used to distinguish the schema [see Wix (1996) and Appendix A]. For example, the attribute bc_ID is declared an unique value and the inv_bcs attribute has one-to-one relationship with the building entities. As a result, a physical entity with an unique bc_ID in the SCHEREC product model should be the only entity which inherit the properties of the same building. The system cannot create another instance with the same name and access corresponding entities accidentally.

---

[1] It is a patented construction plan of Shimizu Corporation, Tokyo, Japan. Please contact Professor Peña-Mora from MIT or Dr. Minemesa from Shimizu Corporation for further detail.

As a result, the SCHEREC product model template defines the dimension, location and other static properties. Moreover, the relationship between SCHEREC entities is encapsulated. The transition of bc_state can be managed by the cooperation of the SCHEREC product and process model, which will be discussed in Section 4.2.

```
ENTITY building_Component

SUBTYPE OF building_component_group;

--Attributes
        bc_ID          :      BC_ID;
        bc_name        :      NAME;
        bc_location    :      Location_Description;
        bc_geometry    :      ACAD_Geometry;
// Follow Autodesk AutoCAD Definition.
        bc_volume      :      Volume_Measure;
        bc_material    :      Material_Description;
        bc_state       :      ENUM of [ PreliminaryDesigned,
                                        Designed,
                                        UnderConstrution,
                                        Implemented,
                                        Temporary,
                                        Demolished ]

  // 1. "UnderConstruction" should be divided into more detailed states when the building
  //    component needs more than one construction task to implement it.
  // 2. "Temporary" and "Demolished" are used by temporary facilities only.

        UNIQUE
        bc_ID          :      BC_ID;
        INVERSE
        inv_bcs        :      SET [1:1] OF Building

  --Decomposition

  subtype_building_component1 : OPTIONAL SET [1:?] building_component_group1
  subtype_building_component2 : OPTIONAL SET [1:?] building_component_group2
```

```
TYPE CONSTRUCTION_SCHEME  = ENUMERATION OF (RC, SRC, PC)
// Concrete-Reinforcement(RC), Steel Concrete-Reinforcement(SRC) and
// Precast-Concrete (PC)
END_TYPE

  DERIVE
  bc_construction_tasks = SELECT (CONSTRUCTION_SCHEME)
  bc_materials = SELECT ( MATERIAL_USED_BY_CONSTRUCTION_TASKS )

  END_ENTITY;
```

Table 4-1 SCHEREC Product Class Template

4.1.2 Creating the Building Component Hierarchy

After making instances of physical entities from drawings and other design documents, the SCHEREC Building Component Hierarchy is introduced to perform the decomposition of the targeted Shimizu Smart Tower building and organize these physical components into a systematic classification.

As illustrated in Figure 4-2, seven individual levels were used to classify building components from a building entity to different levels of detail.[2] The root of the hierarchy is the project entity. In the next level, different building is identified and two instances: office building A and B are created. Afterward, different floors of the same building are categorized in the construction zone level. Moreover, different rooms, serving different functions, are classified in the functional group level. In the same room, different

[2] The SCHEREC Building Component Model is a revised form of the RATAS product model of Finland's VTT in the late 1980s [see Bjork (1992)].

structural components, such as column, beam and slab, are further derived in the structural group level. At last, all building components are instanciated, according to the SCHEREC product class template, at the component level.

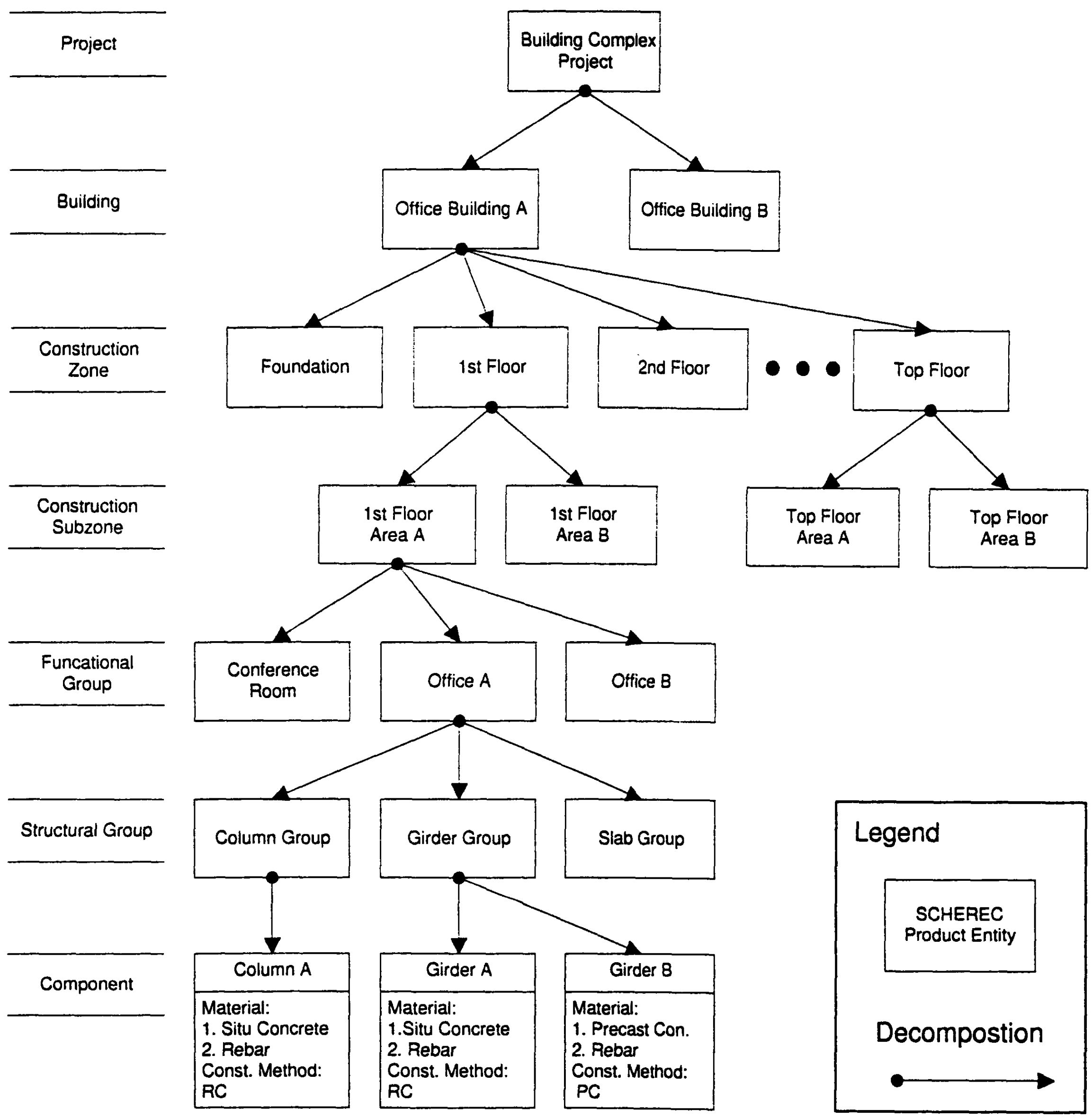


Figure 4-2 SHEREC Building Component Hierarchy

4.2 Process Model

The process model augments the evolution of the project life cycle through the use of project schedules. Unlike the product model, only a fewer number of studies for the integrated A/E/C models have been dedicated to creating an "activity-centric" process model rather than a "data-centric" product model, through the life cycle of an infrastructure project [see Froese (1996)].[3]

Besides, the process model is the key to the simulation of project progress, which is essential to many project management tasks, such as cost estimation and resource management. The traditional activity-centric approaches, such as PERT/CPM, define a systematic way to manage the project schedule independently. However, the process model of an integrated A/E/C model should also monitor, control and update the information of the product and resource model in order to supervise the project expenditure and the resource usage to optimize the performance of the ongoing projects.

Chapter 2 states that most modern object-oriented modeling methods, such as UML and OMT, provide tools to address the dynamics of static object models, but only a few integrated A/E/C models take advantage of those dynamic modeling tools. On the other hand, most dynamic modeling techniques are designed to address systems whose scenarios are limited and inflexible, such as the Automated Teller Machine system and

[3] For example, ISO STEP, VTT's RATAS and EU-based COMBINE project are concentrated on the creation of the product models, but do not define a general approach for the corresponding process models.

payroll systems. Nevertheless, the changeable nature of an A/E/C project makes it hard to utilize existing dynamic modeling tools appropriately.

In real world cases, dozens of change orders are frequently issued after the execution of the contract. That is, entities in the product or process model are inclined to change in both the design and the construction phases. Unfortunately, none of the existing process modeling tools can address this problem well. As a result, a number of large-scale international projects usually use other process modeling techniques, and even develop their own systems. For example, ISO STEP has chosen the IDEF0 to model activities in support of project integration. Similarly, like ISO STEP, the development process of IAI IFC uses its own process diagrams to capture usage scenarios. The IFC core model contains both the abstract product and process class, but the mechanism to link these activity classes with the product models has not been well-established [see IFC Specifications Volume III – IFC Model Exchange (1997)].

The entities in the process model, which install the related elements in the product model, and its supporting entities strongly depend on the selected construction methods. Based on this concept, a great deal of research was undertaken to develop either decision support systems or expert systems to automatically generate construction schedules.

J. Cherneff, R. Logcher and D. Sriam of MIT attempted to integrate computed-aided design (CAD) software with computer-generated construction schedules [see Cherneff, Logcher and Sriam (1991)]. The Object-model-based Project Information

System (OPIS), developed by the Stanford Center for Integrated Facility (CIFE), also tried to develop an expert plan-generation system [see Froese and Paulson (1994)]. Since the generation of a practical schedule requires a lot of knowledge and expertise, none of the implementations based on their approaches has been widely applied to the complicated real world projects.

Thus, for this research, existing schedules, which are generated through experience and expertise of the project managers, are employed to create the process model until the maturing of current computer-based schedule-generation theory allow for computer-generated schedules to be used. Therefore, the linkage among the product model, the resource model and the human-created schedules is the key to the success of the integration of SCHEREC sub-models.

The reminder of this section is divided into three parts as follows: Section 4.2.1 examines the dynamics of A/E/C projects. The generic class template in the process model in Section 4.2.2 aims to encapsulate the project dynamics, as outlined in Section 4.2.1. Section 4.2.3 presents the development of the construction task hierarchy, applying the generic class template in Section 4.2.2.

4.2.1 The Dynamics of A/E/C Projects

The dynamics of a conventional A/E/C project are outlined in its project schedules and scope changes. Because numerous activities are usually needed to complete the project, an A/E/C project often consists of schedules in different levels of detail.

The level of detail in a construction schedule is determined by its intended use. Three different levels are often identified: organizational, project and process. The organizational level involves only enough project milestones to monitor the timely completion of the project, such as Table 4-2 [see the Agreement between Owner and Contractor for the Hotel at the World Trade Center, Boston, MA (1995)]. Therefore, the concise organizational level schedules are often used to support the making of overall strategic decisions beyond a single project.

Milestone	Description	Date
1	The construction of the slurry wall is completed.	Nov 17, 1995
2	The excavation and tie-backs are completed.	March 20, 1996
3	The structural steel is topped off.	June 1, 1997
4	The building is permanently closed in.	December 10, 1997
5	Final completion of the Work.	May 1, 1998

Table 4-2 An Organizational Level Schedule

The more detailed project level schedule may identify activities, such as the excavation of foundations and the pouring of concrete for foundation as Table 4-3 illustrates [see Section 5.3]. On the other hands, the process level contains the field's daily activities [see Stumpf, et al (1996)].

Activity ID	Description	Duration	Start Date	Finish Date
BA640	Site Preparation	3	1/19/98	1/21/98
BA650	Excavation	5	1/22/98	1/26/98
BA660	1F Structural Frame Erection	8	1/27/98	2/2/98
BA670	2F Structural Frame Erection	9	1/27/98	2/3/98
BA680	3F Structural Frame Erection	11	2/3/98	2/13/98
BA690	4F Structural Frame Erection	15	2/14/98	2/29/98
BA700	Scaffolds Hanging	4	3/2/98	3/5/98

Table 4-3 A Project Level Schedule

This thesis is focused mainly on the project and the process level schedule. The generation of the organizational level schedule needs a higher level of integration between different ongoing projects, and thus is beyond the scope of this research.

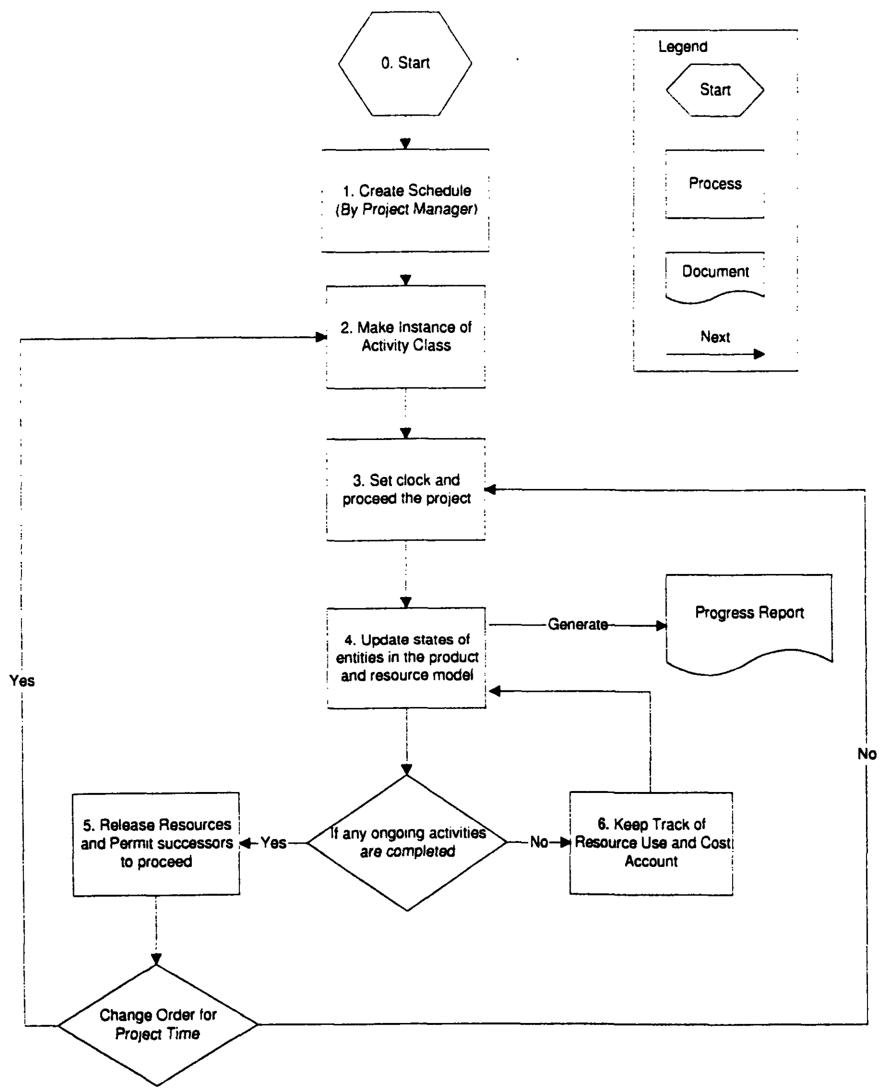
A mechanism for the development of the process model is developed in a conventional flow chart as Figure 4-3 shows:

After the start of an A/E/C project, the project managers and his/her staffs according to the product model, described by the bidding data, will create the project-level schedules.[4] Then, the SCHEREC system will create instances of activity class, whose template is defined in Section 4.2.2. Once the SCHEREC system starts, the

[4] It is assumed that the electronic schedule is stored in Primavera Project Planner 2.0 or other compatible formats.

transition of the SCHEREC product entities, namely the construction sequence, are managed by the SCHEREC process model.

Thereafter, entities of the product and resource model are recursively updated in agreement with the process model, which is always consistent with project schedules updated by change orders. In the meanwhile, the resource assignment and cost account can be traced and recorded in resource use objects, which are described by the SCHEREC resource model in Section 4.3.



4.2.2 The Generic Class Template in the Process Model

As discussed above, the process model should not only simulate the project progression, but also update the resource assignment, cost account and the building components. Thus, a generic class template in the process model is developed as Table 4-3 shows:

```
ENTITY construction_activity

SUBTYPE OF construction_schedule;

--Attributes
     activity_ID                      :     Activity_ID;
     activity_name                    :     NAME;
     activity_description             :     Activity_Description;
     activity_duration                :     TIME;
     activity_early_start             :     DATE;
     activity_late_start              :     DATE;
     activity_early_finish            :     DATE;
     activity_late_finish             :     DATE;
     activity_slack_float             :     TIME;
     activity_total_float             :     TIME;
     DERIVE
     activity_independent_float       :     TIME;
     activity_interferance_float      :     TIME;
     //Schedule Recovery Parameters
     overtime_percentage              :     PERCENTAGE;
     overtime_salary_ratio            :     RATIO;
     overtime_productivity            :     RATIO;
     newhire_percentage               :     PERCENTAGE;
     newhire_salary_ratio             :     RATIO;
     newhire_productivity             :     RATIO;

     activity_predecessors :   OPTIONAL SET [1,?] construction_activity
     activity_successors   :   OPTIONAL SET [1,?] construction_activity
     construction_method   :   Construction_Method_Description;

     UNIQUE
     activity_1                       :     activity_ID;
     INVERSE
     inv_activities                   :     SET [1:1] OF Schedule
```

```
--Decomposition
subtype_construction_activity1 : OPTIONAL SET [1:?] building_activity_group1
subtype_construction_activity2 : OPTIONAL SET [1:?] building_activity_group2

--Relationships
abstract_resource_use : OPTIONAL SET [1:?] of resource_use

abstract_construction_scheme : MANDATORY SET [1:?] of construction_method
abstract_building_component : MANDATORY SET [1:?] of building_component

END_ENTITY;
```

Table 4-3 SCHEREC Process Class Template

4.2.3 Creating the SCHEREC Construction Task Hierarchy

After reviewing a series of real-world construction schedules used in the Shimizu Corporation[5], the following points were concluded:

1. In the project-level schedules of a building construction project, only a limited number of significant building components are directly identifiable, such as columns, slabs, girders and walls. Moreover, temporary components, such as scaffolds and molds, need to be incorporated into the schedules. Besides, subordinate components, which take fewer tasks to implement and cost only a minor portion of the project budget, are not described in the project-level schedules in order to reduce the complexity of the process model.

[5] Most of these schedules are used by a series of building projects in Tokyo, Japan. Since they are strictly confidential, please contact Prof. Peña-Mora of MIT or Dr. Minemesa of Shimizu Corporation for further information.

2. Modular construction tasks prevail in the project-level schedule. For example, the "column construction" task in the Shimizu Smart Tower project is further decomposed into:

    a. marking
    b. erection
    c. bolting
    d. welding
    e. re-bar fixing
    f. forming
    g. concrete pouring
    h. exterior polishing

3. Construction tasks consist of a series of sub-tasks. For example, the "column construction" task incorporates:

    a. steel works, which include marking, erection, bolting and welding;
    b. re-bar and concrete works, which include re-bar fixing, forming and concrete pouring; and
    c. non-structural works, which include polishing, painting and HVAC work.

Consequently, the activity objects can be systematically grouped into sub-tasks and classified by modular construction tasks. Therefore, the activities to implement physical entities in the product model form the construction task hierarchy for the Shimizu Smart Tower project as shown in Figure 4-4. The similarity between the Building Component Hierarchy and the Construction Task Hierarchy provides a simple framework to create the links between the product and process model. The linking mechanism will be discussed in Section 4.4.

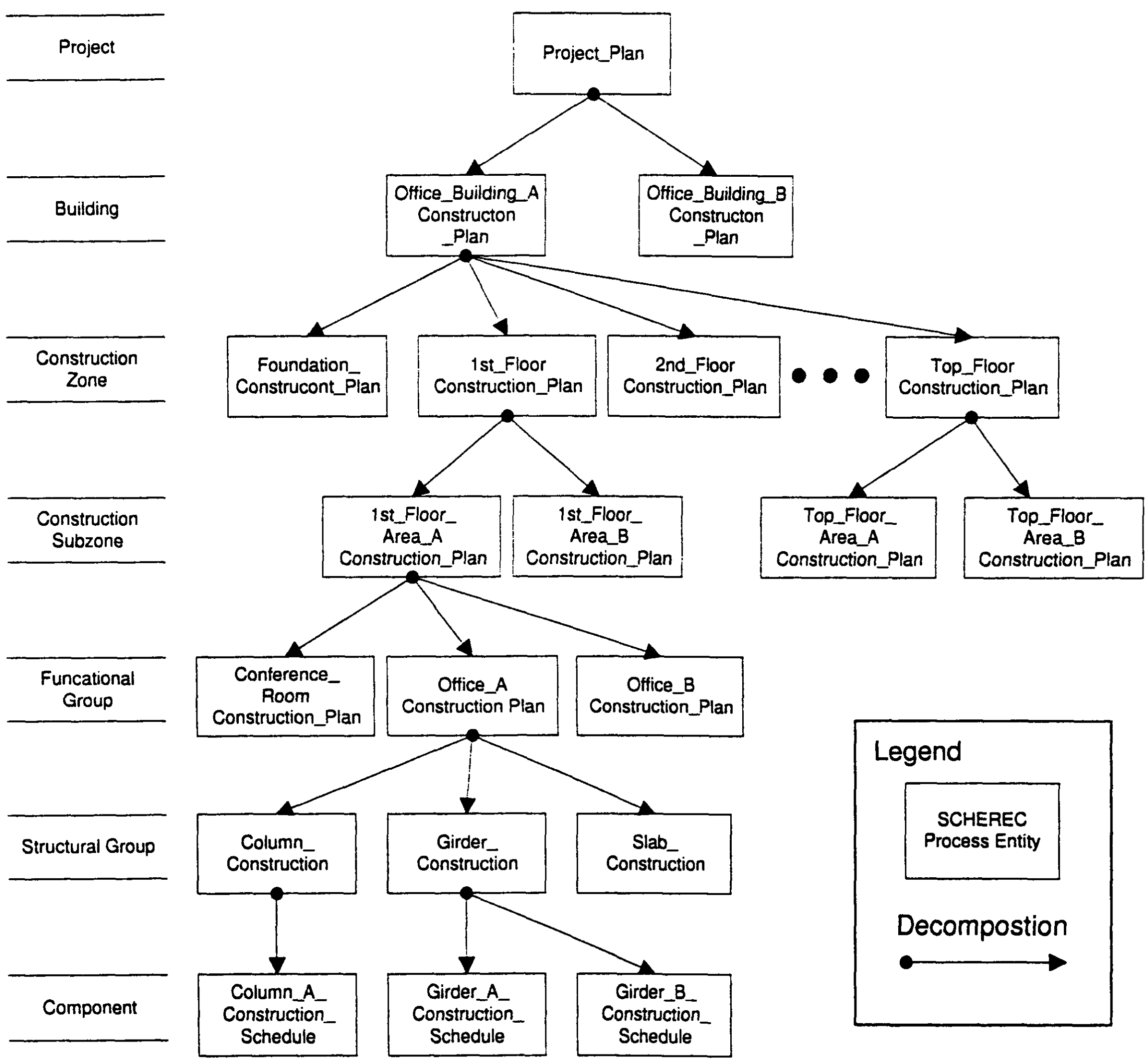


Figure 4-4 SHEREC Construction Task Hierarchy

4.3 Resource Model

The Resource Model aims to identify and organize resource entities used by the primary entities of the product and process model. As the project proceeds, the implementation process of the building components in the product model consumes material resource objects (such as concrete, steel beam and re-bar). Meanwhile, the construction activities, defined by the SCHEREC process model, also need labor resource, like carpenter, electrician and steelworkers, and equipment resources, such as crane, bulldozer and trunks.

The organization of this section is divided into three parts. Section 4.3.1 outlines the creation of the resource database, allowing the persistence of the resource entities. Section 4.3.2 presents two generic class templates in the resource model: resource and resource_use. At last, Section 4.3.3 discusses the SCHEREC resource model, which is composed of the resource and resource_use class templates.

4.3.1 Creating the Resource Database

The SHEREC resource model needs to introduce resource use and cost account objects to estimate the cost of installation and manage the usage of project resources. The following diagram, as shown in Figure 4-5, is used to create a resource database to maintain the persistent resource object in support of the resource model.

Once the project building component hierarchy and the construction task hierarchy have been created, the SCHEREC system can identify the quantity and types of

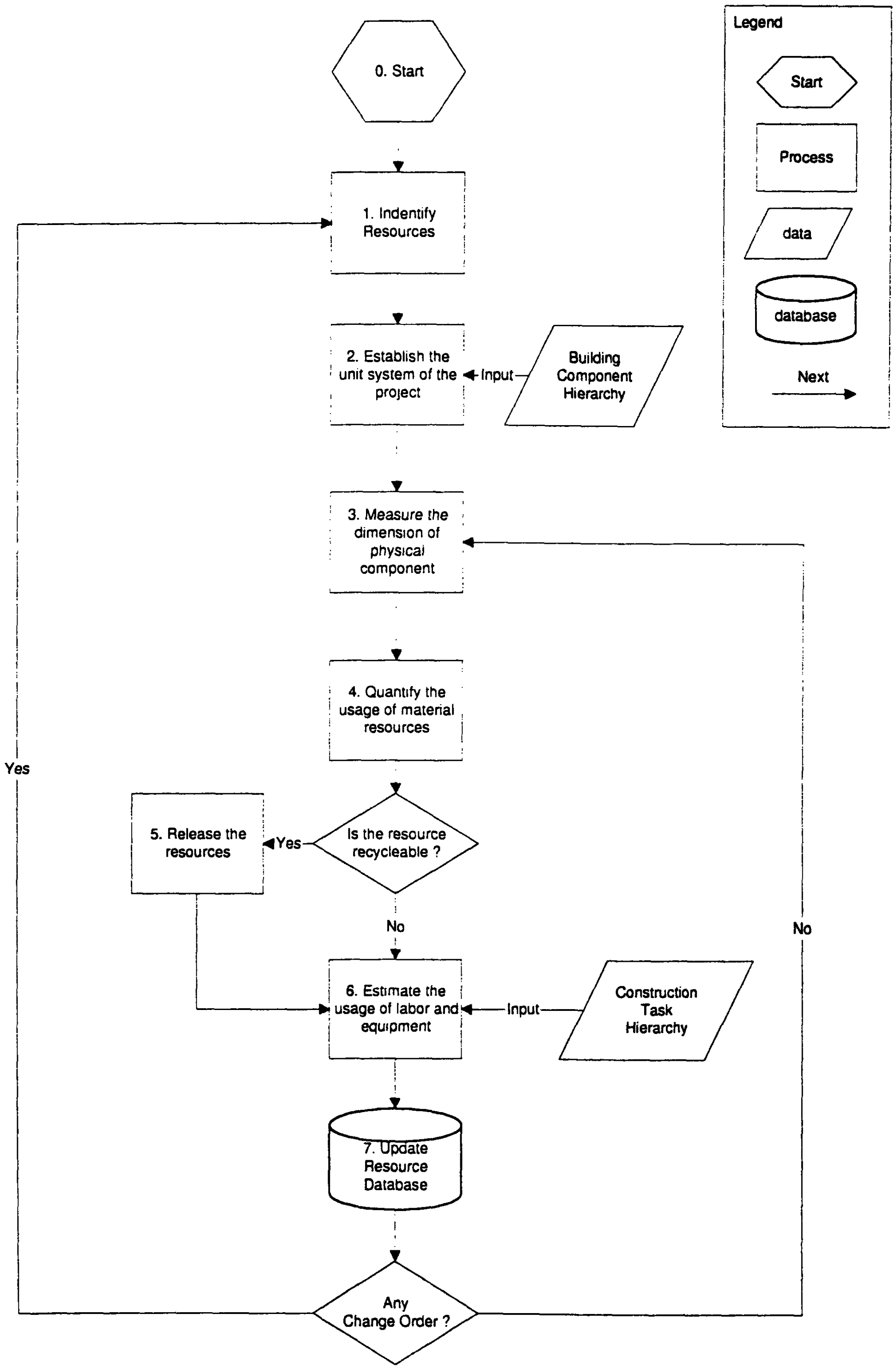


Figure 4-5 SCHEREC Resource Database Creation

material resources from the product model as shown Step 3 and 4, classify the labor and equipment resources from the process model as Step 6, and establish the resource database for the resource model as Step 7. In Step 6, if a temporary resource is marked as recyclable, the system will release it in Step 5 and other entities will be able to reuse it.

The resource database, illustrated in Step 7, needs to contain the following tables:

- Labor and equipment needed to install entities in the product model;[6]
- Unit price of each resource item, which includes each types of material, labor and equipment;
- Productivity of labor and equipment; namely, throughput in a fixed period.

Since data of such a database can be similar from an A/E/C project to another project in adjacent areas, if the local resource market doesn't change dramatically within the project life cycle, the data of the three tables could be created from historical data of the resource suppliers.[7]

The database can be simply implemented by object-oriented and relational database systems.[8] In any case, the entity-relationship model used by relational database systems provides a good mapping to the object-oriented resource model. Afterwards, the resource usage objects and cost account objects should have the accessibility to the three tables.

---

[6] The dimension and quantity of material should be obtained from the product model, not from the resource database, in order to consistently conform to any design change.

[7] Further studies are required to verify the coherence of historical data. Right now the resource data of the SCHEREC systems are manually inputted.

[8] However, most available and robust database products, such as ORACLE, DB2, Sybase, Informix, Microsoft Access and SQL Server, are relational database system (RDBS).

4.3.2 The Generic Class Templates in the Resource Model

The resource class template defines properties of a resource entity, as Table 4-4 demonstrates.

```
ENTITY resource_item

SUBTYPE OF resource_group;

--Attributes
      resource_ID                   :       RESOURCE_ID;
      resource_Unit                 :       Unit;
      resource_Time_Unit            :       Time_Unit;
      resource_Unit_Price           :       Unit_Price;
      resource_Productivity         :       Productivity;
      resource_Time_Stamp           :       Time_Stamp;
      resource_Supplier             :       A/E/C Actor
// Used to record the date updated from database
      UNIQUE
      resource_ID                   :       RESOURCE_ID;
TYPE RESOURCE_TYPE = ENUMERATION OF (MATERIAL, LABOR,
EQUIPMENT)
END_TYPE
      DERIVE
      Resource_Type = SELECT(RESOURCE_TYPE)
END_ENTITY;
```

Table 4-4 SCHEREC Resource Class Template

Similarly, the resource_usage class template describes the usage of resources as Table 4-5 shows.

```
ENTITY resource_usage
SUBTYPE OF resource_usage_group;
--Attributes
      resource_Usage_ID             :       RESOURCE_USAGE_ID;
      resource_Usage_Unit           :       Unit;
      resource_Usage_Quantity       :       Quantity;

      UNIQUE
      resource_Usage_ID             :       RESOURCE_USAGE_ID;
--Relationships
      abstract_Usage_task           :       OPTIONAL SET [1:?] of construction_activity
END_ENTITY;
```

Table 4-5 SCHEREC Resource_Use Class Template

4.3.3 The SCHEREC Resource Model

Based on the two class templates discussed in Section 4.3.2, the relationship among resource, resource_usage, cost_account and other objects can be modeled in EXPRESS-G as Figure 4-6[9] demonstrates. The resource_usage objects create the links to the building component hierarchy for material resource assignment, such as the type and quantity of concrete, and to the construction task hierarchy for labor and equipment resource assignment, such as the quantity of field engineers to supervise the installation of an entity in the building component hierarchy. Meanwhile, the cost account objects keep track of the expenditures of resources according to the resource_Usage_Quantity attribute of the resource_use class and the resource_Unit_Price attribute of the resource class.

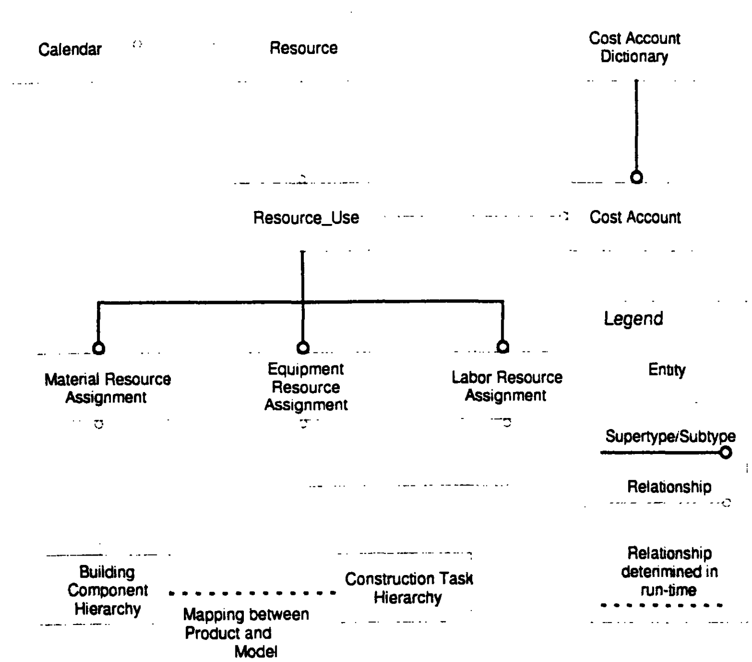


Figure 4-6 SHEREC Resource Model

[9] The relationship between the SCHEREC product and process model will be discussed in Section 4.4.2.

4.4 The Integration of SCHEREC Sub-models

The relationship between a product entity and its corresponding activity entities is determined by the construction method. Different construction methods utilize different set of activities and resources to install an identical product entity. Consequently, the selection of construction methods, which will be discussed in Section 4.4.1, is the key to the integration among the product, process and resource model. After the selection processes, discussed in Section 4.4.1, the SCHEREC core model is presented in Section 4.4.2.

4.4.1 The Selection of Construction Methods

The construction methods of a building component define the construction tasks to install the physical component. In other words, the construction methods govern the instance construction of activity objects. Moreover, the selection of construction methods determines the linkage among the product, process and resource model.

Most research concerning the generation of construction schedules relies on the assumption that the selected construction methods and involved activities comply with either rules from experience or mathematical formula. For example, the depth, the width of excavation, and the rule, "If the soil moisture content is dry or moist, or if the soil is not firm but the depth of excavation is less than 3.6 meters, then no bracing is required and the natural slope of the soil can be used." [see Stumpf et al (1996)], are critical considerations of the creation of the project schedules.

However, some parameters, required by those rules and formula, are difficult to quantify. That is, it is hard to encode these ambiguous criteria, which are gained through human experience, into the "rules" of a rule-based computer system. A variety of studies have been undertaken to address this problem [see Froese and Paulson (1994)].[10] However, most studies have not taken into account an underlying integrated information model. Consequently, their applications are limited to specific areas of project management, such as schedule-generation systems and cost estimation systems. Therefore, these specific applications can not benefit from the interoperability with other systems.

The SCHEREC system does not determine the construction methods arbitrarily, but allow construction manager to select the best construction means. The selection of construction methods is the key measure to the development of the construction plans, namely the context of the process model. The selection process relies on disparate factors, such as cost, time, local regulation, general site condition, the degree of site congestion, the transportability and availability of labor, material and other resources, and even the crew's familiarity with preferred methods. For example, in the Tokyo metropolitan area or other big cities in the world, due to the limitation to transportability, the use of long steel beams is always avoided. Hence, the alternative construction methods, such as short-span steel structures, are considered. [11]

---

[10] For example, the AutoPlan system, conducted by the center for integrated facility of Stanford University, is an expert system application for the generation of construction plan [see http://www-leland.stanford.edu /group/CIFE/ (1997)].
[11] It is based on the opinion of Dr. Minemasa and Mr. Kenji of Shimizu Corporation.

In consequence, well-implemented construction method databases is in need. Such a database would allow the selection of construction methods and the elaboration of the product and process models more efficient. In further analysis, the construction method database should account for the following information:

1. Construction methods and their element construction tasks.
2. Employed entities in the product and process model.
3. The measure of consumed labor force, and the links to the resource usage objects in the resource model
4. The required temporary facilities and the links to the permanent building components in the product model.

As a result, the construction method database not only is comprised of static construction data, but typical construction schemes as well. After construction managers' selection of construction method, all details of the building components in the building component hierarchy of the product model will be further defined and the preliminary construction plans will be organized into the construction task hierarchy for the use of the process model.[12] Meanwhile, the type of the construction method determines the connections between the product and process model [see Fisher (1997)]. Consequently, the entities in the building component hierarchy can be mapped into the corresponding

[12] In current implementation, the selection of construction method is manually inputted by the system users without the support of such a database.

entities, which is imposed by project managers' selection of the construction methods, in the construction task hierarchy.

An experimental scheme to create the link among the SCHEREC sub-models has been developed as Figure 4-7 shows. After following the steps in Figure 4-7, the links among the product, process and resource models can be created and the relationships among them will be well established.

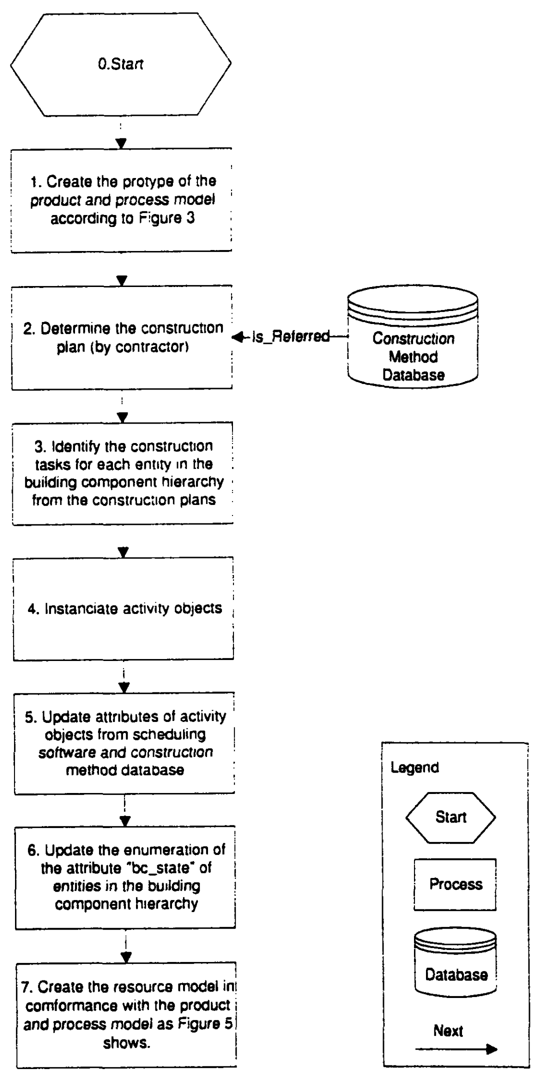


Figure 4-7 SCHEREC Sub-models Integration

4.4.2 The SCHEREC Core Model

The major role of the SCHEREC core model is to integrate project information into the product, process and resource models. In the earlier part of this chapter, the major entities found within the A/E/C project life cycle have been identified. Moreover, a mechanism to capture both the explicit and implicit design, such as geometrical data and construction information has been proposed. The foregoing section defines an applicable process for the integration among the product, process and resource models. Figure 4-8, in EXPRESS-G, 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 effortlessly perform the model-driven takeoff. After the creation of the SCHEREC Core Model, any change to the original design will generate new construction schedules and estimate the equitable cost.

Since the SCHEREC Model doesn't 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 developed in Section 4.4.1, 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 SCHEREC product, process and resource entities are generated in accord with the predefined object-oriented diagram in Figure 4-8.

Finally, such an integrated model provides project managers a valuable tool to integrate project information, supervise the progression of the project, even optimize the performance of project entities.

4.5 Summary

The concept of the SHEREC model sheds new light on the integration of data and knowledge of an infrastructure project through the entire project life cycle. The three SCHEREC sub-models manage the product, process and resource information for different aspects of project management. The SCHEREC core model presents a systematic way, which conforms to the IAI IFC 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 SCHEREC model.

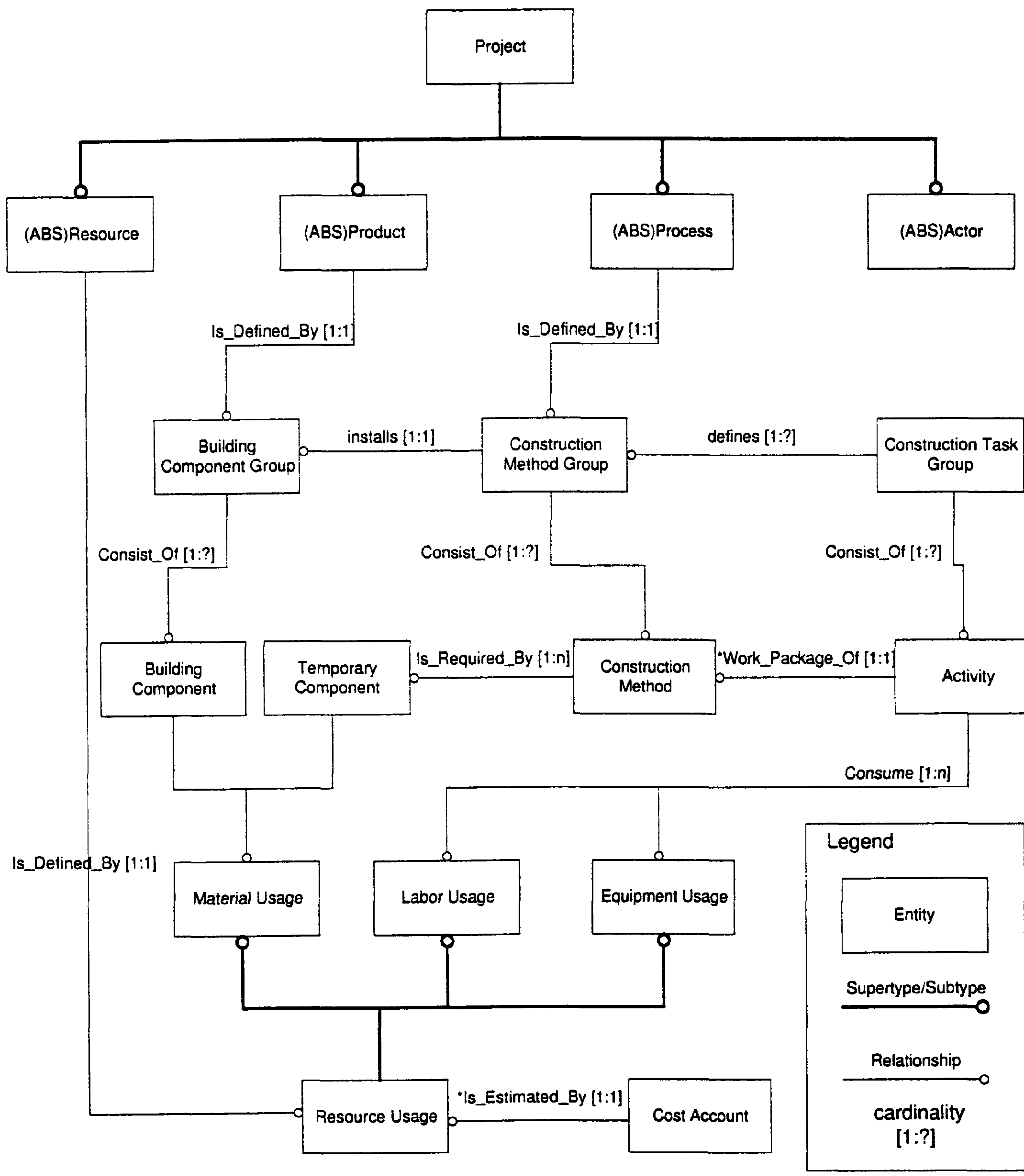


Figure 4-8 SCHEREC Core Model in EXPRESS-G

Chapter 5 will discuss the implementation strategies of the SCHEREC model. The proposed SCHEREC system will utilize the object-oriented technology, discussed in Chapter 2, and take advantage of existing A/E/C information-sharing standards, discussed in Chapter 3, to enable the interoperability between managerial and engineering computer applications. The collaboration of A/E/C applications promises to improve the efficiency and quality of the A/E/C project-delivery process.

Chapter 5

Implementing Collaborative Systems

The recent development of information technology makes the interoperability of engineering computing systems practical. That is, interoperable software systems across diverse domains can integrate information of A/E/C community. Moreover, if the interoperability of A/E/C systems complies with the same information-sharing standards, such as the SCHEREC model, project managers can tap into the power of the interoperability among individual A/E/C systems. Now, the management of multiple A/E/C tasks can be achieved simultaneously.

The reminder of this chapter covers the current standards of software interoperability in Section 5.1. In conformance with these interoperable standards, Section 5.2 outlines the utilization of automation engines of A/E/C systems. At last, Section 5.3 presents the Shimizu Smart Tower as a sample project of the system.

5.1 Standards of Interoperability

The attempt to integrate different A/E/C systems has been proposed since the 1980s [see Bjork (1992)], but the main integration mechanisms were limited to file exchange[1] and shared software libraries[2]. Due to the lack of a generic A/E/C information

[1] For example, ISO IGES has been widely regarded as the official standard for exchanging CAD related information [see http://www.iso.ch (1997)].

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.

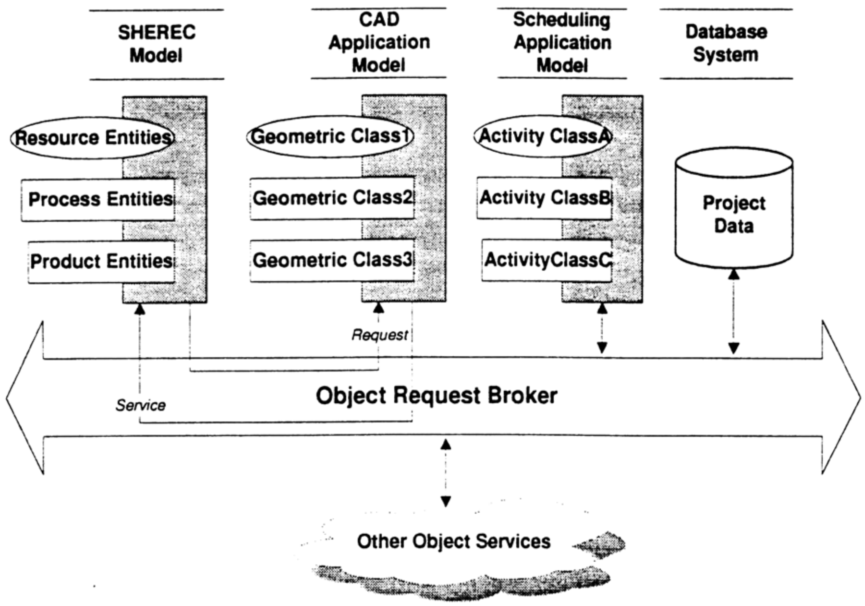


Figure 5-1 ORB Architecture

As discussed in Chapter 2, the evolution of object-oriented technology has promoted the coming of software interoperability. The Object Management Architecture (OMA), defined by Object Management Group (OMG), provides a high level of abstraction (OMG, 1990) to allow the integration of a wide variety of objects. The core of the OMA is the Object Request Broker (ORB)[3], which delivers requests to server objects and returns the result to the client objects. According to this concept, an ORB-based architecture of the proposed SCHEREC system is illustrated in UML as Figure 5-1.

[2] For instance, OpenGL, developed by Silicon Graphics, is used as a solid modeling graphic library throughout industry.

5.2 Using Automation Engines

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 [see http://www.iona.com (1997)], IBM DSOM [see http://www.ibm.com (1997)] and Microsoft COM, allow the deployment of interoperability. Due to the current availability of interoperable software components,[4] the SCHEREC system has been implemented in Microsoft Component Object Model (COM).[5]

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.[6] That is, COM defines a common way for software components to access software services (Microsoft, 1997).

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.

---

[3] It is a part of Common Object Request Broker Architecture (CORBA) standard.
[4] Both Autodesk AutoCAD and Primavera P3 Engine are built on the Microsoft COM.
[5] ActiveX and OLE are the most widely used COM-based technologies.
[6] Actually, a COM interface is defined by the Interface Definition Language (IDL), which is borrowed from the Open Software Foundation's Distributed Computing Environment (OSF DCE).

Figure 5-2 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 SCHEREC system is launched through this kind of automation (Primavera, 1997).
- Local Automation, in which COM services are provided in a separate process running on the same machine. For example, AutoCAD provides its function to SCHEREC system through such a mechanism.

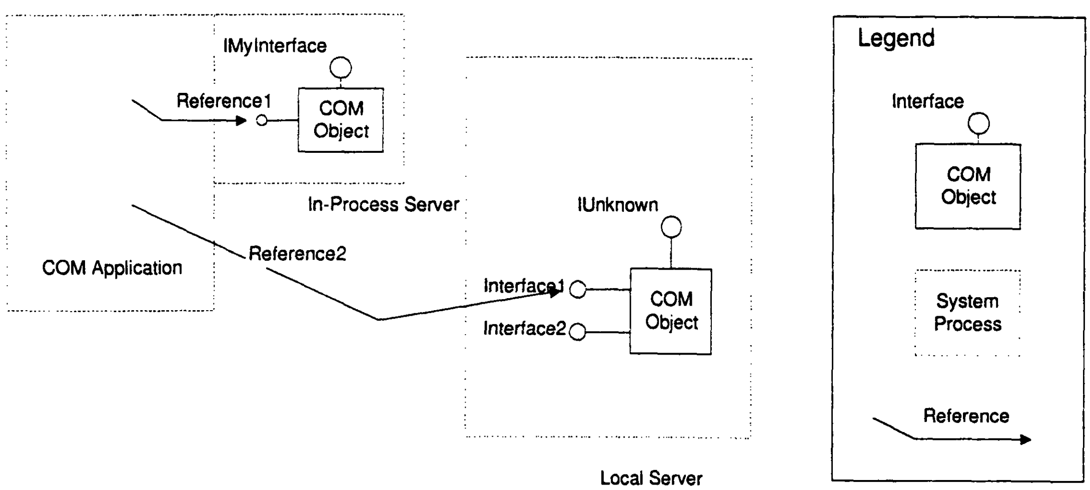


Figure 5-2 COM-based Automation

The reminder of this section outlines the incorporation of the AutoCAD ActiveX Automation Object Model into the SCHEREC Model in Section 5.2.1. Section 5.2.2 presents the RA class model as the scheduling mechanism for the SCHEREC system. Section 5.2.3 examines the Microsoft Access 97 as the project database. At last, Section 5.2.4 presents the worksheets in the Microsoft Excel 97 to generate a variety of progress reports.

5.2.1 AutoCAD ActiveX Automation Object Model

AutoCAD, developed by Autodesk, Inc. [see http://www.autodesk.com (1998)], is one of the most popular three- dimensional modeling systems for a variety of engineering domains, such as Architectural Design and Mechanical Design. However, like all other computer-aided drawing systems, AutoCAD can only recognize the meaning of a line, an arc or other geometrical shapes rather than identifying the meaning of engineering products and process. How to interpret and group geometrical elements within AutoCAD and other systems is the first challenge for the SCHREC system.

Due to the release of AutoCAD ActiveX Automation Object Model[7], AutoCAD drawing data can be retrieved and manipulated through COM interface as shown in Figure 5-3. This SCHEREC system adopts AutoCAD as the graphical Automation Engine, and thus the geometrical elements hosted by AutoCAD COM objects can be incorporated into the SHEREC product model. Thereafter, all geometrical elements, which are encapsulated by the building component class in Visual Basic [see http://www.microsoft.com (1997)] as Table 5-1 illustrates, are identified in accord with their dimension and shapes to build the SCHEREC building component hierarchy.

[7] The AutoCAD ActiveX Object Model, included in the AutoCAD Release 14, is still a beta version for testing, and Autodesk doesn't provide any technical support in that version at this time.

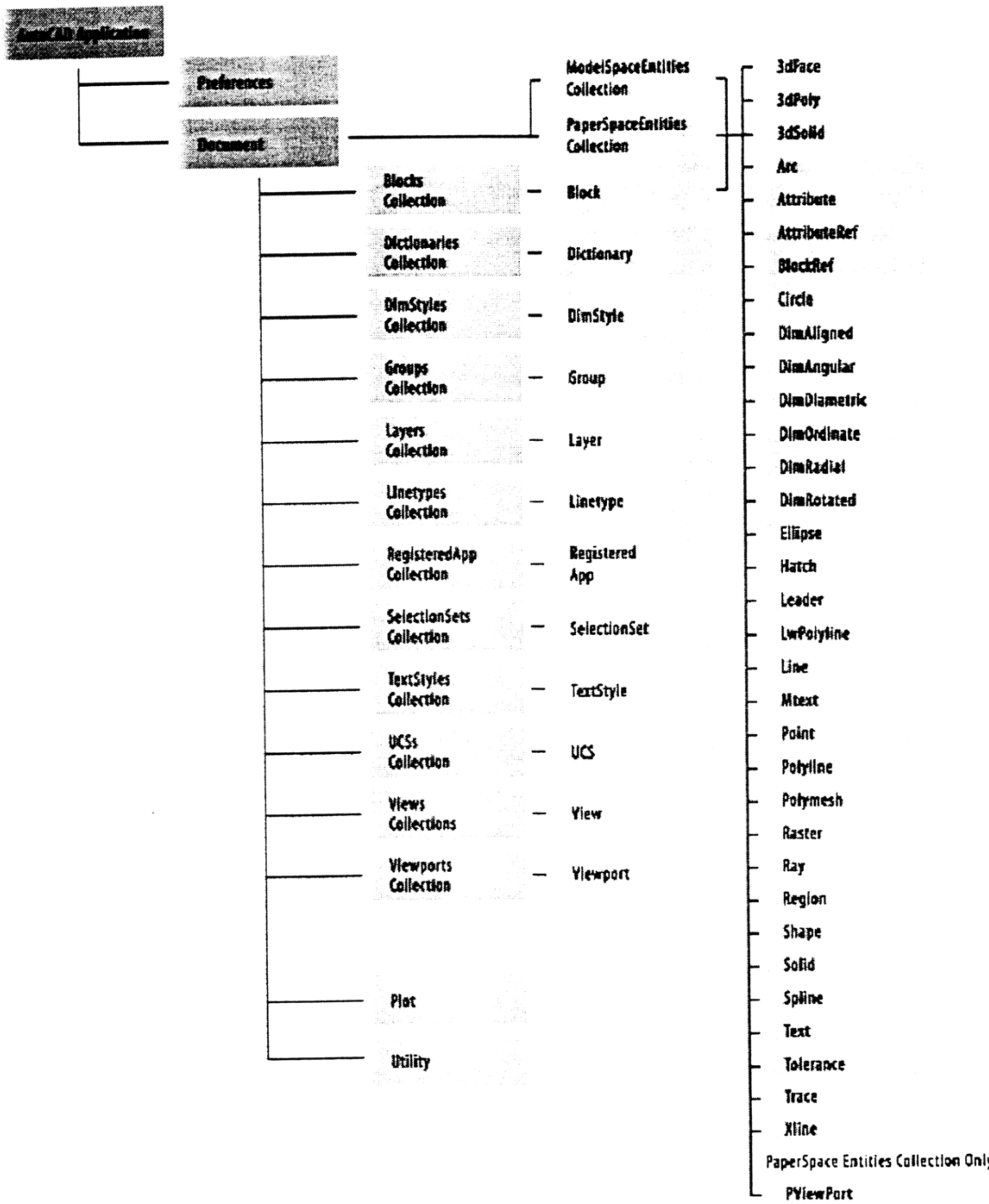


Figure 5-3 AutoCAD ActiveX Automation Object Model [Autodesk, 1997]

```
Option Explicit

Private mstrID As String
// Private Unique ID

Public bcName As String
Public bcType As String

//Dimensional Attributes
Public bcWidth As Long
Public bcLength As Long
Public bcHeight As Long

//Reference Point Attributes
Public RefX As Double
Public RefY As Double
Public RefZ As Double

// Associated Material Objects
Private mMaterial As New Collection
// Associated Construction Task Objects
Private bcConstructionTask As New Collection

Public bcGraphicObject As Object
// Pointer to AutoCAD geometric object

Property Get ID() As String
//Get Private Member Attribute
   ID = mstrID
End Property

Property Let ID(strNew As String)
//Set Private Member Attribute
   Static blnAlreadySet As Boolean
   If Not blnAlreadySet Then
      blnAlreadySet = True
      mstrID = strNew
   End If
End Property
```

Table 5-1 Building Component Class in Visual Basic

5.2.2 RA (P3 Engine) Class Model

RA Automation Engine, developed by Primavera, Inc. [see http://www.primavera.com (1998)], grants access to Primavera Project Planner's scheduling and activity-related actions through its COM interfaces. The development of RA is based on Microsoft OLE 2.0 and Open Database Connectivity (ODBC) specification [see http://www.microsoft.com (1997)]. Hence, the SCHEREC system takes advantage of the functionality of RA Automation Engine to manage the process model.

Like other COM-based Automation Engines, the RA (P3 Engine) Class Model defines the underlying object-oriented architecture for the interoperability as Figure 5-4 shows. By using the RA Software Developer Kit, developers can access and manipulate project data captured in the RA Class Model. Figure 5-4 illustrates a simplified layout of RA functional modules in EXPRESS-G.

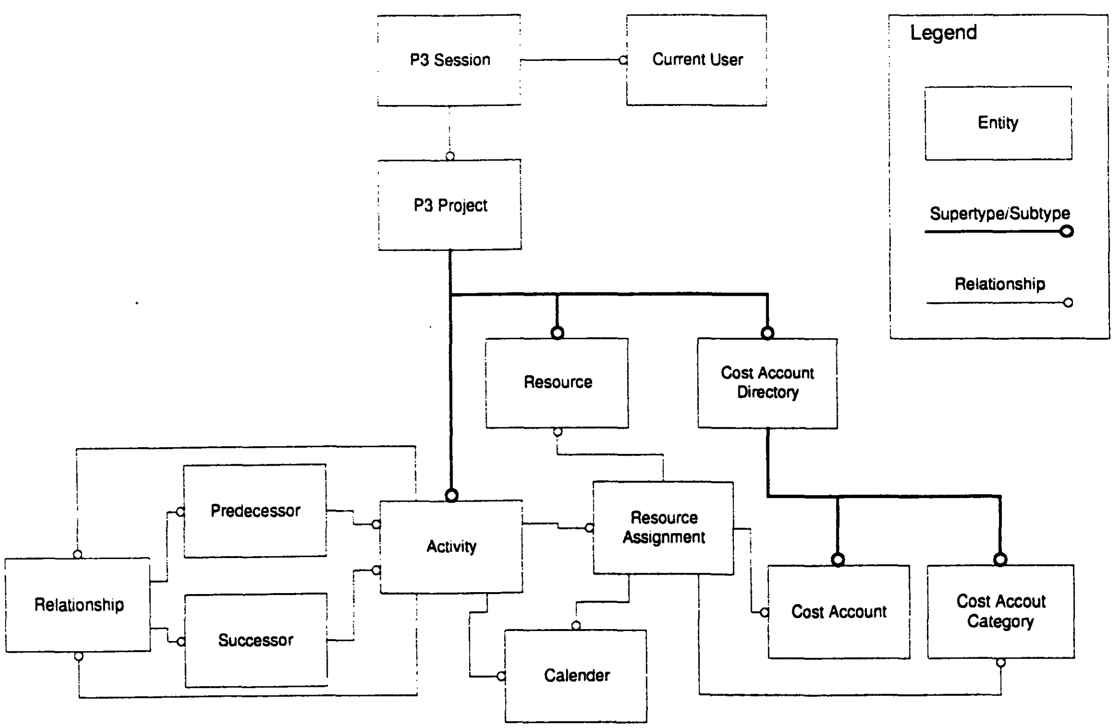


Figure 5-4 RA Class Model

After the SCHEREC system registers the system user in the "Current User" class, the SCHEREC activity class, as Table 5-2 shows, can utilize the RA class model to manage the SCHEREC process model.

```
Option Explicit

Private mstrID As String
// Private Unique Member ID

// Schedule Recovery Parameters for New Hire
Public acNHAllResource As Boolean
Public acNHSelectedResource As String
Public acNHNormalWorkingHours As Integer
Public acNHPercentage As Double
Public acNHSalaryRatio As Double
Public acNHProductivity As Double
Public acNHOverTime As Boolean

//Schedule Recovery Parameters for Over Time
Public acOTAllResource As Boolean
Public acOTSelectedResource As String
Public acOTNormalWorkingHours As Integer
Public acOTPercentage As Double
Public acOTSalaryRatio As Double
Public acOTProductivity As Double


// Pointer to RA Activity Object
Public acActivityObject As Object

Property Get ID() As String
   ID = mstrID
End Property

Property Let ID(strNew As String)
   Static blnAlreadySet As Boolean
   If Not blnAlreadySet Then
      blnAlreadySet = True
      mstrID = strNew
   End If
End Property
```

Table 5-2 SCHEREC Activity Class in Visual Basic

5.2.3 Microsoft Access 97

Microsoft Access 97 is a relational database for the Windows NT and Windows 95 environment [see http://www.microsoft.com/access, (1997)]. The implementation of the SCHEREC system utilizes the Open Database Connectivity (ODBC) technology to access the Microsoft Access 97 [see http://www.microsoft.com/access, (1997)].

Table 5-3 demonstrates the structure and sample data of the Unit_Price Table for the SCHEREC resource model. In order to expedite the system performance, the current implementation take advantage of the Resource and Resource_Use class, as Table 4-3 and 4-4 present, to manage the resource usage and reduce the amount of database queries.

ID	Resource ID	Type	Description	Unit	Unit Price
1	RS110	Labor	Concrete Worker	Man/Hrs	25.0
2	RS120	Equipment	Tower Crane	Equipment/ Hrs	2101.0
3	RS130	Labor	Crane Operator	Man/Hrs	62.5
4	RS140	Labor	Steel Worker	Man/Hrs	28.0
5	RS150	Labor	Field Engineer I - Senior	Man/Hrs	92.0
6	RS160	Labor	Plumber	Man/Hrs	27.0

Table 5-3 Unit_Price Table

5.2.5 Microsoft EXCEL 97

Microsoft EXCEL 97 [see http://www.microsoft.com/excel (1997)] provides its rich formats of reports and charts to the SCHEREC system through Microsoft DCOM interface. As the user asks the SCHEREC system to generate reports, the cells of different EXCEL spreadsheets are linked to the corresponding SCHEREC objects. Table 5-4 is a snapshot of the Gantt Chart worksheet for the Shimizu Smart Tower project. Table 5-5 presents a snapshot of the daily cash flow worksheet. Each cell in the second column represents a chronicle of the daily project expenditure. Similarly, Table 5-6 shows the cumulative cash flow worksheet. Table 5-7 outlines the daily resource allocation worksheet, presenting cost percentage of daily resource consumption. At last, Table 5-8 documents the resource assignment worksheet, presenting the resource usage through the project life cycle.

Activity	Early Start	Duration
Site Preparation	1/18/98	3
Excavation	1/22/98	5
1F StructuralFrame Erection	1/29/98	8
2F StructuralFrame Erection	1/29/98	9
3F StructuralFrame Erection	2/11/98	11
4F StructuralFrame Erection	2/26/98	15
All Column Bolting	3/19/98	5
All Column Welding	3/26/98	2
Scaffolds Hanging	3/19/98	4

Table 5-4 Gantt Chart Worksheet

Date	Daily Cost
1/20/98	$231,552.00
1/21/98	$231,552.00
1/22/98	$262,080.00
1/23/98	$262,080.00
1/24/98	$262,080.00
1/25/98	$262,080.00
1/26/98	$262,080.00
1/27/98	$262,080.00
1/28/98	$262,080.00
1/29/98	$2,126,720.00

Table 5-5 Daily Cash Flow Worksheet

Date	Cumulative Cost
1/20/98	$231,552.00
1/21/98	$463,104.00
1/22/98	$725,184.00
1/23/98	$987,264.00
1/24/98	$1,249,344.00
1/25/98	$1,511,424.00
1/26/98	$1,773,504.00
1/27/98	$2,035,584.00
1/28/98	$2,297,664.00
1/29/98	$4,424,384.00

Table 5-6 Cumulative Cash Flow Worksheet

Resource	Cost
FLD ENG*	$4,000.00
EQUIPMNT	$1,760.00
IRWK	$8,960.00

Table 5-7 Daily Resource Allocation Worksheet

DATE	EQUIPMNT	EXCAVATR	FLD ENG	FLD ENG2	[illegible]
1/20/98		$5,632.00	$800.00		
1/21/98		$5,632.00	$800.00		
1/22/98	$1,760.00	$4,928.00		$800.00	
1/23/98	$1,760.00	$4,928.00		$800.00	
1/24/98	$1,760.00	$4,928.00		$800.00	
1/25/98	$1,760.00	$4,928.00		$800.00	
1/26/98	$1,760.00	$4,928.00		$800.00	
1/27/98	$1,760.00	$4,928.00		$800.00	
1/28/98	$1,760.00	$4,928.00		$800.00	
1/29/98	$1,760.00		$4,000.00		$8,960.00

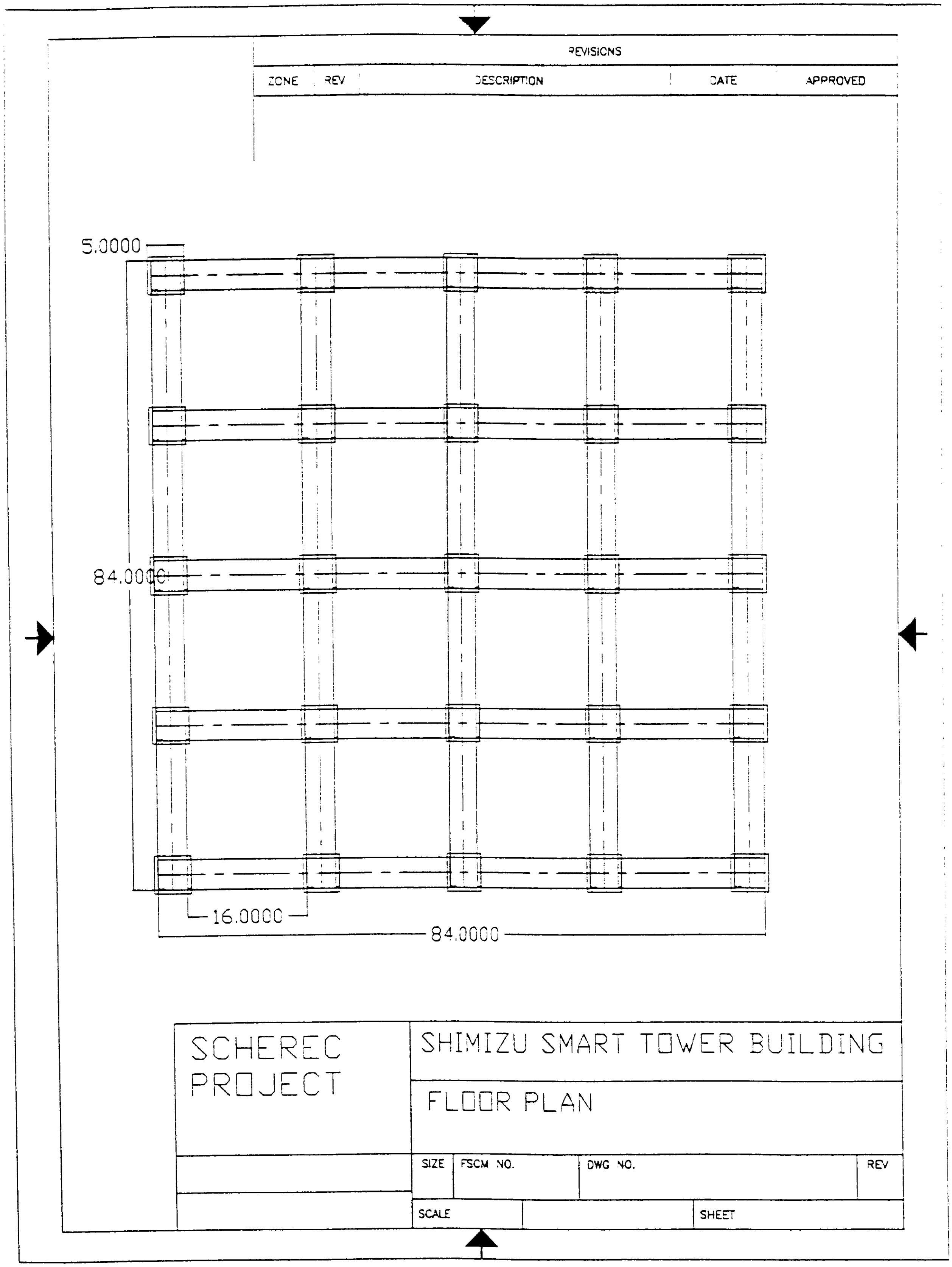
Table 5-8 Resource Assignment Worksheet

5.3 Case Study

5.3.1 The Shimizu Smart Tower

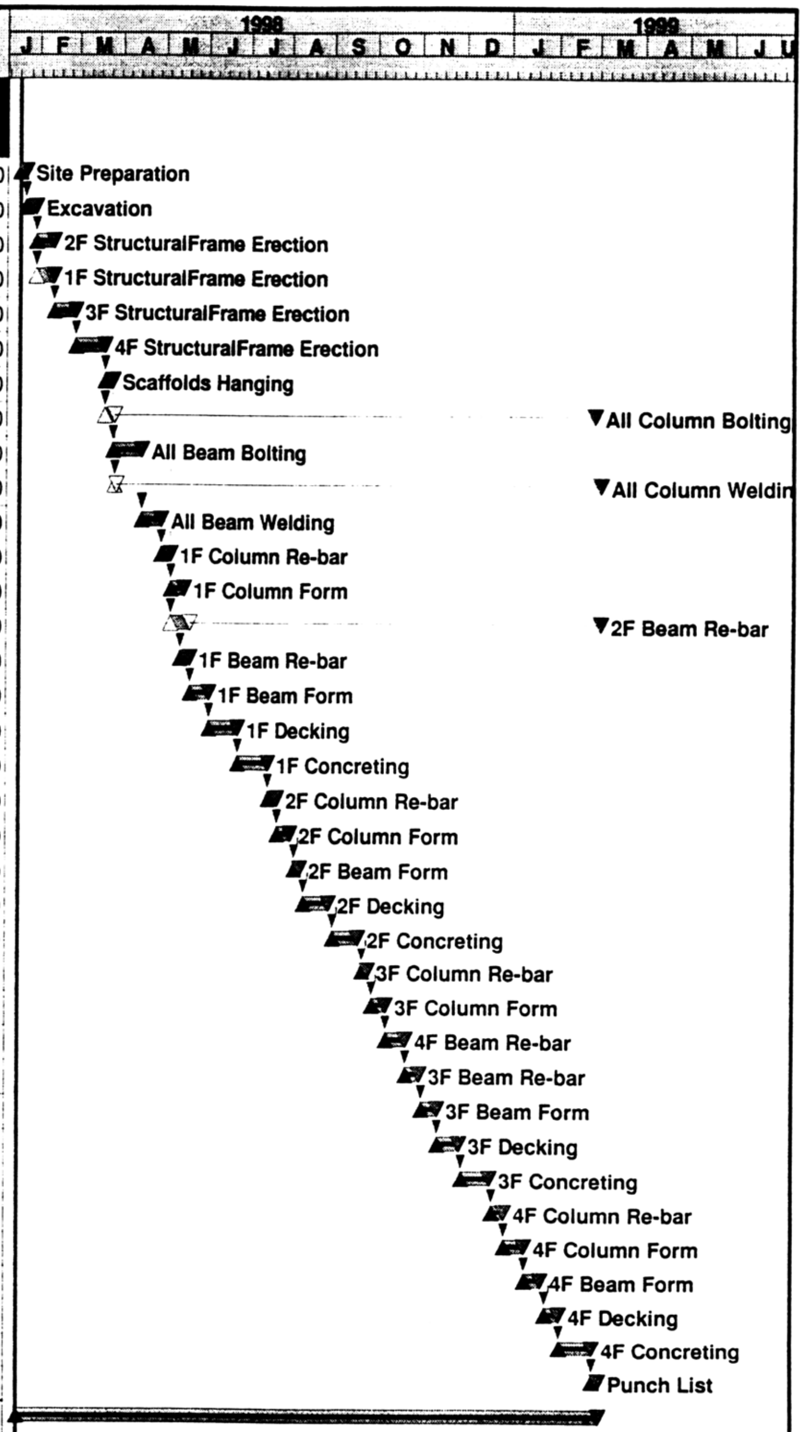
The Shimizu smart tower is located in the western part of Tokyo. The project began in 1990 and finished in 1996. The general contractor of this project was Shimizu Corporation. Since the detailed construction processes are confidential to Shimizu Corporation, some project data pertaining to cost and resource usage will not be available in this thesis.[8]

The SCHEREC product model incorporates the floor plan of the Shimizu Smart Tower as Figure 5-5 shows. The basic structure of this building is constructed as steel reinforced concrete. The logic and relationship of activity-related data, required to create SCHEREC process model, is retrieved from the real world schedules as Figure 5-6 shows. In order to expedite the simulation of construction progress, the experimental schedule starts from the January of 1998 and finished in the February of 1999 after shortening activities' duration.



[8] Please contact with Professor Peña-Mora of MIT or Dr. Minemasa of Shimizu Corporation Research for any information about the project.

Activity ID	Activity Description	Orig Dur	Rem Dur	%
Construction Department				
BA640	Site Preparation	3	3	0
BA650	Excavation	5	5	0
BA670	2F StructuralFrame Erection	9	9	0
BA660	1F StructuralFrame Erection	8	8	0
BA680	3F StructuralFrame Erection	11	11	0
BA681	4F StructuralFrame Erection	15	15	0
BA710	Scaffolds Hanging	4	4	0
BA690	All Column Bolting	5	5	0
BA712	All Beam Bolting	14	14	0
BA700	All Column Welding	2	2	0
BA720	All Beam Welding	10	10	0
BA731	1F Column Re-bar	5	5	0
BA750	1F Column Form	8	8	0
BA800	2F Beam Re-bar	10	10	0
BA770	1F Beam Re-bar	5	5	0
BA780	1F Beam Form	10	10	0
BA790	1F Decking	15	15	0
BA810	1F Concreting	15	15	0
BA820	2F Column Re-bar	5	5	0
BA830	2F Column Form	8	8	0
BA831	2F Beam Form	5	5	0
BA841	2F Decking	15	15	0
BA851	2F Concreting	15	15	0
BA861	3F Column Re-bar	5	5	0
BA871	3F Column Form	8	8	0
BA881	4F Beam Re-bar	10	10	0
BA891	3F Beam Re-bar	7	7	0
BA901	3F Beam Form	9	9	0
BA911	3F Decking	11	11	0
BA921	3F Concreting	15	15	0
BA931	4F Column Re-bar	5	5	0
BA941	4F Column Form	10	10	0
BA951	4F Beam Form	9	9	0
BA961	4F Decking	8	8	0
BA971	4F Concreting	16	16	0
BA960	Punch List	2	2	0
Subtotal		0	289	0



Project Start	24NOV97
Project Finish	01MAR99
Data Date	19JAN98
Run Date	30JAN98

Early Bar
Float Bar
Progress Bar
Critical Activity

SST4

Sheet 1 of 1

Shimizu Smart Tower

Coutesy of Shimizu Corporation

Float Bars by Department, Resp

5.3.2 Project Management Using SCHEREC System

After inputting the product and process data of the Shimizu Smart Tower project, discussed in the Section 5.3.1, the following scenario is presented to explore the capability of integrated project management system.

Once the project starts, project managers will launch SCHEREC system to perform the project management tasks in the Microsoft Windows NT environment. Figure 5-7 shows the main screen of the system, and users are asked to select the geometrical engine[9], scheduling engine[10], accounting engine[11] or dynamically loading[12] according to their specific needs.

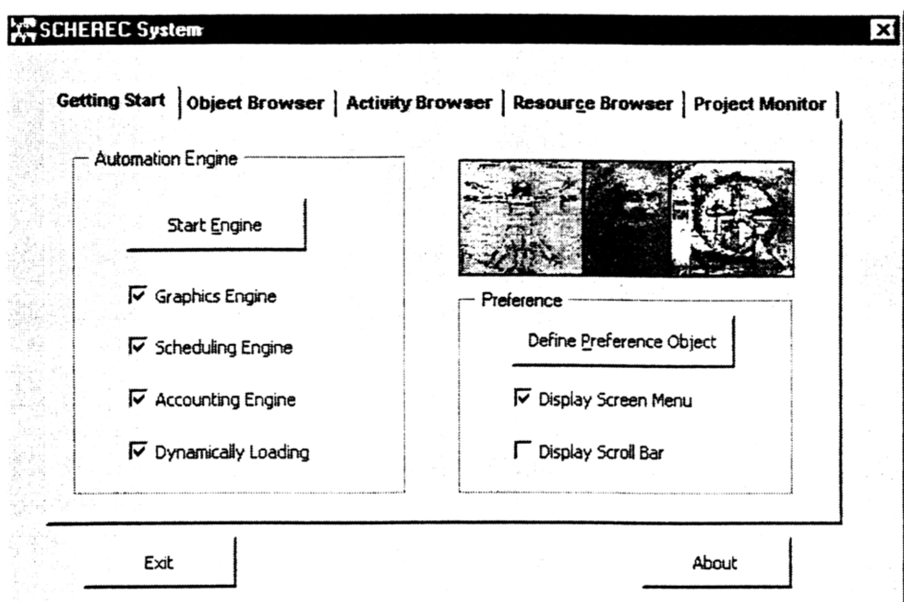


Figure 5-7 SCHEREC Main Screen

[9] In the current implementation, the SCHEREC system uses Autodesk AutoCAD Release 14 as the geometrical engine.
[10] Primavera RA (P3 Engine) 2.0b is adopted to perform scheduling tasks.
[11] Microsoft ACCESS 97 and EXCEL 97 manage both cost and resources in the current system.
[12] If the first initialization for an automation engine fails, the dynamically loading allow the SCHEREC system to connect the automation engine later.

After the initialization of selected automation engines, the SCHEREC system will ask user to select the project template for different types of project as Figure 5-8 shows.

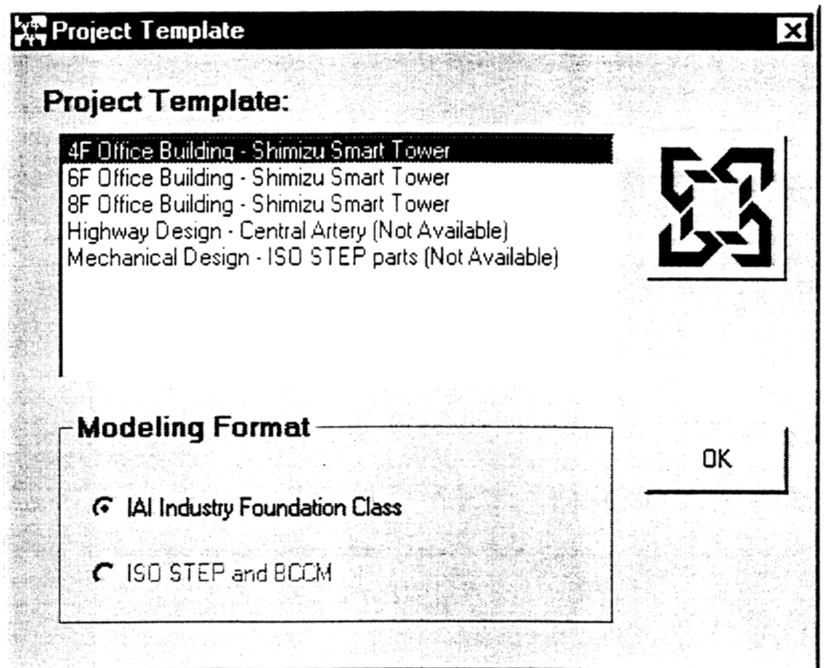


Figure 5-8 Project Template Selection

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 construction of SCHEREC classes are followed the IAI IFC standard in the current version of the SCHEREC system. In future version, the SCHEREC system 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 SCHEREC product model as demonstrated in Figure 5-9. The SCHEREC system uses different colors to denote different types of class in the AutoCAD.

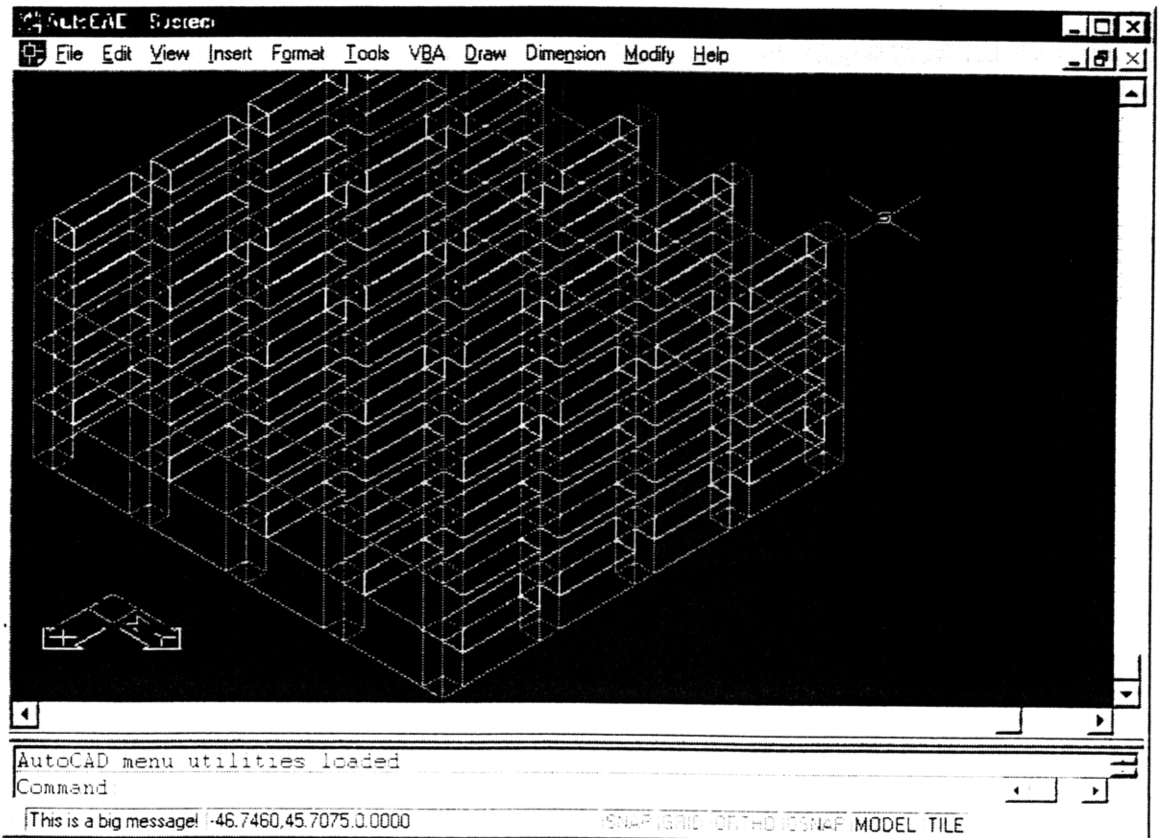


Figure 5-9 SCHEREC Product Model in AutoCAD

Thereafter, the two arrow buttons are employed to navigate the SCHEREC building component hierarchy and the object editor helps users to modify the attributes of the located entity as Figure 5-10 and 5-11 shows.

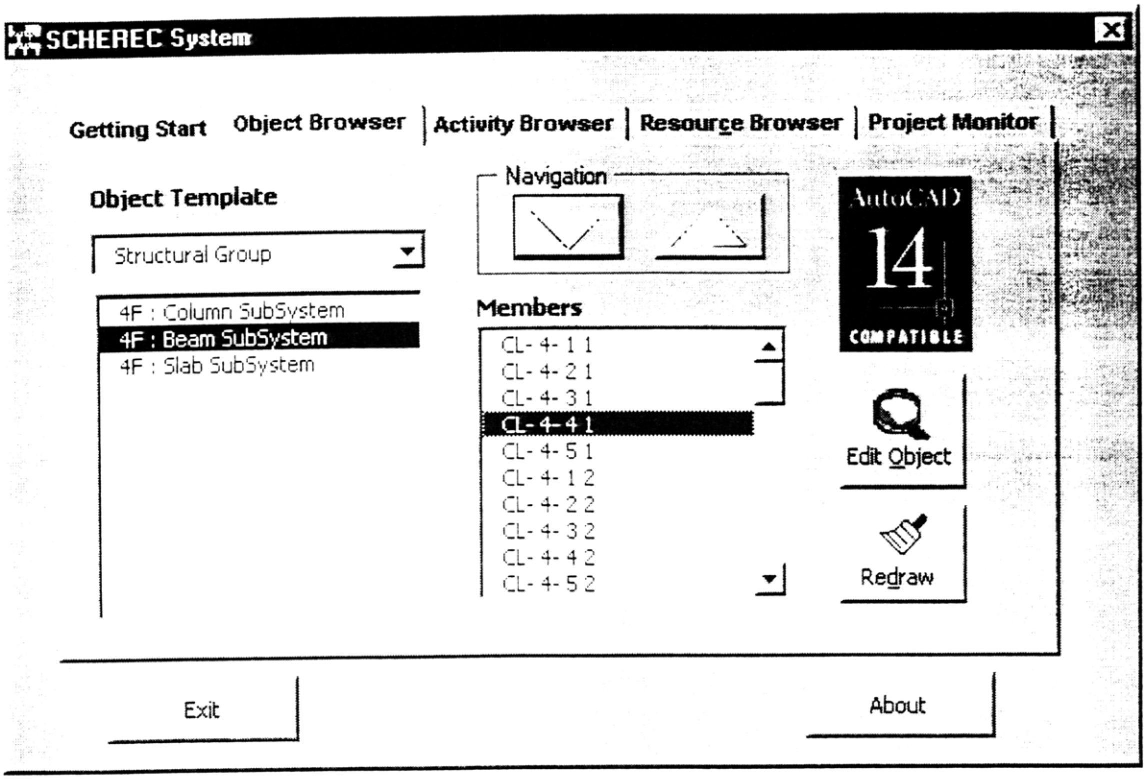


Figure 5-10 SCHEREC Object Browser

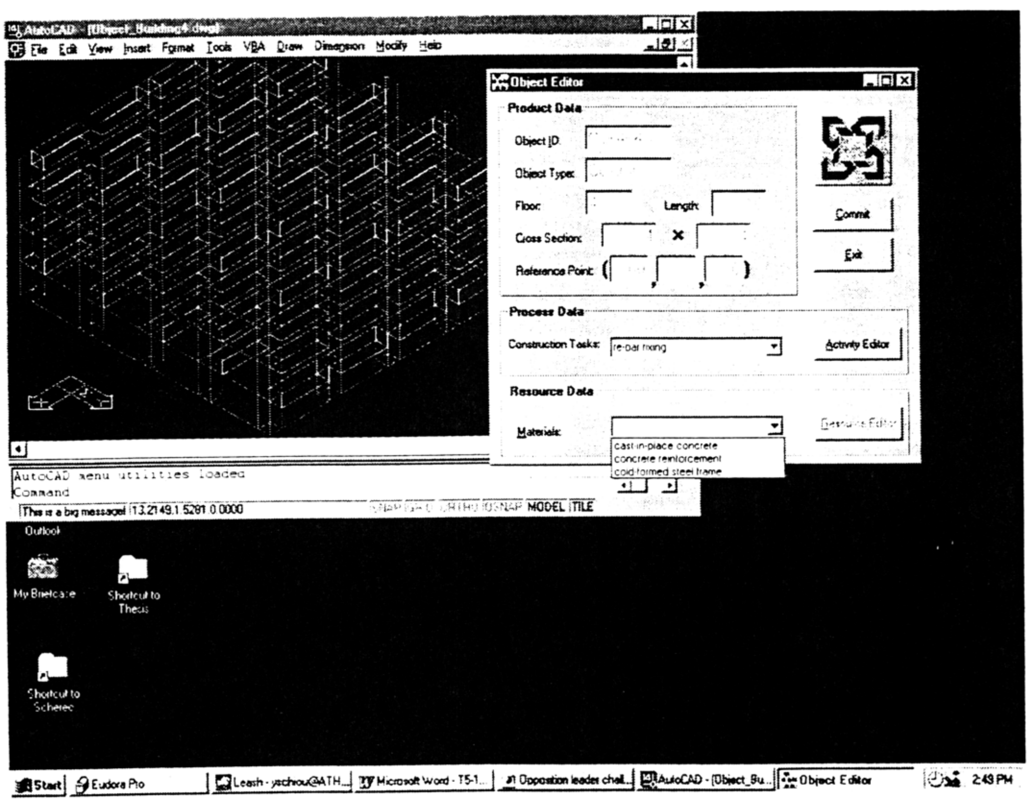


Figure 5-11 SCHEREC Object Editor

After the establishment of the SCHEREC 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 5-12 shows. Meanwhile, the SCHEREC system will organize the activities in the schedule into the SCHEREC construction task hierarchy as discussed in section 4.2. Since the SCHEREC building component hierarchy has been created in AutoCAD, the process entities in the SCHEREC construction task hierarchy are mapped to the corresponding product entities in the same position in the SCHEREC building component hierarchy.

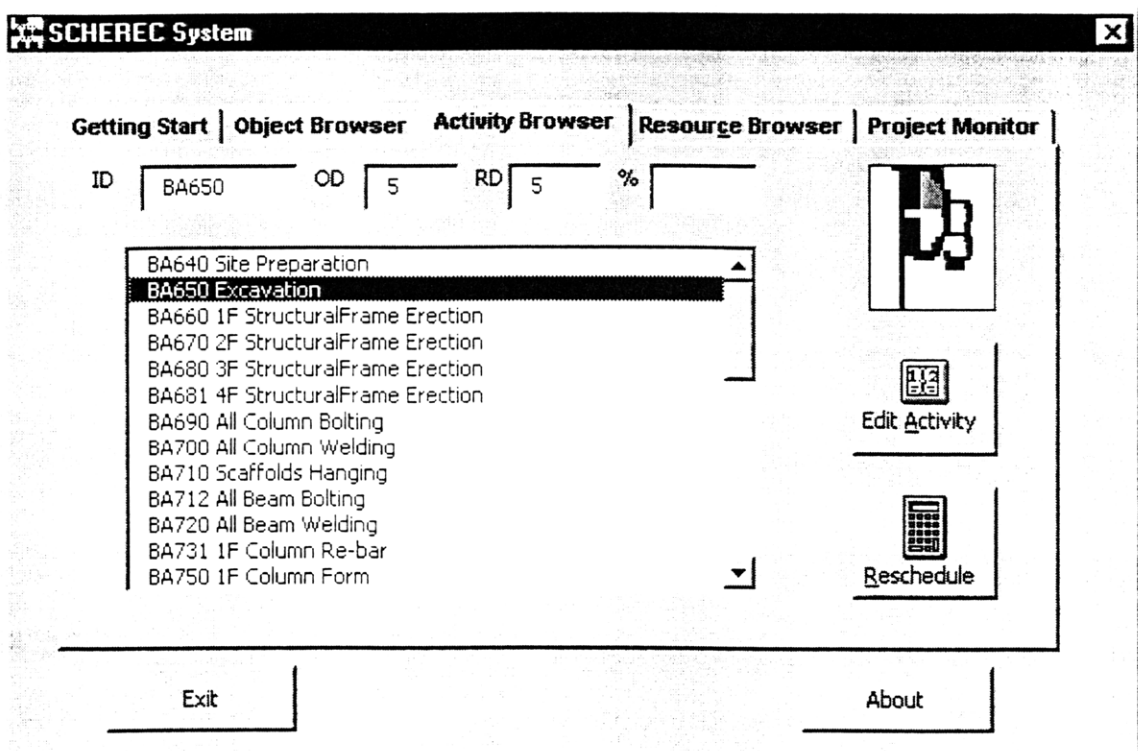


Figure 5-12 SCHEREC Activity Browser

Project managers may highlight a specific activity and press the "Edit Activity" button to launch the activity editor. As Figure 5-13 illustrates, the activity editor allows users to redesign properties of activities, such as CPM/PERT, resource assignment and schedule recovery parameters. Afterwards, user my recalculate the updated schedule by pressing the "Reschedule" button as illustrated in Figure 5-14.

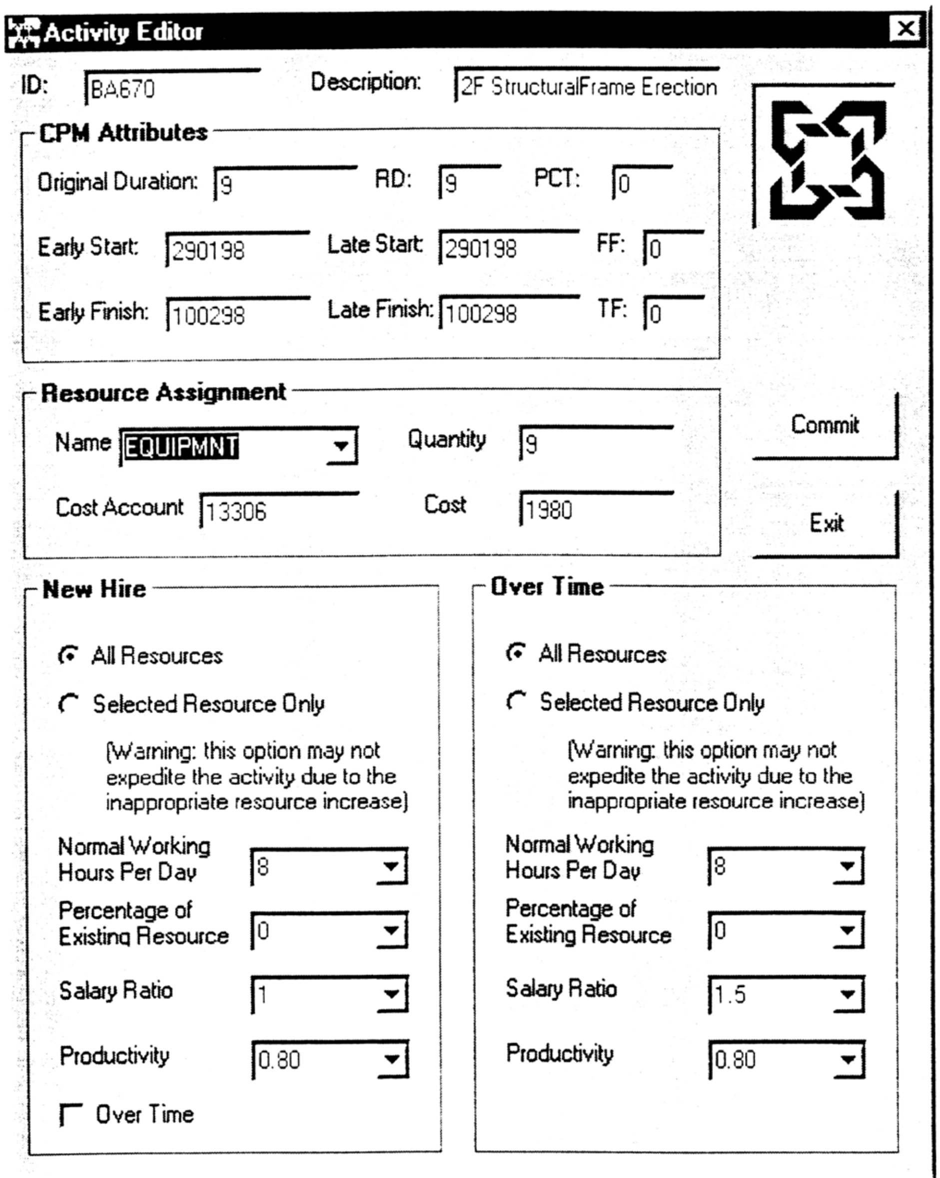


Figure 5-13 SCHEREC Activity Editor

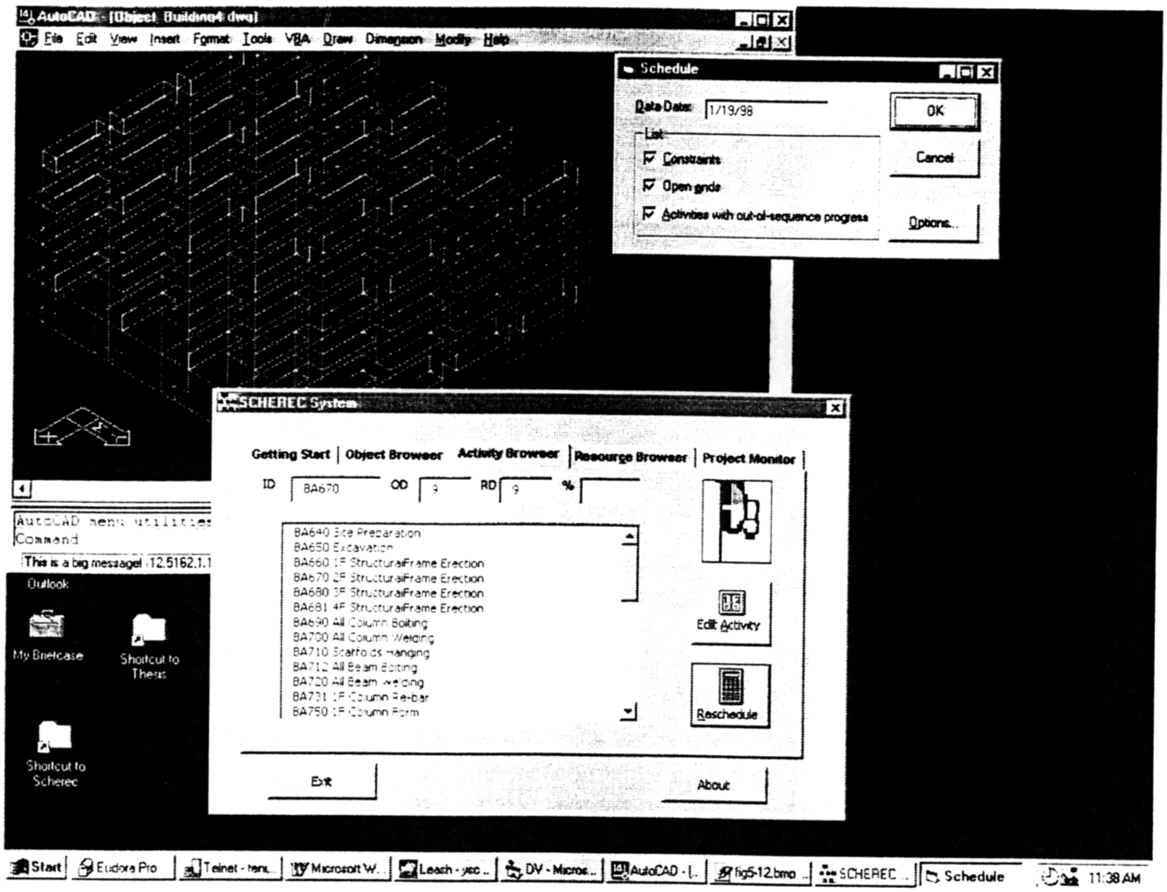


Figure 5-14 SCHEREC Rescheduling

Once the SCHEREC product and process model has been created, the resource browser instanciates 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 5-15.

Moreover, after the establishment of the product, process and resource models, the SCHEREC Object Editor enables users to monitor the corresponding activity in the

SCHEREC Activity Editor by pressing the "Activity Editor" button in Figure 5-11. Thus, this functionality allow project managers to specify, create, monitor and modify the connections between the product entity and its corresponding activities.

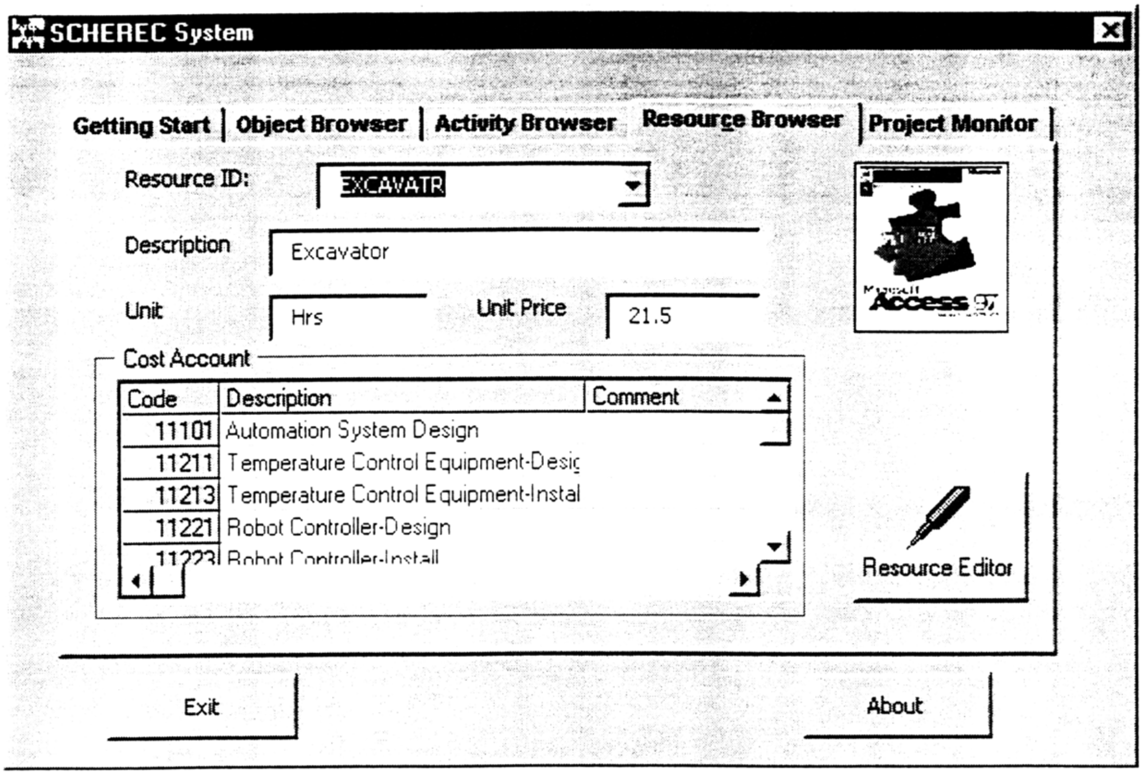


Figure 5-15 SCHEREC Resource Browser

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 5-16. When project proceeds, the construction sequence simulator help user control the speed of progress and monitor the project at any specified time as Figure 5-17 illustrates. The Gantt chart of the project is displayed in Figure 5-18 to present the graphical schedule information. The daily cash flow diagram estimates the project expenditure on a daily basis as Figure 5-19 shows. Similarly, the cumulative cash flow diagram, as Figure 5-20 shows, demonstrates the spent budget so far. The daily resource allocation diagram is illustrated in Figure 5-21, 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 5-22 shows.

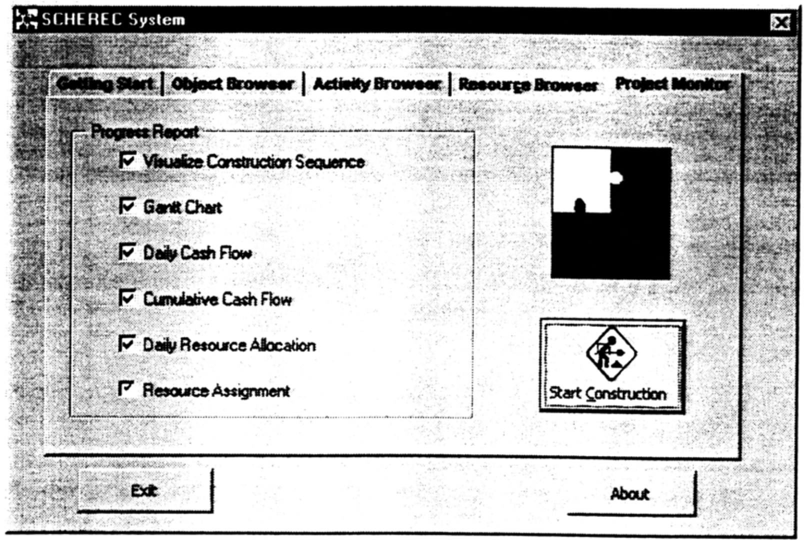


Figure 5-16 SCHEREC Project Monitor

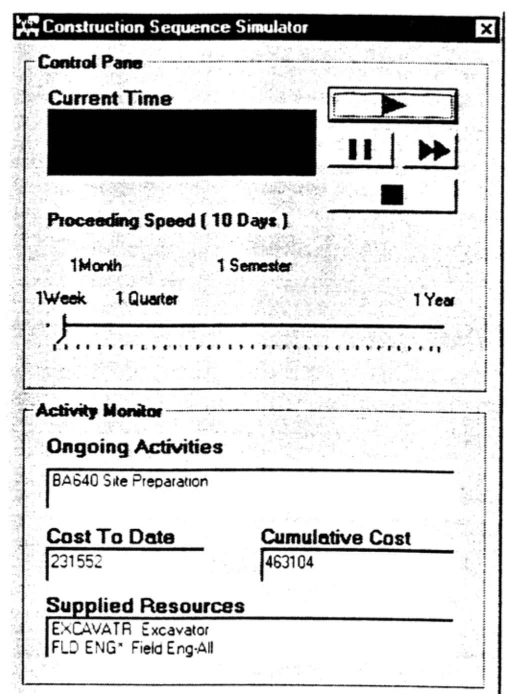


Figure 5-17 SCHEREC Construction Sequence Simulator

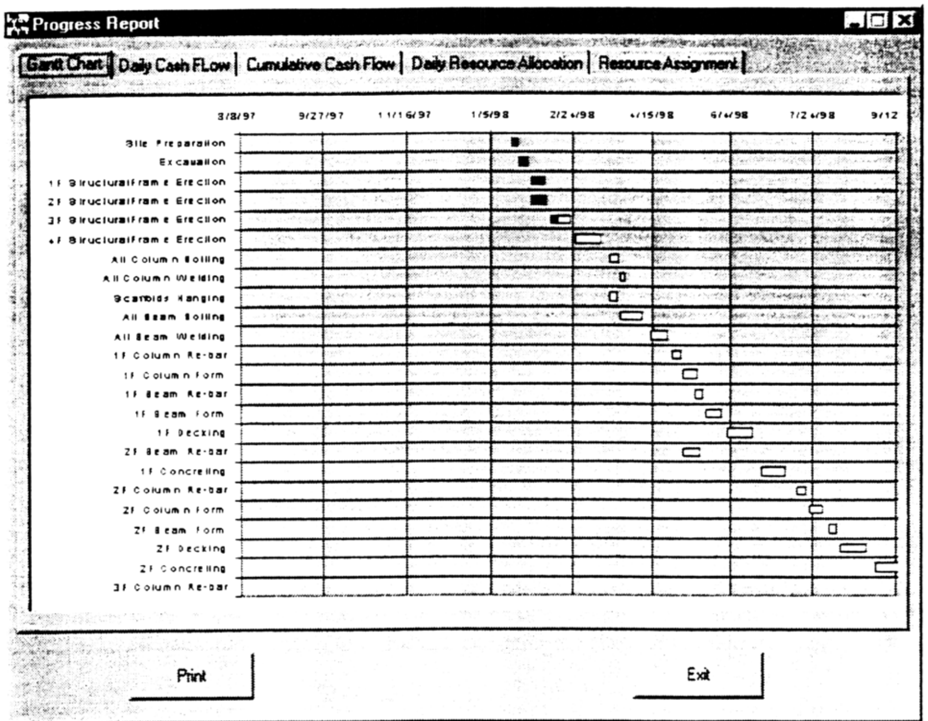


Figure 5-18 Gantt Chart

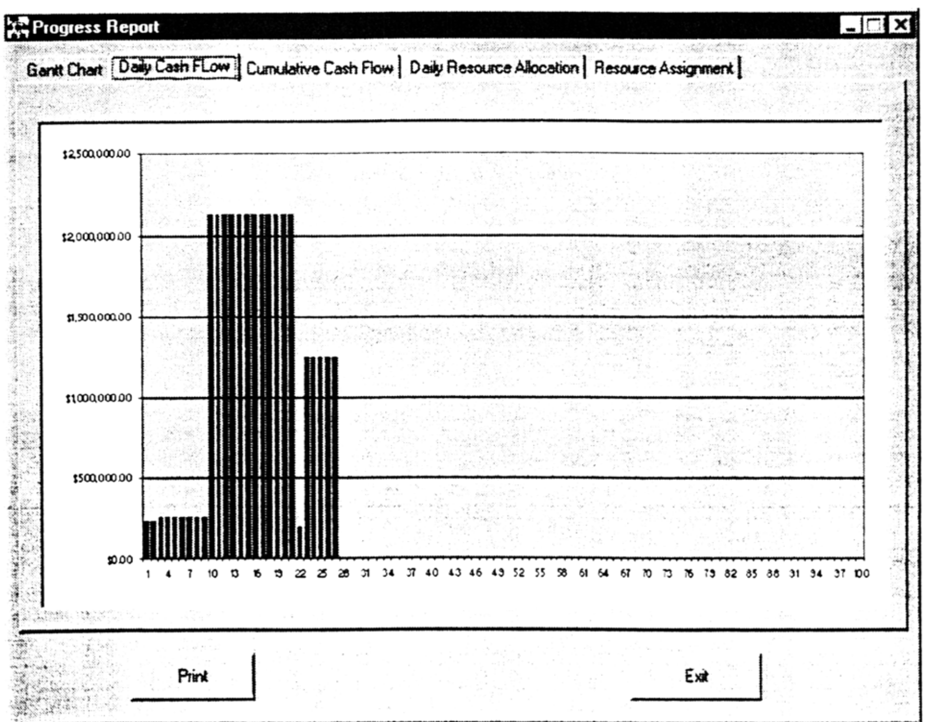


Figure 5-19 Daily Cash Flow

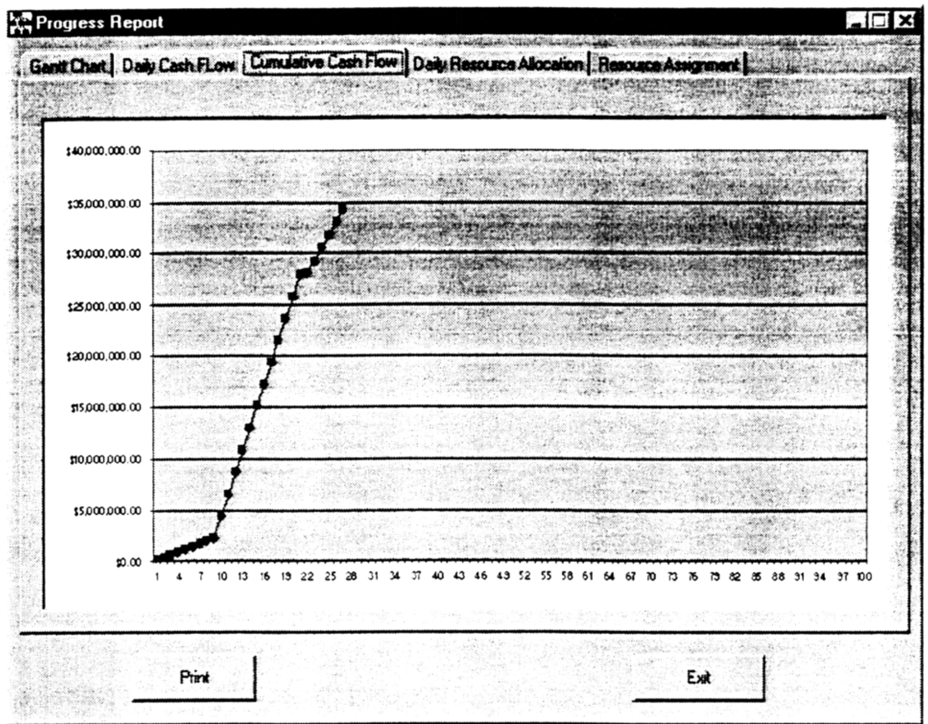


Figure 5-20 Cumulative Cash Flow

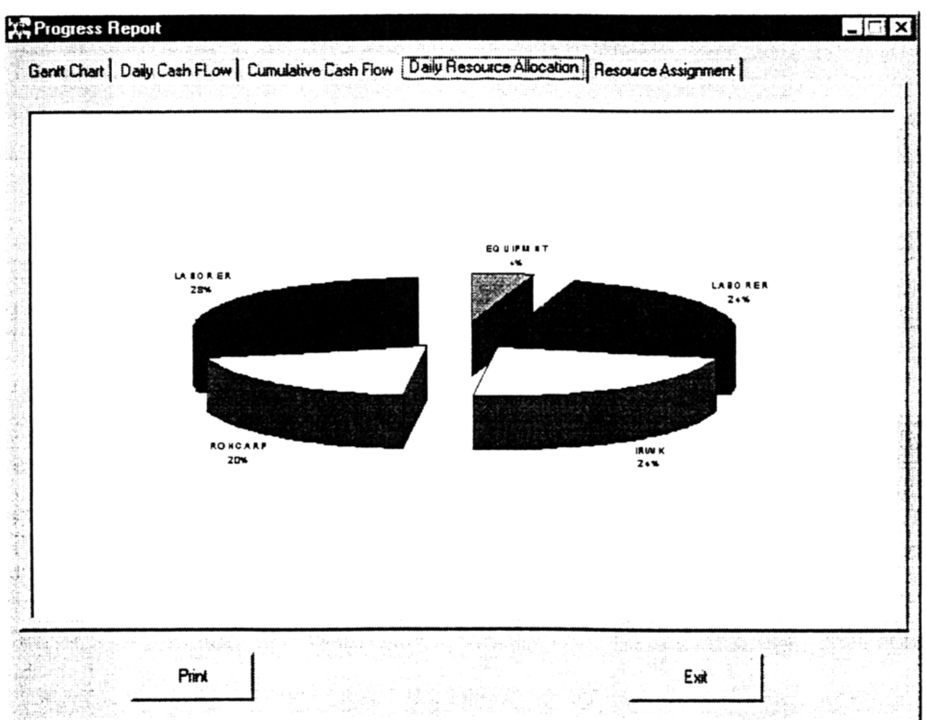


Figure 5-21 Daily Resource Allocation

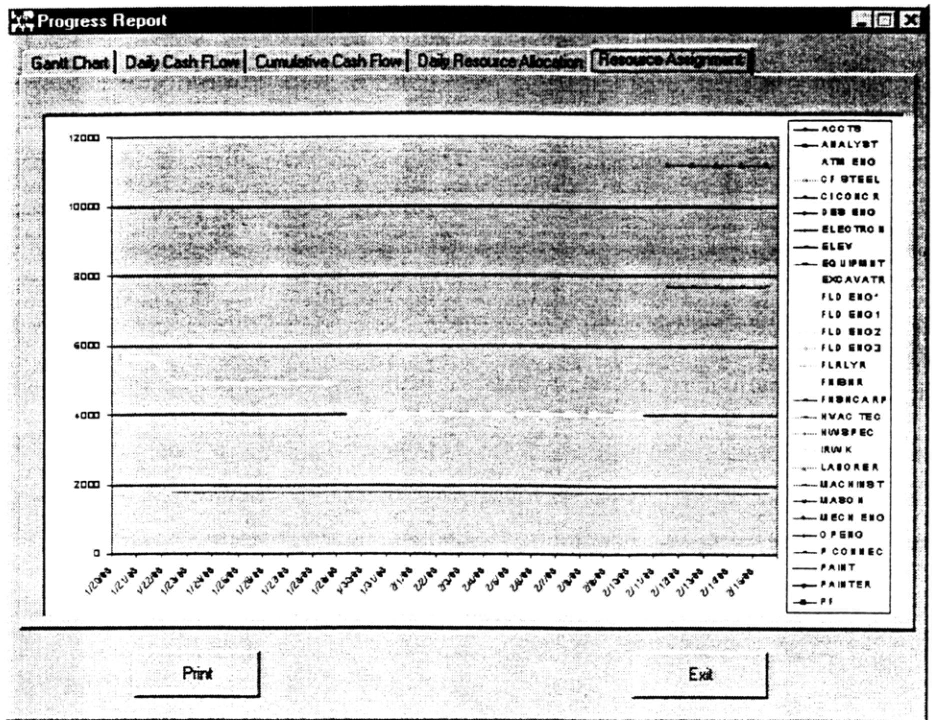


Figure 5-22 Resource Assignment

5.3.3 Schedule Recovery

The Critical Path Method (CPM) was designed to control both the time and cost aspect of a project [see Meredith and Mantel (1995)]. However, the current CPM tools do not define a clear mechanism to update the change in the architectural drawings or resource assignment. Due to the implicit linkage between project cost and CPM tools, the trade-off between time and cost is usually established by the experience of project managers.

On the other hand, 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 SCHEREC system, 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. The other method is to hire more new resources. The following formulas are adopted to estimate the impact to the entire project [see Chen (1996)].

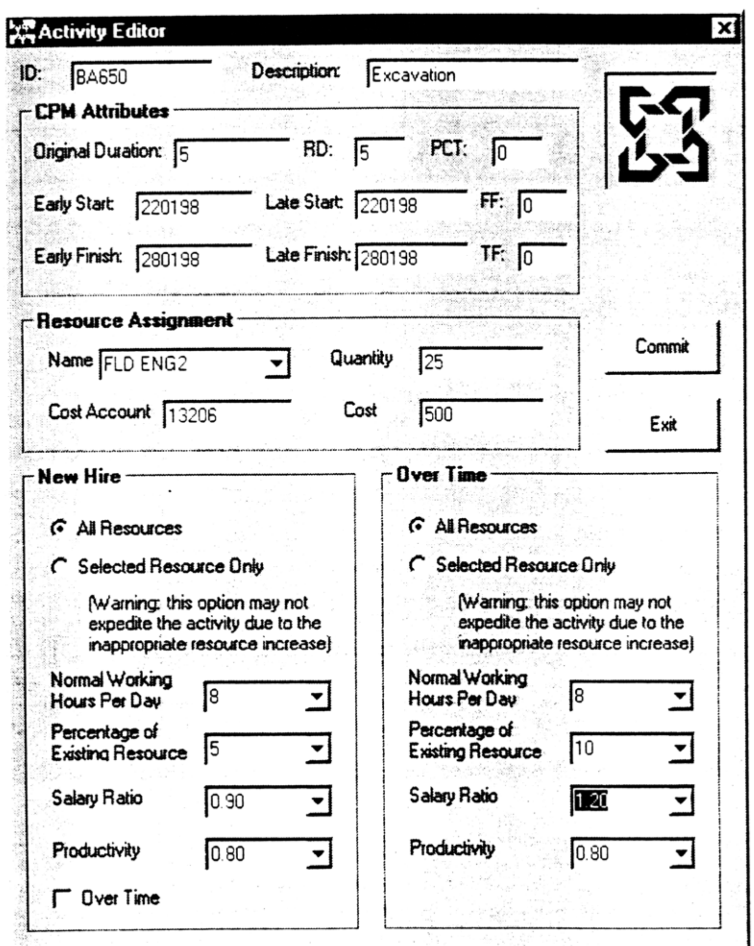


Figure 5-23 Schedule Recovery I

New Duration = Old Duration

* 1/(1+Overtime Percentage * Overtime Productivity)

* 1/(1+New Hire Percentage * New Hire Productivity)

Daily Resource Cost = Unit Price * Normal Working Hours * Resource Quantity

+ Overtime Unit Price * Overtime Hours * Resource Quantity

+ New Hire Unit Price * New Hire Normal Working Hours * New Hire Percentage * Quantity

+ New Hire Overtime Unit Price * New Hire Overtime Hours * New Hire Percentage * Quantity

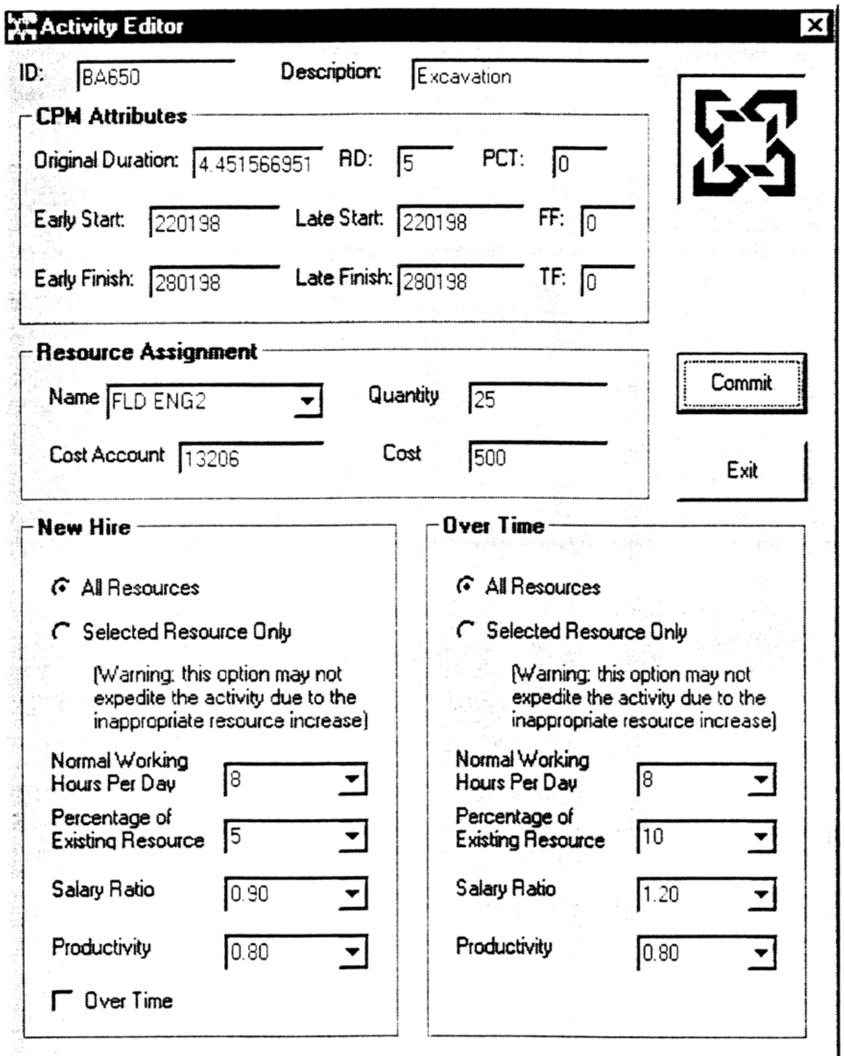


Figure 5-24 Schedule Recovery II

As the proposed activity class template discussed in Section 4.2.2, these parameters of new hire and overtime are outlined by the attributes of an activity object. Thereafter, the project monitor of the SCHEREC system 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 SCHEREC system. 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 5-23 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 5-23 illustrates, the SCHEREC activity editor allow project managers to hire 5% more new resource with salary ratio 0.9 and make existing resource 10% overtime with salary ratio1.10 for the activity, "excavation". After pressing committing, the duration of the activity will be shortened to 4.45 days as Figure 5-24 shows. Figure 5-25 and 5-26 illustrates that the additional $45,000 is in need to crash the activity for 0.55 day.

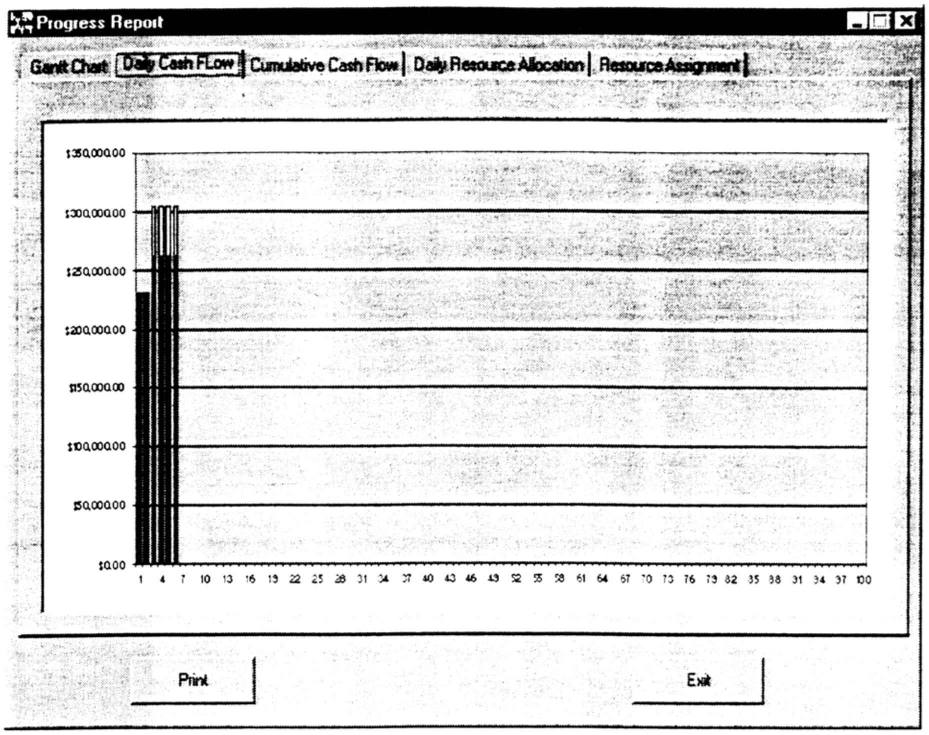


Figure 5-25 New Daily Cash Flow

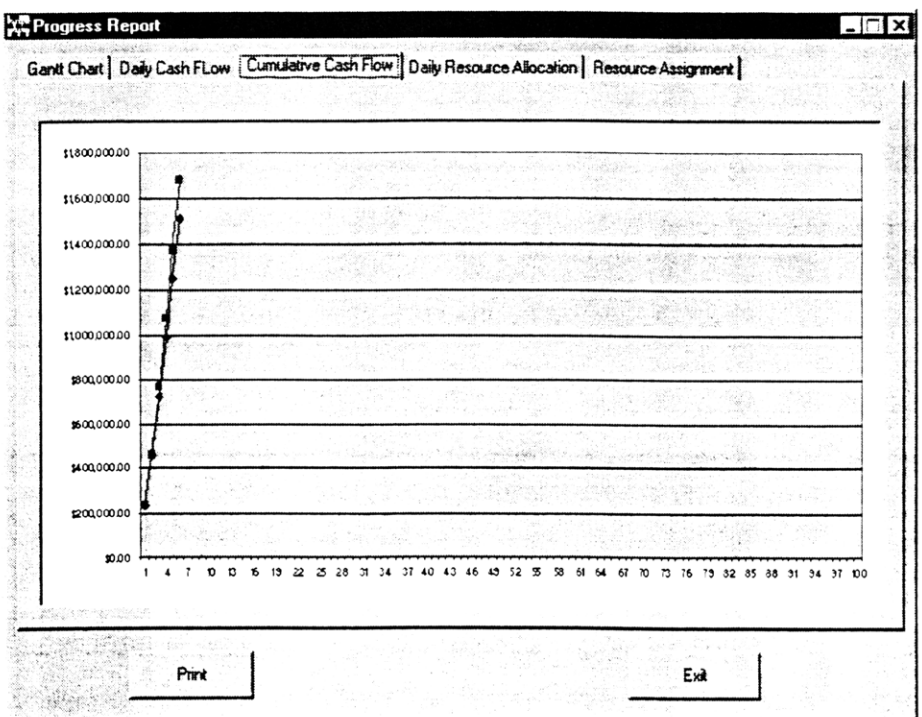


Figure 5-26 New Cumulative Cash Flow

5.4 Summary

The implementation of the SCHEREC system demonstrates the cooperation of engineering design systems and project management systems. The interoperability among AutoCAD, Primavera P3 (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 system for the real world project.

Chapter 6

Conclusions and Further Directions

The SCHEREC system presented in this thesis has demonstrated a conceptualized framework to improve the integration of project information. The underlying SCHEREC model brings a new level of openness to A/E/C domain practitioners through object-orientation and software interoperability.

The object-oriented structure of the proposed SCHEREC model is categorized into three different aspects of project management: product, process and resource. The integration of these three sub-models enhances information-sharing and data exchange from multiple electronic systems. Thus, the concept of the SCHEREC model can be extended to the cooperation of diverse project management domains, such as design, scheduling, resource assignment, budgeting, cost estimating and change management systems.

Moreover, 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[1] 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 [see http://www.omg.org (1997)].

A preliminary architecture is illustrated as Figure 6-1 shows.[2] 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 [see http://www.microsoft.com (1997)].

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.

[1] For example, Finite Element Analysis (FEA) systems [see http://www.adina.com (1998)] and negotiation tools, such as MIT CAIRO system [see Hussein (1997)].
[2] The diagram is very preliminary. Please do not quote.

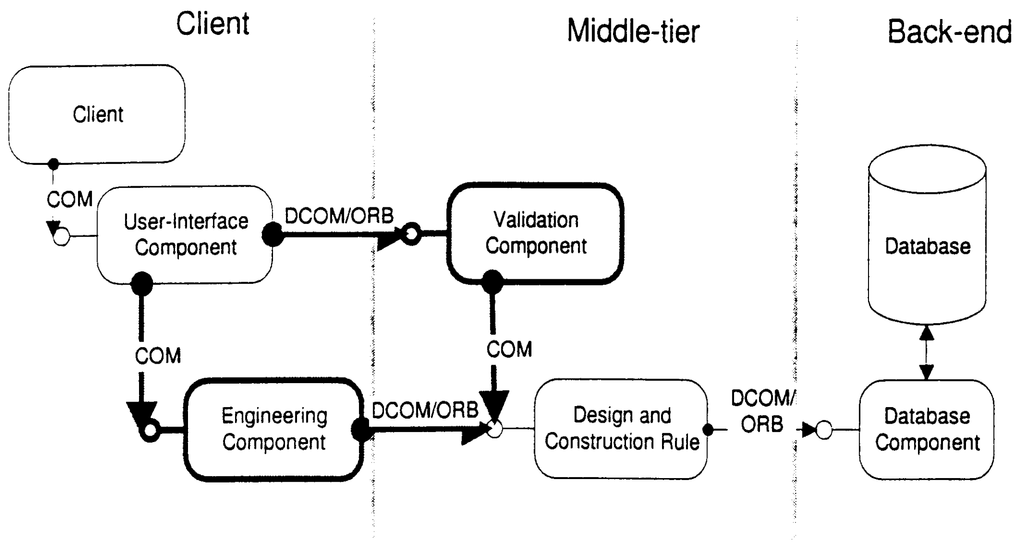


Figure 6-1 Distributed Computing Architecture

Bibliography

[1] Autodesk, Inc. (1997), "Introduction to ActiveX Automation in AutoCAD Release 14," *http://www.autodesk.com*

[2] Adina R&D, Inc (1998), "Streamlined CAD with Adina," http://www.adina.com

[3] Booch, Grady. (1991) "*Object-Oriented Design with Applications*." Benjamin and Cummings Pub. Co., Redwood City, CA.

[4] Booch, Grady. (1996) "*Best of Booch: Designing Strategies for Object Technology.*" Rational Software, Santa Barbara, CA

[5] Bjork, B.-C. (1992) "The Topology of Spaces, Space Boundaries and Enclosing Structures In Building Product Data Models." CIB W78 workshop on Computer-Integrated Construction, Montreal, Canada.

[6] Chappell D. (1996), "*Understanding ActiveX and OLE*," Microsoft Press, Microsoft, Redmond, WA.

[7] Chen, H. (1996), "*Web Based Project Management System,*" Master Thesis, MIT, Cambridge, MA.

[8] Cherneff, J. M. (1988) "*Automatic Generation of Construction Schedules from Architectural Drawings*," Master Thesis, MIT, Cambridge, MA.

[9] Denning A. (1997), "ActiveX Controls Inside Out", Microsoft Press, Redmond, WA.

[10] Field, R.C. (1997), "*Knowledge Represetation for Constructed Facilities,*" Master Thesis, MIT, Cambridge, MA.

[11] Fowler, M., and Scott, K. (1997), "*UML Distilled : Applying the Standard Object Modeling Language,*" Addison-Wesley Object Technology Series, Santa Barbara, CA.

[12] Froese, T., and Waugh, L. (1991) "Project Management and Computers in the Year 2010." *Proceeding of Annual Conference of the Canadian Society of Civil Engineering*, Canadian Society of Civil Engineering, Vancouver, Canada, pp. 435-444.

[13] Froese, T., and Paulson, B. (1994) "OPIS, an Object-Model-Based Project Information System." *Microcomputer in Civil Engineering*, 9(Feb.), pp. 13-28, Elservier, New York, NY.

[14] Gann, D., Hansen, K. L., Bloomfield, D., Blundell, D., Crotty, R., Groak, S. and Jarrett, N. (1996) *"Information Technology Decision Support in the Construction Industry: Current Developments and Use in the United States,"* Department of Trade and Industry Overseas Science and Technology Expert Mission Visit Report, University of Sussex, Brighton, UK.

[15] Graham, I. (1994) *"Object-Oriented Methods."* Swiss Bank Corporation, London, UK.

[16] Hussein, K. (1995) "*Communication Facilitators for a Distributed Collaborative Engineering Environment.*" Master Thesis, MIT, Cambridge, MA 02139

[17] International Alliance for Interoperability (1997), "AEC Processes Supported by IFC," *Industry Foundation Classes Release 2.0*, International Alliance for Interoperability, Washington D.C.

[18] International Alliance for Interoperability (1997), "IFC Object Model For AEC Projects," *Industry Foundation Classes Release 2.0*, International Alliance for Interoperability, Washington D.C.

[19] International Alliance for Interoperability (1997), "IFC Model Exchange," *Industry Foundation Classes Release 2.0*, International Alliance for Interoperability, Washington D.C.

[20] International Alliance for Interoperability (1997), "IFC Model Software Interfaces", *Industry Foundation Classes Release 2.0*, International Alliance for Interoperability, Washington D.C.

[21] International Organization for Standardization (1997), "Introduction to ISO," *http://www.iso.ch*

[22] Knowledge Based Systems, Inc. (1997), "The U.S. Air Force IDEF Methods," *http://www.idef.com*

[23] Martin, F., and Aalami, F. (1996), "Scheduling with Computer-Interpretable Construction Method Models", *Journal of Construction Engineering and Management*, American Society of Civil Engineers, Vol. 122, No. 4.

[24] Object Management Group (1997), "CORBA: Non-Standardized Application-Specific Interfaces", *http://www.omg.org*

[25] Peña-Mora, F., Sriram, D. and Logcher R. (1995), "Design Rationale for Computer-Supported Conflict Mitigation," *Journal of Computing in Civil Engineering*, American Society of Civil Engineers, Vol. 9, No. 1., pp. 57 – 72.

[26] Primavera, Inc. (1997), "RA Primavera Automation Engine", *http://www.primavera.com*

[27] Rational Software, (1997), "Unified Modeling Language Summary," http://www.rational.com

[28] Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F. and Lorensen, W. (1991), "*Object-Oriented Modeling and Design*." Prentice Hall, Englewood Cliffs, NJ.

[29] South Carolina Research Authority, (1997), "Standard for the Exchange of Product Model Data," http://www.scra.org

[30] Stumpf, A., Chin, S., Liu, L., and Ganeshan, R. (1995), "Use of a Relational Database System to Integrate Product and Process Information during Construction." *Modeling of Buildings through Their Life-Cycle; Proceeding of CIB W78/TG10 Workshop,* CIB Publication No. 180, CIB, Rotterdam, The Netherlands, pp. 316-326.

[31] Wix, Jeffrey (1996) "*The EXPRESS Data Definition Language Introduction*", CEIC/IAI, International Alliance for Interoperability, Washington D.C.

[32] Wix, Jeffrey (1997) "Purpose of a Core Model", *Computerised Exchange of Information in Construction Project*, UK Department of the Environment, London, UK.

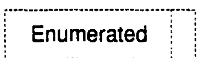
Appendix A

Modeling Notations

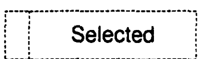
A.1 EXPRESS-G Notations

-entity

-data type



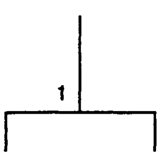
-enumerated



-selected

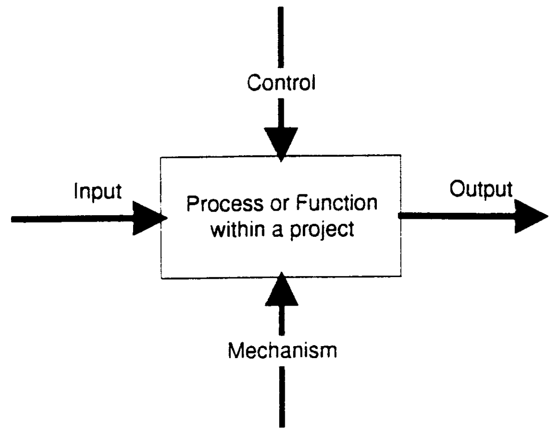
-subtype/supertype

-relationship

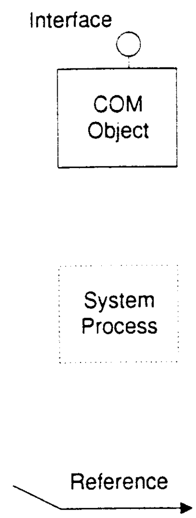


-tree structure

A.2 IDEF0 Notations



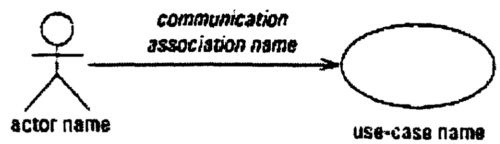
A.3 Microsoft DCOM Notations



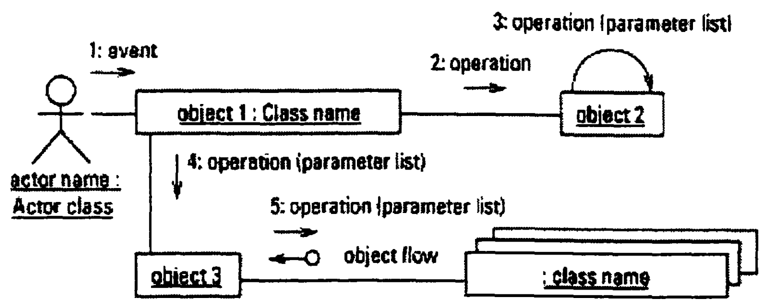
A.4 UML Notations [taken from UML Quick Reference, Rational Software (1997)]

Use Case Diagram

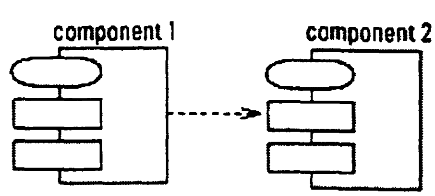
Actor, use case, and association



Collaboration Diagram



Component Diagram



Appendix B

The Source Codes of the SCHEREC System

For all the software code used in this thesis, please contact Professor Feniosky Peña-Mora for further information.

B.1 Class Modules

BcActivity.cls
BcEntity.cls

B.2 Visual Basic Forms

frmActvityEditor.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

B.3 Visual Basic Module

HierarchyOperation.bas

B.4 Visual Basic Designer Module

Frm4Dmain.dsr

B.5 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

B.6 Microsoft Office Files

Resource.mdb
CashFlow.xls
CumulativeCashFlow.xls
DailyCashFlow.xls
Schedule.xls

B.7 Primavera P3 and RA Files

Schcac2.p3
Schcacc.p3
Schcact.p3
Schccal.p3
Schcdir.p3
Schcdst.p3
Schchol.p3
Schcitm.p3
Schclog.p3
Schclay.p3
Schclog.p3
Schcplt.p3
Schcrel.p3
Schcrep.p3
Schcres.p3
Schcrlb.p3
Schcspr.p3
Schcstr.p3
Schcstw.p3
Schcttl.p3
Schcwbs.p3
Users50.p3