Model-Based Integrated Total Project Management Systems,

with an Emphasis on Materials Management

By

ABDOLHAMID GHANBARI

B.Arch.Eng., Architectural Engineering and Urban Planning and Design,
Shahid Beheshti University, Tehran, Iran, 1985

M.Eng., Building Engineering, Concordia University, Montreal, Canada, 1995

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

THE FACULTY OF GRADUATE STUDIES

(Civil Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

May 2006

Abstract

This dissertation addresses the field of computer integrated construction (CIC), which encompasses interoperability among architecture, engineering, construction, and facility management (AEC/FM) computer systems for the purpose of improving the efficiency and effectiveness in the AEC/FM industry. It focuses on model-based integrated total project management (PM) systems: a class of computer systems that include a suite of applications that support a wide range of PM functions and can flexibly and openly contribute to and draw from a shared pool of project information (referred to as a unified project object model) irrespective of the type of environment.

Over the past two decades, many CIC projects have worked towards developing the basic building blocks of CIC; i.e., the formalization and standardization of AEC/FM project information. While the results are encouraging, they have not yet reached the level of richness and comprehensiveness required by CIC. Among the major problems identified by this research are: lack of coverage of the full range of PM functions, lack of standardization of the processes through which information flows, and lack of integration of business process and information models. The overall objective of the research was to investigate (within the context of CIC) how AEC/FM product and process information could be integrated into a unified AEC/FM project model that could support a wide range of construction PM functions, with an emphasis on materials management (MM). This was done through a coupling of top-down analytical activities (from high-level conceptual analyses to detailed modeling) and bottom-up activities (from data collection activities to analysis and design). The bottom-up investigations, which were critical in informing, providing the research criteria, and validating the models produced in the top-down activities, include a review of current project and construction management software applications, software prototyping, field data-and- process analyses, and a Process Modeling Practice Workshop.

This dissertation includes a comprehensive, critical literature review and a description of the research activities and deliverables. The primary research deliverables presented are: 1) a conceptual AEC/FM Project Core Model (APCM), 2) a Conceptual PM Functions Framework (CPMF) for classifying PM functions, 3) a classification of PM/MM functions, 4) a model for the integration of AEC/FM business and software system models, called the Integrated Total Project Management Systems (ITPMS) model, 5) an implementation of the ITPMS model in a CASE (Computer-Aided Software Engineering) tool, and 6) a set of MM domain object models and process models. A special emphasis is placed on the "modeling process" and derivation of information through business process modeling (i.e., explicit linkage of the two types of models) for a better model management. The proposed solutions collectively represent both a proof-of-concept and the means of realization of model-based integrated total PM systems. It is envisioned that they will be informing and will contribute to the CIC initiatives.

TABLE OF CONTENTS

List of Tables

List of Figures

List of Abbreviations

AEC:	Architecture, Engineering, and Construction
AEC/FM:	Architecture, Engineering, Construction, and Facility Management
AP:	Application Protocol
APCM:	AEC/FM Project Core Model
API:	Application Programming Interface
BCCM:	Building Construction Core Model
BPR:	Business Process Reengineering
CAD:	Computer-Aided Design
CASE:	Computer-Aided Software Engineering
CIC:	Computer Integrated Construction
CIM:	Computer Integrated Manufacturing
CM:	Construction Management
CPMF:	Conceptual Project Management Framework
DBMS:	Database Management System
FFB/P:	Form, Function, and Behavior (or Performance)
HR:	Human Resource
HRM:	Human Resource Management
IAI:	International Alliance for Interoperability
IFC:	Industry Foundation Classes
IDEF:	ICAM (Integrated Computer Aided Manufacturing) Definition
$IDEF_0$:	Integrated Definition for Function Modeling
IDEF1X:	Integrated Definition for Information Modeling Extended
IR:	Integrated Resource
ISO:	International Organization for Standardization
IT:	Information Technology
ITPMS:	Integrated Total Project Management Systems
JDS:	Jigsaw Distributed System
JMT:	Jigsaw Modeling Tool
MDI:	Multiple Document Interface
MM:	Materials Management
MS-VB:	Microsoft Visual Basic™
MS-Access:	Microsoft Access™

OOAD:	Object-Oriented Analysis and Design
OOSE:	Object-Oriented Software Engineering
PM:	Project Management
PMBOK:	Project Management Body of Knowledge
PMPW:	Process Modeling Practice Workshop
PMI:	Project Management Institute
QM/QC:	Quality Management/ Quality Control
RM:	Resource Management
SADT:	Structured Analysis and Design Technique
SGML:	Standard Generalized Markup Language
STEP:	STandard for the Exchange of Product model data
UI:	User Interface
UML:	Unified Modeling Language
XML:	eXtensible Markup Language

Acknowledgement

In The Name of God, The Compassionate, The Most Merciful

This dissertation was made possible with contribution of many individuals and organizations that deserve my sincere thanks. Specifically, I would like to express my sincerest gratitude to:

- My research supervisor, Dr. Thomas Froese, for his encouragement throughout the course of research and his valuable guidance and comments on several drafts of the dissertation.
- My other research committee members, Dr. Alan Russell (Civil Engineering) and Dr. Jerzy Wojtowicz, (Architecture) for their time spent on reviewing the dissertation and for their profound comments, which helped improve the quality of the dissertation dramatically.
- Dr. Richard Rosenberg (Computer Science), Dr. Tarek Sayed (Civil Engineering), Dr. Farraokh Sassani (Mechanical Engineering), and the extarnal examiner, Dr. Brenda McCabe (Civil Engineering, University of Totonto), for examining the dissertation.
- The construction organizations and practitioners, especially the University of British Columbia Properties Trust, Mr. Robert Malcolm (the principal and CEO of the Timberline Construction Group Western, TCGW), and Robert Currie and Steve Patitucci (the TCGW's Superintendents), for providing me the access to their construction fields and information resources.
- Ensemble Systems Inc., for its generous provision of a free copy of Ensemble Streams™ to the research, and the invaluable supports of its technical staff.
- The many other software companies, who responded to my requests with their invaluable information and provided trial- or full-version copies of their software solutions.
- And last but not the least, my beloved ones: my dearest, my mother, who dedicated herself to build the foundation of my current position and encouraged me to continue my studies; my true love and my source of hope, my wife, for her endless love, understanding, and encouragement, and for providing me the possibility to concentrate on my research; and my endless source of joy and inspiration, my children, for their patience and understanding in this journey.

I wish them all the blessings of God and a real success in their personal, social, and business life.

Abdolhamid Ghanbari

Vancouver, Canada

May 2006.

CHAPTER 1 Introduction to the Research

This dissertation addresses the field of *computer integrated construction* (CIC) with a focus on *model-based integrated total project management (PM) systems*: a class of computer systems that include a suite of applications that support a wide range of PM functions and can flexibly and openly contribute to and draw from a shared pool of project information irrespective of the type of environment. More specifically, it examines how various types of construction project information could be modeled in an integrated manner to support integration of various functions involved in the management of the project throughout its life cycle. This chapter provides an overall description of the research—the background, problem statement, objectives, scope, methodology, and deliverables—and the structure of the dissertation.

1.1 Introduction

Construction projects are characterized by their complexity, which provides the basis for fragmentation in the AEC/FM (architecture, engineering, construction, and facility management) industry. Among the factors that collectively contribute to this complexity are the uniqueness and relatively short life span of projects, the high volume of (often non-standardized) information exchanged among a large number of ever-changing project participants, and the different views of participants on the project information. In this context, information management is a key to technical integration (e.g., data integrity and consistency) as well as managerial integration (e.g., who is responsible for what, where, and when).

Information Technology (IT) has been used as an effective strategy to improve both efficiency and effectiveness in organizations [Cash et al. 1994]. In the AEC/FM industry, however, despite its use in a variety of application areas, computers have not been employed to their maximum potential. Many reasons for such failures have been identified, and potential solutions have been suggested [Howard et al. 1989; Bedard et al. 1991; Teicholz and Fischer 1994; Amor and Faraj 2001]. In particular, *computer integrated construction* (CIC) has been proposed as a strategy to overcome the inherent fragmentation of the industry (section 1.2.1) [Howard et al. 1989; Teicholz and Fischer 1994].

Existing commercial software application systems serve a variety of design, engineering, and PM functions (e.g., estimating, scheduling, accounting, etc.); however, integration between these functions has not yet occured [Paulson 1995; Eastman 1999], a situation that is referred to as "islands of automation". Some of the areas for potential computer-based integration have been addressed by researchers [Fischer et al. 1994; Amor and Faraj 2001]. Central to this integration is the *formalization and standardization of information.*

Different approaches have been taken towards this formalization and standardization of information in the AEC/FM industry. Many models have been proposed and many prototypes have been developed [e.g., Bjork et al. 1993; Stumpf et al. 1995; Rezgui and Debras 1995]. However, the use of *standard objects* and the development of *core models* integrating some related *application models* (section 2.4.1.3) has received the most attention [Froese 1992; Teicholz and Fischer, 1994; Wix and DP 1995; Froese 1996a &1996b; Rezgui and Cooper 1998; IFC R2x, 2001]. One of the recent efforts in this *information standardization* is the on-going IAI (International Alliance for Interoperability) project [IAI 2005]: an industry project parallel to the ISO standard 10303, STEP (STandard for the Exchange of Product model data) [Froese 1996c; Wix and DP 1995]. It has applied the STEP-based technology to the industry's software products to create a project data standard called the Industry Foundation Classes (IFC's) aimed to provide software *interoperability* in AEC/FM industry at the international level.

Due to the central role of *product* information (e.g., the physical and geometrical attributes of a building and its physical components) on an AEC/FM project, *product modeling* has been a major focus to most of AEC/FM information modeling efforts (as opposed to other types of project information such as processes, costs, organizational information, etc.). Product information is normally a central reference to operations of most CM (Construction Management) functions; however, such information is not explicitly represented in current CM systems and, thus, is not readily available to these functions [Fischer et al. 1995; Ghanbari and Froese 1999]. In general, product information is dealt with and modeled more from a designer's perspective and less from a construction manager's view.

Moreover, due to the very close relationship between different types of information in a project, a need has also been recognized for the integration of all types of information (i.e., product, process, cost, resource, etc.) into a *common project information/object model* (section 1.2.2.1) [Teicholz and Fischer, 1994; Fischer et al. 1995; Froese 1996a &1996b]. Yet, there still seems to be gaps between existing models and those required for the true CIC environment. The methodology adopted by current CIC projects involves a loose link between the modeling workflows and their related artifacts, especially the process models (specifying the way the business runs) and information models (specifying the structure of the information communicated between processes). This issue can impact the effectiveness of the resulting models. This lose link is observed in the IAI project's methodology [IFC R2.0, 1999c]. In the current IFC object model [IFC R2x2-1, 2004], which represents the state-of-the-art model, there is a lack of coverage of all functions involved in the management of an AEC/FM project and their related information requirements. There is a need for a framework that comprehensively covers all possible PM functions (i.e., not limiting to estimating, planning, and scheduling), and there is a need to examine how PM functions view and use project information.

This dissertation investigates *model-based integrated total PM systems*. It examines *how AEC/FM product and process information can be integrated into a unified AEC/FM project model that can support a wide range of construction project management (PM) functions, with an emphasis on materials management, within the context of a CIC environment*. It places an emphasis on the *process* of modeling; i.e., integration of *business process modeling* and *software engineering* [Jacobson et al. 1999] in the context of CIC *integrated total PM systems development*. This dissertation presents the research results as follows:

- an overall description of the research—the background, problem statement, objectives, scope, methodology, and deliverables of the research—and the structure of the dissertation (this chapter);
- an extensive critical review of current literature and models related to the knowledge areas of the research and the research's views on identified issues and to a set of research criteria (Chapter 2);
- a description of the bottom-up (i.e., from specific to general) investigations that contributed to the understanding of the requirements for integrated project management systems in general and materials management in particular: examination of current PM/CM software applications, software prototyping, field investigations (observation of processes, collection and analysis of field documents, and models, and field interviews), and a Process Modeling Practice Workshop (Chapter 3);
- a top-down examination (i.e., from general concepts to detailed requirements) of current AEC/FM information models to identify their level of support for PM functions and requirements of an AEC/FM project information model (Chapter 4);
- a conceptual AEC/FM Project Core Model (APCM) addressing the requirements (Chapter 5);
- a Conceptual Project Management Framework (CPMF) proposed and used by the research for classification of PM functions (Chapter 6);
- an Integrated Total Project Management System (ITPMS) model for integration of models of AEC/FM business and software systems, and an overall explanation of its application (Chapter 7);
- the results of an implementation of the ITPMS model in a Computer-Aided Software Engineering (CASE) tool—i.e., a set of Materials Management (MM) process and object models (Chapter 8);
- an evaluation of the research results against the research criteria and the requirements defined in earlier chapters (Chapter 9); and
- a conclusion to the dissertation (Chapter 10).

The proposed models (APCM, CPMF, ITPMS, and MM process and object models) correspond to successive layers (from abstract to concrete) that build towards the creation of IT solutions for AEC/FM interoperability, and each model builds upon, implements, and validates (in part) the previous

model. The research encompasses a coupling of top-down and bottom-up approaches, and places a special emphasis on derivation of information requirements through business process analysis and modeling. It is envisioned that the results of this research will contribute to the CIC movement by informing on-going, major CIC projects.

1.2 A General Background

This section presents the key definitions and research ideas and highlights the various dimensions of the research. It is intended to facilitate the definition of the research's problem statement and plan.

1.2.1 Definitions of Basic Terms

In this dissertation, the term *project* is used to refer to a temporary endeavor undertaken to build a unique constructed facility. *Construction management* (CM) involves the project delivery activities, from conceptualization to construction and hand-over of the product to the owner. Moreover, the term *project management* (PM) is broadly used in the context of construction projects to refer to the management of any portion of the development life cycle of any type of facility [Kavanagh et al. 1978; PMI 1996].

The term "AEC" stands for architecture, engineering, and construction [Howard et al. 1989], and refers to *construction and related industries*. "AEC/FM" broadens the scope to include a wider life span of construction projects, including facility management. In this dissertation, AEC, AEC/FM, and construction are all used interchangeably to refer to *construction* projects (including their complete life-span from cradle to grave) and their related industries.

The construction industry is distinguished from other industries (especially from manufacturing) by its complex nature, which is mainly due to a series of interrelated characteristics of the industry [Hendrickson and Au 1989; Howard et al. 1989; Bedard et al. 1991; Teicholz and Fischer 1994]:

1) *Project-Orientation*: The industry is project based, as opposed to ongoing repetitive work.
2) *Uniqueness*: Each project is unique (e.g., in products, processes, organization, and resources).
3) *Short Duration of Business*: Projects generally have a short, finite life cycle (typically a few years, which is a short time period relative to the lifespan of the large organization carrying out the project).
4) *Work Place*: Construction projects take place on sites that are different for each project and that generally involve challenging outdoor work environments.
5) *Business Instability* (in terms of organization and processes): Resources are usually aquired on an as-needed basis, while this is not normally practiced in a hospital or manufacturing environment, for instance. Moreover, the initiation of a project generally means the start of a new business venture with a new set of players coming into the overall business of the enterprise.

6) *Technological Diversity*: The industry embraces a wide range of technologies (e.g., building materials, equipment and tools).

7) *Diversity of Skills and Views*: The industry comprises a large number of small groups of narrow-skilled trades with widely differing views and interests and levels of technological capabilities and adaptability (i.e., a desire to change). The industry is heavily dependent on individuals' knowledge, which widely varies from one person to another.

8) *Non-Standardization and Fragmentation*: A lack of standardization exists in the all dimensions of the industry (in terms of products, resources, processes, procedures, organization, and information). This contributes to the fragmentation of the industry in all its dimensions.

The sum of the aforementioned characteristics contributes to the inherent complexity of the construction industry, causing many challenges for the planning and management of AEC/FM projects.

Computer Integrated Construction (CIC) has been put forward as a strategy to overcome the inherent fragmentation of the industry through technical integration (e.g., data integrity and consistency) and, consequently, managerial integration (e.g., what is where, and who is responsible for what), which results in enhanced efficiency and effectiveness of processes. CIC is defined as a business process that *links the project participants in a facility project into a collaborative team through all phases of a project*. Use of information technology to share data and knowledge between team members, in a real-time manner, is an integral part of CIC [Teicholz and Fischer, 1994]. CIC is analogous to Computer Integrated Manufacturing (CIM), used in other branches of industry.

1.2.2 The Key Research Ideas

1.2.2.1 Interoperability and The Unified Project Object Model

Today's typical computer programs in the AEC/FM domains could be described as being largely stand-alone application systems owning and managing their own data. These systems range from general-purpose tools (e.g., word processing, image processing, accounting, drafting, and scheduling) to specialized tools (e.g., construction-oriented drafting and design, construction cost estimating and cost control packages). They have some limited ability to exchange data with other programs through some specific file formats (e.g., using DXF file format in CAD—computer aided drafting/design—programs), but they are not interoperable to any significant degree, and human interpretation and manual translation of data is required to move data from one program to another. In most cases, there is no common language for communication among related applications; i.e., *no consensus on project concepts/objects* (e.g., material, product, process, resource, and order) *and their relationships* (e.g., a process uses one or

more resources) among the programs that use and manipulate them. To overcome this problem, the idea of a *unified project object model* has been evolved [Teicholz and Fischer, 1994; Froese 1996b; IAI 2005].

Figure 1-1 illustrates this emerging idea, which was fundamental to this research, by a scenario in which a set of related PM software application systems *logically* share the *unified project object model.* The systems can exchange data with each other using the vocabulary (i.e., the AEC/FM domain objects and their relationships) defined in the commonly shared model. The model is a communication mechanism providing for understandability of the information that is being communicated, rather than a physical database (e.g., a physically centralized database from which the users of the application systems pull data). The model acts like a logical language for communication between related software agents. The physical databases of projects and/or enterprises are built based on the definitions made in the model. Such databases may potentially be managed in a centralized database. Alternatively, they may be physically distributed among the systems.

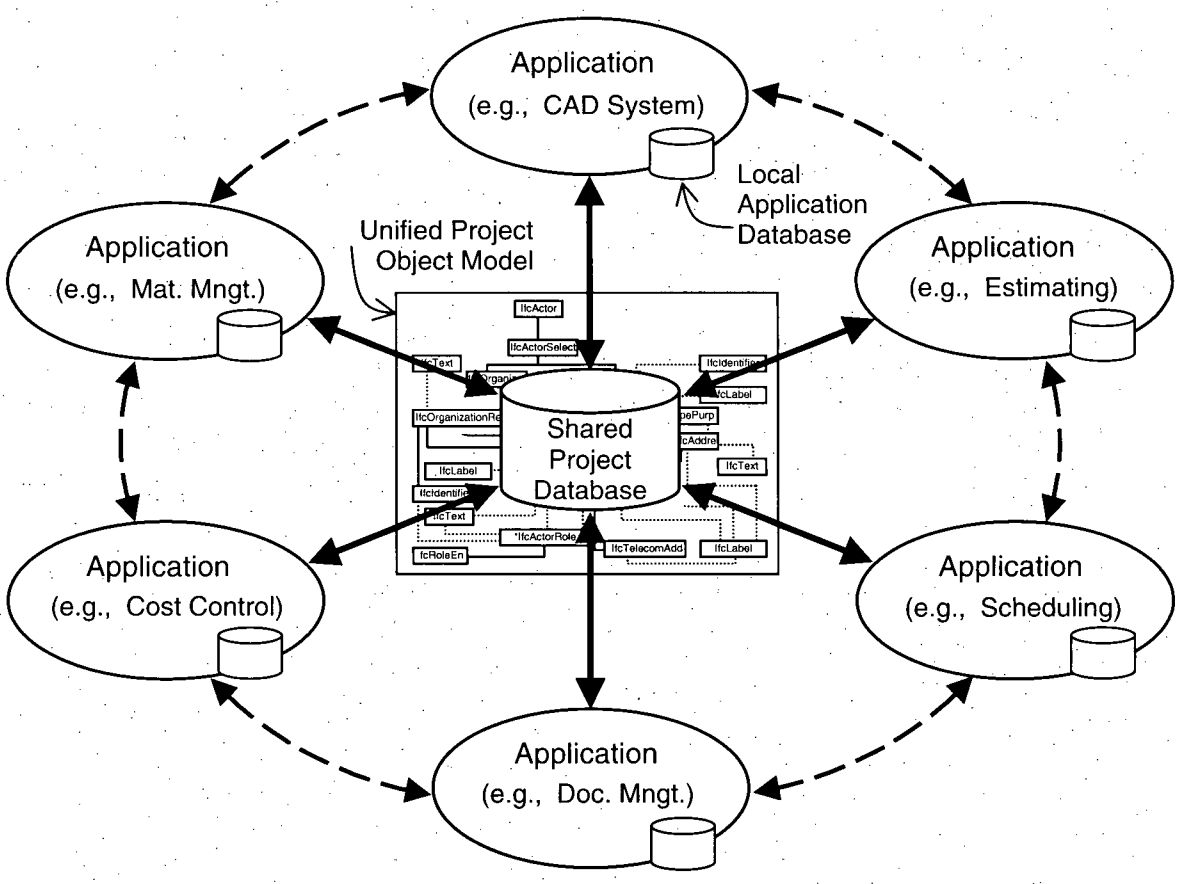


Figure 1–1: The Basic Idea: A Unified Project Object Model for Communication of Project Information

1.2.2.2 Integration of Business Modeling with the Software Development Process

The process of software development has been described in different ways, depending on the methodology adopted [Sommerville 1997; Coad and Yourdon 1991a; Jacobson et al 1992; Jacobson et al. 1999]; however, the new emerging methodologies for software system development propose an integrated process approach, which tightly integrates processes and their resulting models from definition of the problem to the design and development and realization of a solution (i.e., the supporting information system). Since a software system is usually intended to be used and to support some business process (what people physically do in the organization), *business process modeling* has been suggested to be not only a part but also the driving vehicle of the software development process [Jacobson et al. 1995; Eriksson and Penker 2000]. The current CIC projects have not fully and formally benefited from the idea of *integration of business process modeling and software engineering processes* (see section 1.2.5).

The current software development methodologies differ in their focus and extent of coverage of the workflows of the software development process. Figure 1-2 schematically shows this coverage for some of the key methodologies in comparison with this research by illustrating the basic workflows (before testing and deployment) and their related models in the upper part of the figure. These workflows and their related models as well as the methodologies are elaborated in Chapter 2 of this dissertation.

The dissertation goes through most of the workflows of the software development process in the context of construction PM. However, a special emphasis is placed on the interface between business process modeling and software engineering, a concept that is emerging within the framework of the Unified Process [Jacobson et al. 1999; Jacobson and Bylund 2000]. The frameworks (i.e., principles or ideas) and models (i.e., representations) developed in this research encompass the *integration of business process modeling with the software development process*, especially the requirements analysis, and system analysis stages. The software design and implementation was also addressed through a prototype system, which served as a proof-of-concept for certain elements of the research (Appendix C).

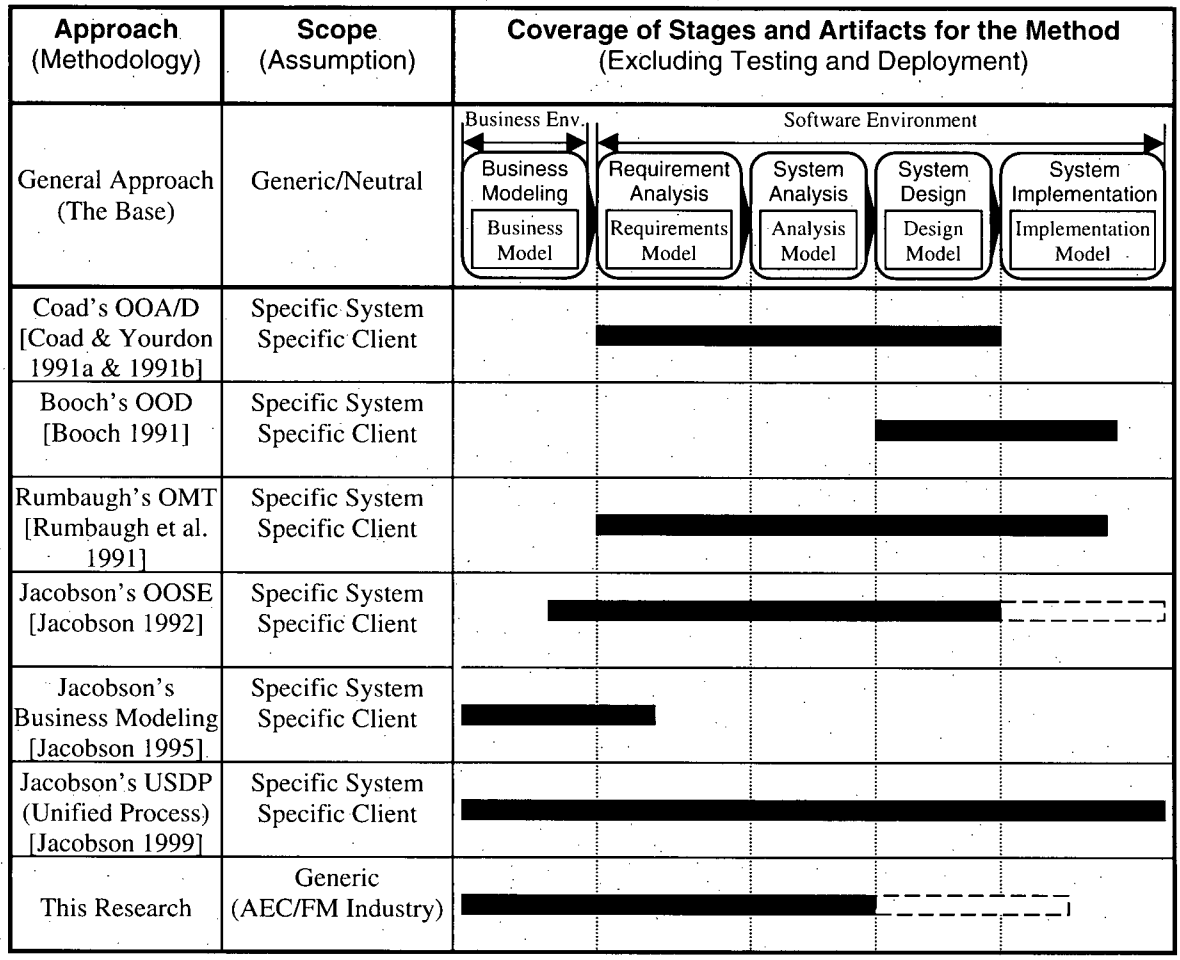


Figure 1–2: The Research's Modeling Approach Compared with Software Development Methodologies

1.2.3 Project Management Functions and the CIC Projects

Effective information management is a key to the integration of PM functions, and it requires the on-time availability of the right information. Such a requirement is often not met in construction projects for many reasons. For example, MM processes are usually performed in several different parts of organizations such as purchasing, production control, transportation, warehousing, materials handling, operations, receiving, and logistics, and materials information is scattered throughout these organizational departments and sections [Stukhart 1995]. The information is represented from different perspectives in different formats and systems. The consequences of this include little consolidation of materials information and, thus, inconsistencies in the materials information. These problems propagate to other related PM functions such as planning, scheduling, quality assurance/control, cost estimating, and cost budgeting and control, resulting in efficiency and effectiveness losses in various levels of organizations (the project, enterprise, and industry) [CII 1986]. Standard models of materials information integrated with other types of project information can help to overcome such consequences.

Project management (PM) has traditionally emphasized a few managerial functions (e.g., planning, controlling, and administration processes) [Kavanagh et al.1978], while operational functions (e.g. physical production and logistics) have received less focus. Over the past two decades, many research and development and data standardization projects have tried to achieve the goals of CIC using different approaches and techniques. Some of these projects have adopted a narrow scope (in terms of the type of information, application areas, and geographical location of the concern), while others have taken a somehow larger scope (e.g., information spanning the whole project and at an international level). These efforts range from information modeling and implementation of a specific application area to integration of several applications within a specific domain of AEC/FM (e.g., design of HVAC systems) and integration of the whole AEC/FM domain at different institutional, national, and international levels.

Examples of these projects include ATLAS (EP-2780; for large scale engineering) [ATLAS 1997], COMBINE (COmputer Models for the Building INdustry in Europe) [Augenbroe 1995], COMMIT (COnstruction Modeling and Methodologies for the Intelligent inTegration of information) [Brown et al. 1996], CONDOR (ESPRIT 23 105) [Rezgui and Cooper 1998], GARM (General AEC Reference Model) [Gielingh 1988], GDCPP (Generic Design and Construction Process Protocol) [GDCPP 2001], IBPM (Integrated Building Process Model) [Sanvido 1990], ICON (Integration of Information in CONstruction) [ICON 1994], IAI (International Alliance for Interoperability) [IAI 2005], and STEP (STandard for the Exchange of Product model data) [ISO 1992].

As will be discussed in chapters 2 and 4 of this dissertation, however, the proposed models have not yet fully addressed the goals of CIC. There has been more effort on formalizing and standardizing product information (e.g., a physical building and its elements) reflecting a designer's view and less on other types of information (e.g., processes, resources, costs, and organizational information) involved in the management of projects and the integration of the two. Also, there is a lack of coverage of most PM functions in current models; thus requiring an extension to the scope of such models to cover a wider range of application areas [Froese et al. 1997a; Yu et al. 2000; Yu 2002]. Moreover, the existing AEC/FM information modeling projects have generally adopted a *top-down* approach to formalizing and standardizing information; thus it is suggested that their effectiveness should be examined through *bottom-up evaluation* approaches (i.e., by reviewing, testing, and implementation of the models) [van Nederveen and Tolman 2001].

1.2.4 The Materials Management Function

The MM function was selected as a subject area in this research for a number of reasons. First, it has a high impact on construction productivity [Thomas et al. 1999]. The cost of materials and equipment usually runs about 60% of total project costs with construction labor costs at about 25% [CICE 1982]. A

10-12% increase/decrease in labor productivity has been reported in construction projects due to the quality of MM. This labor cost savings originates from a reduction in nonproductive labor time spent waiting or searching for materials [CICE 1983a; CII 1986; Ballard et al. 1994].

Second, the preliminary investigation of this research showed that, with few exceptions for specific application areas, MM processes (e.g., materials ordering, receiving, storing, and testing) have not been generally supported in current PM systems (section 3.1). Consequently, limited aspects of materials are separately modeled and scattered throughout a few applications (e.g., materials cost in estimating and cost accounting). Moreover, like other types of information, the materials information generated by one application system is not easily transferable to other potential systems.

Third, the MM function has not previously been addressed by the existing AEC/FM information standardization efforts (section 4.1.6). Materials information has not been modeled as used in construction processes (e.g., the bulk of material received and processed by construction personnel) but rather as used in the design phase of the project (i.e., material properties).

Finally, information about the final-target product (e.g., a building, a wall, etc.) is closely related to the materials incorporated into the product (e.g., door, window, concrete, brick, etc.) (section 4.1.6.1). This relation has loosely been represented in some existing PM software and AEC/FM standard data models. This research has aimed at investigating suitable ways of representing these two interrelated concepts so that the resulting model could be shared among related application systems (e.g., specification, purchasing, and storing/warehousing).

1.2.5 The State-of-the Art in CIC Developments

With the formation of the International Alliance for Interoperability (IAI) as a not-for-profit international consortium in September 1995, the standardization efforts in the AEC/FM industry entered a new stage. The IAI, a major industry effort to standardize AEC/FM project information, has resulted in the Industry Foundation Classes (IFC's)—a computer data standard that enables *software interoperability* across different AEC/FM industry domains in the whole life cycle of a building project. The IFC's is, in fact, the specification of an object-based data model of a building project to be commonly used by multiple vendors of software applications for data exchange. The software products using the IFC's could then share and exchange information using this common data model. So far, the IAI has published four major releases of the IFC, R1.0, R1.5, R2.0 and R2.x [IFC R2x, 2000a]. The latest version of the last release is R2x2-1 (Release 2x, Edition 2, Addendum 1) [IFC R2x2-1, 2004; IAI 2005].

The IAI project represents the state-of-the-art AEC/FM information standardization effort. Its formation was, in fact, a response to the slow progress of earlier standardization efforts in the AEC/FM

industry [van Nederveen and Tolman 2001]. There is usually a gap between IT and its standardization process[1], mainly due to the *slowness* of the development and adoption of the standards [Ng et al. 2001; Samuelson 2002]. The higher level of involvement of related software companies in the standardization process can increasingly reduce this gap. Ultimately, it is the software companies that should take action to apply the standards and bring them into use. The effectiveness of the process can be further enhanced with participation of a wide range of the required expertise (e.g., internationalization and many disciplines). To a large extent, the IAI project embraces all these characteristics, and thereby has the potential to reduce the gap.

Moreover, the IAI project has built upon the experiences of its predecessors, though with a larger scope (in terms of geographical coverage, involvement of software vendors and required expertise and other resources). It inherits from such standardization projects as COMBINE [Augenbroe 1994; COMBINE 1995], ATLAS [ATLAS 1997], and STEP (specifically BCCM, AP225) [ISO 1995a; ISO 1997]. The modeling techniques and methodology applied in the development of the IFC's were drawn largely from the STEP project. For example, the EXPRESS language and its graphical component, EXPRESS-G [ISO 1991] (Appendix A), were chosen for definition and specification of the IFC object model. Adoption of a layering approach to data modeling is another example of this commonality between the IAI and STEP projects. An overview of the IAI project and the IFC model is presented in Chapter 2 of this dissertation.

The IFC model has aimed at identifying, defining, and representing *all the common information requirements* of *all industry processes* across *all AEC/FM domains*. However, despite the many advantages offered by the model, the following major issues remain to be resolved:

1) The coverage of *all processes across all AEC/FM domains* has not yet taken place (e.g., facility management [Froese et al. 1997a; Yu et al. 2000]). This could be the result of limited resources. *Construction management* functions have not been fully dealt with. At present, only two functions of estimating and scheduling have received attention within the North American PM group of IAI. Therefore, there is a need to consider the *total scope of PM* within a single framework.
2) Reflecting the emphasis of IAI projects to date, the IFC object model is heavily design oriented (section 4.1.6.3), mainly due to the number of design-domain projects involved in its development. By definition, the IFC's should address the information requirements of all AEC/FM domains, not only building design and a few other functions. Moreover, true CIC calls for an object model that

[1] The information standardization process generally involves model development (by modelers), model implementation/adoption (by software vendors), evaluation (by AEC/FM companies), model standardization, (by standards organizations), and standards adoption (by software vendors).

reflects the information requirements of all activities in the delivery and facility management processes of any type of AEC/FM project, not only buildings.

3) Business processes, through which information flows, are not explicitly modeled in the current IAI standardization effort. Word processing and spreadsheet applications are generally the main tools used for managing the modeling process workflows, and links between process and object models are not made explicit—e.g., no mapping of the information input and output represented in $IDEF_0$ diagrams and the objects defined in the object model [IFC R2.0, 1999c]. A framework that provides a comprehensive picture of all AEC/FM project processes and their information requirements as well as their explicit links could considerably raise the efficiency and effectiveness of both the modeling process and the resulting models. It helps in the standardization of both processes and information.

The aforementioned issues are not intended to undermine the importance and advantages of the IFC's but to identify opportunities for improvement and to highlight the directions of this research.

1.3 Problem Statement

Many research and development projects have worked towards the formalization and standardization of the information involved in the management of AEC/FM projects to provide for software interoperability in a CIC environment. The initial review of these projects as well as their related knowledge areas resulted in the identification of a number of significant areas for improvement, as summarized in the following.

1) The *top-down* approach of the existing AEC/FM information modeling projects needs to be coupled with *bottom-up evaluations* (i.e., reviewing, testing, and implementation of the models).

2) The need to *integrate product and process information* into a *unified project information/object model* has already been realized, but current initiatives do not go far enough. Such initiatives need to be reviewed and critiqued, and effective solutions for the continuation of this work need to be addressed.

3) Despite their central role to almost all CM processes, product information and materials information have previously been modeled mainly from a designer's view, rather than from the perspective of CM functions. There is a need to model *product and material information as used by CM*. The MM function is one of most important functions that has received little attention in current information modeling projects and PM software systems. Therefore, the *MM function* is a good candidate for detailed study and modeling as a significant, representative component of a *unified project object model* and *integrated total PM systems*.

4) Best practices in software engineering methods generally suggest a progression from business process modeling to information modeling. The reliability and usefulness of the information models and resulting information systems are dependent on how formally and explicitly such information is elicited from the business process models [Eastman 1999]. The current AEC/FM information modeling projects have used a variety of development techniques and with varying degrees of formality and explicitness. An *examination of the information modeling process* itself can help to find the ways to enhance the usefulness of AEC/FM information models.

5) In current systems and AEC/FM models, PM functions have been traditionally limited to a few functions such as estimating, planning and scheduling. There is a need for *a framework that provides for modeling all possible PM functions and their information requirements in an integrated manner.* Due to the central role of *business process modeling* to the identification, definition, and specification of the information, a special emphasis needs to be placed on integration of this workflow with other workflows involved in the software development process and integration of the artifacts (models) of the whole process (section 2.1.3).

1.4 The Research Plan

This section describes the research plan in terms of research objectives, scope, methodology, and deliverables and contributions.

1.4.1 The Objective and Research Questions

Considering the previous problem statement, the *overall objective* of this research was *to investigate how AEC/FM product and process information could be integrated into a unified AEC/FM project model that could support a wide range of construction project management (PM) functions, with an emphasis on materials management, within the context of a CIC environment.* This objective has been addressed through the *research hypothesis*: *the distinct views of the functions involved in an AEC/FM project on the project information can be brought together into a unified project model, which can be used as a central repository for a wide range of PM application systems reflecting the CIC environment; this may be accomplished through the modeling of PM business processes and their information requirements, within a unified software development process.*

The basic research challenge posed by the objective (*how to model project information in order to be useful for PM functions*) was further decomposed into the following research questions:

1) What is the state-of-the art in process and information modeling (concepts, terminologies, methods, techniques, tools, and models), especially in the AEC/FM industry?

2) What are the (product and process) information requirements of PM functions?

3) How could the information requirements of PM functions be formalized into a conceptual AEC/FM project information model?
4) What are the full range of PM functions, and how could they be classified?
5) How could models of AEC/FM systems (i.e., process and information models of both business and software systems) be integrated into a single framework to support the integrated total PM systems development process?
6) Given a specific PM context (i.e., the MM function), how could the proposed models be implemented (i.e., using the frameworks to model the functional and information requirements of MM systems)?

The basic approach, then, was to survey a broad range of references (from literature, data collection field studies, etc.) in order to identify the strengths and weaknesses of each, then to develop a series of models that combine the best features and solve the primary problems of the references. The primary research challenges lay in the acquiring, reviewing, analyzing, and synthesis of the very wide range of references, and in developing the resulting models, which are complex because of their broad scope, their multiple layers of abstraction, and the numerous relationships between them (particularly linking the process views with the information views).

The above challenges required the scoping/rescoping of the research subjects and deliverables at various levels of abstractions (e.g., conceptual modeling versus detailed object modeling). They also required considerable amount of time for planning (e.g., methodology preparation), executing, and controlling (e.g., revising) of the various research activities to achieve the predefined objectives. The resource-and-time use of major AEC/FM information modeling projects is noticeable in this regard—e.g., 70 person-years in the COMBINE project [Augenbroe 1995], which had a much narrower scope than the ongoing IAI project [IAI 2005].

1.4.2 The Scope

Considering the research objectives, six major knowledge areas were considered within the scope of the research, as illustrated in Figure 1-3. These knowledge areas helped define the points of departure for the research (Chapter 2). Figure 1-4 further illustrates the underlying domains of various modeling activities of the research. Other subjects were considered out of scope for the research, such as the following:

1) business process re-engineering [Hammer and Champy 1993];
2) business process modeling for purposes other than information modeling (e.g., workflow management, process planning/scheduling);

3) implementation of software based on the research results (other than the preliminary prototype described in Appendix C);

4) the strategic aspects of IT; and

5) low-level information definitions such as geometry representation (of, for example, a building and its elements), representation of material properties (e.g., density, vapor resistivity, thermal conductivity, etc.), and presentation objects (e.g., tables, graphs, etc.).

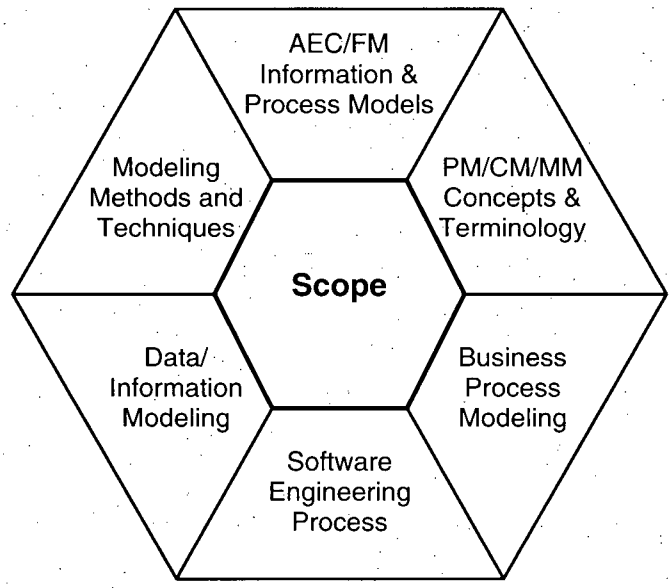


Figure 1–3: The Knowledge Areas Within the Scope of the Research

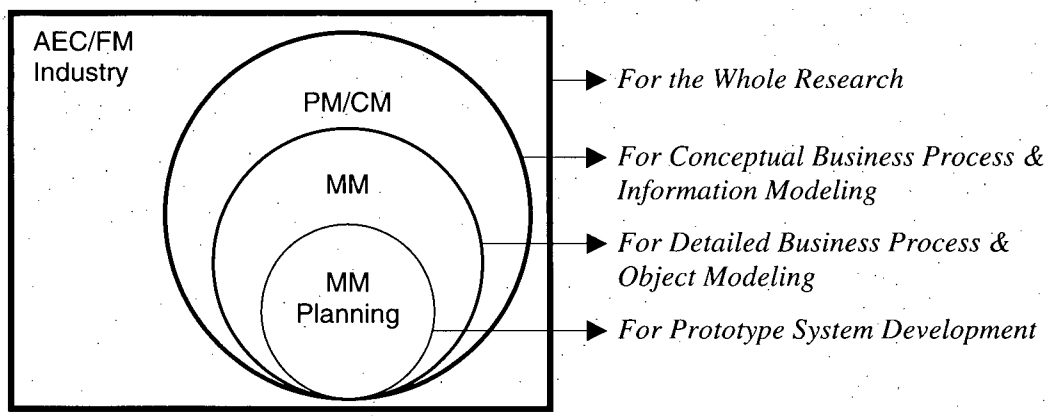


Figure 1–4: The Modeling Domains of the Research

1.4.3 The Methodology

This section describes the research methodology in terms of the basic research activities and workflows, the modeling techniques and tools used, and the major deliverables of the research.

1.4.3.1 Research Activities and Workflows

The major research activities and workflows are schematically shown in Figure 1-5 in the form of a UML (Unified Modeling Language) activity diagram [UML 1.4 2001] overlaid on dashed boxes, which group the activities and workflows into the major knowledge areas of the research. In order to avoid complexity, however, the figure does not show all decision points and feedback loops. In the UML notation (described more fully in Figure A-6 of Appendix A of this dissertation), a round box represents an *activity*, while a rectangular box represents the *input* and *output objects* to and from activities (underlined items represent the primary research deliverables). Simple arrows indicate transition (i.e., completion of one activity and start of another). Dashed arrows are used to mean *object flow* and *dependency* between elements. Diamonds are *decision* points and indicate a branch or merge in the flow. Horizontal thick lines illustrate *synchronization*—a branching of the flow into parallel flows or a joining of parallel flows which all complete before the flow continues. The *start state* and *end states* are represented with a filled circle and a filled circle in a blank circle, respectively.

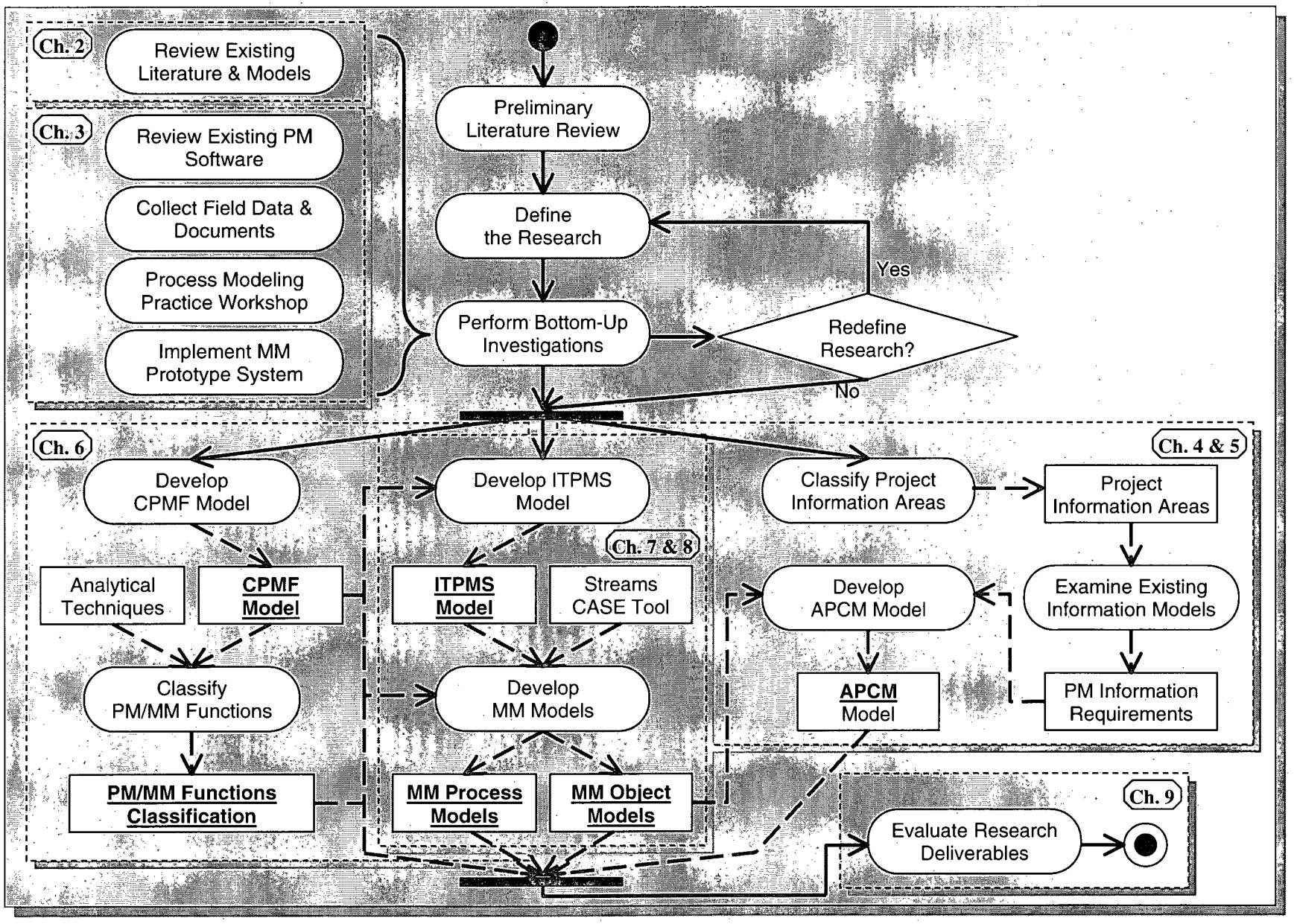


Figure 1–5: The Research Activities and their Workflows

1.4.3.2 Modeling Techniques and Tools

The research used UML notations [UML 1.4, 2001] and EXPRESS-G notations [ISO 1991] (Appendix A) extensively for process and information modeling purposes. It also benefited from the promising advantages of CASE (Computer-Aided Software Engineering) tools (section 2.3.4) for both modeling and documentation purposes.

The UML use case modeling and activity diagramming techniques were specifically used for modeling of processes and objects within both business and computer environment. The reasons for selecting these techniques are elaborated in Chapter 2, with the main points being as follows:

1) They are a part of an important defacto standard (UML), which is intended to bring all software development methodologies into a shared language of communication and which is becoming an international standard through ISO [OMG 2005].
2) Use cases can portray an external-functional view of systems of any kind.
3) Use cases focus on *what* the system does, independent of *how* and *why* concerns.
4) When the external view is established and agreed upon, the internal view (i.e., the "*how*" issues such as sequencing, objects and their flows, etc.) may become of the concern. Activity diagrams can portray this view.

EXPRESS-G (the graphical-notation component of EXPRESS modeling language, a part of the STEP standards [ISO 1991]) was selected for presenting existing information models as well as the object models suggested by this research for two major reasons:

1) its simplicity (to draw and comprehend), and
2) its use for many other existing AEC/FM data models (so no conversion would be required for comparison purposes).

Ensemble Streams™ (Streams) [2000] (Appendix A), which is a CASE tool that supports UML notations, was used for implementation of the ITPMS model. More specifically, it was used for:

1) listing and modeling of business processes in the form of use case models,
2) describing the internal behavior of use cases, which represent processes, in the form of activity diagrams (i.e., activities and their possible workflows), and
3) eliciting and listing of data objects involved in business processes (especially in MM) by adding object flows to the activity diagrams.

The Chapter 2 of this dissertation provides the background information about current modeling methods, techniques, and tools, especially those used in this research.

1.4.3.3 Research Deliverables

The research objectives and methodology suggest the following major deliverables:

1) A *comprehensive, critical review* of the relevant knowledge areas; specifically process and information modeling technologies (concepts, terminologies, methods, techniques, tools, and models) and PM/MM concepts.
2) A *conceptual AEC/FM Project Core Model (APCM)*, encompassing product and process information requirements of PM functions.
3) A *conceptual PM Functions Framework (CPMF)*, useful for eliciting PM functions.
4) A *classification of PM functions and MM Processes.*
5) An *Integrated Total Project Management System (ITPMS) model architecture*, capable of capturing the functional (i.e., processes) and information requirements of AEC/FM systems (i.e., both business and computer environments) in an integrated manner.
6) An *implementation of the ITPMS model for the domain of MM using the Streams CASE tool*, yielding a set of *MM process models and domain object models*, and providing a test of the ITPMS model.

Of these, items 2 to 6 have the highest importance and are regarded as the *primary research deliverables*. In addition, there are other intermediate deliverables and activities that played an important role in validating the primary deliverables. For example, the *construction field data and documents* (collected in the field investigation activity) served as materials used in the development of the primary deliverables (e.g., MM object models) and are not listed above.

1.4.3.4 The Evaluation and Validation of The Research

The methodology and primary deliverables of the research were supported by several *evaluation/validation* mechanisms. First, the frameworks and models of the research were developed in an *iterative* manner, which resulted in their continuous evaluation-and-reshaping. Second, the top-down development of the primary deliverables generally evolved in a chain of *concretization*; i.e., one model being used as a framework for development of, and thus testing by, another model. For instance, the validity and usefulness of the CPMF model was tested by being used as a framework for development of the ITPMS model and its implementation. The deliverables also gained their validity through a third mechanism: the *bottom-up investigations.*

The bottom-up investigations (Chapter 3), which were carried out to complement the extensive critical literature and models reviews, included observation of real business processes, field interviews, collection and analysis of field documents and models, software prototyping, examination of current PM/CM software applications, and a Process Modeling Practice Workshop. The results of these activities

served as input materials for the evaluation/validation and reshaping of the models produced in the top-down development approach.

1.5 The Organization of the Dissertation

The structure of this dissertation generally reflects the steps described in the research methodology (section 1.4.3). The dissertation is organized into ten chapters plus appendices:

- *Chapter 1:* provides an overall description of the research—the general background, problem statement, objectives, scope, methodology, and deliverables of the research—and the structure of the dissertation.
- *Chapter 2:* presents the foundation used for building the frameworks and models of the research. In addition to reviewing other work, this chapter provides definitions, explanations, and discussions of the terms, concepts, methods, and technologies related to the knowledge areas of the research.
- *Chapter 3:* describes the bottom-up investigations that contributed to the understanding of the requirements of integrated project management systems in general and materials management in particular: examination of current PM/CM software applications, software prototyping, field investigations, a Process Modeling Practice Workshop, and the research criteria.
- *Chapter 4:* focuses on the information modeling part of integrated AEC/FM application systems development at a framework level. In a top-down analytical approach (from general concepts to detailed requirements), it examines current AEC/FM information models to identify their level of support of PM functions and lays out the requirements of an AEC/FM project information model.
- *Chapter 5:* presents a synthesis of the earlier analytical materials and describes the scope, structure, and characteristics of a conceptual project information model (called the AEC/FM Project Core Model, APCM), which addresses the requirements identified earlier.
- *Chapter 6:* focuses on the process modeling part of the total PM systems development process at a framework level. It describes the first step of the research towards formalizing PM processes, i.e., the development of an analytical framework (called Conceptual PM Functions Framework, CPMF) within which a fairly comprehensive list of PM functions could be explored. It lays down the theoretical bases of the framework, introduces the CPMF model, and describes how the model was used to identify and classify PM functions.
- *Chapter 7:* describes the architecture and underlying structure of the Integrated Total Project Management Systems (ITPMS) model, which is proposed by the research for integration of models of AEC/FM business and software systems, as well as its application.

- *Chapter 8:* advances another step towards concretization of the ideas and models suggested by the research by exploring the applications of the previous results (the CPMF model, the PM functions classifications system and the ITPMS model) in a specific PM context (i.e., the MM function), based on a set of predefined criteria. It presents the results of implementation of the ITPMS model in a CASE tool (i.e., a set of MM process and information models useful for integrated MM systems development).
- *Chapter 9:* presents an evaluation of the research results (i.e., primary research deliverables) mainly by referring to the research criteria defined in earlier chapters. It highlights the advantages and implications of the deliverables.
- *Chapter 10:* concludes the dissertation with recommendations, contributions and benefits of the deliverables of the research. It addresses the future research directions suggested for a full realization of the promises of this research.
- *References and Appendices:* include a list of references cited by the research as well as a set of appendices, which provides more detailed information about various dimensions of the research. The appendices include: 1) a description of several modeling techniques and tools that are used or referred to in the research; 2) a list and attributes of some of the commercial PM/CM software applications reviewed in the research; 3) a description of the prototype MM system tool developed in the research; and 4) a list of MM business objects definitions suggested by the research.

1.6 Chapter Conclusions

This chapter introduced the main dimensions of the research aimed at the integration of product and process information for the purpose of development of integrated computer systems supporting construction PM functions. It described the objectives, methodology, and deliverables of the research. Some of the significant characteristics of the research include its amalgamation of *top-down analyses* and *bottom-up investigations* and exploitation of the advantages of the latest modeling languages (e.g., the UML) and a CASE tool in the AEC/FM systems modeling context.

Given the objectives and methodology of the research, the study of related topics and development of the listed deliverables are not intended to define or introduce new technologies, per se, but rather to specify an environment in which new, as well as established, technologies can be feasibly applied so that the results can contribute to current CIC movements. The *integration of process and information modeling processes* (i.e., emphasis on the *systems development process* and its effectiveness) as well as *integration of product and process information* (i.e., emphasis on effectiveness of the models and systems) are believed to be the major opportunities promised by the research.

CHAPTER 2 Points of Departure and Research Views

This chapter serves as a point of departure to the main body of the research and critiques aspects of the underlying knowledge areas of the research (listed in section 1.4.2). Formalization and standardization of AEC/FM information and processes are the very first steps towards CIC, and they require methods, techniques, and tools. Therefore, a major part of this chapter is dedicated to elaborating on some of the terms and concepts related to this formalization and standardization process and the methods and techniques involved in the process.

This chapter includes seven major sections. The first section focuses on software development process in general and elaborates on the workflows and artifacts (i.e., models) of the process. The second section explains some of the terms, concepts, and issues related to business process modeling (the first step in software development). The third section describes different modeling approaches, techniques and tools that are used in software development and are relevant to this dissertation. The fourth and fifth sections focus on information and process modeling in the AEC/FM industry respectively. They present a general overview of a number of major AEC/FM modeling projects aimed at developing CIC environments (though with different methods, techniques, and results). Reflecting on the issues associated with the models, the fifth section also explains the criteria and methodology of the research for process modeling. The sixth section elaborates on another major theme of the research, i.e., MM in the CIC context, by explaining the key concepts as well as presenting a critical review of MM literature and models. It also describes the research approach to modeling MM systems. Finally, this chapter ends with some concluding remarks.

2.1 Modeling and Software Development (Terms and Concepts)

This section focuses on software development process in general and elaborates on the workflows and artifacts (i.e., models) of the process. First, it gives a general background on the evolution of software development methods and techniques. Next, it highlights the focal point of this section (and also the chapter), i.e., software engineering and software development process, by defining some of the related basic terms and concepts. Then, it explains the software development process and its related workflows and artifacts. Due to its central role in this research, the requirements model, which is the first specification of a software system, is further explained in more detail.

2.1.1 Some Basic Definitions

In order to better clarify the focus the discussions in the upcoming sections, some basic terms and concepts are defined in the following.

2.1.1.1 Data, Information, and Knowledge

A three-level hierarchy may be considered between data, information, and knowledge [Beynon-Davies 1993]. *Information* is defined as: "facts, concepts, or instructions", while *data* is "a representation of facts, concepts, and instructions in a formal manner [i.e., 'in accordance with explicit rules prior to use'] suitable for communication, interpretation, or processing by human beings or computers" [ISO 1992]. These definitions suggest that: Data is a *representation* of some information.

A datum (a unit of data) is one or more symbols used to represent something, while information is interpreted data (i.e., data placed within a meaningful context) [Beynon-Davies 1993]. Data consists of known attributes of objects whose values can be measured or determined. In order to be interpreted, data is *represented* (using symbols or signs) and physically stored on the hard disk of a computer or on a hard copy. For example, "3.00" is a datum, which can represent many things. It can represent the height of a specific room in a specific building, the price of a product, or capacity of a bucket, for instance.

Knowledge is the interpreted information (in the form of rules, heuristics, and algorithms) about objects (e.g., products, resources, and processes) and their relationships that is used to reason and make decisions about objects [Beynon-Davies 1993]; e.g., performing operations such as change, add, delete, etc. on objects and/or their attributes and their relationships. Howard et al. [1989] consider two types of knowledge: Procedural knowledge (e.g., used in a structural program) and heuristic knowledge (e.g., used in an expert system application). In order for information to be useful for reasoning and decision-making, it should be interpretable into knowledge.

In summary, data, information, and knowledge are interrelated. Data, at the lowest level of abstraction, is some known facts or results about objects, while information relates to the *semantics* (meaning) of data— much like a container within which data would make sense. A datum is essentially a value for a given attribute of an object. The totality of the object and its attributes and, possibly, the attributes' values is called information. Knowledge is the interpretation of this totality used for reasoning and making decisions.

2.1.1.2 Process

The term *process* is defined as a "series of actions or operations performed in order to do, make, or achieve something (e.g., unloading cargo, reforming the education system)" [Oxford 1989]. In a business context, Davenport [1993] defines a process as "a structured, measured set of activities designed to produce a specified output for a particular customer or market. ... It is a specific ordering of work activities across time and places, with a beginning, an end, and clearly identified inputs and outputs: a structure for action" [Davenport 1993]. More specifically, in the context of building construction, IFC

model defines a process as "an action taking place in building construction with the intent of acquiring, constructing, or maintaining products. Processes are placed in sequence (including overlapping for parallel tasks) in time" [IFC R2x, 2000a].

Various suggestions have been made for naming a process when modeling them [e.g., $IDEF_0$ 1981; Hammer and Champy 1993; Jacobson et al. 1995]; however, this research follows the common practice of the AEC/FM modeling community: plain/distinctive verb forms, in an imperative mood, and its object (e.g., "Order Materials").

2.1.1.3 Object-Oriented Modeling

The object-oriented modeling approach looks at entities as a composition of data and activity. Applying the *encapsulation* mechanism, object-oriented methodologies encapsulate both data (i.e., facts) and the operations on the data (i.e., activity or process) into one single concept of an *object*. *Objects* are then considered as instances of the *classes* they belong to (e.g., object *ABC Company* as an instance of class *Organization*), referred to as *instantiation* mechanism [Jacobson et al. 1992].

Objects are described with their *properties*, a part of which denotes information about the objects' *relationships* with other objects. Objects communicate with each other through their relationships, and a relationship between two objects means one object is describing, constraining, or using the services of another object. Two objects may be related in a model *directly* or *indirectly* (association through an intermediate relationship object) [Fowler 1997]. That part of data that is directly attached and is available to an object is called *attributes* of the object. A class defines *attributes* (e.g., *Address*) and *services* (sometimes called methods, representing activities; e.g., *Update Address*).

Other object-oriented concepts include *aggregation*, *classification*, *inheritance*, *association*, and object *communication* (for objects to request services from one another). Using these concepts, classes are presented in the form of aggregation and classification hierarchies with their required associations and object communications. Different terms and graphical notations are used for these concepts by different object-oriented techniques. An overview of some popular object-oriented methods is presented elsewhere [Jacobson et al. 1992, Martin and Odell 1998].

Theoretically, in object-oriented technology, an *object* represents a particular instance of a class. The term *class* usually comes into use at the software design and implementation (i.e., programming) stages. In practice (as in this dissertation), however, the term *object* (or *business object*) has flexibly been used in business modeling and software analysis contexts to mean a business entity (e.g., *Purchase Order*), a class of things, or an instance of a class.

2.1.1.4 Software Engineering, Software Reengineering, and Information Engineering

Software engineering is a disciplined approach to developing computer software systems. Its focus, however, has dramatically changed during the ages, from a single-stage-process of programming to more sophisticated integrated software-development processes suggested by different methodologists and schools of thoughts [Kubeck 1995; Sommerville 1997].

Software reengineering is an approach to analyzing and restructuring legacy (existing) software applications (e.g., code, rather than their underlying business processes) to attain dramatic improvements in terms of functionality, performance, etc. It involves understanding the software's logic and converting it to a new language or platform, using software engineering standards [Kubeck 1995]. Such a process may include *reverse engineering* (i.e., generating analysis/design models from the program source code) and *forward engineering* (i.e., from model to source code).

Information engineering is a methodology popularized by James Martin [1982] and further developed by others [e.g., Coad and Yourdon 1991a and 199b; Rumbaugh et al. 1991; Jacobson et al. 1992; Martin and Odell 1998]. It generally involves analyzing existing automated systems, and its scope extends beyond computer applications. Viewing the system as a *black box*, its major concern is to understand *what* the system does, i.e., "*data input*" and "*data output*", not *how* it does it (e.g., process and code consideration). Kubeck [1995] compares information engineering with using a fax machine, whose internal operations need not be known by its user.

Information engineering was further developed into *information* (or *data*) *modeling* methodologies [e.g., Coad and Yourdon 1991a and 199b; Rumbaugh et al. 1991; Jacobson et al. 1992; Martin and Odell 1998]. *Information modeling* is an approach to identifying and representing (graphically and/or textually) the physical and conceptual objects and their relationships in a business or computer system, in a way that the resulting models (i.e., data structure) could be used for developing computer applications or data sharing among related applications.

In summary, information engineering, software engineering, and software reengineering are interrelated, and they all concern information management. Though, concerning computer application systems, their scopes vary: *information analysis/modeling*, *coding and implementation*, and the *software development lifecycle* respectively. Information modeling provides the blueprint for building the systems.

2.1.2 Evolution of Software Development Methods and Techniques

Computers are used to assist businesses to perform their processes more efficiently and effectively than otherwise would be achieved manually. Thus, every software information system is initially defined with some functionality, which is derived from, and should add value to, the business

processes. From the early introduction of computers, different approaches has been proposed and practiced to develop such systems. These approaches range from a very informal, single-view oriented (e.g., code-centric) process to an integrated process. The first was practiced in early times, when technical issues were more of the concern. Applications were simple, discrete programs, and number crunching in a very limited domain was their ultimate goal. Programming was the major task, and modeling techniques were limited to assisting programmers in modeling single programming procedures and processes. Flowcharts are a typical example of such techniques [Capron and Williams 1984; Laudon and Laudon 1988; Kendall and Kendall 1995; Sommerville 1997].

The narrow-scoped programs were not satisfying the goals and strategies of the businesses growing in complexity, and more innovative, integrated application systems were needed [Davenport 1993; Cash et al. 1994]. The scope of software project management was extended beyond implementation and coding (with little or no modeling) to cover the whole project life cycle. Analysis and design become the very important initial phases of the system development life cycle. System analysts took on such primary roles as "consultant, supporting expert, and agent of change" in businesses, and structured analysis and design was sought to provide a systematic approach to designing and building quality computer systems [Kendall and Kendall 1995].

This highlighted the importance of modeling data and processes as a first step, and resulted in the emergence of new modeling techniques, to assist analysts in portraying and communicating the way in which business activities are both conducted (i.e., "as-is" models) and intended to be conducted (i.e., "to-be" models), for the purpose of process improvement and reengineering [Hammer and Champy 1993; Kubeck 1995]. Structured Analysis (SA) techniques such as Data Flow Diagrams (DFD) [Laudon and Laudon 1988; Kendall and Kendall 1995], Structured Analysis and Design Techniques (SADT) [Marca and McGowan 1998], and $IDEF_0$ [1981] came into existence and were used by business and software analysts. Very soon, in the early 1990s, business and software analysts were equipped with a class of productivity tools—so called Computer-Aided Reengineering (CARE) and CASE tools (Computer-Aided Software/System Engineering) tools—aimed explicitly at improving their routine work through the use of computer automated support [Kendall and Kendall 1995].

In order to address the ever growing complexity of businesses and their supporting computer systems, object-oriented concepts, which had already been in use in programming since 1960s, were applied and formalized into different object-oriented methods and techniques for carrying out and presenting the analysis and design of software systems in 1990s [Coad and Yourdon 1991a and 1991b; Booch 1991; Rumbaugh et al. 1991; Jacobson et al. 1992; Martin and Odell 1998]. The planning and control of software projects became the focus of software engineers, and more innovative modeling

methods and techniques were required to not only define and specify the project elements (i.e., systems, products, and processes) but also integrate and manage the elements throughout the whole software development process [Jacobson et al. 1999; Jacobson and Bylund 2000]. The worldwide popularity of object-oriented technology, especially its emergence in business and software development modeling, is an indication of the power this technology [Jacobson et al. 1995].

2.1.3 Software Development Process

A software system is developed through its own lifecycle stages. The process of software development has been described differently, depending on the methodology of the concern [Sommerville 1997; Coad and Yourdon 1991a; Jacobson et al 1992 and 1999]. The new emerging software engineering methodologies, however, suggest an integrated process approach, which tightly integrates processes and their resulting artifacts (models, model elements, or documents) from definition of the problem to the design and development and realization of a solution (i.e., the supporting information system).

A previous "methods war" of many competing software development methodologies has recently seen a cease-fire with the introduction of the UML, and the Unified (Software Development) Process, proposed as a formalized, integrated software development process [Jacobson et al. 1999; Booch et al. 1999; UML 1.4, 2001]. The process is described in terms of process *phases* (horizontal axis) and process *workflows* (vertical axis), as shown in Figure 2-1. The shaded curves represent the workflows' relative degrees of focus over time).

The workflows are classified into two groups: *core workflows* (business modeling, software requirements modeling, analysis, design, implementation, test, and deployment—each having its related model, e.g., business model—and *supporting workflows* (configuration and change management, project management, and environment). The workflows take place over four phases: *inception* (i.e., development and presentation of system's "seed idea"), *elaboration* (i.e., detailed system development), *construction* (i.e., physical production), and *transition* (i.e., hand in the system to it user). Moreover, every phase may include a number of *iterations*, each defining a milestone for a release of the artifacts. The process phases may be roughly mapped into the phases of a construction project: conceptual planning and design, detailed design and construction planning, construction, and operation and maintenance.

Figure 2-2 further schematically illustrates the core workflows, and their related major actors (i.e., roles) and artifacts in the form of an activity diagram. The figure, however, does not show the iterations (or feedback loops) between activities. Excluding the contents of the model artifacts, traditionally, the first workflow has been the concern of *business process modeling* [Marca and McGowan 1993; Jacobson et al. 1995], while the rest has been the focus of the *software engineering* domain

[Jacobson et al. 1992; Sommerville 1997]. Most software engineering literatures agree on analysis, design, implementation, and testing stages. The *Unified Process* methodology [Jacobson et al. 1999; Jacobson and Bylund 2000], however, encompasses the whole workflows and their related models as an integral part of software development process. It also encourages the evolutionary and incremental nature of the software development process. Modeling business processes can help better appreciate the functional and information requirements needed by the software systems, which are to support the processes [Jacobson et al. 1995; Eriksson and Penker 2000]. Information models (i.e., object models) developed in the form of analysis and design models are the basic building blocks of the desired systems.

The research generally involves working through almost all of the workflows (Figures 2-1 and 2-2) in the context of construction PM. However, regarding the prototype system, it extends only to the elaboration phase, as it is not intended to produce commercial software. For a comparison of the extent to which this research's coverage of the workflows to those of the existing object-oriented modeling methods Figure 1-2 (in Chapter 1) may also be consulted. Due to its relevance to the scope of this research, the next section briefly describes the requirements model and its connections with other artifacts of the software development process.

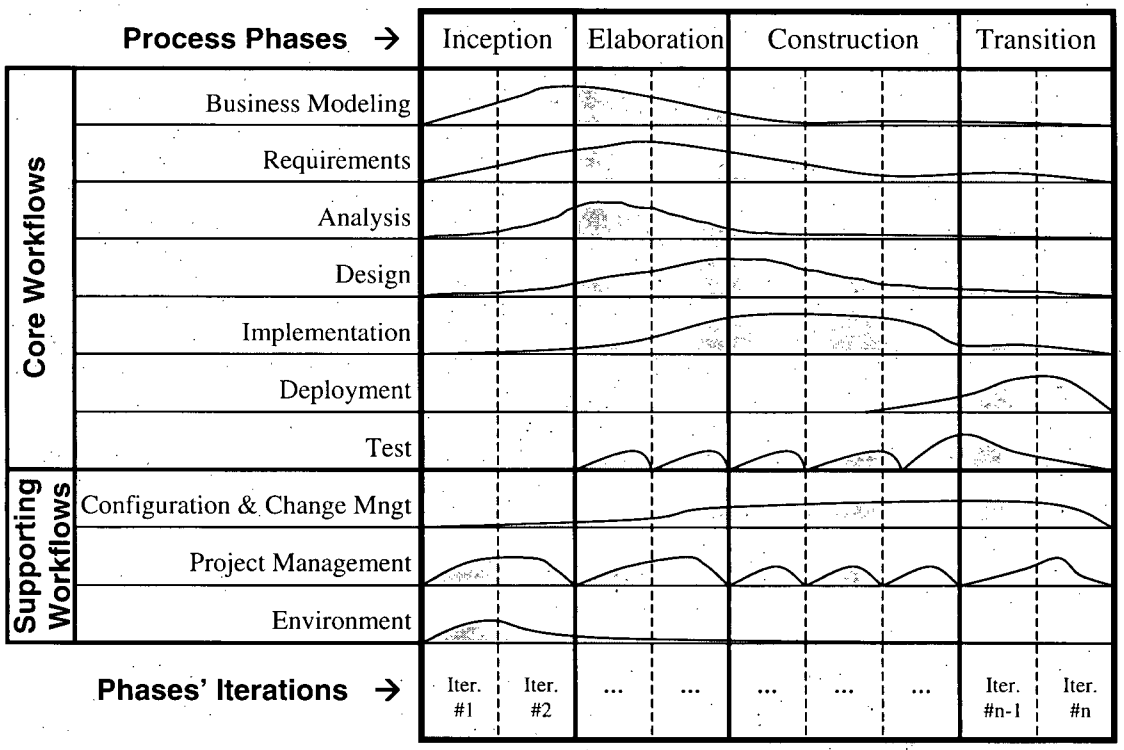


Figure 2–1: The Unified Software Development Process: Phases and Core Workflows

[Based on Booch et al. 1999, and Jacobson et al. 1999]

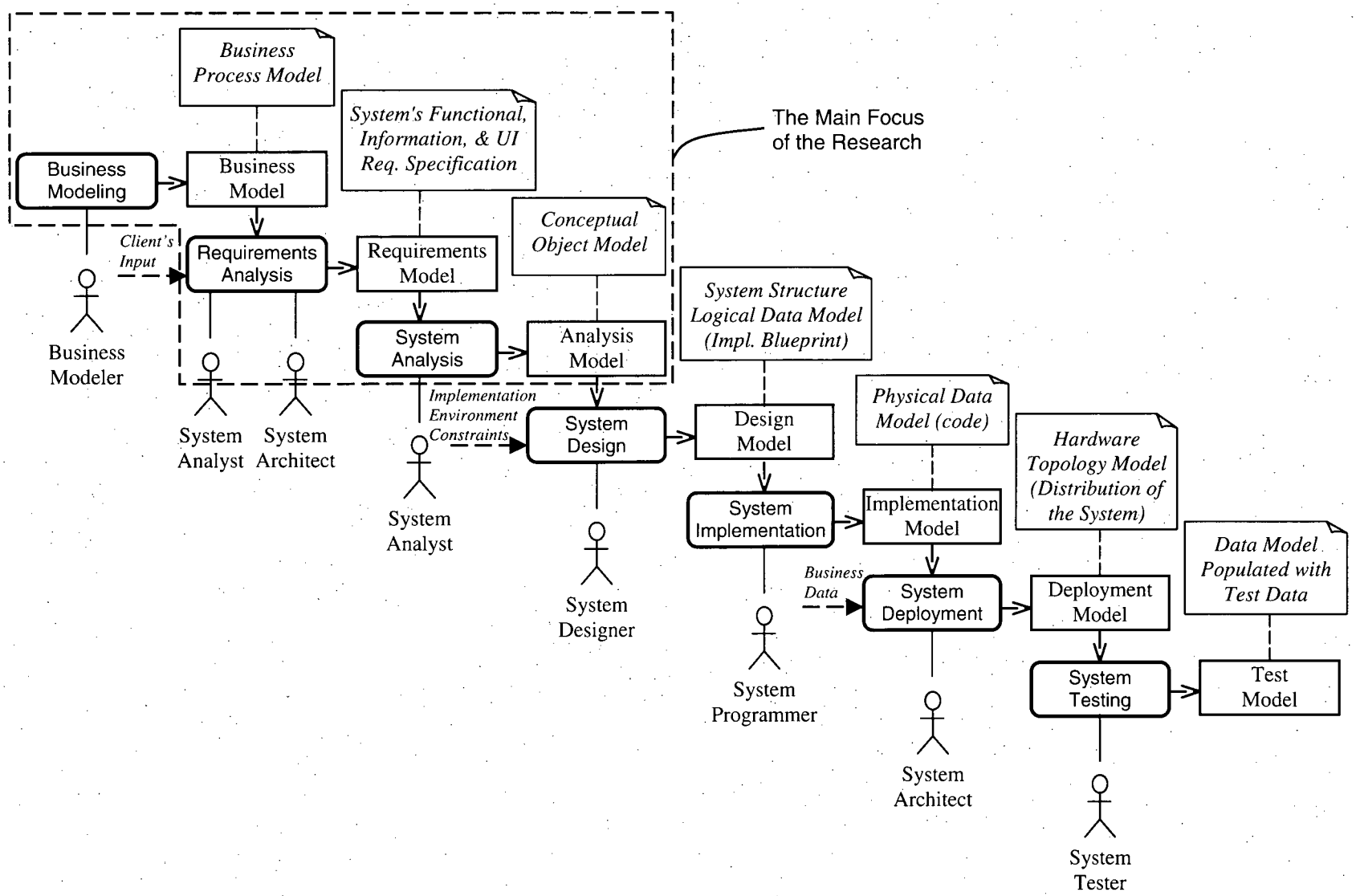


Figure 2–2: A Schematic Activity Diagram of the Core Workflows within each Phase of a Software Development Process

2.1.4 From Requirements Model to the Design Model

The artifacts produced in a software development process (Figure 2-2) flow from one phase to another and are subjected to changes and iterations. The ***requirements model*** defines an *external* view of the software system; it encompasses the developer's view of what the user expects from the system to do (i.e., system's functionality). According to Jacobson et al. [1992], the requirements model consists of: 1) the *use case model* (i.e., actors, use cases, and their relationships); 2) the *domain object model* (i.e., domain objects with or without any of their relationships); and possibly 3) the *interface descriptions* (i.e., UI screens and sequence of interactions between the user and the system). Due to the relevance of the first two components of the requirement model, they are elaborated in the next sections. The requirements model acts as a schematic blueprint for development and verification of other models (i.e., analysis, design, implementation, testing, and deployment models) throughout the system development.

In contrast with the requirements model, the ***analysis model*** defines an *internal* view of the software system. Assuming ideal conditions (i.e., no implementation environment constraints), the analysis model describes the internal structure of the real system and how it works, as opposed to what the system is and does. It is mainly an elaboration of the domain object model into a more detailed object model describing software classes. Jacobson et al. [Jacobson et al. 1992; and 1999] categorize these objects into three groups: 1) *entity objects* (i.e., equivalent to domain objects surviving after software operations; e.g., an *Approval*); 2) *interface objects* (i.e., short-life user interface objects created and destroyed in a user-computer session; e.g., an *Approval Form* used for data entry), and 3) *control objects* (i.e., those handling coordination, sequencing, transactions, and business logic; e.g., *Approval Handler*).

The ***design model*** is "a refinement and formalization of the analysis model", with consideration of the constraints imposed by the implementation environment. At the design stage, the consequences of implementation environment (e.g., the language and platform) start to be considered. A design model, thus, reflects both the implementation environment constraints and results of the analysis model.

2.1.4.1 Domain Object Model

The problem domain object model (or *domain object model*, or just *domain model*) is simply a conceptual class diagram that shows the logical view of the system by presenting the structure of the domain objects. It is like an entity relationship diagram [Kendall and Kendall 1995] with object-oriented concepts such as inheritance relationships and operation content. *Domain objects* represent the *things* that exist or *events* that transpire in the environment in which the system works. They have direct counterparts in the application system, and the system must know about them [Jacobson et al. 1992 and 1999].

Domain object models have been treated somewhat differently in various object-oriented methodologies. In general, the OOA/D [Coad and Yourdon 1991a and 1991b], OOD/Booch [Booch 1991], and OMT [Rumbaugh et al. 1991] methods focus entirely on identifying domain object (using heuristic methods) and specifying them onto an object model, which are used directly as a base for the design and actual implementation. The domain objects are directly mapped onto software classes for implementation; i.e., the requirements analysis and system analysis stages become one stage. In the OOSE [Jacobson et al. 1992] and Unified Process [Jacobson et al. 1999], on the other hand, domain object models are independent of software-details and portray the business environment; i.e., the system analyst focuses on basic domain objects (information requirements) away from their design and implementation details. The model is then used as a basis for both developing and verifying the system's analysis model (i.e., a more robust software object model), design model and so on. Nevertheless, such domain objects are identified through analysis of functionality of the software system, represented in software *use case models*; i.e., they are not totally independent of the software system. They are, in fact, defined in a dual software-business modeling environment.

2.1.4.2 Use Case Model

Use case modeling is an approach to formalization of systems functionality. A *use case model* presents an external view of a system. It is a diagram or model that represents actors (i.e., roles outside the system and interacting with the system, e.g., an estimator or superintendent), use cases (what the actors should be able to do with the system, e.g., order materials), and their relationships in the system. section 2.3.3 elaborates more on use case modeling.

The concept of *use case* is generally applied in two contexts of software engineering and business modeling. Use case modeling was initially applied in the context of computer application systems. The OOSE method [Jacobson et al. 1992] suggests use case modeling as an initial important step of the object-oriented software engineering process and as a means of formalization of functionality of software systems. Due to the close dependency of software requirements on the business context, the idea of use case modeling was then extended to be used for business modeling (covered in the next sections) [Jacobson et al. 1995]. With the advent of process-centric software development and the integration of software development processes, use case modeling has received more attention and become the core of the *Unified Process* methodology [Jacobson et al. 1999; Jacobson and Bylund 2000].

2.1.5 Summary of Modeling and Software Development

The above elaborated on the software development process, its workflows, and artifacts (i.e., models). With a special emphasis on the relationship between business process modeling and software

engineering, this research generally works through most of the workflows of the software development process in the context of construction PM. The frameworks and models developed in this research encompass the integration of business process modeling into the software development process, especially the requirements analysis and system analysis stages. The next sections elaborate on the two major components of this integration (i.e., business process modeling and information modeling) and their related concepts, methods, techniques and tools in general and in the AEC/FM industry.

2.2 Business Process Modeling (Concepts and Views)

The existing theories and methodologies in process modeling all strive to explain how a business is structured and run, and how it can be improved, especially through exploiting IT in the business [Hammer and Champy 1993; Davenport 1993; Kubeck 1995; Jacobson et al. 1995; Kendall and Kendall 1995; Eriksson and Penker 2000]. A typical process model may include a set of activities and their related information: e.g., prerequisites (e.g., inputs, resources, and other criteria), outcomes (i.e., outputs), sequencing, timing, and other constraints. *Business process modeling* views the business from a process point of view—i.e., how business processes interconnect—and it results in a *process definition* [Eriksson and Penker 2000], which can be used for eliciting the functional and information requirements of the software systems supporting the business processes. This section presents some background and research views on business process modeling, as relates to software development.

2.2.1 From Functional Orientation to Process Orientation

The hierarchical structuring of organizations into functional units and sub-units has historically been the main focus of organizing businesses, particularly in the 70s and 80s. Each unit would be responsible for its function and have directly assigned resources for accomplishing it. The modeling of a business in this manner would then typically result in an organizational chart: a diagram including a number of labeled boxes at nodes of a hierarchical tree, each node representing an organization unit (e.g., department, section, or division) and its involved organizational role or personnel.

Over time, it has been realized that this approach to modeling organizations, known as "*functional orientation*", may not be optimal for the overall business [Hammer and Champy 1993]. In particular, the approach has been criticized for lack of a business-wide perspective and for introducing organizational and communication barriers between units, slower working, resource buildups, and intensive administration to the business.

Thus, emphasizing organizations as dynamic systems rather than static hierarchies of functional units and roles, processes became the focus in modeling and improving businesses; what is known as "*process orientation*". Business processes were used for defining the workflow across organizational

boundaries; the *process team* (rather than *department team*) was used as a base for organizing resources; and business goals and objectives are shared. The software applications aimed at supporting businesses were not exempted from this way of thinking and generally followed more or less the same path [Jacobson et al. 1992]. Central to developing effective computer systems remains: 1) assuming a distinction between the two business and computer environments, 2) basing computer systems on models of business processes and objects. These are explained in the following sections.

2.2.2 Business System versus Software System

There exists some confusion among practitioners involved in modeling *software systems* on the distinction between a *software system* and a business system and their related models and model elements. Any software system is defined in a business context and is intended to support the business system's processes. This implies a close relationship between software systems and the business systems in which they are defined and developed. In order to avoid confusions, this research suggests a distinction be made between business systems and their supporting software systems. Such a distinction would then need to be extended to the concepts and models related to these systems, as explained in the following.

2.2.3 Business Process versus Software Process

In capturing the functionality of a computer or business system, *business processes* and *software (application) processes* are sometimes mistakenly described and modeled together or one in place of another. Hammer and Champy [1993] define a *business process* as "a collection of activities that takes one or more kinds of input and creates an output that is of value to the customer." More briefly, Jacobson et al. [1995, p. 3] define business process as "the set of internal activities performed to serve a customer." Several inferences may be made from these definitions. First, the term *internal* implies the system's *boundary*. Second, the term *customer* is used in an extended meaning, and it refers to both those who are *internal* (e.g., an employee) and those who are *external* (e.g., a supplier) to a business (e.g., a construction company) and somehow interact to its processes. Finally, the process should have an output with a *measurable value* to the customer. In some literature [e.g., Chambers and Rand, 1997], the term *business function* is used to mean a *departmental function* within an organization (e.g., payroll and purchasing departments). In this dissertation, however, the terms *function* and *process* are used interchangeably to refer to the actions involved in the business rather than to some departmental units.

The definition of the term *software process* is largely view-dependent. A programmer, for instance, may view a process as a sequence of operations on some data, usually represented in a flowchart. However, getting closer to the view of a business analyst, a software analyst may apply the definition of the business process, but in a computer environment. Thus, in general, a software process

can be defined as a collection of software activities (or operations) that offer some value to the user of the system. Such activities usually take place through a series of interactions between the user and the system. The value resulted from the process would be one of the major concerns to the system analyst.

In summary, by definition, there is a close relationship between the two concepts of business process and software process. Software processes are the central focus of software engineering, while business processes are of concern in business process modeling. In order for a software process to deliver its intended value, however, the context (i.e., the business process) within which the software is defined needs to be studied; i.e., the necessity of *integration of business process modeling onto the software development process* (Figure 2-2).

2.2.4 Business Object Model versus Software Object/Class Model

A major part of the information represented in software systems relates to the *business logic* of the domain, referred to as *business objects*. Such objects are identified and modeled differently in various software engineering methodologies. Moreover, *business objects* and *software objects/classes* and their related models may be mistakenly referred to or used in place of another, while their contexts vary.

In this dissertation, the term *business object* (or *business entity*, as named by Jacobson et al. [1999]) is used to refer to a real world thing that exists or an event that transpires in a business environment (not a software environment) and flow in business processes. It can be a physical object (e.g., building, door, road, and purchase order), or a conceptual object (e.g., project, role, employment, and credit). A *business object model* represents the structure and meaning of such objects.

On the other hand, the term *software class* denotes an object that exists in a software environment. A *software object/class model* represents the structure and meaning of such objects, most of which are counterparts to the business objects identified in the business process modeling stage. At the requirements analysis stage, such a model could be equated to the *domain object model*, as defined elsewhere [Coad and Yourdon 1991; Booch 1991; Jacobson et al. 1992] (section 2.1.4).

Emphasizing on *business objects* as a backbone of the systems, this research assumes three major benefits for establishing a distinction between business and software systems and their related process and object models (Figure 2-3). First, it helps emphasize the focus of a model; i.e., either a business or computer environment. Second, business models can be developed independent of the internal behavior of their possible supporting software systems; i.e., less complexity in the modeling process. The most important of all, the business logic represented in the systems can be elicited directly through studying their related business contexts; thus, enhancing the reliability and usefulness of the systems developed.

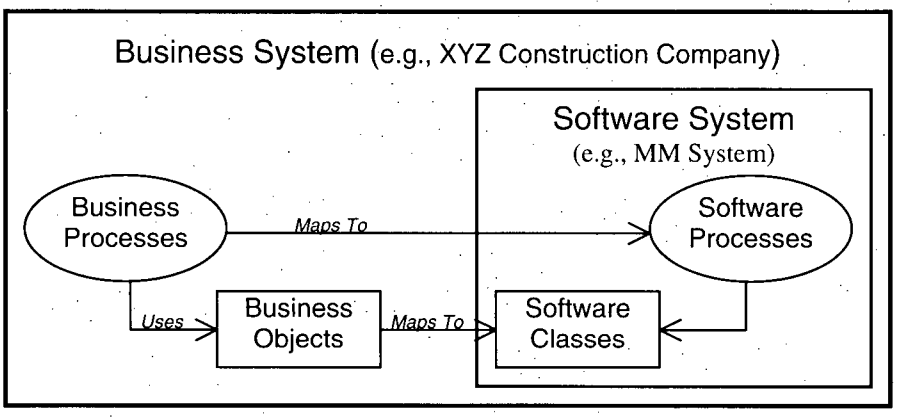


Figure 2–3: Business Process and Software Process in a Business System

2.2.5 Process Reuse, along with Object Reuse

Processes are reused, as data are. The *reusability* concept has long been applied in different contexts, especially in computer programming (i.e., software implementation; e.g., using subroutines as a module of code called in several parts of a program [Etter 1992]); however, it is relatively new in process modeling [Jacobson et al. 1995]. Reusing of processes is very similar to reusing of building designs and specifications from one project to another; e.g., a door type used in several projects. Similarly, information and knowledge about existing processes may be maintained and reused for creation of new processes; e.g., construction methods modifications.

In a business process-modeling context, processes may be defined based on other process definitions. For example, processes such as *Approve Design*, *Approve Invoice*, *Approve Order*, and the like commonly share the basic process of *Approval*. Therefore, the earlier ones can be defined upon the definition of the latter. The same analogy is equally applicable to process reuse in a computer environment (i.e., for software processes). Object-oriented concepts (section 2.1.1.3) such as inheritance provide a good mechanism for modeling the process reuse.

2.2.6 Advantages of Basing Information Systems on Process Models

There are several **advantages** for using process modeling as a basis for information systems. Eriksson and Penker [Eriksson and Penker 2000] explains the advantages of basing information systems on the same basic business model, as summarized in the following:

1) The information system becomes an integrated part of the overall business, supporting the business and enhancing the work and the results.
2) The systems integrate easily with each other and can share or exchange information.

3) The systems are easier to update and modify as dictated by changes in the surrounding environment, goals of the business, or improvements or innovations to the business model. This, in turn, reduces the cost of maintaining the information systems and of continuously updating the business processes.

4) More importantly, business logic can be reused in several systems.

Among the above, the chief advantage seems to be the last one (i.e., *reuse of business logic*) since it can provide for most of the others. Although a one-to-one mapping may not always be the case, most of the objects elicited and presented in the business model would ideally translate or map to information system objects; i.e., *traceability of objects*. For instance, a *quotation* object captured in a business process model would definitely need to be represented in an application that is intended to support a *quoting* functionality. The direct capturing and translation of objects from business models to software information systems would tightly integrate the resulting information systems into the supported business. At a very high level, integration among all supporting information systems of a business can be captured if the systems are built within a total model of the business.

Modeling and documenting business processes up front and getting it approved by the users can considerably prevent information systems development projects from drastic changes and from overall failure to meet project goals. For example, after a few months or years of being in progress, the user of the system may say, "This is not what I had in mind and wanted!" Such a situation may be avoided by incremental modeling of the requirements through *appreciation of the business domain* and establishment of a firm base for system requirements. The produced models can be used as a *means of communication* between systems analysts and users of the system. Moreover, since functionality of the supporting systems are elicited from business process models, they can be traced in the implemented systems by their examination against the process models; i.e., *traceability of functionality*.

The advantages outlined above may be warranted with the use of advanced modeling methods and techniques built on object-oriented concepts, as elaborated in section 2.3.

2.2.7 Summary of Business Process Modeling

This section clarified some basic concepts and research views relating to business process modeling in relation with software system modeling and without concerning their related techniques and tools. In this research, a *business process model* is viewed as *a means of presenting the context for supporting information systems*. The research also makes a distinction between models of *business processes* (what people physically do in the organization), and *software processes* (what a software system does) and their related *object models*. In some instances, however, the term *information model* is used flexibly to refer to a *business object model* or a *software object model*.

The interest of this research generally centers around the use of business process modeling to determine and define the functional and information requirements of business systems and their supporting computer systems. It applies business process modeling as a means of identifying and defining the right business objects, as used in AEC/FM processes, for computer systems development. The materials presented in this section provided the necessary background for explaining the modeling techniques and tools used for software system development in the next section.

2.3 Modeling Techniques and Tools

This section presents an overview of the modeling techniques and tools used for software systems development.

2.3.1 Modeling Techniques

Models are representations of some reality at some level of abstraction. Similar to any other physical product (e.g., a construction facility), the development of software systems involves producing models of the systems and their underlying business contexts. Such models may generally be categorized into *information models* capturing the data requirements of a system) and *process models* (capturing the functional aspects of the system). The modeling techniques used for producing such models may be explained under these two categories as follows.

2.3.1.1 Information Modeling Techniques

The major information modeling techniques used at the conceptual level are IDEF1X (ICAM Definition Language Version 1 Extended) [IDEF1X 1987], NIAM (Nijssen Information Analysis Methodology) [Nijssen & Halpin 1989], and EXPRESS and its graphical component of EXPRESS-G [ISO 1991]. IDEF1X, NIAM, and EXPRESS-G are graphical-based, while EXPRESS is a text-based modeling language. The UML [UML 1.4, 2001] is another modeling language that is intended to support all functionality provided by other modeling languages.

The underlying data model of the graphical notation IDEF1X is an enhanced version of the Entity-Relationship (ER) model, which provides some typical data structures of relational databases [Kendall and Kendall 1995]. It also offers such data structuring mechanisms as generalization/ specialization abstraction (i.e., the enhancement over ER diagram), cardinality (i.e., 1-m and m-m relationships), and type-of relationship Entity types and relationships are named using nouns and verbs respectively. It also provides the possibility to show attributes and (main and foreign) keys in the model.

NIAM is a data modeling technique that offers an extensive list of capabilities for data modeling [Eastman and Fereshetian 1994]. It was heavily used within the STEP project [Froese 1996c], but its use

was diminished after the development of EXPRESS and its graphical component (EXPRESS-G) in the mid 1990s. Being a part of STEP, EXPRESS and EXPRESS-G have become the basic data modeling technique used within STEP. COMBINE [1995] is another CIC project that used EXPRESS as its modeling language, while using NIAM for graphical presentation of its models.

Graphical data modeling representations such as EXPRESS-G and NIAM are usually used on the earlier system modeling stages, when communication between developers is most important (i.e., for analysis and design purposes). EXPRESS is in a textual format, and thus, it is a computer-interpretable language. EXPRESS and other non-graphical modeling languages are usually used for the later stages, when stability and completeness of the software becomes important. Due to its heavy use in this research, EXPRESS-G is elaborated upon in Appendix A.

2.3.1.2 Activity/Process Modeling Techniques

Process modeling techniques provide the concepts and modeling elements required to capture the structure of activities and their relationships in a business or software system. In its simplest form, a process model could include a set of activities (represented by boxes) and their relationships (represented by a set of arrows connecting the boxes) on a single drawing. This technique is very much like a *flowchart* similar to those used by computer programmers for modeling the flow of data in a program [Capron and Williams 1984; Laudon and Laudon 1988]. This technique, which is referred to as a *flat, box-and-arrow flowchart* in this dissertation, allows one to explicitly formalize the sequencing of the activities and workflows involved in a process in the form of graphical models. However, it may not be suitable for modeling a large and complex system.

In such situations, other process-oriented techniques—referred to as Structured Analysis (SA) techniques—such as Data Flow Diagrams (DFD) [Laudon and Laudon 1988; Kendall and Kendall 1995], Structured Analysis and Design Techniques (SADT) [Marca and McGowan 1998], and $IDEF_0$ [1981] may be used. In comparison with the flat, box-and-arrow flowchart, these techniques offer possibilities to model more dimensions of processes (e.g., inputs, outputs, and other constraints), and they dealt with complexity through hierarchical breakdown of processes.

$IDEF_0$ and DFD techniques are similar in some aspects, but they differ in use. DFD is a graphical technique for describing the operation of *computer software*, more specifically *databases*, in terms of the flow of *data* between different *information processing* activities and between participants. It uses four basic symbols for graphical presentation: external entity, flow of data, process, and data store. It identifies the data in a business system, and the flow path to/from processes acting on data (thus, all input and

output of processes are data). The technique is good for analysis of data flow in well-defined, small domains but not for complex situations like modeling of construction projects.

On the other hand, the purpose of $IDEF_0$ models may be one of many, since its modeling method is applicable to any *system* (e.g., computer, business, department, etc.) comprised of *things* and *happenings*. Moreover, $IDEF_0$ can capture types of elements other than just data (e.g., processes and their constraints, resources, inputs, and outputs). Nevertheless, both techniques use a strictly hierarchical approach to breakdown processes. $IDEF_0$ has been widely used for a variety of purposes; however, considering the important role of information management in managing businesses, they have been widely used by software development projects for eliciting information requirements of software systems through modeling business and software processes. It has also been used as a process modeling technique in the AEC/FM modeling community [e.g., Sanvido 1990; ISO 1995a and 1997; IAI 2005]. The $IDEF_0$ technique and its advantages and disadvantages are explained in Appendix A.

The SA techniques are, nonetheless, labeled as "traditional" and criticized as being restricted to "functional decomposition" approaches and not suitable for object-oriented software development [Jacobson et al. 1992]. The traditional business process modeling approaches and techniques, which hail from the structural analysis (or functional decomposition) days of computing, forced the business analysts to think like a computer analyst, while this is not how a business works. Various modeling methods and techniques originated from this way of thinking and built on the use of object-oriented technology in both process and information modeling [Jacobson et al. 1992; 1995; and 1999; Booch et al. 1999; Taylor 1995; Eriksson and Penker 2000], and a collective merging of the prominent ones became the basic foundations of the UML [UML 1.4, 2001]. This is the subject of the following sections.

2.3.2 Advantages of Object-Oriented Concepts and Technology

Several advantages have been realized to using object-oriented concepts and techniques in business and software modeling [Jacobson 1992; Jacobson et al. 1995; Eriksson and Penker 2000]. First, object-oriented technology has proven to be powerful enough in modeling large and complex systems.

Second, using object-oriented techniques, a system can be viewed and captured from different views (e.g., organizational, process, and resource, etc.) rather than just from an organizational structure perspective, captured by traditional organizational charts, for instance. The *process flow*, which is commonly known as "horizontal flow", may not be modeled with traditional organizational charts.

Third, business concepts can be directly mapped to objects and relationships and interactions between objects. Fourth, because of this direct mapping of concepts, the business and software models can be used as an understandable means of communication between modelers (i.e., business modeler and

system analyst, designer, etc.) and the user; i.e., filling the semantic gap between the parties. Consequently, the use of a single set of concepts (i.e., object, class, relationships, etc.) eliminates the need for model translations, and thus, facilitates the transition between the basic stages of system development (i.e., requirement analysis and systems analysis, design, implementation, etc.).

Finally, by first modeling businesses in an object-oriented way, and then building the IT support systems according to object orientation, a further advantage may be achieved [Jacobson et al. 1995]: improving productivity, quality and effectiveness in the whole information system development process. The aforementioned benefits may be realized by using a single object-oriented modeling notation standard capable of modeling both processes (i.e., requirement analysis and definition, design, implementation, etc.) and artifacts (i.e., models and systems) of the software development process. The next section describes this single standard.

2.3.3 The Unified Modeling Language: A Technology for Integration of Models

The Unified Modeling Language (UML) [UML 1.4. 2001] is a general-purpose graphic language primarily designed for object-oriented modeling, though it also allows to model other paradigms or implementation environment. It brings all software development methodologies into a shared language of communication, and it is becoming an international standard (i.e., ISO) [OMG 2005]. The UML can be used for analysis and design of a variety of development processes; however, its constructs have been best used for specifying, visualizing, constructing, and documenting software-intensive systems (i.e., analysis, design, implementation, and testing models) [Fowler and Scott 1997; Booch et al. 1999; UML 1.4 2001].

The concept *Use Case* is one of the central modeling elements of the UML that can represent the *functional* requirements of systems of any type (e.g., software, hardware, and real-world business systems) and any level of complexity [Jacobson and Bylund 2000]. Use case modeling originally came into existence in the domain of software development [Jacobson et al. 1992], and they were later applied to business process modeling (still for the purpose of software development) [Jacobson et al. 1995]. More recently, they have also been used as a means of integration and management of both workflows and the artifacts of the software development process [Jacobson et al. 1999, UML 1.4, 2001; Rational 2002].

Use cases can be used as a means of capturing, documenting, communicating, and controlling the requirements throughout the software development life cycle. They also provide a framework for specification of non-functional requirements (e.g., performance issues) and project details (e.g., development responsibilities). Such system requirements can be visualized in *Use Case Diagrams*, which present an external view of the system in the form of *Actors* (a person or system agent external to and interacting with the system), *Use Cases*, and their relationships [Booch et al. 1999].

Business use cases describe the functionality of a business system; much like the way *software use cases* (often just called "use case") describe the functionality of the software system used in the business. A *business use case model* describes the structure of business processes (i.e., sequence of activities) of a company in terms of business use cases and business actors (i.e., customers).

Despite the fact that use-case modeling concepts are still under evolvement (unlike other well established techniques such as $IDEF_0$ [1981], Appendix A), they offer great advantages [Fowler and Scott 1999; Booch et al. 1999; Cockburn 2001; Jacobson 1992; Jacobson et al. 1999, Jacobson and Bylund 2000; Schneider and Winters 1997]. Use cases and their various relationships (e.g., *generalization*, *extend*, and *include*) provide a mechanism for model *reusability* (through factoring out of *commonalties* of processes and reusing them as needed), and thus a means of managing system *complexity* and *extensibility*. Such a mechanism is not available in other process modeling techniques, such as $IDEF_0$ and DFD (which are strictly hierarchical; section 2.3.1.2). The possibility of linking this functional view and other views of the system (e.g., information) can further warrant system-model *maintainability*.

While use case models focus on the external view of the system away from implementation details (i.e., *what* the system does, rather than *how* it does it), other UML modeling constructs, diagrams, and mechanisms may be used for other purposes; e.g., *Activity Diagrams*, *Objects*, and *Class Diagrams* for representing the internal view of the system (i.e., how system tasks are performed), *Packages* for grouping and structuring of various model elements, and extensibility mechanisms (i.e., *Stereotypes*, *Tagged Values*, and *Constraints*) for extending the semantics (i.e., the meaning, not the structure) of a model element through defining new conventions without introducing new UML concepts.

The UML's extensibility mechanisms have been used for defining *UML profiles*—devised at OMG, Object Management Group Inc. [OMG 2005] as "a predefined set of Stereotypes, Tagged Values, Constraints, and notation icons that collectively specialize and tailor the UML for a specific domain or process (e.g., the Unified Process profile)" [UML 1.4, 2001]. The currently defined profiles are: 1) *Unified Process* (which focuses on the whole *process* of software systems development), and 2) *Business Modeling* (which focuses on business process modeling, mostly using UML *Activity Diagrams*, for preparing the "as-is" and "to-be" models of business systems). Though, not all of the modeling elements suggested by the Business Modeling profile was found useful for the purpose of this dissertation. For example, this research does not focus on modeling of specific organization instances, and it has not aimed at modeling of human resources organizations and responsibilities.

Overall, the UML offers many advantages over other modeling techniques. In particular, concepts such as *Package*, *Use Case*, *Use Case Model*, *Object Model*, and *Stereotype* were found relevant and useful to the research objective of "information modeling through business process modeling". However,

for the sake of clarity, these concepts are applied slightly differently in the dissertation. For example, considering the discussions made in section 2.2, this research has used the term *business use case* (instead of *use case*) versus *software use case* (Chapter 7). Use cases were specifically found as a good candidate for the purpose of functional modeling. The research suggests use case modeling as an effective technique for classifying, organizing, and managing PM functions and coping with the complexity inherent in the functions as well as handling variations of process setups and views of modelers (section 3.4), which are commonplace in less standardized environments such as AEC/FM processes. Appendix A provides a brief background, an overview of modeling constructs, and advantages of the UML. Moreover, chapters 7 and 8 describe how the research made a use of the UML to achieve its objectives.

2.3.4 CASE Tools

In the early 1990s, system analysts were the first group to benefit from new productivity tools created explicitly to improve their routine work through the use of computer automated support; so-called CASE tools (Computer-Aided Software/System Engineering). Distinguishing CASE tools from Computer-Aided Reengineering (CARE) tools (used in reverse engineering), Kendall and Kendall [1995] classify CASE tools into three groups:

1) *Lower CASE Tools*: are used mainly for program source code generation (often with multiple language support), from the project model information stored in the CASE repository.
2) *Upper CASE Tools*: allow the analyst to modify the system design by accessing the information stored in the CASE repository.
3) *Integrated CASE Tools*: assist two or more of the software development functions in an integrated manner; i.e., automatically feeding of products from one process to the next or previous one.

Some of the existing CASE tools are very sophisticated and support a wide range of operational and managerial functions involved in the management of software engineering/reengineering projects in an integrated manner. Such functions include project management (e.g., estimation, and scheduling of the project), documentation, requirement analysis and modeling, design, code generation (i.e., forward engineering), reverse engineering, model management, testing, configuration and version/release management, etc. They usually support a number of modeling methodologies (e.g., James Martin, Coad and Yourdon, Rumbaugh, Booch, Jacobson, Martin/Odell, and UML) and multiple languages (e.g., C++, Java, and Visual Basic) [e.g., Rational Rose 2002; Together 2002; ArchStyler 2002].

The new generation of integrated CASE tools deal with the whole software development process, as viewed by PM of a software development project. Such *process oriented* CASE tools support such functions as project coordination management, configuration (version and release) management, etc.

Objectory is an example of such products built on the OOSE methodology [Jacobson 1992]. Rational Unified Process (RUP), a product by Rational Software Corporation, is a newer, refined and renamed version of Objectory product that has been under continuous development. Integrated with a suite of other Rational applications, it is intended to support all managerial and core process workflows, including business process modeling (Figure 2-1), involved in software development projects.

There are several *benefits* listed for using CASE tools. Among them are those listed by Kendall and Kendall [1995]:

1) Increasing analyst productivity (in production and distribution of model diagrams)
2) Improving analyst-user communication
3) Integrating the software development process's activities and handling change management.
4) Accurately assessing maintenance changes.

To the above, however, two more benefits may be added.

1) Lowering the time and cost of forward and reverse engineering (i.e., generating software code from model and vise versa), and
2) Improving project communication at the project level (i.e., among all stakeholders: end users, analysts, developers, system integrator, testers, technical writers, and project mangers).

The benefits of CASE tools may be constrained by their extent of coverage of software development process workflows, integration level of the workflows (i.e., models interoperability), and software-related issues (i.e., user-friendliness, performance, etc.). As a CASE tool, Ensemble Streams™ [2000] was used for business process modeling and object modeling parts of this dissertation. This tool is briefly described in Appendix A. Moreover, chapters 7 and 8 describe the results of this application.

2.3.5 XML (eXtensible Markup Language) Technology

XML (eXtensible Markup Language) is a simplified subset of the Standard Generalized Markup Language (SGML) defined by the World Wide Web Consortium (W3C) specifically for Web applications (www.w3.org/XML). XML is a *data language* that provides a standard format to describe different types of data (e.g., a purchase order, an invoice, a book, a person, etc.). It specifies just the static data structure (i.e., *physical data*) in an ASCII-based text (rather than a compiled object code), regardless of its *format* (i.e., presentation of data to the user). Therefore, an XML file's data content can be easily understood. XML parser tools can easily parse through the content of the file to dissect, interpret, and manipulate XML data (www.w3schools.com/xml).

XML can be used in conjunction with HTML (Hypertext Markup Language), which provides the *format* of data, and with HTTP (Hypertext Transport Protocol), which provides *transport* of data. It creates structured data, which can be interpreted by both client and server side applications. XML is also used in conjunction with CSS (Cascading Style Sheets) or XSL (eXtensible Stylesheet Language) schemas, which are used for defining XML formats; i.e., *shared vocabularies* for displaying XML documents.

In order to meaningfully apply XML in any industry, the business objects of the domain must be defined onto a set of accepted customized data tags (i.e., domain-specific DTDs). A few of such current XML-related technologies are briefly described in the following.

1) ***ebXML (E-Business XML)***: is aimed at providing standards-based business document sharing for a global economy (www.ebxml.org); i.e., focusing on business processes and their associated messages and contents, business process sequences, company profiles, and trading partner agreements.
2) ***cXML (Commerce XML)***: is aimed at providing standards-based business document sharing in e-commerce applications (www.cxml.org); i.e., a set of DTDs to facilitate processes such as purchase orders, change order, acknowledgments, status updates, shipping notifications, and payment transactions. It is based on web client-server communications, tied closely to HTTP.
3) ***BizTalk***: is a Microsoft initiative that is now open. It is a part of Microsoft.NET initiative (www.microsoft.com/biztalk). It provides guidelines for common methods for publishing XML schemas and for incorporating XML messages in software.
4) ***BLIS-XML (Building Lifecycle Interoperable Software-XML)***: is an XML encoding initiative under the BLIS (Building Life Cycle Integrated Systems) project [BLIS 2002] (www.blis-project.org), which was formed in July 1999 to support and coordinate the implementation of IFC-R2.0-based software applications [IFC R2.0, 1999a] in the AEC/FM industry. BLIS-XML was conceived with this recognition that, as "a business reality", STEP's EXPRESS data definition language and its related physical file format (i.e., Part 21) cannot be adopted as an effective means for data exchange over the Internet [BLIS_Update_2000]. BLIS-XML is, in fact, a methodology for encoding EXPRESS-based information (i.e., the IFC object model) in an XML format, so that a BLIS-XML schema can be automatically generated from an EXPRESS schema. XML-supporting applications (e.g., architectural design, HVAC design, and quantity takeoff / cost estimating) could then read/write IFC model data in XML format. The BLIS-XML was later renamed as IfcXML R2.0.
5) ***ifcXML***: is an international XML representation of the IFC object model. ifcXML is the first official IAI methodology for an automatic translation of the IFC R2x object model [IFC R2x, 2000a] from its EXPRESS format to an XML format (technically termed a language binding).

6) ***aecXML***: is an implementation methodology to define the physical file format for specific AEC/FM e-business transactions (i.e., "*packets of information*", e.g., ordering goods, not the entire project information) over the Internet. It is a consortium of a number of AEC/FM software companies and builds on the experiences of ifcXML and BLIS-XML, and it is carried out by the North American Chapter of the IAI project [IAI-NA 2005]. Similar to EDI approach (Electronic Data Interchange) [Trus, and Collica 1995; Gibson and Bell 1990; Stukhart 1995], the aecXML approach involves transferring information about single business transactions from one location to another. It features the addition of using the IFC object model, the XML, and the Internet technologies.

It has been suggested that XML "has a greater chance of success in domains where the data model is relatively simple and has been generally accepted by software providers in that domain" [Khemlani 2001]. This confirms the aecXML approach over the ifcXML and BLIS-XML approaches. The aecXML approach focuses on *data exchange* between two parties or systems, rather than *data sharing* within a discipline or among all project participants.

In summary, the XML technology is taking over and improving upon the concepts introduced by EDI, so that the results could be integrated with the advantages suggested by Internet technology. It provides for interoperability of *content* (i.e., business data) and *style* across applications and platforms, and multiple views of the same content can be created. In order to exploit the XML's advantages, however, the identification and formalization of the *contents* (i.e., information) and *contexts* (i.e., business processes) of such data structures in an integrated manner remains to be a real challenge, especially in a fragmented industry such as the AEC/FM industry; a challenge that is attempted in this research. Appendix C explains how XML was used in the prototyping part of the research.

2.3.6 Summary of Modeling Techniques and Tools

This section presented some background on the modeling techniques and tools used in the process of developing a software system to specify and communicate the various aspects (e.g., information and process views) of the system and its business environment. The extent of coverage of these aspects may vary from one technique or tool to another. So far, UML has been recognized as a new general-purpose *modeling language* (as opposed to only a *diagramming technique*) and that inherits the capabilities of its preceding information and process-modeling techniques exploits the advantages of object-oriented technology. It can be used for modeling any system or development process, though it has been best used for specifying, visualizing, constructing, and documenting software-intensive systems (i.e., the analysis, design, implementation, and testing models). Nevertheless, the AEC/FM project modeling efforts have not exploited the UML's advantages.

This research, which involves both process modeling and information modeling, uses the UML's modeling constructs for business and software process modeling, while EXPRESS-G diagramming is used for presenting information models of existing CIC projects as well as those suggested by the research. The choice of the UML in the research is due to its advantages over other modeling techniques as well as its potential in satisfying the criteria set for the framework and models of this research (elaborated in Chapter 4). The selection of EXPRESS-G was based on two major reasons: 1) its simplicity (to draw and comprehend), and 2) its use for many other existing AEC/FM information models, especially IFC object model (so no conversion would be required when they are referred to). Moreover, the research uses Ensemble Streams™ as a CASE tool for the purpose of process modeling and object specification in an integrated manner (chapters 7 and 8). The next section presents an overview of AEC/FM project modeling.

2.4 AEC/FM Information Modeling

This section focuses on information in the AEC/FM industry. It describes different types of information models, and presents a general overview of a number of major AEC/FM modeling projects (addressing the CIC environment) and their resulting models. Detailed technical discussions of the models and their related issues, however, are presented in Chapter 4.

2.4.1 Types of Information/Data Models

Types of information models developed in the AEC/FM industry are elaborated by Froese [1996a] and Hannus and Pietilainen [1995]. Some of the major types of AEC/FM information models, which are widely referred to in the literature, are summarized as follows.

2.4.1.1 Meta-Models

Meta-models are models of models. "A meta-model is a model that tells you how to develop a model" [IFC R2.0, 1996c]. It defines the *modeling language* for representing definitions in the form of a conceptual model, for instance. It includes basic, generic data modeling constructs, such as objects, classes, attributes, and different types of relationships (e.g., inheritance, decomposition/ aggregation and associations); e.g., Hannus et al. [1996] and section 7.2 of this dissertation.

2.4.1.2 Conceptual Models

Conceptual models are information models at a high level of abstraction. They concern the logical or semantic (as opposed to syntactic) description of the data item types and their interdependencies in some domain (i.e., basic objects, their relationships, and possibly a few attributes of

the objects). They do not, however, hold business rules that change (e.g., an employee's minimum salary), and they are not directly implementable; i.e., they are not sufficiently detailed for direct instantiation.

2.4.1.3 Core (or Reference) Models versus Application Models

With the emergence of the idea of common object models and their use for CIC [Teicholz and Fisher 1994], core (or reference) models were introduced as an integrating mechanism between related application systems (e.g., structural engineering, HVAC design, material management, etc.) [Wix and DP 1995]. One of the definitions given for a *core model* is: "an information model that captures the requirements and knowledge of a number of different domains for a specific industry sector" [ISO 1995c]. Unlike a core model, an *application model* is usually intended to describe the structure of the actual data in an application area. Some related application models use the concepts defined in the high-level core model to exchange information. An example of such core models in the STEP project is Building Construction Core Model (BCCM) [ISO 1997], which is used by Application Protocols (APs) such as AP225 [ISO 1995a] (section 2.4.4.4). In the IFC model [IFC R2x, 2000a], the notion of core model is realized in a number of layers (section 2.4.4.7).

IRMA (Information Reference Model for AEC) [Luiten et al. 1993; Froese 1994a] and GARM (General AEC Reference Model) [Gielingh 1988] are two more examples of AEC core models. While IRMA is a very high-level core model attributed as a process data model, GARM is oriented towards modeling AEC products. A description of GARM is presented in section 2.4.4.1.

Core models are a type of conceptual model that are not generally intended to be instantiated. They are used as a reference by some related application models to share and exchange information. The existing AEC/FM standard information models concern, at most, a specific industry (e.g., building industry, road construction, process plants, etc.); nonetheless, it is not uncommon that a construction project consists of multiple types of construction (e.g., a project including such tasks of construction of a number of buildings, development of the site, and construction of an access road to the site, etc.). There are some concepts shared between all types of construction projects.

2.4.1.4 Type Models versus Instance Models

Type models deal with *classes* of objects (e.g., building, wall, window, etc.), while instance models concern *individual* object instances (e.g., my house and UBC Main Library). *Type models*, which are created using modeling languages (such as EXPRESS), are usually used as data type declarations by software systems for instantiation (i.e., for creating actual project information objects, containing actual data). Most AEC/FM models are type models. *Instance models*, on the other hand, define the data for real individual instances of objects, instead of classes of objects. The instances could be either real (tangible or

intangible) occurrence objects (e.g., the door WD206 in room 206 of building XYZ; bedroom #2) or type objects (e.g., a door type WDS-T1, defined in a door schedule) (section 4.2.2.1).

Instance models are not as elaborate as type models. They are usually targeted as a short cut to avoid the issues associated with type models. The "slowness" of development and adoption of type models (such as STEP) has been considered as one of the most important reasons for developing instance models [van Nederveen and Tolman 2001]. However, the issue of slowness may have considerably been reduced in such information modeling projects as the IAI (discussed in section 1.2.5). Moreover, instance models are limited to just creation of project information, for example, for an individual project (e.g., UBC Main Library) and may not be reused as a template for other projects. Therefore, the modeling activity should be repeated for each new project. This is not a problem for a type model, which can be used over and over as a template for creation of information about multiple projects.

2.4.2 Product and Product Model

Product models are devised as a means of capturing product information in some domain. The term *product*, in a materialized sense, is defined as "thing or substance produced by a natural or manufacturing process" [Oxford Dictionary 1989]. This definition is close to the one given in a data modeling sense by ISO [1992]: "a thing or substance produced by a natural or artificial process". *Product information model* is, thus, "an information model which provides an abstract description of facts, concepts, and instructions about a product or set of products" [ISO 1992]. IAI defines product as:

> "*Any object, manufactured, supplied or created for incorporation into an AEC/FM project. This also includes objects that are created indirectly by other products, as spaces are defined by bounding elements. Products can be designated for permanent use or temporary use, an example for the latter is formwork. Products are defined by their properties and representations. Products occur at a specific location in space. They can be placed relatively to other products, but ultimately relative to the world coordinate system defined for this project*" [IFC R2x, 2000a].

The above definition of product applies to *occurrences of products* (e.g., the "west external wall" of a building), as opposed to *product types* (e.g., foundation type "FS2")—discussed in section 4.2.2.1).

2.4.3 Product (data) Model versus Process (data) Model

The term *product model* has been referred to in the literature more consistently than the term *process model*. The former is usually used to refer to models that convey information about the structure and meaning of some physical entities produced in a process (e.g., buildings and roads), independent of how they are built. The term *process model*, however, is generally used in two different, but relating, contexts. Some [e.g., Fischer and Froese 1996; Froese 1996a] have used the term to refer to *process data/information models* (i.e., information about the structure and meaning of the actions and non-product

elements involved in a process—e.g., resources, documents, and regulations), while others [e.g., IFC R2x, 2000a] have used the term to mean *activity models* or *business process models* (section 2.2), which portray the structure of the set of activities involved in a process and their flows.

Product data models and process data models describe two complement views of project information. Though tightly interwoven, the total *project data* can be divided into *product data* and *process data*. A product data model gives information about WHAT the physical product of a project is, while a process data model describes HOW the product is developed (Figure 2-4). Examples of the models that are oriented towards product information include GARM [Gielingh 1988] RATAS [Bjork 1989a & 1989b], COMBINE [Augenbroe 1995], and most STEP models [Froese 1996c]. Also, examples of proposed process data models are IRMA [Luiten et al. 1993], and ICON [Aouad et al. 1994]. More examples are provided by Froese [1996a].

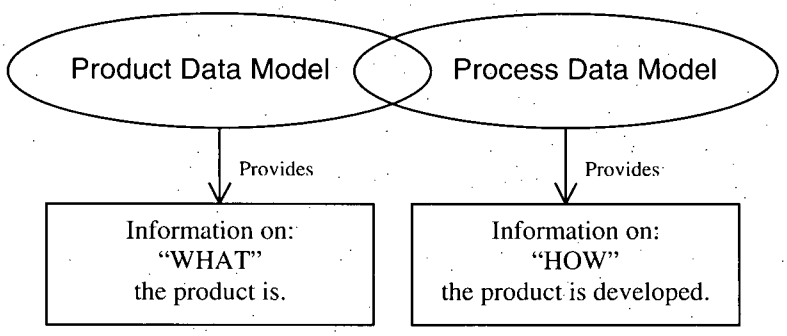


Figure 2–4: Product Data Model versus Process Data Model

In practice, product and process data models overlap. For example, "Pour *concrete* in the *second-floor slab* of a *building*", as a construction *Activity*, makes no sense if there would not be any indication about the *product*. Such product information could be commonly referenced and reused by many PM functions. In fact, it would act as a linking element between PM functions. Two instances of such referencing are 1) in cost control: "Cost of *concrete* used for *superstructure* of the *building*", and 2) in quality control: "Quality of *concrete* used for *second floor* of the *building*". This example highlights a number of points including:

1) The essential role of product model, as a kernel of reference, in integration of PM functions;
2) Dependency of process (data) model on product model and, thus, necessity of approaching towards *project models*, encompassing both product and process information [Fischer and Froese 1996; Froese 1996a]; and
3) Possibility of defining the *common information* into a *core model* (section 2.4.1.3).

In this dissertation, in order to be in agreement with the CIC literature, the terms *product model* and *process data/information model* are used to refer to information models of AEC/FM products and processes respectively. To refer to information models that encompass product and process information, the term *project model* is used. On the other hand, the terms *process model*, *activity model*, and *business process model* are used interchangeably to refer to the models that picture a set of activities and their structure and flows (e.g., breakdown and sequencing) in a process.

2.4.4 An Overview of Some of the Existing AEC/FM Information Models

This section briefly describes some of the major AEC/FM modeling projects and their models. The projects are selected from the most famous product, project, and process modeling initiatives, which vary in their impact levels (international, national, and research and development projects) and domains and scopes (e.g., core models versus application models; total AEC/FM versus building industry). In some cases (e.g., the COMBINE model), for the sake of uniformity with other models presented in the dissertation, the models were transformed form their original representation technique (e.g., NIAM) to EXPRESS-G. The detailed discussion and elaboration of the models, however, is left to next chapters, as the subject matter requires.

2.4.4.1 GARM [Gielingh 1988]

GARM (General AEC Reference Model) is an abstract, high-level data model, which is supposed to be general enough to serve the needs of all AEC application areas. It deals with modeling of conceptual design of AEC products (more specifically buildings). It originally focused on the modeling of buildings, building systems, and building elements, but it was generalized such that it could be applicable for other product types, such as civil works, plants, and ships [Gielingh 1988].

GARM is a framework that deals with a number of architecture ideas within the STEP project. It sets down a number of basic modeling principles and concepts, which include:

1) Introduction of the concept of a Product Definition Unit (PDU), which highlights the gradual changes of product in its life cycle.
2) Separation of the ideas of Functional Unit and Technical Solution; along with the PDU principle.
3) Introduction and elaboration of the generic/specific/occurrence paradigm, which may be mapped into the idea of type/occurrence objects in this research.

GARM models are presented and formalized using IDEF1x (for graphical notations) and EXPRESS modeling language. For explanation purposes, however, the IDEF1x diagrams include more entities than what is within the scope of the GARM formalized into its EXPRESS models.

Figure 2-5 shows an EXPRESS-G version of the most high level entities of GARM. The dotted boxes denote those entities that are not included in the scope of GARM. As a basic and the most generic product entity, *PDU (Product Definition Unit)* is defined as "a product or any subsystem, part or feature of a product, interesting enough to record information about". Thus, no decomposition is pre-defined in GARM, and decisions as how PDU's are related to each other is assumed to be modeled by the user (i.e., the designer/architect). In order to support the product life cycle, PDU is specialized into seven fundamental types of PDU's, each corresponding to a specific stage of PDU in the product development process. Nevertheless, GARM deals only with the first two of them: *Functional Unit* (FU, i.e., as-required PDU), and *Technical Solution* (TS, i.e., as-designed PDU) subtypes. Considering levels of abstractions, the PDU is also subtyped into three classes: *Generic* (i.e., a parametrically defined PDU), *Specific* (i.e., a generic PDU with fully defined parameters' values), and *Occurrence* (i.e., containing only place and orientation of a PDU). A PDU is considered to have a set of *Characteristics*, which are specified by an *Aspect* (e.g., strength, cost, safety, and durability). A Characteristic can be *required* (identical to the Functional Requirements), *expected* (identical to analysis results in the conceptual design of products), or *measured*. The "*expected characteristics*", however, is not included within the scope of GARM.

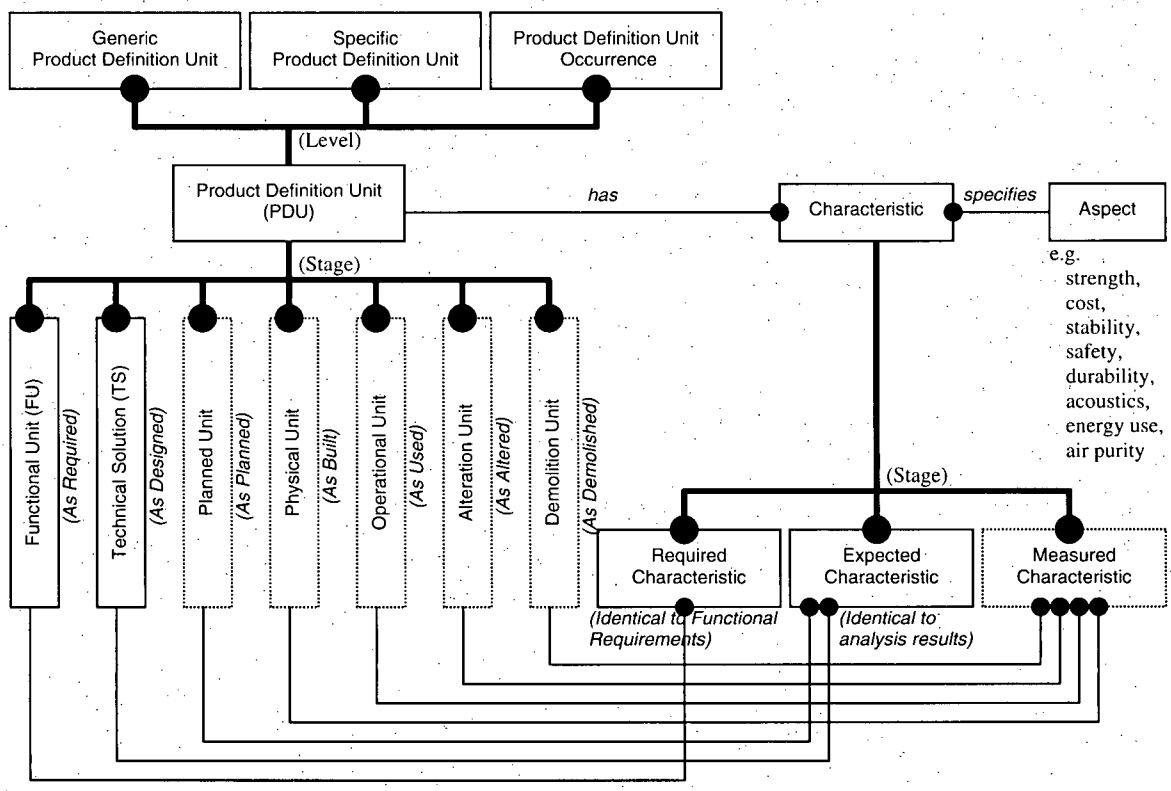


Figure 2–5: High-Level Entities in the GARM Model

2.4.4.2 STEP (ISO 10303) [ISO 1995a]

STEP (Standard for the Exchange of Product Model Data) is an International Standard (established in 1983) for the computer-interpretable representation and exchange of product data. Its objective is to provide a neutral mechanism capable of describing product data throughout the life cycle of a product, independent from any particular system. The result of this effort was intended to be used as a basis for implementing and sharing product databases and archiving [ISO 1995a, Froese 1996c].

STEP has a layering approach to product data modeling. Its data models are generally in either of two forms: Integrated Resource (IR) or Application Protocol (AP). The IRs define a group of high level resource constructs used as the basis for product data [ISO 1992]. Building Construction Core Model (BCCM) [ISO 1995b and 1997], as an IR, and AP225 [ISO 1995a], as an AP, are two models that were developed under the umbrella of Building Construction Group of the AEC Team of STEP.

2.4.4.3 BCCM (ISO 10303-106) [ISO 1995b and 1997]

The Building Construction Core Model (BCCM), or Part 106, is an Integrated Resource (IR) within the STEP standards. It is a high-level information model that describes information of common interest to all applications throughout the domain of building construction, which is defined as "everything which is constructed or results from construction operations and which has as its primary purpose the provision of shelter for its occupants or contents and is usually enclosed and is designed to stand permanently in one place". BCCM was intended to be implementation independent, and thus, it is only implemented indirectly via Application Protocols (AP), which use BCCM's constructs as a resource [ISO 1995b and 1997]. It covers both product and process related information. Figure 2-6 shows a partial classification hierarchy of product-oriented objects defined in the latest version of BCCM [ISO 1997]. In this figure, "bc" and "ABS" stand for building construction and abstract entity respectively.

Concerning products, BCCM deals with both physical (i.e., tangible) and spatial elements in buildings. It also reflects the shape-related (i.e., topological and geometrical) information using functional classification of the product. The classification of building assemblies and their geometrical information accounts for a major part of the model. Although it is theoretically defined as an IR (independent of and used by APs), BCCM has some references to AP225, which is an AP itself in the STEP project.

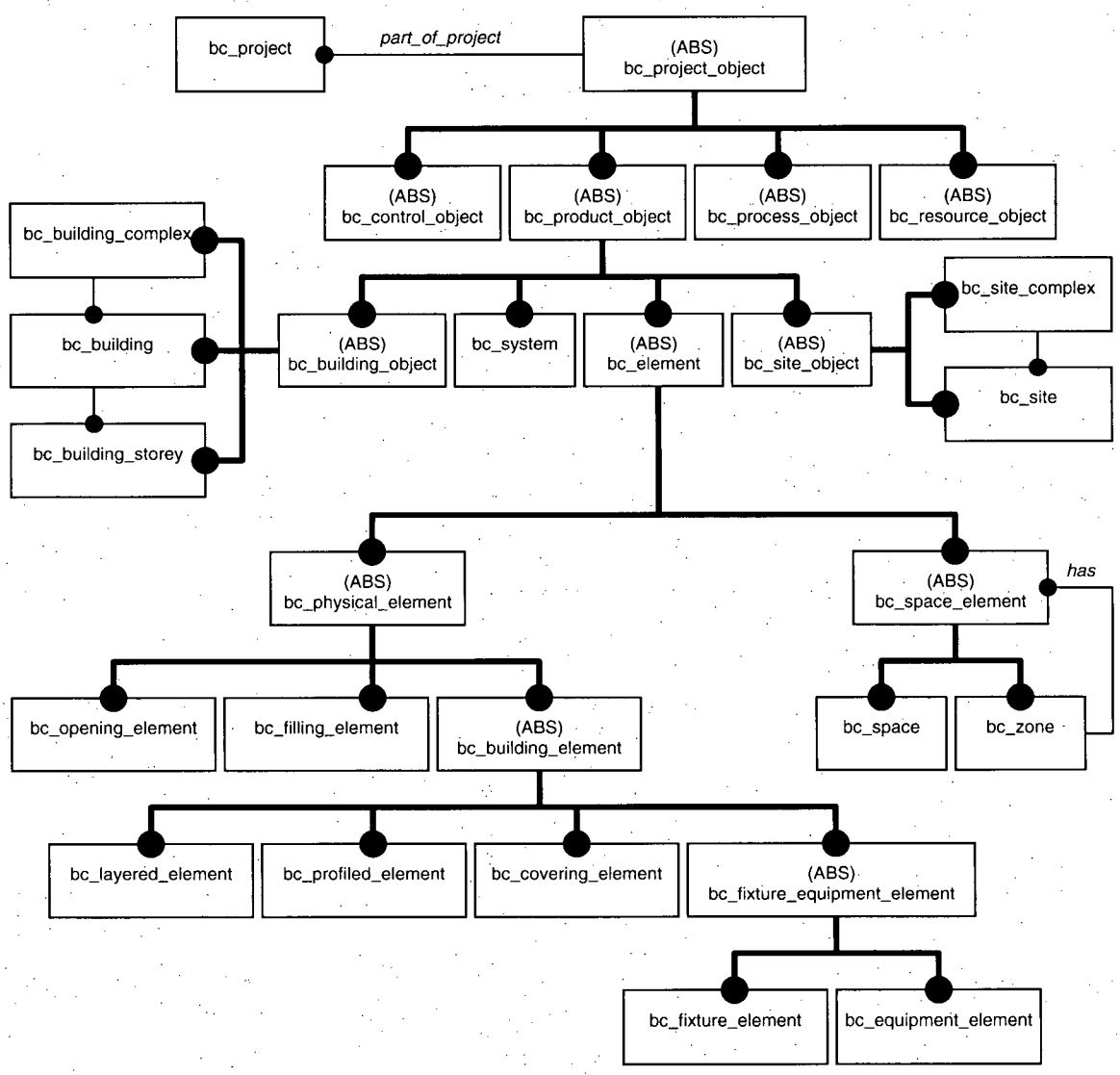


Figure 2–6: A Partial Classification Hierarchy of High-Level Objects in BCCM

2.4.4.4 AP225 (ISO 10303-225) [ISO 1995a]

AP225 (or ISO 10303-225, i.e., Part 225 of STEP) is an application protocol (AP) developed in the Building Construction Group of STEP in the early 1990s for "Building Elements Using Explicit Shape Representation". It uses BCCM as one of its resources, and defines the shape-related product information in a building project at an elemental-detailed level (i.e., building element), as required in detailed design. It specifies an AP for exchanging building element *shape*, *property*, and *spatial arrangement* information between AEC applications using explicit (not parametric) 3D shape (or geometrical) representations. Figure 2-7 illustrates a partial simplified model of product view in AP225 and its basic entities and their relationships.

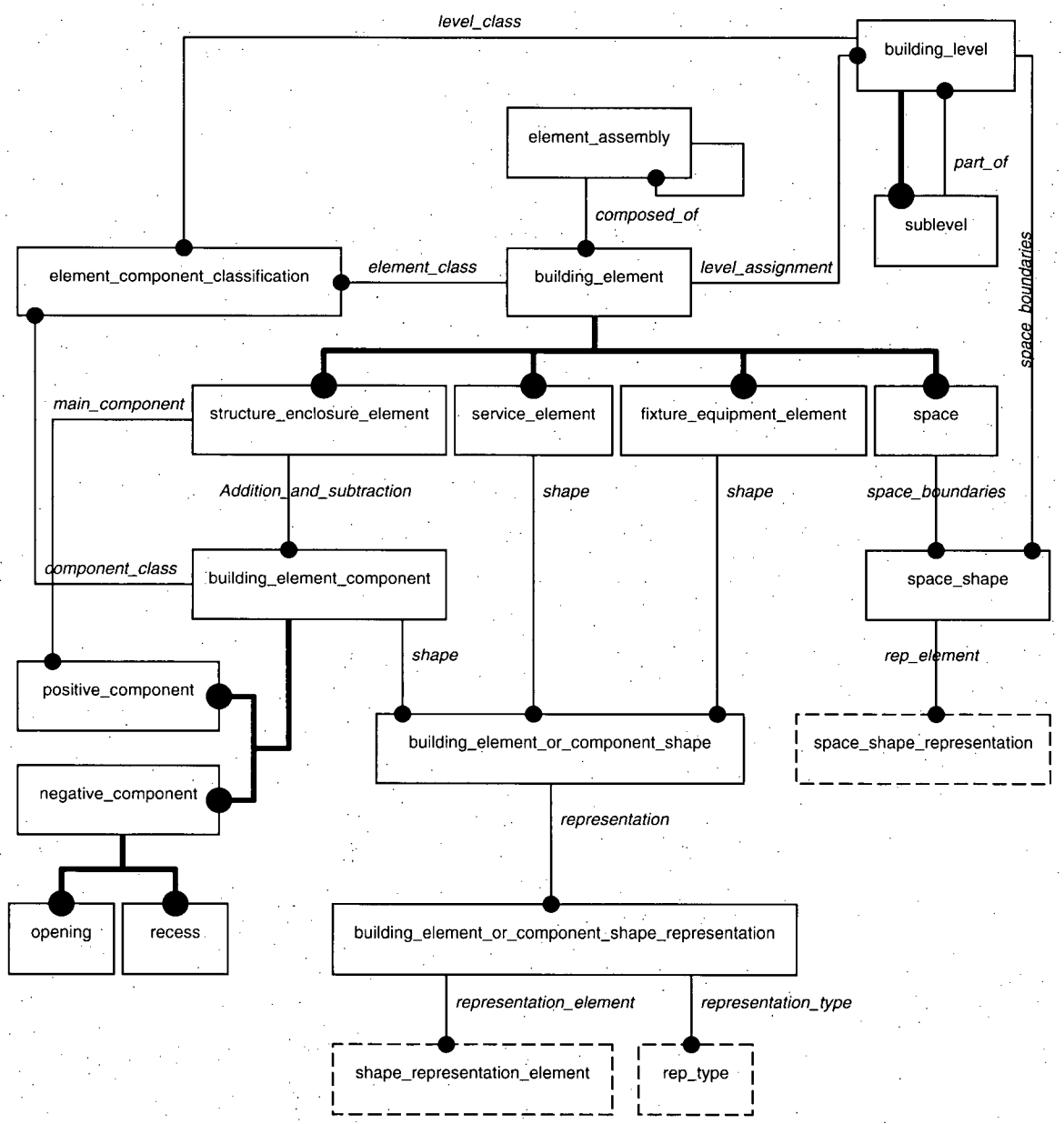


Figure 2–7: A Partial, Simplified Model of Product View in AP225

2.4.4.5 COMBINE [1995]

COMBINE (COmputer Models for the Building INdustry in Europe) is a product-modeling project, mainly concerned with energy analysis and HVAC systems in the early stages of building design. Its aim was to develop an operational computer-based Integrated Building Design System (IBDS), using STEP technology and a number of existing design tool prototypes shared around a central common data repository, called Integrated Data Model (IDM) [Dubois et al. 1994]. The actual interfaces between the IDM and the design tools were implemented through STEP neutral file exchange. Having no industrial

client, the project was carried out by building engineers at the Technical University of Delft, the Netherlands, and it took five years to complete (1990 to 1995), with approximately 70 person-years of resources [Augenbroe 1994; Augenbroe 1995].

Object-oriented methods and technologies were used in the development of models and implementation of prototype systems, and NIAM and EXPRESS were used as graphical formalism and formal modeling language respectively. The Integrated Data Model (IDM) was developed in the form of an EXPRESS file [Augenbroe and Rombouts 1994; COMBINE 1995]. For the sake of uniformity with other models presented in this dissertation, however, the models of the project were transformed form NIAM to EXPRESS-G diagrams. Figure 2-8 shows a partial, simplified model of the product view with its high-level generic product entities. The IDM includes a total of 310 entities and 210 types.

Overall, COMBINE formalizes the space-enclosure view of the building into a fairly rich information model; i.e., the typical space view of the architect during spatial design and early fabric detailing. It gets into the modeling of space element of a building and its functional, shape, and performance characteristics. In terms of STEP methodology, it could be considered as a composition of an Integrated Resource (IR) and an Application Protocol (AP). It is more than an AP (e.g., AP225, explained earlier), since it has an integrating approach between some design applications. Compared with BCCM and IFC model, it does not cover the full range of views of products, though it gives a richer topological view of space. It does not deal with process views of a building project either. However, it demonstrated the possibility of integrating computer applications using information modeling.

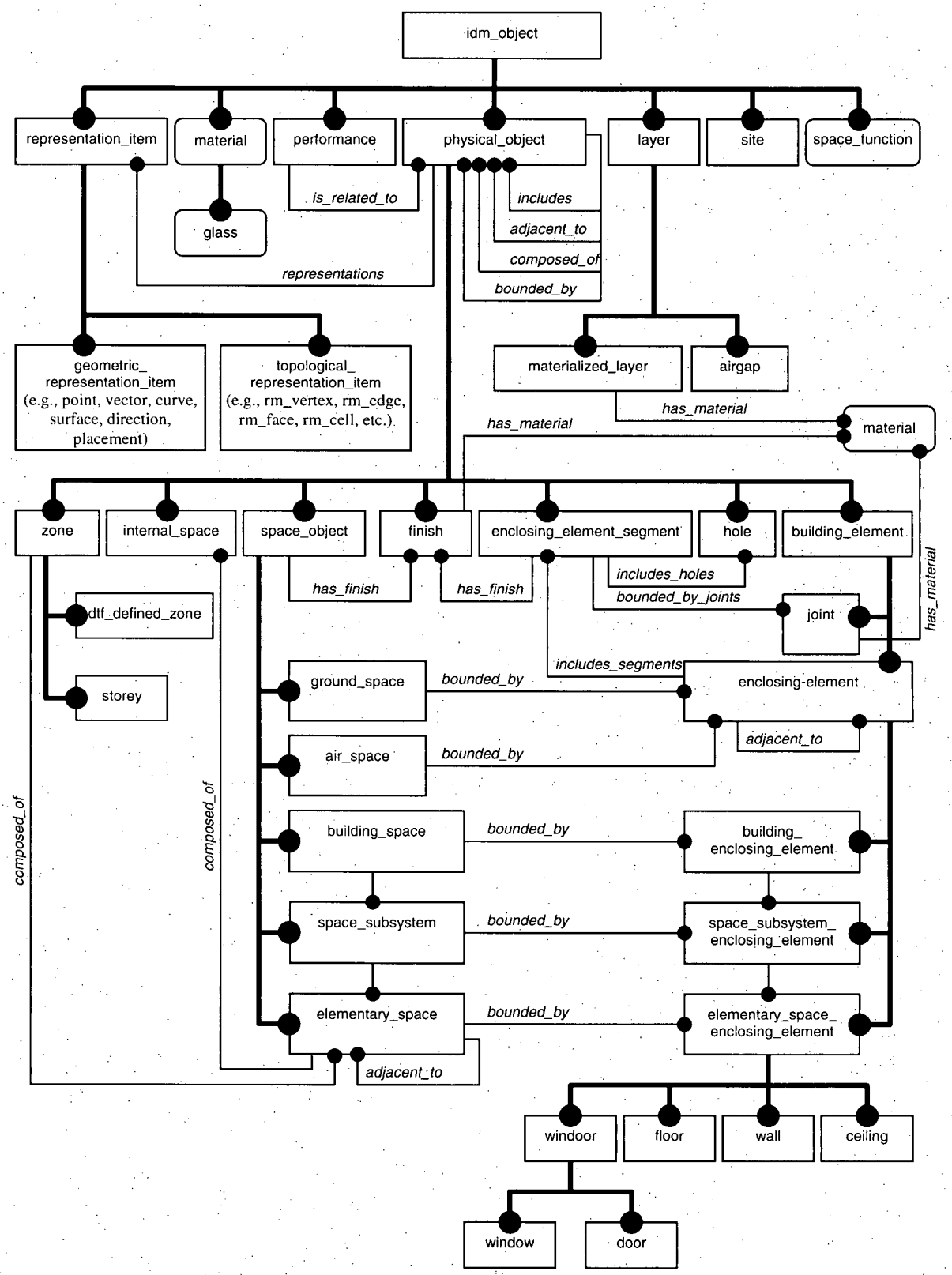


Figure 2–8: A Partial, Simplified Model of Product View in COMBINE

2.4.4.6 CONDOR (ESPRIT 23 105) [Rezgui and Cooper 1998]

The European Esprit CONDOR project (ESPRIT 23 105) has a specific focus on *document management*. It draws a migration path from *document-based* to *model-based* approaches to document management. In the *document-based* approach, documents are treated as "black-boxes" (e.g., dealing with information such as type, name, category, owner, issued/approval date of the document), and the focus is on computer-based support for grouping, storage, retrieval (using referencing), versioning, and approval of documents [e.g., Bjork et al. 1993]. It is well suited to the apparent proliferation of Electronic Document Management (EDM) systems, though it does not deal with the systems' limitations (e.g., document cross-referencing within and between EDM systems and the obligation of the end-users to use the same one system, similar to Electronic Data Interchange systems [Gibson and Bell 1990]). The *model-based* approach resolves such limitations by structuring and representing the *information content* of project documents (e.g., drawings and specifications) in object models [Rezgui and Debras 1995]. The information is then contained in some form of integrated database, which is populated with real project data and is ultimately used for production and authoring of project documents.

The CONDOR project suggests an information exchange between the existing EDM systems through an Application Programming Interface (API). The API is defined using existing services (i.e., document categorization, archiving, retrieval, versioning, and approval) provided by the legacy EDM systems. The CONDOR methodology involves capturing and generalizing the business processes of document management from three sample projects into business process models of document management using $IDEF_0$ diagramming technique. The UML's graphical techniques (i.e., use-case models, sequence diagrams, and class diagrams) are also used to model the functional, behavioral and structural aspects of the CONDOR system.

2.4.4.7 IAI [2005]

Another on-going standardization effort is the activities of IAI (Industry Alliance for Interoperability), which is closely related to the STEP's BCCM development. The IAI is made up of over 600 international organizations that have targeted a definition of building industry object classes called the Industry Foundation Classes (IFC's). The IFC's are considered to be a library of commonly defined objects, representing project data, and supporting the whole life cycle of building development [IAI 2005]. The largest focus of this project has been towards modeling design areas, and a small portion of the model defines those objects supporting construction management functions. A general description and characteristics of the IAI project was presented in section 1.2.5 of this dissertation. The following provides more information on the project and its IFC model.

2.4.4.7.1 A Brief Background

Seeking "*interoperability*" between different AEC/FM software applications, the IAI (International Alliance for Interoperability) project was formally established as a not-for-profit international consortium in September 1995. At present, the IAI has 9 Chapters worldwide, each representing an international region, with membership of over 600 companies from over 20 countries. Contributors to the project range from domain experts to technical experts from different types of organization in the AEC/FM industry; i.e., architects, engineers, contractors, building owners and managers, building product manufacturers, software vendors, information providers, government agencies, research labs, and universities [IAI 2005].

The IAI has defined its Mission as "*to define, promote and publish a specification for sharing data throughout the project life cycle, globally, across disciplines and across technical applications*" [IFC R2.0, 1999b]. This would require the establishment of some standard communication mechanism; i.e., defining a common language and dictionary for the software applications so that they could talk together (i.e., read and write). The IFC object model provides this mechanism.

2.4.4.7.2 IFC Model Architecture

The IFC Model architecture consists of four conceptual layers: *Resource, Core, Interoperability*, and *Domain/Application* layers, listed from down to top (Figure 2-9). Each *layer* consists of one or more "*model modules*", each physically specified in the form of a schema. A strict referencing hierarchy, originated from so-called "*ladder/gravity principle*", has been enforced between these layers, and thus between their included model components (i.e., schemas). Following this principle, the following two basic rules are generally in effect for relationships within and between the modules:

1) A class may reference another class at the same or lower level of this hierarchical layering but may not reference a class from a higher level. In order to support the self-containment goal for each resource, inter-resource referencing is not recommended; however, references of this type (but independent of other layers) exist in the model.
2) With the exception of the Core layer, a class of a higher level can be a subtype of another class in the same or lower level. I.e., the Core classes can be subtypes of (or use) other classes in Core layer but cannot be subtypes of classes in Resources.

Such a layering hierarchy is intended to provide for modularity, reusability, and maintainability of the model. This hierarchy of the layers and the constituent modules of each layer in the Release 2.x of IFC Object Model [IFC R2x, 2000a], are depicted in Figure 2-9. A brief description of the four layers is provided in the following.

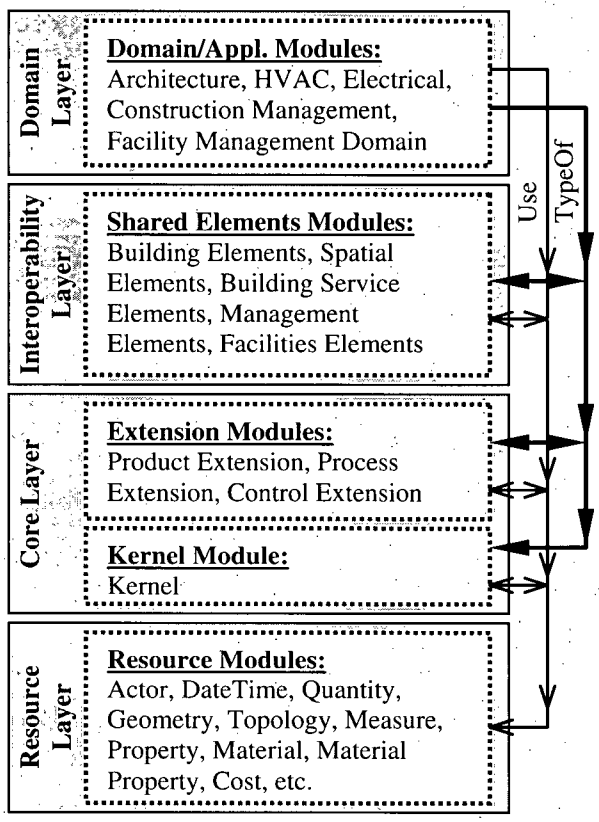


Figure 2–9: IFC Model Layers and Modules/Schemas

1) **Resource Layer**: The concepts commonly used and referenced by classes of components of higher layers are defined in Resources. There are 20 schemas defined in this layer.

2) **Core Layer**: This layer includes the Kernel Model and its subsets of Core Extensions. There are 4 schemas defined in this layer.

3) **Interoperability Layer**: Applying a "plug-in" concept, the Interoperability Layer includes those modules that define the specialized interfaces (i.e., concepts or objects) that are commonly used by two or more domain/application models. These modules provide for interoperability between application models. There are 5 schemas defined in this layer.

4) **Domain/Application Layer**: This layer includes a number of modules, each separately modeling more detailed concepts, which are specific to an AEC/FM domain process or to a type of application. Each module may use or reference any class defined in the Core and Resource layers. External Domain Models, which are not fully harmonized with IFC core model, will be considered at this Domain Layer too. In order to use the IFC model framework, such models must provide appropriate Adapter Plug definitions (as explained above). There are 5 schemas defined in this layer.

2.4.4.7.3 *IFC Specification Development Process*

The formal process of IFC specification development may be summarized into four principle stages: *propose project*, *specify requirements* (for individual projects), *integrate models* (of projects into the IFC object model models), and *implement model*. The requirement specification for each project involves six major activities: *break down processes*, *define terms and conditions*, *add usage scenarios*, *analyze model requirements* (using $IDEF_0$ technique), *develop project object model*, and *develop test criteria*. The process is extensively elaborated in IFC R2.0 [1999c].

A *usage scenario* is a textual description that sets down the information requirement based on a real world situation. It provides the basic input to the identification of classes, attributes, relationships, properties and interfaces. The project object model defines classes, attributes, relationships, properties and software interfaces. The model may be formally represented in a structured spreadsheet and/or in EXPRESS-G [ISO 1991]. Word processing and spreadsheet applications are the main tools used for managing the modeling process workflows, and links between process and object models are not made explicit (e.g., no mapping of the objects defined in the object model and the information input and output represented in $IDEF_0$ diagrams).

2.4.4.7.4 *The IFC Object Model*

The release R2x of the IFC object model includes 34 interrelated schemas (or modules), which are specified in the EXPRESS modeling language and graphically presented in EXPRESS-G modeling notations. The model comprises of 370 Entities, 23 Select Type, 117 Enumeration, and 89 Defined Types. Figure 2-10 shows a subset of the major objects defined in the Kernel of the model [IFC R2x, 2000a]. The entities shown in rounded boxes are those whose relationships and attributes are not fully presented in the figure. Also, the data types shown in a solid-line, rounded box contained in a dashed-line, rectangular box are those that are referenced from other model modules (or schemas) than the Kernel. Appendix A may be consulted for a full explanation of the notations. The abstract class of *IfcRoot*, at the highest level, provides for the fundamental concepts of identification, ownership and change information, naming and description. Any entity is rooted to this class.

In the first-level specialization hierarchy, three fundamental abstract entity types (i.e., no instantiation) are considered as subtypes of the *IfcRoot*: *IfcObject*, *IfcRelationship*, and *IfcPropertyDefinition*. *IfcObject* represents generalization of any semantically treated thing (or item) within the IFC model. It stands for all physically tangible built items (e.g., walls, windows, roofs, etc.), physically existing items (though intangible, e.g., spaces), conceptual items (e.g., grids, cost items, etc.), processes (e.g., tasks), resources (e.g., labor and equipment), and actors (i.e., persons and organizations).

IfcPropertyDefinition represents generalization of all characteristics that may be assigned to objects; i.e., the specific information of an object type, as opposed to the occurrence information of the actual object (generalized by *IfcObject*) in the project context (section 4.2.2.2). Such information may be assigned to objects through subtype of *IfcRelationship* (more specifically through subtypes of *IfcRelDefines*); not shown in the figure. *IfcRelationship* represents generalization of all relationships among objects that are treaded as *objectified relationships* (i.e., being classes on their own right) in the IFC model. This allows relationship-specific properties of objects (i.e., those originating as a consequence of relationships between objects) to be separated from the object attributes and kept at the relationship object. This may be considered as one of the most powerful aspects of the IFC model.

Each of these three abstract classes is further specialized into other classes. The subtypes of *IfcObject* within an AEC/FM project are *IfcProject*, *IfcProduct*, *IfcResource*, *IfcProcess*, *IfcControl*, *IfcActor*, and *IfcGroup*. Referring to the IFC R2x object model documentation [IFC R2x, 2000a; 2000b], a brief description of each of these classes is provided in the following.

- *IfcProject*: a concrete class that represents an undertaking of some engineering activities leading towards a product. It establishes the context for representation of an AEC/FM project's concepts.
- *IfcProduct*: an abstract class that captures the concept of physical items that are incorporated into an AEC/FM project either directly as supplied or through construction/assembly of other products. It includes tangible objects designated for permanent use (e.g., a door) or temporary use (e.g., a formwork), and objects created indirectly by other products and physically existing (e.g., spaces).
- *IfcResource*: an abstract class that captures the concept of "use of things" within an AEC/FM project. It can be seen as the idea of things that are used within a process or by a product. It contains the information needed to represent the costs, schedule, and other impacts from the use of a thing in a process. This class is not intended to be used to model the general properties of the things themselves; which are normally represented by subtypes of *IfcProduct*. Such information is suggested to be captured through the *IfcRelAssignsToResource*.
- *IfcProcess*: an abstract class that captures the concept of activities undertaken within an AEC/FM project and handles the idea of work being carried out over a period of time. Processes are placed in sequence, with the possibility of overlapping in time. All processes may incorporate a measure of productivity for the process.
- *IfcControl*: an abstract class that captures the concept of a thing that controls or constrains other objects (i.e., products or processes) within an AEC/FM project. Examples include specification, regulation, constraint, and other (product/process) requirements that must be fulfilled.

- *IfcActor*: a concrete class that captures the concept of people and organizations that are active within an AEC/FM project. In fact, it makes use of the resource part of the IFC object model to encompass information about individual agents (e.g., a person or an organization) and their roles (e.g., a designer).
- *IfcGroup*: a concrete class that captures the concept of an arbitrary collection of objects so that they can be treated as a single object. It is a general mechanism for grouping and does not define a common behavior for its members. Examples of some of the subtypes of this class that are defined in other IFC model layers (i.e., outside Kernel layer) are *IfcInventory*, *IfcAsset*, *IfcSystem*, and *IfcZone*.

IfcPropertyDefinition is considered to reflect either an "*object type*" (e.g., door type) or a "*partial type*" (i.e., property sets) within the context of an AEC/FM project. Thus, it is specialized into *IfcTypeObject* and *IfcPropertySetDefinition* respectively.

Finally, *IfcRelationship* is specialized into five subtypes: *IfcRelConnects*, *IfcRelAssigns*, *IfcRelDefines*, *IfcRelDecomposes*, and *IfcRelAssociates*. These subtypes, which are further specialized to establish particular semantic meanings of the relationships (not fully shown in the figure), are briefly explained in the following (based on IFC R2x object model documentation [IFC R2x, 2000a; 2000b]).

- *IfcRelConnects*: an abstract class that represents generalization of the concept of connectivity between objects (e.g., processes or products) under some criteria. Its subtypes define the applicable object types for the connectivity relationship and the semantics of the particular connectivity. For example, *IfcRelSequence* handles the concatenation of processes over time.
- *IfcRelAssigns*: an abstract class representing a generalization of "*link*" relationships among instances of objects (i.e., *occurrence-to-occurrence* relationship). A link means the specific association through which one object (the client, denoted as the "relating object") applies the services of other objects (the suppliers, denoted as the "related objects"), or through which one object may navigate to other objects. This abstract relationship, which is bi-directional and does not imply any dependency relationship among the participating objects, is further specialized into six concrete subtypes: *IfcRelAssignsToProduct*, *IfcRelAssignsToProcess*, *IfcRelAssignsToResource*, *IfcRelAssignsToControl*, *IfcRelAssignsToGroup*, *IfcRelAssignsToActor*. Considering some constraints defined by these subtypes, the subtypes establish links between objects that supply information to the specific type of object. Using these objectified relationship subtypes, instances of different types of objects may be assigned to an instance of a specific type of object. Examples include assignment of an elevator shaft (i.e., as subtype of *IfcProduct*) to a number of building stories (i.e., a subtype of *IfcProduct*), and assignment of a number of construction materials (as subtype of *IfcObject*) to a wall (as subtype of *IfcProduct*) using the *IfcRelAssignsToProduct* objectified relationship.

- *IfcRelDefines*: an abstract class that, through its subtypes (not shown in the figure), uses a type definition (represented by *IfcTypeObject*) or property set definition (represented by *IfcPropertySetDefinition*) to define the properties of an object instance. It is a *type-to-occurrence* relationship and implies dependencies (i.e., occurrence objects depend on specific properties).
- *IfcRelDecomposes*: an abstract class that defines the general concept of elements being composed or decomposed. The decomposition relationship denotes a whole/part hierarchy with the ability to navigate from the whole (the composition) to the parts and vice versa; so, it implies a bilateral dependency between the whole and its parts. It has two subtypes. *IfcRelNests* handles the nesting of all subtypes of *IfcObject*, provided that the whole and the part are of the same object type (e.g., activity breakdown structure). *IfcRelAggregates*, on the other hand, provides for aggregation of objects of different kinds (e.g., a house as a composition of walls, roofs, slabs, doors, etc.).
- *IfcRelAssociates*: a concrete class that refers to external sources of information (e.g., a classification, library, or document) and associates it to objects (i.e., subtypes of *IfcObject*) or property definitions.

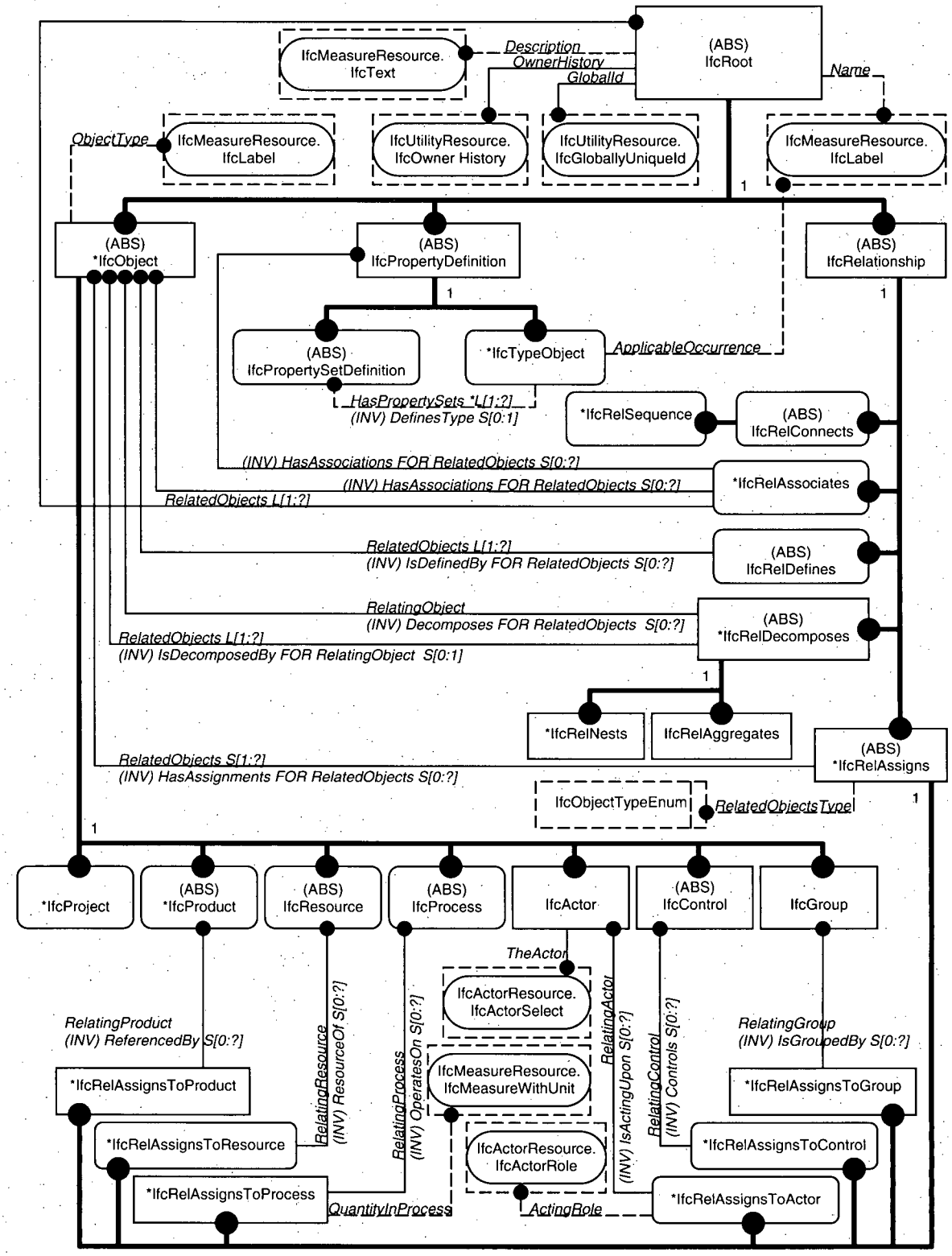


Figure 2–10: A Partial Model of the Kernel Module of the IFC Object Model

2.4.5 Summary of AEC/FM Information and Process Modeling

This section elaborated on information modeling in the AEC/FM industry. It explained different types of information models and a general overview of a number of major AEC/FM information modeling projects aimed at CIC. Detailed technical discussions of the models, however, are covered in the next chapters, as the subject matter requires. Overall the presented models are core (reference) models. They are high-level information models that may be used as a unifying mechanism by a number of application models to exchange AEC/FM information. Some of the models can support actual implementations, while they still act as a core for their other related parts. The IFC model is a layered information model that defines a hierarchy of specifications of project information from resource to core and application level objects. The model may be considered as the state-of-the-art model that inherits from its predecessor models, which targeted formalization and standardization of information through the provision of type models. The key immediate impact of the IFC's development has been the incorporation of the IFC's in various AEC/FM prototype systems as well as commercial software systems [IAI 2005]. The next section focuses on AEC/FM process modeling.

2.5 AEC/FM Process Modeling (PM Functions)

A review of current AEC/FM literature and software systems and models shows that the extent of coverage of PM functions and their integration has been minimal; *planning* (especially scheduling) and *controlling* (of project time, cost, quality) functions have generally been the main focus [CICE 1983b; Paulson 1995; Eastman 1999]. However, there is a wealth of references that their review highlights the extent to which such coverage can be improved. This section focuses on AEC/FM process modeling in the context of CIC. It presents an overview of a number of AEC/FM process models and discusses some of their related issues. It also generalizes the issues and suggests a set of guidelines for resolving them. Finally, it introduces the method and techniques used by the research to address the issues (i.e., to model AEC/FM processes). The materials presented in this section are intended to provide the necessary background for communication of the research deliverables described in chapters 6 and 7.

In this dissertation, the term "*function*" is used to refer to a *system* or *process*, rather than a functional *organization*, such as a *department* (e.g., transportation department). Thus, PM functions are the processes exerted in the course of managing a project rather than physical organizations performing tasks. The term "*process*" is used in this chapter to refer to the sum of the application of project resources and PM techniques for provision of the target product of the project. At an operational level of a project, however, a process may be referred to as an *activity*. The term "*function*" is also used to refer to a *process* at any level of detail.

2.5.1 A Review of Selected AEC/FM Process Models

Focusing primarily on classification of PM functions, this section describes and discusses some of the major AEC/FM process models and their related issues. The discussions are preceded by definition of project and PM. For the purpose of uniformity of presentations and in order to facilitate the overview and discussion of process structures of the models, a tree format textual representation is devised. The hierarchical numbering of the processes provides for better readability and referencing of the models.

2.5.1.1 Project and Project Management

Grouping *works* performed by organizations into *operations* (ongoing and repetitive) and *projects* (temporary and non-repetitive), the Project Management Institute (PMI) [1996, p. 1.3] defines a *project* in terms of its distinctive characteristics relative to other types of operations: "A project is a temporary endeavor undertaken to create a unique product or service." It further gives a definition of *project management* (PM): "the application of knowledge, skill, tools, and techniques to project activities in order to meet or exceed stakeholder needs and expectations from a project." Meeting or exceeding the needs and expectations would involve balancing competing demands among project objectives, stakeholders' interests, and requirements. The IAI project [IAI 2005], however, uses the term PM in the context of building projects, from inception to the operation and maintenance of the facility.

Reflecting a North American perspective, *construction management* (CM) is defined as "the composite of all modern project management methodologies having as their objective the control of time, cost, and quality in the design and construction of a new facility" [Kavanagh et al. 1978, p. 2]. This definition does not consider the management of the constructed facility (referred to as *facility management*, FM) within the scope of CM. Moreover, it emphasizes three functions: *time control*, *cost control*, and *quality control* during design and construction.

In this dissertation, the term *project* is used to refer to a temporary endeavor undertaken to build a unique constructed facility. Also, CM refers to the activities that take place in a facility project delivery, from conceptualization to construction and hand-over of the product to he owner. Moreover, PM is broadly used in the context of construction projects to refer to the management of any portion of the development life cycle of any type of facility.

2.5.1.2 The IBPM Approach [Sanvido 1990]

The IBPM (Integrated Building Process Model) is a product of a CIC project aimed at providing "an open information architecture to support the provision of a facility." It was sought to "accurately represent the *essential functions required to manage, plan, design, construct, operate, and maintain a facility*" [Sanvido 1990, p. 1-1], and this was done using $IDEF_0$ diagramming technique [1981]. Based on

project phases, and at the highest level of the process breakdown structure, the "*Provide Facility*" process is divided into five main processes (Figure 2-11), each of which further broken down into lower-level processes. Each process and it related inputs and outputs are explained in text format.

Figure 2-11 summarizes the process breakdown structure in a textual, hierarchical tree format. As this figure shows, the model provides an extensive breakdown of many AEC/FM processes. Moreover, it includes managerial activities (under A1. *Manage Facility*) as well as technical and operational activities. However, *repetition of processes* can be observed in the model. For example, the processes of *Receive and Inspect Resources* (A155) and *Distribute/Store Resources & Manage Inventory* (A156), which are listed under process *Acquire/Provide Resources for Facility* (A15), are also repeated under *Provide Resources* (A43) as A433 and A434 respectively. Another example is the process *Acquire Services to Provide Facility* (A14) and its sub-processes (A141 to A145) that are all somehow repeated in *Acquire Construction Services* (A41) and its sub-processes (A411 to A415).

A0. Provide Facility

- **A1. Manage Facility**
  - **A11.** Establish Management Team
    - **A111.** Determine Internal Capabilities/Operations
    - **A112.** Develop Preliminary Facility Work Scope
    - **A113.** Structure & Staff the Management Team
  - **A12.** Develop Work Scope and Needs
    - **A121.** Understand Owner Needs
    - **A122.** Define Facility Work Scope & Needs[1] (Planning function supplies input to this phase.)
    - **A123.** DVLP. Strategy for Resource & Service Acquisition
  - **A13.** Plan/Control Facility
    - **A131.** Understand Work Scope & Performance Criteria
    - **A132.** Develop Facility Management Plan
    - **A133.** Administer Contracts/PO's Agreements
    - **A134.** Implement & Supervise Work (Contracts)
    - **A135.** Monitor Facility & progress
    - **A136.** Analyze Performance of Facility
  - **A14.** Acquire Services to Provide Facility
    - **A141.** Identify Services Needed
    - **A142.** Identify Sources of Services
    - **A143.** Prepare Invitation to Bid and Submit Proposal
    - **A144.** Review Proposals & Select Agent/Contractor
    - **A145.** Execute Contracts and Agreements
  - **A15.** Acquire/Provide Resources for Facility
    - **A151.** Identify Resource Needs[3] (Resources include Funding, Time, Material, Manpower, etc.)
    - **A152.** Identify Sources of Resources
    - **A153.** Prepare Purchase Requisitions & Submit Proposals
    - **A154.** Review Proposals, Select Vendor and Execute P.O.
    - **A155.** Receive and Inspect Resources
    - **A156.** Distribute/Store Resources & Manage Inventory
- **A2. Plan Facility**
  - **A21.** Assign Planning Team
  - **A22.** Study/Define Needs
    - **A221.** Study User Requests
    - **A222.** Evaluate Existing Facilities
    - **A223.** Determine Needs
    - **A224.** Generate Alternatives
  - **A23.** Study Feasibility of Alternatives
    - **A231.** Study Economic Feasibility
    - **A232.** Study Technical Feasibility
    - **A233.** Study Environmental Feasibility
    - **A234.** Communicate Results/Decisions
  - **A24.** Develop Program
    - **A241.** Gather Information
    - **A242.** Define Scope
    - **A243.** Develop Design Criteria
    - **A244.** Develop Site Criteria
    - **A245.** Communicate Program

Figure 2–11: The IBPM's Process Breakdown Streucture (p. 1/4)

- **A25.** Select & Acquire Site
  - **A251.** Identify Candidate Sites
  - **A252.** Evaluate & Select Site
  - **A253.** Acquire Site
  - **A254.** Investigate Site (for Design)
- **A26.** Develop Project Execution Plan (PEP)
  - **A261.** Identify Required Services
  - **A262.** Study Construction Market Conditions
  - **A263.** Develop Project Plan
  - **A264.** Develop Contracting Plan
  - **A265.** Communicate Project Execution Plan

A3. Design Facility

- **A31.** Understand Functional Requirements
  - **A311.** Assimilate and Analyze Information
  - **A312.** Establish Project Objectives
  - **A313.** Establish Design Parameters
- **A32.** Explore Concepts
  - **A321.** Perform Preliminary Studies
  - **A322.** Prepare & Develop Concepts
  - **A323.** Coordinate Concepts
  - **A324.** Evaluate & Select Concepts
- **A33.** Develop System Schematics
  - **A331.** Develop Standard Systems Schemes
  - **A332.** Coordinate to Find Compatibilities
  - **A333.** Develop Integrated Schematics
  - **A334.** Evaluate & Select Schematics
- **A34.** Develop Design
  - **A341.** Perform Systems Development & Layouts
  - **A342.** Perform Studies & Reviews
  - **A343.** Develop Outline Specs. Drawings, Schedules
  - **A344.** Acquire Design Approval
- **A35.** Communicate Design to Others
  - **A351.** Develop Post-Design Drawings
  - **A352.** Develop Post-Design Specifications
  - **A353.** Perform Document Reviews
  - **A354.** Deliver Post-Design Docs & Acquire Approval
- **A36.** Maintain Design Information and Models
  - **A361.** Collect Data
  - **A362.** Store Data
  - **A363.** Retrieve Data
  - **A364.** Update Information
  - **A365.** Transit Information

A4. Construct Facility

- **A41.** Acquire Construction Services
  - **A411.** Identify Qualified Parties
  - **A412.** Provide Work Scope Information
  - **A413.** Prepare & Submit Proposals
  - **A414.** Review Proposals & Select Constructor
  - **A415.** Execute Contracts/Agreements

Figure 2-11: The IBPM's Process Breakdown Streucture (p. 2/4)

- **A42.** Plan & Control the Work
  - **A421.** Develop the Construction Plan
    - **A4211.** Determine the Scope of Work & Coordinate the Planning
    - **A4212.** Select Work Methods
    - **A4213.** Estimate the Work
    - **A4214.** Schedule the Work Activities
    - **A4215.** Analyze the Plan
  - **A422.** Implement the Plan
  - **A423.** Monitor Performance
  - **A424.** Analyze Performance
- **A43.** Provide Resources
  - **A431.** Mobilize
  - **A432.** Acquire Resources
  - **A433.** Receive & Inspect the Resources
  - **A434.** Store the Resources & Manage the Inventory
  - **A435.** Prepare & Maintain the Resources
  - **A436.** Allocate the Resources
- **A44.** Build the Facility
  - **A441.** Plan the Daily Work
  - **A442.** Distribute the Resources
  - **A443.** Do the Physical Work
    - **A4431.** Identify the Location for the Work
    - **A4432.** Set up the Work Area
    - **A4433.** Prepare the Resources
    - **A4434.** Perform the Work
    - **A4435.** Clean-up the Work Area
  - **A444.** Inspect & Approve the Work
  - **A445.** Turn Over the Completed Work

A5. Operate Facility

- **A51.** Manage Operations
  - **A511.** Review Data
  - **A512.** Plan Operations
    - **A5121.** Understanding Operations Plan & User Needs
    - **A5122.** Schedule Operations
    - **A5123.** Determine User Needs
    - **A5124.** Assign Operations Execution Team
  - **A513.** Acquire Operations Services & Resources
- **A52.** Monitor Facility Condition & Systems
  - **A521.** Select Critical Points/Areas to Monitor
  - **A522.** Select Monitoring Mechanism
  - **A523.** Collect Data
  - **A524.** Reduce to Information in Correct Format
- **A53.** Evaluate Conditions & Detect Problems
  - **A531.** Evaluate Information Against Standards
    - **A5311.** Understand User Standards
    - **A5312.** Classify/Sort Information
    - **A5313.** Compare Information with Critical or Expected Conditions
    - **A5314.** Determine if the formation is Within the Tolerances
  - **A532.** Locate & Identify Problems
  - **A533.** Notify Problem Solving Mechanism

Figure 2-11: The IBPM's Process Breakdown Streucture (p. 3/4)

A54. Develop Solutions
- **A541.** Understand the Problem
- **A542.** Determine Necessary Information & Skills
- **A543.** Assemble Necessary Information & Skills
- **A544.** Develop /Design Solutions
- **A545.** Analyze Implications
- **A546.** Present Alternatives

A55. Select Plan of Action
- **A551.** Understand Alternatives & Their Implications
- **A552.** Select Alternative
- **A553.** Commit Services & Resources

A56. Implement Plan
- **A561.** Distribute Resources
- **A562.** Do the Work
- **A563.** Inspect Work

Figure 2-11: The IBPM's Process Breakdown Streucture (p. 4/4)

2.5.1.3 The ICON Approach [1994]

ICON (Information/Integration for CONstruction) was a research project aimed at investigating the feasibility and design of a prototype framework for integrated information systems (i.e., *databases*) to support *design*, *procurement*, and *construction management* functions [ICON 1994, Final Report]. It takes a phase-oriented approach into modeling processes. In activity modeling, as a first step in information modeling, ICON suggests an *activity hierarchy* that "defines the activities carried out during a construction project and is concerned with what is done, not by whom, when, where, or how. Also, the decomposition does not imply a sequential ordering of activities" [ICON 1994, Manual of Models, pp. 4-6]. The intention for production of this activity hierarchy was set as:

1) To show the *context model* for ICON.
2) To observe different *perspectives* of various construction project participants.
3) To be used for production of *object models* for each perspective.

Figure 2-12 shows the activity hierarchy suggested by ICON in a textual, hierarchical tree format. At the highest level of the hierarchy, seven activities are suggested: *managing*, *defining*, *procuring*, *designing*, *constructing*, *commissioning*, and *maintaining*; though three of them (*designing*, *procuring*, and *constructing*) are considered within the scope of ICON and are broken down into more detailed activities. While the model provides an extensive breakdown of many AEC/FM processes (within its scope), the following points can be observed:

1) *Repetition of Processes*: Processes are sometimes repeated with or without the same levels of detail. For example, the process *Appointing Participants* is repeated in two places (A22 and A33). Other examples are *Determining Procurement System* (repeated in A21 and A33), *Checking Ground*

Conditions (repeated in A35121 and A43312), *Checking Ground Levels* (repeated in A35122 and A43313), and *Checking Ground Dimensions* (repeated in A35123 and A43314), *Checking Access* (repeated in A35124 and A43315), *Assess Work Standards* (repeated in A4723131 and A4723231), and *Assess Work Outputs* (repeated in A4723132 and A4723232).

2) *Repetition of Process Concepts*: Repetition of the main concepts of some processes can be observed in the activity hierarchy. Examples include "*evaluation and selection*" (e.g., of suppliers) in the processes of A22, A33, A4721 and A4722, and "*scheduling*" in A4153 and A373.

3) *Scattering of Activities of Processes*: In addition to the repetitions (mentioned above), the related activities of some processes are spread at different levels of the activity hierarchy. An example is tendering-related activities. The notion of "*tendering*" is seen under *Procuring* (e.g., A24 and its subs) and *Constructing/Estimating and Tendering* (e.g., A431, A432, 434, and 435).

4) *Places for Extensions*: Despite the extensive process breakdown, places for extensions exist in the model. For instance, Receiving, Storing, Distributing, Testing, and Inventorying are some of the example process areas that can be added under *Obtaining and Organizing Material* (i.e., A47234).

In a second step, in object modeling, ICON introduces the term "*domain perspective*" to refer to "a small manageable object-oriented model, which expresses the information requirements of a single well defined activity" [ICON 1994, Manual of Models, p. 1]. However, each activity is considered as a perspective of a process; i.e., a one-to-one relationship between an activity and a perspective. The idea of perspective is similar to the "*viewpoint*" considered in the SADT process modeling technique [Marca and McGowan 1993] (Appendix A), but in an object-modeling format.

In ICON, a class may have different definitions in various perspectives. There may be classes that appear in more than one perspective (i.e., activities). These identical classes may have the same name but may represent subtly different concepts, or different aspects of the same concepts. Their descriptions may be very similar, with only the relationships with other classes being different. However, the meaning of the classes may diverge as further attributes and operations are added to the classes. In fact, the sharing of information is defined within separate "*integration perspectives*", which translate information defined in domain perspectives into the integrated object model (called "*canonical model*") [ICON 1994, Manual of Models, pp. 1 and 29].

A0. Construction Project Activity

- **A1. Defining**
- **A2. Procuring**
  - **A21.** Determining Procurement System
    - **A211.** Assess Client Requirements
    - **A212.** Assess Project Constraints
    - **A213.** Assess Market Conditions
    - **A214.** Suggest Procurement System
      - **A2141.** Accept Procurement System
      - **A2142.** Suggest Alternative
    - **A215.** Decide on Type of Contract
    - **A216.** Decide on Form of Contract
  - **A22.** Appointing Participants
    - **A221.** Appoint Product Supplier
    - **A222.** Appoint Service Supplier
    - **A223.** Establish Organization Structure
  - **A23.** Prepare Documentation
    - **A231.** Prepare Pre-Contract Documentation
    - **A232.** Prepare Post-Contract Documentation
    - **A233.** Prepare Tender Documentation
    - **A234.** Prepare Contract Documentation
  - **A24.** Undertake Tendering Procedure
    - **A241.** Select Tendering Method
    - **A242.** Prepare Tender List
    - **A243.** Invite Tenders
    - **A244.** Assess Offers
      - **A2441.** Accept Offer
      - **A2442.** Reject Offer
  - **A25.** Administer Contract
    - **A251.** Arrange Site Meeting
    - **A252.** Check Progress
    - **A253.** Issue Certificate
    - **A254.** Evaluate Work
    - **A255.** Issue Certificate
    - **A256.** Settle Final Account
- **A3. Designing**
  - **A31.** Managing the Design Process
    - **A311.** Scheduling the Design Process
    - **A312.** Costing the Design Process
    - **A313.** Coordinating Drawings
  - **A32.** Determining Clients Requirements
    - **A321.** Establish Functional Requirements
    - **A322.** Agreeing Site
    - **A323.** Establishing Cost Limits
    - **A324.** Agreeing Working Methods
    - **A325.** Agreeing Timetable
    - **A326.** Producing Brief
  - **A33.** Appointing Participants
    - **A331.** Determine Procurement System
    - **A332.** Determine Participants Required
      - **A3321.** Determine Finance Available
      - **A3322.** Determine Size of Project
      - **A3323.** Determine Purpose of Building

Figure 2–12: The ICON's Process Breakdown Structure (p. 1/4)

- **A333.** Appoint Service Supplier
- **A34.** Liaising with External Bodies
  - **A341.** Obtain Outline Planning Permission
  - **A342.** Obtain Detailed Planning Permission
  - **A343.** Obtain Building Regulations Approval
  - **A344.** Obtain All Other Consents
- **A35.** Undertaking Feasibility Studies
  - **A351.** Assembling Site Information
    - **A3511.** Checking Site Services
    - **A3512.** Performing Land Survey
      - **A35121.** Checking Ground Conditions
      - **A35122.** Checking Ground Levels
      - **A35123.** Checking Ground Dimensions
      - **A35124.** Checking Access
    - **A3513.** Checking Town Planning Requirements
    - **A3514.** Performing Engineering Site Investigation
    - **A3515.** Checking Legal Constraints
      - **A35151.** Checking Boundaries
      - **A35152.** Checking Rights of Way
      - **A35153.** Conducting Local Search
    - **A3516.** Determining Existing Information Sources
  - **A352.** Examining Viability
  - **A353.** Producing Feasibility Report
- **A36.** Designing
  - **A361.** Spatial Design
    - **A3611.** Conceptual Design
    - **A3612.** Physical Design
  - **A362.** Structural Design
  - **A363.** Technical Design
  - **A364.** Sketch Design
  - **A365.** Outline Design
  - **A366.** Detailed Design
  - **A367.** Production Information
  - **A368.** Analysis
  - **A369.** Synthesis
  - **A3610.** Evaluation
- **A37.** Producing Production Information
  - **A371.** Producing Building Specification
  - **A372.** Producing Production Drawings
  - **A373.** Producing Production Schedules

A4. Constructing

- **A41.** Construction Planning
  - **A411.** Establish Baseline Program
  - **A412.** Determine Construction Activities
  - **A413.** Perform Conceptual Planning
  - **A414.** Establish Approximate Budget
  - **A415.** Establish Project Plan
    - **A4151.** Estimate Activity Duration
    - **A4152.** Estimate Activity Dependency
    - **A4153.** Schedule Activities
    - **A4154.** Determine Resources Required
    - **A4155.** Check Resource Availability
    - **A4156.** Calculate Costs

Figure 2-12: The ICON's Process Breakdown Structure (p. 2/4)

- **A42.** Cost Control
- **A43.** Estimating And Tendering
  - **A431.** Decision to Tender
    - **A4311.** Assess Project Information
    - **A4312.** Assess Company Situation
    - **A4313.** Assess Market Conditions
    - **A4314.** Make Decision To Tender
  - **A432.** Prepare Tender Plan
  - **A433.** Undertake Project Study
    - **A4331.** Assembling Site Information
      - **A43311.** Checking Site Services
      - **A43312.** Checking Ground Conditions
      - **A43313.** Checking Ground Levels
      - **A43314.** Checking Ground Dimensions
      - **A43315.** Checking Access
    - **A4332.** Check Drawings
    - **A4333.** Confirm Methods of Working
    - **A4334.** Establish Sequence of Events
    - **A4335.** Establish Key Dates
  - **A434.** Preparing Tender
    - **A4341.** Collect Cost Information
      - **A43411.** Establish Labor Costs
      - **A43412.** Establish Material Costs
      - **A43413.** Establish Plant Costs
      - **A43414.** Establish Subcontractor Costs
    - **A4342.** Decide Estimating Method
    - **A4343.** Prepare Estimates
      - **A43431.** Estimate Labor Costs
      - **A43432.** Estimate Material Costs
      - **A43433.** Estimate Plant Costs
      - **A43434.** Estimate Subcontractor Costs
      - **A43435.** Estimate Facilities Costs
    - **A4344.** Adjudicate On Tender
      - **A43441.** Allow Profit
      - **A43442.** Allow Overheads
  - **A435.** Submitting Tender
- **A44.** Bidding
- **A45.** Cash Flow
- **A46.** Site Operations
- **A47.** Managing Construction
  - **A471.** Financial Management
  - **A472.** Resource Management
    - **A4721.** Evaluate Resource Suppliers
      - **A47211.** Evaluate Material Suppliers
      - **A47212.** Evaluate Plant Suppliers
      - **A47213.** Evaluate Facility Suppliers
      - **A47214.** Evaluate Subcontractor
    - **A4722.** Select Resource Supplier
      - **A47221.** Select Plant Supplier
      - **A47222.** Select Material Supplier
      - **A47223.** Select Facility Supplier
      - **A47224.** Select Subcontractor

Figure 2-12: The ICON's Process Breakdown Structure (p. 3/4)

A4723. Obtain Resources
- **A47231.** Obtaining and Organizing Labor
  - **A472311.** Review Project Plan
  - **A472312.** Establish Requirement
  - **A472313.** Supervise Labor
    - **A4723131.** Assess Work Standards
    - **A4723132.** Assess Work Outputs
- **A47232.** Obtain and Organize Subcontractors
  - **A472321.** Review Project Plan
  - **A472322.** Establish Requirement
  - **A472323.** Supervise Subcontractors
    - **A4723231.** Assess Work Standards
    - **A4723232.** Assess Work Outputs
- **A47233.** Obtaining and Organizing Plant
  - **A472331.** Review Project Plan
  - **A472332.** Establish Requirement
  - **A472333.** Order Plant
    - **A4723331.** Establish Delivery Dates
    - **A4723332.** Establish Delivery Arrangements
    - **A4723333.** State Spec and Quality
  - **A472334.** Check Plant
    - **A4723341.** Check Delivery Dates
    - **A4723342.** Check Performance
  - **A472335.** Organize Plant Maintenance
- **A47234.** Obtaining and Organizing Material
  - **A472341.** Review Project Plan
  - **A472342.** Establish Requirement
  - **A472343.** Order Material
    - **A4723431.** Establish Delivery Dates
    - **A4723432.** Establish Delivery Arrangements
    - **A4723433.** State Spec and Quality
  - **A472344.** Check Material
    - **A4723441.** Check Delivery Dates
    - **A4723442.** Check Quality
    - **A4723443.** Check Quantity
- **A47235.** Obtaining and Organizing Facility
  - **A472351.** Review Project Plan
  - **A472352.** Establish Requirement
  - **A472353.** Obtain Facilities
    - **A4723531.** Arrange Access
    - **A4723532.** Arrange Site Transport
    - **A4723533.** Arrange Site Accommodation
    - **A4723534.** Arrange Site Services
    - **A4723535.** Arrange Site Temporary Works

A473. Site Management

A5. Commissioning

A6. Maintaining

Figure 2-12: The ICON's Process Breakdown Structure (p. 4/4)

2.5.1.4 The MoPo Project [Karstila and Bjork 1999]

The MoPo project (Models for the construction process) was an international collaborative research project in the domain of construction process modeling during 1999 to 2001 [Karstila and Bork 1999]. It aimed at developing methods, IT tools, and reusable generic construction process models useful for *construction process analysis and planning*. The dimensions and results of this project have been summarized by Karhu et al. [2002]. The project focuses on both *representation* and *presentation* (i.e., visualization) of processes. It uses the $IDEF_0$ technique for representation of generic processes.

2.5.1.5 The GDCPP Approach [2001]

The GDCPP (Generic Design and Construction Process Protocol) project was a high-level process modeling research project in the UK that aimed at identifying the IT requirements needed to support the process of design and construction of projects. A particular approach from the manufacturing industry was adopted as a consistent means of formulating project development process [GDCPP 2001; Kagioglou et al. 1998a and 1998b; Wix and Liebich 2001], following this motto:

> "*We believe that if every body involved in a project can work to an agreed set of processes and procedures, then we will not only be more efficient, but we will be in a much better position to meet the client's business needs*" [GDCPP 2001].

Aiming at a generic framework for the development process of any construction project, the GDCPP project suggests four main *stages* for development life cycle of a construction project: *Pre-project*, *Pre-construction*, *Construction*, and *Post-construction*. Then, it divides these stages into ten *phases*, numbered from zero to nine: 0) demonstrating the need, 1) conception of need, 2) outline feasibility, 3) substantive feasibility study and outline financial authority, 4) outline conceptual design, 5) full conceptual design, 6) coordinated design, procurement and full outline financial authority, 7) production information, 8) construction, and 9) operation and maintenance.

The term "*gate*" is also introduced to represent a review process, which has to be undertaken after each stage or phase before proceeding to the next stage or phase. The start of the next stage or phase is subjected to the outcome of a gate: "*Go*" (i.e., continue, unconditional, to next phase), "*No Go*" (i.e., project scrapped), "*Conditional Go*" (i.e., pending further work; continue, subject to certain actions being taken), or "*Conditional No Go*" (i.e., pending further work; corrections and quality improvement and a further review are needed). The gate is suggested to be either "*hard*" (i.e., full completion of all activities within the phase or stage before passing the gate) or "*soft*" (enabling concurrency; i.e., non-completion items are identified and carried over until the next gate is reached).

The GDCPP also suggests nine "*activity zones*". Activity zones are oriented towards management functions (e.g., resource management, facility management, etc.) rather than disciplines (e.g., architectural design, structural design, etc.), for instance. They are defined as a "structured set of sub-processes involving tasks which guide and support work towards a common objective (for example, to create an appropriate design solution)" [Kagioglou et al. 1998a, p. 15]. They are considered as overlapping and interacting with each other (e.g., a product model input from *Design Management* to *Production Management* and *Facilities Management*). An activity zone's tasks are a structured set of tasks and processes task (e.g., supply, production, etc.) and cover the whole spectrum of functional skills (e.g., architects, engineers, constructors, etc.) needed for a construction project. In total, nine activity zones are suggested. Table 2-1 provides a listing of the activity zones and their responsibilities.

The GDCPP project suggests an "IT/process map". The process map includes the stages, phases, activity zones, and the required IT support for different phases of a project.

Each phase includes a number of sub-processes, which are organized into a three-level hierarchical format [Fleming et al. 2000; GDCPP 2001]. Each sub-process belongs to a specific activity zone (i.e., the process owner). Logical dependencies are also considered between related sub-processes. However, sub-processes are very generic. Figure 2-13 illustrates an example of such sub-processes for Phase 0 (i.e., Demonstrating the Need).

In general, the GDCPP model may well fit under *workflow management* models in AEC/FM projects domain. The model is an attempt in formalization of the knowledge (i.e., processes and procedures) involved in construction projects in a very generic format for the purpose of process management. The model is mainly a general description of a generic construction project in terms of the involved phases and processes, and identification of the deliverables and interactions of the processes. An Activity Zone is assigned to each process as "process owner" (i.e., responsible for the process). Activity Zones are looked at as *organizational bodies*. An example of such a treatment is a suggestion made to include some of them (i.e., *Project Management*, *Process Management*, and *Change Management*) along with chairperson, client, and activity zone leaders as members of the "Phase Review Board" [Kagioglou et al. 1998a, p. 26]. In essence, the model was intended to be used as a generic process framework for the development of more refined processes, which would suit the organization's needs and working practices.

The modeling technique used in the representation of process structure in the GDCPP model is very similar to the $IDEF_0$ diagramming technique. The breakdown of processes into a process hierarchy, and consideration of input/output and sequencing links between processes are some of the points of similarity. However, the linkage of processes to Activity Zones and assigning process owners are new

concepts introduced in the model. These concepts can help the model is applied as a process protocol for managing workflows in AEC/FM projects processes.

The GDCPP project may be generally categorized as a workflow-management modeling project, which had an attempt in formalization of processes at a very high level of abstraction. It had a special emphasis on coordination among processes, change management, and performance measurement.

Table 2–1: Activity Zones and their Responsibilities in GDCPP

Activity Zone	Responsibility
• Development management	• Creating and maintaining business focus throughout the project, which satisfies both relevant organizational and stakeholder objectives and constraints.
• Project management	• Implementing the project effectively and efficiently to agreed performance measures, in close collaboration with process management.
• Resources management	• Planning, co-ordination, procurement and monitoring of all financial, human and material resources including establishing the overall budget.
• Design management	• The design process which translates the business case and project brief into an appropriate product definition. It guides and integrates all design input from other activity zones.
• Production management	• Ensuring the optimal solution for the buildability of the design, the construction logistics and organization for delivery of the product.
• Facilities management	• Ensuring the cost efficient management of assets and the creation of an environment that strongly supports the primary objectives of the building owner and/or user.
• Health and Safety, Statutory and Legal management	• Identifying, considering and managing all regulatory, statutory and environmental aspects of the project.
• Process management	• Developing and making operational the Process Protocol together with planning and monitoring of each phase of the project.
• Change management	• Communicating project changes effectively to all relevant activity zones and the development and operation of the legacy archive.

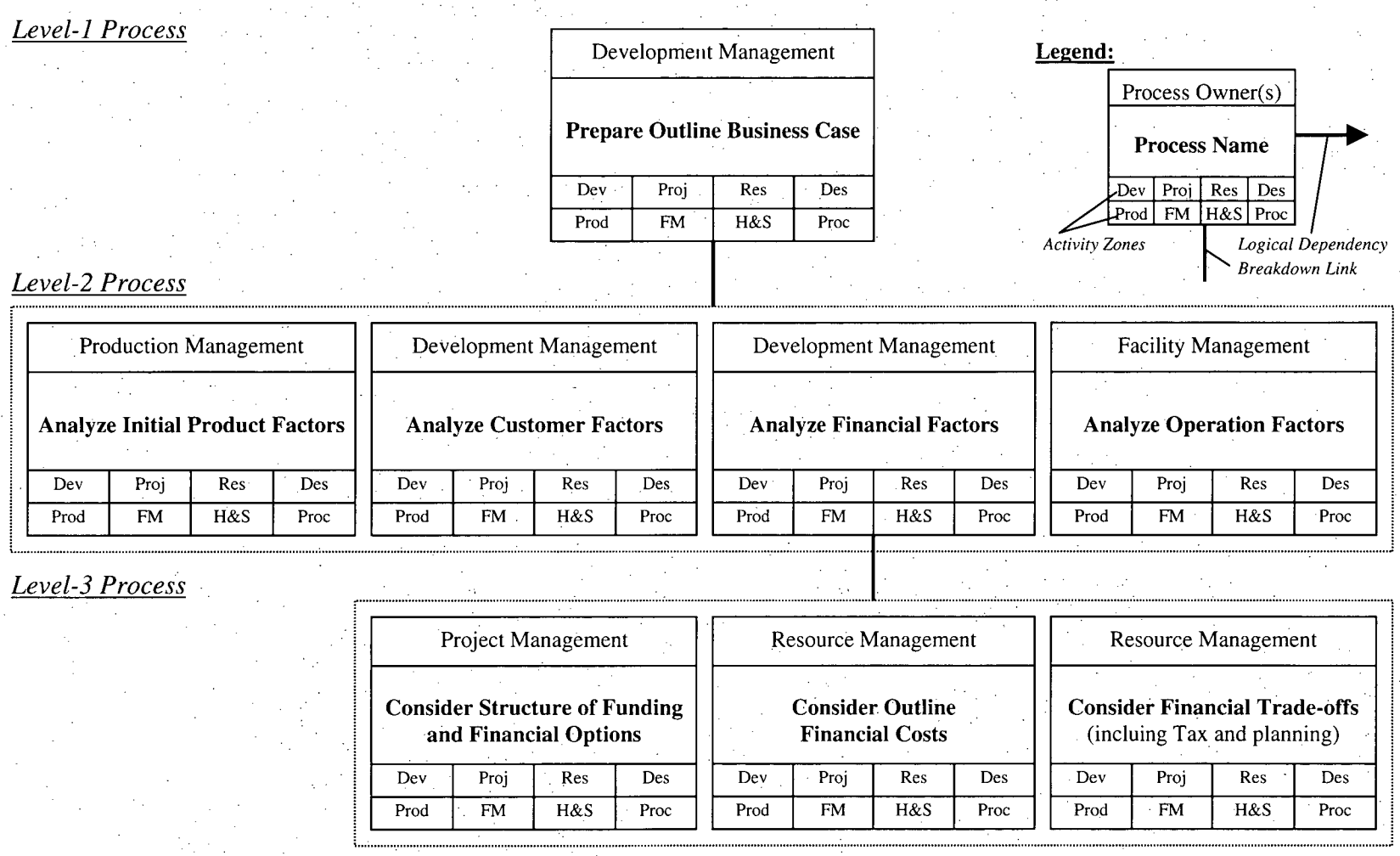


Figure 2–13: An Example of Hierarchy of Sub-Processes for the Phase 0 in GDCPP Model

Note: The figure is based on GDCPP [2001], and it does not show the logical dependencies between processes. A "Process Owner" is, in fact, an "Activity Zone" responsible for the related Process.

2.5.1.6 The PMBOK Approach [PMI 1996]

In "the Project Management Body of Knowledge" (PMBOK), the Project Management Institute (PMI) divides project processes into two categories: "*PM processes*" (i.e., those concerned with describing and organizing the "work" of the project); and "*product-oriented processes*" (i.e., those concerned with specifying and creating the project product) [PMI 1996, p. 27]. The PMBOK suggests that the processes in the first category are generically applicable to most projects, whereas the latter ones are typically defined by the project life cycle vary by application area.

Concentrating on PM processes, the PMBOK organizes them into five "process groups": *initiating* (i.e., project need recognition and commitment), *planning* (devising and maintaining project plan), *executing* (i.e., coordinating resources to carry out the plan), *controlling* (ensuring project objectives are met), and *closing* (i.e., formalization of acceptance of the product and ending of the project). Figure 2-14 shows a UML activity diagram of the process groups and the flow of documents and documentable items between them (represented by arrows). Each phase of a project (e.g., design, implementation, etc.) is suggested to include these groups of processes.

The PMBOK also introduces nine knowledge areas, each consisting a number of processes, each of which belonging to one of the process groups. Being more project-objective oriented, the knowledge areas are: 1) *Project Integration Management, 2) Project Scope Management, 3) Project Time Management, 4) Project Cost Management, 5) Project Quality Management, 6) Project Human Resource Management, 7) Project Communication Management, 8) Project Risk Management,* and 9) *Project Procurement Management.*

The knowledge areas and their processes are very conceptual and generic, and they are dealt with at the project level. They are intended to be independent of any context or application area (i.e., not referring to any specific industry, project type, phase or product). In analyzing processes, their inputs and outputs (in the form of documents and documentable items) and mechanisms (i.e., tools and techniques applied to create output from input) have been of concern in the PMBOK [PMI 1996].

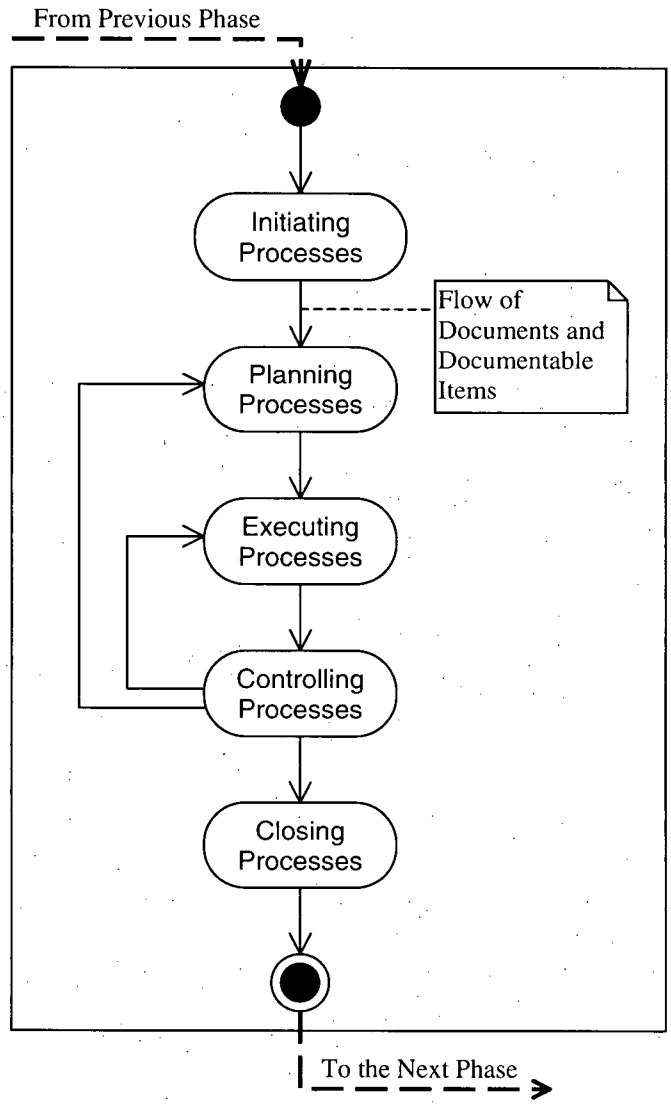


Figure 2–14: The Process Groups within a Project Phase

The PMBOK is deemed to describe PM in general. It covers the major areas and processes of PM common to all types of projects such as software development projects, chemical processing plant projects, biopharmaceutical research projects, educational projects, etc. However, it falls short in covering all processes performed in construction projects. Equipment management and materials management are two examples in this regard. Materials handling and distribution plays a significant role in the progress of most types of projects (including construction) as materials procurement, which is already considered in the PMBOK in the form of acquiring goods and services. This could be resolved, perhaps, by substituting procurement with two other areas of materials and equipment management. Another example is safety management, which is of a high importance in construction management but not included in the PMBOK. Thus, some modification may be needed in order for the PMBOK to be applied for construction

management. Two of the required modifications expressed in the PMBOK are a "need for application area extensions" and a need for "customizing process interactions". Moreover, the PMBOK concentrates heavily on managerial processes, but not "product-oriented processes", despite interactions and overlapping between the two groups of processes.

2.5.1.7 The UBC Approach

Focusing on CM functions and their supporting computer application systems, Russell and Froese [1997]—in the Construction Engineering and Management Group of the Department of Civil Engineering, the University of British Columbia (UBC)—group such business functions into four views by "*task disciplines*" (as opposed to life cycle stage, physical component type, etc.): *physical and environmental view*, *cost view*, *process view*, and *as-built view*. Further, two more views are added by Russell [1997] to these four views: *quality view* and *change view*. Each view represents the area of concern of a project participant and the collections of functions required of the participant. However, areas of overlap have been recognized among these views. Moreover, the possibility of defining some other views such as a *business view* (addressing business strategy, profit goals, marketing, etc.) is foreseen in this categorization of CM functions.

Centering on Total Project Systems (TOPS), the TOPS project [Froese et al. 1997a and 1997b] adopts the PMBOK approach [PMI 1996] (section 2.5.1.6) to extend the above six views into a list of categories of PM functions. Consequently, it suggests twelve groups of functions: *Project Scope Management, Time Management, Cost Management, Quality Management, Risk Management, Safety and Environment Management, Human Resource and Organizational Management, Management of Procured Resources, Operational Methods and Processes, Information and Communication Management, Financial Resource Management, and Physical Resource Management*. Furthermore, reasoning that PM involves "little strict sequential ordering of the processes" and that "it makes it difficult to fit into the $IDEF_0$ notion of domain processes", the project applies a tabling format to list the inputs/outputs (i.e., documents and data topics) of the listed function [Froese et al. 1997a].

2.5.2 Issues and Resolutions in Modeling PM Functions

This section formulates the results of the discussions presented in the previous section into a set of issues surrounding AEC/FM process modeling and addresses their solutions. The presented mateials, which highlights the research's process modeling criteria, were used to shape the general research criteria.

2.5.2.1 Complexity and Integration Issues

Modeling the whole AEC/FM industry's processes and their related information at once would lead to a very complex model; therefore, it is wise to approach the problem by breaking it into some parts that can be dealt with separately while considering a possibility for integration of the parts into the whole system. Following a classical approach to systems analysis and design, the system may progressively be broken down into some smaller subsystems until some manageable units are reached.

The classification of AEC/FM processes is more challenging than classification processes in other domains whose system breakdown can be mapped to the physical composition of the system. A building design process, for instance, entails generating a number of coordinated design views of the building (e.g., architecture, envelope, structure, mechanical, and electrical designs). Each subsystem's design represents the related discipline's view into the system. The general AEC/FM process modeling is, however, more challenging as no physical system decomposition is involved. A comparison of different approaches in AEC/FM process modeling—such as the use of "perspectives" in ICON project (section 2.5.1.3), "knowledge areas" in PMBOK (section 2.5.1.6), and "activity zones" in GDCPP (section 2.5.1.5)—proves this challenge. A part of the challenge lies in resolving the overlaps and interactions between subsystems. A major part of the complexity issue is due to the diversity of viewpoints and contextual factors (explained next).

2.5.2.2 Subjectivity Issue

One of the immediate issues in process modeling is *subjectivity*. Process models are abstractions of real processes in some systems that are deemed to serve a purpose; hence, every model presents one view of the system. Eriksson and Penker suggest:

> *"A business model can never be totally accurate or complete, simply because no two observers of a business will have an identical perception of the business or agree on an accurate model.....Not every detail can be captured due to restrictions in the modeling language or the concepts used to build the model."* [Eriksson and Penker 2000, p. 7].

An example may clarify the issue. Focusing on execution and excluding other activities such as planning and control, Figure 2-15 shows some of the major MM process areas. A number of questions could be raised when modeling for MM process: If one were to draw an $IDEF_0$ diagram (Appendix A) for materials management, which one of the processes would be included? How would they be placed in sequence? Where would the "*Expedition*" process be placed in the sequence? How many "*Approval*" processes may be involved in the presented cycle? Where would exactly an approval process be? How many alternative configurations (both in terms of number of processes and their sequencing) could be generated? Would *engineered materials* as well as *bulk materials* be modeled equally?

In general, the answer to most of the above questions would be: "It depends on..." This addresses the subjectivity issue of process models that lies in the contextual factors of AEC/FM processes, the scope of modeling, and the viewpoint of the modeler. Section 3.4, which reports the results of the Process Modeling Practice Workshop arranged in this research, may also be referred to for more elaboration on these issues as relate to the modeling of AEC/FM processes.

In order to address the subjectivity issue, Eriksson and Penker [2000, p. 7] suggest a business-process modeler should focus on "core business tasks" and the "purpose of the model". In order for a model to be more manageable (i.e., less complex) and understandable, the SADT/IDEF$_0$ modeling technique suggests to set a viewpoint (e.g., a foreman, an inspector, or a project manager) at the early stage of modeling (Appendix A) [Marca and McGowan 1993; Feldman 1998]. Every viewpoint has its own interests, and thus, may result in a different process model. The explicit representation and integration of the viewpoints still remain as a challenge.

Figure 2–15: Some of the Major Process Areas in Construction Materials Management

2.5.2.3 Context Issue (Specific versus Generic)

There is a possibility for process models to be oriented towards modeling a specific context or situation (e.g., the as-is processes of a specific company), as opposed to a *generic* process modeling (i.e., the core processes in a generic company). In order for the models to be generally applicable to other similar situations, it has been suggested to model processes in a *neutral* format; i.e., not to reflect a specific environment [Booch et al. 1999; ICON 1994]. For example, ICON uses an activity hierarchy to model generic processes involved in a construction project. ICON suggests that "an activity hierarchy defines the activities carried out during a construction project and is concerned with WHAT is done, not whom, when, where or how" [ICON 1994, Final Report, p. 33].

Seeking *neutrality*, the generic business process modeling may be targeted. However, in order to reduce the risk of ending up with unreal and less-useful models, in a bottom-up approach, the models may be verified against specific business processes and/or other existing models.

2.5.2.4 The Base of Classification

Process models usually involve classification of processes according to some bases, regardless of their purposes [Eriksson and Penker 2000]. Most AEC/FM process models are *project-phase (or stage) oriented*; i.e., based on the normal sequence of activities taking place in the delivery, operation, and maintenance of a facility (e.g., plan, design, construct, operate, and maintain processes). IBPM [Sanvido 1990] (section 2.5.1.2) and ICON [ICON 1994] (section 2.5.1.3) are examples of such models. The process classification in others models are founded on other bases such as *task disciplines* (e.g., design disciplines, construction engineering disciplines, etc.) and *physical project elements* (e.g., product, resources, etc.). For example, PMBOK [PMI 1996] (section 2.5.1.6), Chambers and Rand [1997], Russell and Froese [1997], and Froese et al. [1997a] (section 2.5.1.7) take a *function-oriented* approach to grouping PM functions. The GDCPP model [GDCPP 2001] (section 2.5.1.5) is a special case that is based on project phases, though it associates every process with an organizational unit (so called "process owner" or "activity zone").

A distinction, however, has to be made between *organizational units* and *functional areas* of a business. Organizational units encompass the *operating units* (e.g., factories, construction site, workshops, subsidiary companies, and the main company) and *departments* (such as sales department, purchasing) of the business. A *functional area*, on the other hand, is defined as "a unit in an organization which provides a specific function to the organization. It may or may not coincide with the traditional department structure" [Kubeck 1995, pp. 35-36]. Depending on the nature of the business, functional areas may include marketing, research, sales, distribution, production, accounting, finance, materials, personnel, etc. [Martin 1982; Chambers and Rand 1997]. Though this research does not refer to the term *functional area* as a department structure, but as a major area of activity in an organization. A functional area may include operational and/or managerial components.

The *classification base* issue roots into some other essential issues, which may be used as a basis for comparing various aspects of the two *phase-based* and *non-phase-based* process modeling approaches:

1) **Model Review**: The first approach generally has this advantage that one can follow operations step-by-step and walk through the whole model to check for any missing activity. The second approach, however, may not perfectly reveal this characteristic as it groups processes into some *categories of*

processes, which may include a number of activities whose sequencing may or may not necessarily resemble the sequencing of project phases.

2) **Overlapping and Repetition of Processes**: There is a possibility that processes be (partially or totally) repeated throughout a model (especially AEC/FM models) for many reasons, including intermediate decision-makings and approval processes. Phase-based process models are more prone to this issue. Examples such repetitions exist in the IBPM and ICON models (sections 2.5.1.2 and 2.5.1.3), and no mechanism has been suggested for *reusing* processes (without multiple definitions of a process and with possibility to modify and reuse a process definition, e.g., an approval of a design versus approval of a purchase order).

3) **Sequencing of Processes**: Due to interrelationships and overlaps among process categories, the sequencing of processes in the second approach is generally harder. For example, in *cost management*, different choices may exist for classifying and sequencing the related processes. One initial classification could be *cost estimation*, *cost budgeting*, *cost control*, etc. The *cost estimation* may further be decomposed into cost estimation of *materials*, *equipment*, *labor*, etc. Each of these may be the subject of other process categories (e.g., *materials cost estimation* in *materials management*). Resolving such overlaps would then require judgments, conventions, and standards.

4) **Focus**: Complying with PM practices and requirements, the second approach allows to focus on various aspects of a specific project element, however, this may not be easily possible in the second approach, which is restricted to a specific phase. For example, a project manager might wish to closely control project materials (i.e., its availability, timeliness, surplus and disposals) due to their very large effect on productivity and cost effectiveness of the whole project. The MM system of concern should assist the PM in performing the desired functions for the whole project, rather than just for one phase or stage of the project.

Considering its objectives, this research refers to PM functions as the total processes of a project. It takes a hybrid approach to modeling of PM functions by using different dimensions of PM as a basis for classification of the functions. It also emphasizes integration of processes and information. The next section explains the modeling method and techniques used for this purpose.

2.5.3 The Research Methodology for AEC/FM Process Modeling

This research addresses the identified issues (previous section) by applying a combination of top-down and bottom-up approaches to modeling PM functions, as illustrated in the activity diagram in Figure 2-16. For simplicity and understandability purposes, however, the diagram does not show the feedback loops involved in the process (e.g., refinement of plans as a consequence of further bottom-up

investigations and model developments). The following briefly explains the major activities and their related modeling techniques.

1) *Review Literature and Models*: This activity includes the review of the literature related to the subject and study of existing process models, and one of the main results of this activity was the identification of PM process modeling issues and criteria (presented in previous sections).
2) *Develop the CPMF Framework*: In a top-down approach, an analytical framework (called Conceptual PM Functions Framework, CPMF) capable of providing a fairly comprehensive list of PM functions was developed (Chapter 6).
3) *Explore PM Processes Using the Framework*: The CPMF model was used two ways, to identify PM functions (processes) in two formats: *tables* and *hierarchies* (section 6.3).
4) *Design the ITPMS Model*: In order to synthesize the results of the analyses performed in the previous steps (above), based on the theoretical foundations of the research and applying the CPMF model, the architecture and structure of the Integrated Total Project Management Systems (ITPMS) model was designed. The ITPMS was intended as a means of modeling the processes and information requirements of both business and software systems in an integrated manner (Chapter 7).
5) *Evaluate the ITPMS Model*: The design of the ITPMS model and results of the CPMF development were used to physically develop the ITPMS model in the Ensemble Streams™ CASE tool. Section 2.6.6.2 elaborates this activity more in details.
6) *Perform Bottom-up Investigations*: For validation purposes, the top-down development process of the research was supported by several bottom-up investigations, as described in Chapter 3.

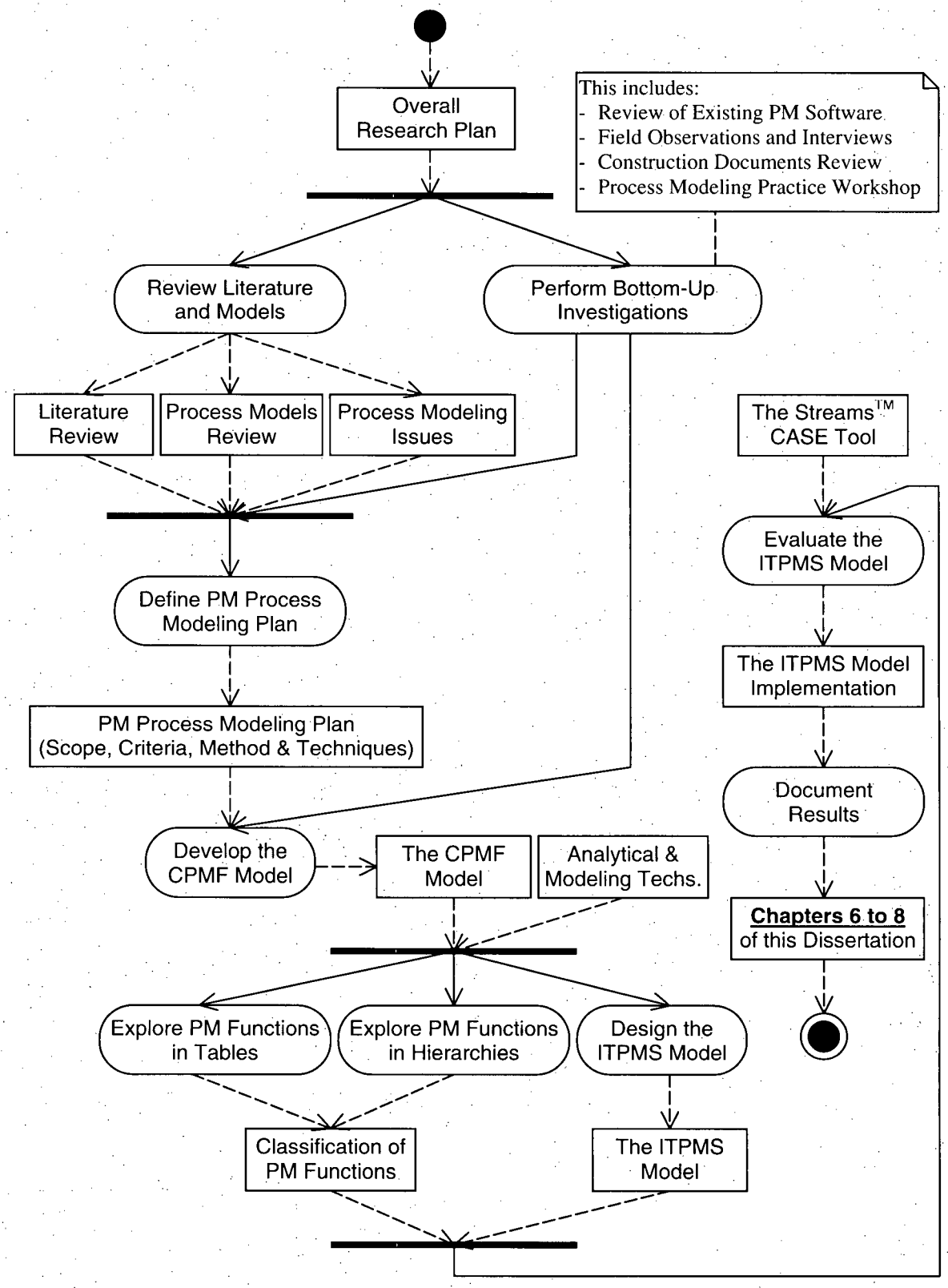


Figure 2–16: The Activities Involved in the PM Process Modeling of the Research

2.5.4 Summary of AEC/FM Process Modeling

This section discussed some of the issues related to AEC/FM business process modeling in the context of CIC through an overview of some of the major AEC/FM process models and their related issues. It also formulated a set of process modeling issues and approaches to their resolutions and explained the method and techniques used in the research for modeling PM functions for the purpose of integrated total PM systems development. Chapters 6 and 7 elaborate on the models developed in this research to address the issues. The next section specifically focuses on MM systems.

2.6 Materials Management Systems

Materials management (MM) plays an important role in construction projects. Materials information is used by AEC/FM project participants for a variety of purposes almost throughout the whole project life cycle, from conceptual planning to design, construction, and operation and maintenance. Considering the results of the investigations performed in this research, MM was targeted as a candidate PM function for detailed process and information analysis and modeling. Motivations for selection of MM function may be summarized into: 1) construction MM has neither been supported in the existing PM software, nor has it been dealt with in the existing AEC/FM information standardization efforts; 2) a very close natural relation exists between information about the final-target product and the materials incorporated into the product whose information is defined in a product or project model; and 3) the relation between materials and the target product has been represented only loosely in the existing PM software and the standard models(section 4.1).

This section is intended to provide the necessary background for communication of the research deliverables described in Chapter 8. It focuses specifically on MM in the context of CIC. It highlights its importance and presents a review and discussion of MM literature and models. It also introduces the approach (i.e., scope and methodology) of the research to modeling the functional and information requirements of MM systems.

2.6.1 Importance of Materials Management

Materials: Products are made of it, and many agents work on it. They market, plan, make, package, and store it. We look for it, find it, and choose it with the hope of *the best choice*. We negotiate and order it with the hope of *the best deal*. We count on it and build on it. They ship it and charge for it, and we pay for it. We follow and expedite it, and finally receive it; though, not always on time. We inspect it (expecting *the right one*), store it (with expensive labor and equipment in expensive stores), and handle non-conforming and defectives. We guard it, watch it, dust it off, and spend time and resources to count it every year. But, sadly, we see damages and wastes; so, we dispose it or replace it. They request it

with hopes. We locate it, haul it and pile it; though, not always in time and at the right place. It waits in the stockyard for a while, with the hope of being treated, used, and (perhaps) maintained as it was intended. All work is usually done in "islands of operations", each one concerning its own territory with little or no communication and sharing information and other resources. The end result, then, comes as a burden to all involved parties at all project, company, industry, national, and international levels.

The high impact of MM in construction productivity improvement is an important stimulus in directing the focus of this research on materials and MM, as one of the most essential PM functions. The Business Roundtable Construction Industry Cost Effectiveness (CICE) project reports: "The cost of materials and equipment usually runs about 60% of total project costs with construction labor costs at about 25%" [CICE 1982, p. 4]. In another study, the value of materials itself has been reported between 40-45% of the total cost of construction [Agapiou et al. 1998].

The criticality of MM function to organizational effectiveness [Tersine 1976] and benefits of an improved MM system [Stukhart 1995 and CII 1986] may not be overlooked. The studies by CICE and Construction Industry Institute (CII) reveal that basic MM system can be expected to produce a 6% improvement in craft labor productivity [CICE 1983b; CII 1986]. Moreover, "when, ... the crafts use the system to plan their work around bulk material availability, an additional 4-6% in craft labor savings can be expected" [CII 1986]. This 10-12% labor cost reduction (also reported by Ballard et al. [1994]) essentially originates from avoiding nonproductive-idling time of the labor waiting or searching for materials. A 20% loss of time has also been reported in some projects as a consequence of lack of materials at right times [Stukhart 1995]. The impacts of materials shortage and excess may also be extended to other dimensions of a project (e.g., production operations, revenue generation, and customer relations), raising the aforementioned rates to much higher levels. Moreover, taking all aspects of MM into consideration, one would appreciate the great potential savings that may be achieved by a good MM.

MM processes are usually performed in various parts of organizations such as purchasing, production control, transportation, warehousing, materials handling, operations, receiving, and logistics [Tersine 1976]. The consequences of this include:

1) *Lack of Consolidation of Materials Information*: Information disperses organizations' departments and sections. This would potentially lead to the problem of lack of *what-is-where knowledge* especially when paper-and/or-verbal media is used as a main means of communication. The poor communications documentation could even give rise to problems. Thus, when required, a specific piece of information may not be available at the right time to an individual person or department.
2) *Incompatibilities and Inconsistencies of Materials Information*: Different types of data about materials are represented from different perspectives in various formats and systems. Each

department, depending on procedural requirements of the organization, may communicate its information with few other departments. These differences in views, representation formats, and information systems could normally result in incompatibility and inconsistency of both data (i.e., facts) and information (i.e., contents) about materials.

The effects of these problems would be directly mirrored and transferred into other related PM functions, such as planning, scheduling, quality assurance/control, cost estimating, cost budgeting and control, etc. [CII 1986]. Due to the importance of materials information in handling other PM functions, management of such information proves to be crucial in the efficiency and effectiveness of the project. Modeling materials information as a central repository for MM system could help integrate among processes of MM function. Such integration could then have a direct positive impact on other related PM function. Standard materials information models can be developed in a way that the related MM application systems can share the model. Such standard models of materials information can then be shared by other related PM systems requiring materials information. In the age of computers, however, the formalization and standardization of MM processes and their information requirements in harmony with other related PM processes prove to be a critical factor in the automation of the processes using current information technology.

2.6.2 Categories of Materials Management Literature

The current MM literature may fall into two major categories: 1) the literature that presents an unformalized (though useful) body of knowledge about the subject area (e.g., textbooks), 2) the research and development literature. The knowledge presented in the first category of MM literature may include definitions of the terms and concepts, explanation of the related techniques and processes, provision of guidelines and recommendations, etc. There are also many textbooks that cover different aspects of MM in general [Davis 1976; Kulwiec 1985]. A large number of them deal with MM in hospital [Ammer 1983; Scheyer 1985; USEPA 1991] and manufacturing [Parbhu and Baker 1986] contexts. Among them, some elaborate only on a specific function of MM in a specific industry (e.g., materials inventory in a hospital environment).

However, there are a very limited number of textbooks specifically dealing with construction MM [e.g., Chandler 1978 and Stukhart 1995]. One of them that explains construction MM terms, concepts, and processes, is the textbook written by Stukhart [1995]. This book, which is aimed at educating experienced project and procurement managers and senior managers, provides a very good body of knowledge, and it was found very useful in this research. Its content was reviewed for both the analyses and the models presented in the dissertation. Nevertheless, to the knowledge of the author of this

dissertation, despite the profound content of the book that is backed up with the long research experience of its author in the subject area, no current research was found having a reference to this book.

The second category of literature (i.e., research and development) that may somehow relate to construction MM may be classified into two major groups, as suggested by Tommelein [1994]: 1) those dealing with the *supply and delivery of materials to the site*; and 2) those that focus on *materials laydown and handling on the site*. The first group concern materials procurement (i.e., supplier-buyer relationship) and/or delivery of materials to the site. Materials requirements planning (MRP) has been generally emphasized as one of the most critical factors in process improvement. Some have looked at the supply chain at a *strategic* level, while others have searched for technical solutions. Pheng and Chuan [2001], for example, present the result of a survey on the delivery of precast concrete components and possibility of just-in-time (JIT) delivery of such products in construction. Cheng et al. [2001] and Kong et al. [2001], on the other hand, focus on the supply chain and present e-business frameworks, models, and systems for its support. Similarly, others search for exploitation of benefits of other related IT in MM processes [Gibson et al. 1994; Cheng and Chen, 2002]. There are also studies that focus on the development of computerized systems for automating and integrating the processes of materials planning, sourcing, and procurement [Sadonio et al. 1998].

At the strategic level, *supply chain* process has become the subject of investigation for finding cost-effective approaches to the improvement of process [Agapiou et al. 1998; Pheng and Chuan 2001]. Traditional contracting has been criticized for emphasizing the lowest bid price rather than considering the larger picture of materials flow and the inevitable costs that must be borne later for logistics (i.e., materials handling and transportation) [Agapiou et al. 1998; Pheng and Chuan 2001]. In order to eliminate all unnecessary cost in the supply and delivery chains, a *logistics approach*[1] to construction MM is suggested. Long-term collaboration and partnership with suppliers, planning of production and deliveries based on the JIT concept, greater use of pre-assembled components, and minimization of materials handling and stocking are listed as some principles for a good logistics practice [Agapiou et al. 1998]. Nevertheless, data management and information exchange has been pointed out as a crucial factor to the success of this approach [Agapiou et al. 1998; Pheng and Chuan 2001].

[1] The concept of logistics was initially developed within the manufacturing industry and now constitutes an important management tool to ensure an overall strategic perspective on the flow of materials in the production process. This definition, however, has been modified to fit the concept for construction industry. "Logistics comprise planning, organization, co-ordination, and control of the materials flow from the extraction of raw materials to the incorporation into the finished building." [Agapiou et al. 1998, p.357]

The second group of the research literature study materials processes on the site and present ideas, patterns, methods and procedures, models (usually process models), and/or some kind of computerized systems that could assist the processes [Bernold 1990; McCullouch and Lueprasert 1994; and Echeverry Beltran 1997; Riley and Sanvido 1995 and 1997; Cheng and Chen 2002; Tommelein and Zouein 1993a].

Among all research subjects, construction space planning may be ranked as one of the most covered subject areas that is the closest to the planning part of MM. Some have focused on the site layout [Tommelein et al. 1993; Tommelein and Zouein 1993a; Tommelein 1994; Riley and Tommelein 1996], while others have dealt with the space use in the constructed facility (i.e., buildings) [e.g., Riley and Sanvido 1995 & 1997]. A number of efforts have also been directed towards algorithm development for site layout design/planning (i.e., allocating site spaces to temporary facilities). Some have tried heuristic approaches [e.g., Cheng and O'Connor 1996], while others have focused on other approaches such as genetic algorithm [e.g., Li and Love 1998; Hegazy and Elbeltagi 1999].

In order to provide the basic grounds for communicating the ideas and models developed by this research, an overview and discussion of some of the important relevant terms, concepts, and models suggested in the literature is presented in the following sections.

2.6.3 Materials and Materials Management: Definitions and Technologies

2.6.3.1 Construction Materials

The term "*material*" is generally defined as "a substance or things from which something else is or can be made; thing with which something is done; examples: *raw materials* (e.g., iron ore and oil) and *building materials* (e.g., bricks, timber, and sand)" [Oxford Dictionary 1989].

In a construction context, Stukhart [1995] defines *material* as "a substance or combination of substances forming components, parts, pieces, and equipment items." He also uses the term *materials*, in a plural form, to refer to "items used in producing a product for service and which include raw materials, component parts, work in process, finished products, packing and packaging materials, supplies, and equipment items". Here, the term *equipment* refers to permanent plant equipment and, possibly, minor equipment, parts, and special tools but not construction (moveable) equipment and its related parts. *Materials* are consumed, while *equipment* (e.g., a crane) is used to make a *product*. The first one is the subject of MM, while the second one is dealt with in equipment management, though there is a very close relationship between these functions.

Stukhart classifies construction project materials three major categories: engineered, fabricated, and bulk materials, as explained in the following.

1) ***Engineered materials/equipment***: Prescribed by the engineering staff of the owner, they are "either engineered and fabricated specifically for the project or that are manufactured to an industry specification and are often stocked by the manufacturer or distributor. These items have a uniquely assigned number such that they can be uniquely referred to through the life cycle of the project, and they are purchased to be used for a particular purpose, as opposed to bulk materials" They are frequently referred to as "tagged" items. "The tags should identify each item as it appears in a drawing, bill of materials or work item, serial number, and purchase order" (essential for control purposes) [Stukhart 1995, p. 290].

   An engineered material may have one or more installation and O&M (operation and maintenance) documents, usually in the form of manuals. These manuals, which are intended for both installation and maintenance purposes, normally relate to items such as mechanical and electrical equipment, operating doors, finish hardware, and folding partitions [Minks and Johnston 1998].

2) ***Fabricated materials***: These materials are manufactured to a project specification at a fabrication shop remote from the site. They are project specific and are identified by the fabricator on the drawings submitted for approval. A fabricated material is an "assembly of basic materials or component parts that are joined together to produce a finished part or a more complicated component, i.e., the building up of complicated shapes from simple stock materials, for example, a steel beam with holes, beam seats, and/or connecting angles added" [Stukhart 1995, p. 291].

3) ***Bulk materials***: These materials are manufactured to industry standards and are purchased in quantity (e.g., by lot, length, weigh, etc.). Being bulk in nature, they are not uniquely identified and tracked by individual items. They are also normally allocated at the time of fabrication and/or construction per schedule priority. Examples are pipes, wiring, and cables. Bulk materials constitute the majority of site materials, and their planning is the most in complexity.

The above classification of materials types seems to be useful in many ways, some of which mentioned above. To name a few more, "many projects divide the responsibilities of materials according to their types. For example, the home office will procure engineered items and initial bulk procurement. The field will purchase the remaining bulks, after requirements have been fully defined. Engineering may control fabricated steel items, to ensure dimensional quality, and control systems because of the specialized knowledge needed" [Stukhart 1995, p. 70].

Second, the type and sequencing of MM activities and, thus, documentation of the process are highly dependent on the type of material involved. For example, in a typical procurement process for

engineered and **fabricated** materials, for each major item listed in the *bill of materials* (e.g., equipment list), a specification is developed. This is followed by a *requisition*, *request for quotation* (RFQ) from suppliers, *quotation* and its *evaluation and approval*, and *purchase order* (PO). After, *acknowledgement* of the PO, the vendor prepares *shop drawings* for approval. Having the drawings approved, the vendor begins *fabrication* and prepares any required *certification*. However, for **bulk materials**, this whole process may not be the same. Many of the processes (e.g., shop drawing and certification preparation and fabrication) may not apply to bulk materials. PO quantities may be developed from either *estimates* or *actual materials takeoff* (usually generated from design documents) depending on the type of the bulk material and other constraints (e.g., availability, criticality and cost of the item). For instance, initial materials may be purchased based on rough estimates at the beginning of a project, much before the design of the facility is completed and actual materials takeoff is prepared. However, for inexpensive, readily available, non-critical bulk items whose demand schedule can be predicted with reasonable accuracy, a "min-max" system may be used. The min-max system defines the lowest and maximum level of inventory that must be considered to maintain a balance between the supply and consumption of materials [Stukhart 1995, pp. 56-57; Ammer 1974, pp. 305-307].

Therefore, in general, bulk materials may require fewer *types* of processes and documents than engineered and fabricated materials. Nonetheless, a higher number of *instances* of the required types of processes and documents may be necessary for bulk materials. For instance, only one PO may be submitted for an engineered material, while many PO's may be sent to the supplier of a bulk material (e.g., gravel, concrete, etc.) at different times during the development of a project.

2.6.3.2 Materials Management and its scope

MM has been viewed differently in the literature. While some limit the scope of MM within the supply-chain and/or materials handling (movement) processes [Tersine 1976; Illingworth and Thain 1988; and Coad et al. 1999], others extend the scope beyond organizational boundaries [CICE 1982; Stukhart 1995]. The Construction Industry Cost Effectiveness Project (CICE) of the Business Roundtable suggests:

> "*Materials management is the management system for planning and controlling all necessary efforts to make certain that the right quality and quantity of materials and equipment are appropriately specified in a timely manner, are obtained at a reasonable cost, and made available at the point of use when needed. It should be noted that materials management is a system, not the organization responsible for performing these tasks. However, it does affect activities traditionally performed by engineering, purchasing and construction departments*" [CICE 1982, p. 24].

This view, which considers MM as a "*system*" or a *process* rather than an "*organization*", has been supported in some other literature as well [e.g., Stukhart 1995]. This view implies the necessity of consideration of the total MM as a set of related subsystems or sub-processes (such as purchasing,

expediting, distribution, handling, and storage) that may cross the functional/ organizational boundaries to achieve a set of business goals. Such sub-processes may include purchasing, production control, transportation, storing, warehousing, materials handling, operations, receiving, and logistics. However, the definition limits the scope of MM systems to the "planning and controlling" functions. Section 6.1.2 explains other possible basic PM functions.

In the context of construction projects, the main elements of MM systems are:

1) The *substance* (in any form: raw, fabricated, or engineered) incorporated into the facility (or product).
2) The material-related *processes* (e.g., selecting, specifying, ordering, testing, storing, distributing, and tracking materials).
3) The *resources* involved in MM processes. Considering the 4Ms concept (i.e., materials, machines, manpower, and money), Stukhart lists such resources as "materials, computers and computer programs, home office and site materials personnel, storage, transportation, automated identification, electronic data interchange, and materials-handling facilities" [Stukhart 1995, p. 29].

The moveable construction equipment (e.g., loader, dozer, crane, etc.) and construction crews involved in application and incorporation of materials into the final product may not be considered within the scope of MM. However, depending on the type and organization of projects, the scope of MM may or may not include the materials used for temporary facilities (e.g., moveable annexes/containers used for site office) and their related processes, materials used for making equipment (e.g., timbers used for concrete formwork, ladder, etc.), and energy and supplies materials (e.g., water, office supplies, etc.).

In summary, MM encompasses parts or all of such activities as defining, specifying, requesting, ordering, purchasing, handling and storing, distributing, and tracking of materials, and it may be of concern at different levels of organization (e.g., corporate versus project level).

2.6.3.3 Evolution of MM Systems and Use of IT

A three-stage evolution may be assumed for MM [Stukhart 1995]. At first, technologies and project organization were not as sophisticated as today, and a simple organization would be sufficient for MM. Individual supervisors handled all of the processes regarding the management of materials. The processes would include identifying needs and acquiring, receiving, and controlling materials.

The second stage coincided with the advent of new construction materials and equipment, which resulted in complications in construction technologies and methods. Variation of materials, especially in large projects, required dealing with more specialty vendors. Manual information processing and low-level communications precluded a very cost-effective project. These all led to a centralized purchasing

system with firm inventory control procedures, which provided for improvement of materials traceability and reduction of shipment cost.

Now, at its third stage of evolution, MM is viewed as the entire range of activities encompassing the total MM, with the goal of getting the right quantity and quality of materials to the customer at the right time and in the most cost-effective way. It is sought that, in its next stage, MM must move into "a *fully automated system*, with materials management playing a key role in the project economics. The *process view* of materials management looks at the way that materials can restructure engineering and construction" [Stukhart 1995, p. 3].

The process view suggests a project organization over functional one; moreover, it emphasizes the role of IT in automation of MM processes. Despite its potential benefits, however, the use of IT in construction MM lags far behind those commonly used by manufacturers [CICE 1983b, p. 56]. Among the solutions offered by Stukhart [1995] are use of Electronic Data Interchange (EDI) [Gibson and Bell 1990], third-party shipment and location tracking, partnering with customers and suppliers, and accurate information exchange with customers and suppliers. Elaborating on the role of computer systems in MM, Bell [1989] describes some other areas of functionality that can be covered by such systems as well as the criteria for such systems. In particular, the areas include materials planning, materials takeoff and engineering interface (e.g., sharing commodity and specification information, conversion of bulk materials coding systems, generation of bill of materials from design), vendor inquiry and evaluation through historical performance, purchasing (e.g., generation of purchase orders from bill of materials), expediting, field material control (e.g., alternate control methods such as "min-max" for bulk materials, warehouse inventorying, and "trial allocation" simulation for warehouse withdrawals). The next section elaborates on how MM functions relate to other PM functions.

2.6.4 Materials Management and other PM Functions

There is an interrelationship between MM functions and other PM functions. Studying modern management systems, the Construction Industry Cost Effectiveness (CICE) project of the Business Roundtable [CICE 1982] considers PM as a comprise of four basic interrelated functions: 1) planning and scheduling; 2) cost estimating, budgeting, and control accounting; 3) quality assurance; and 4) materials management. Emphasizing the interrelationship between the four systems, it then suggests, "Where the use is justified by the size of the project, automatic data processing of systems of these four functions should be designed so that each is self-contained and reports the data required for control of the function. At the same time, the systems should be linked so that a change of data in one function will immediately show its effect on all the other functions" [CICE 1982, p. 4].

How are these functions related? Highlighting the key role of MM in project information communication, Stukhart [1995] elaborates on communication links among the control components of MM, scheduling, cost accounting, engineering, and CM functions (Figure 2-17). The necessity of establishing such links (e.g., between the contractor's home office and the field, purchasing function and engineering and the contractor, etc.) is further emphasized.

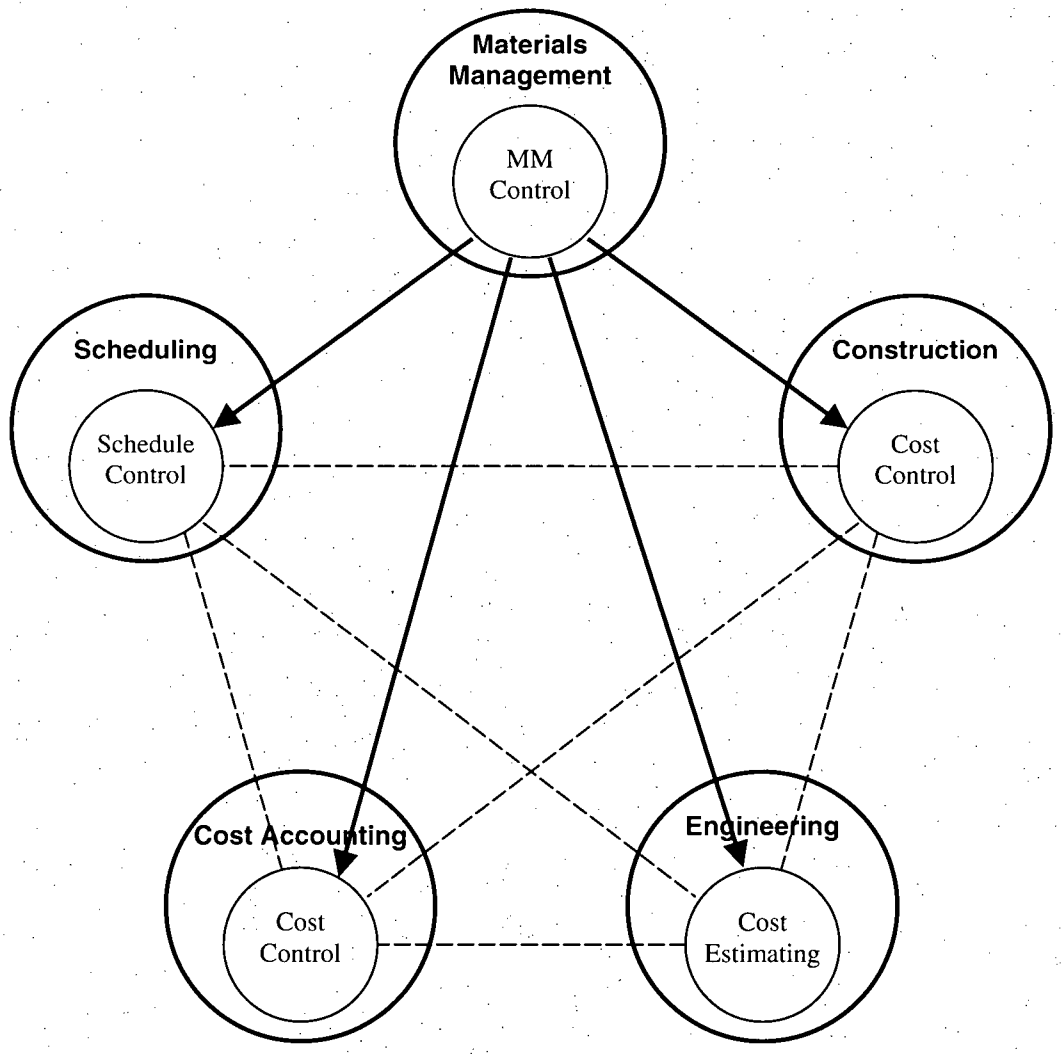


Figure 2–17: Communication Links among Organizational Functions

[Based on Stukhart 1995, p. 50]

Nevertheless, many more functions and links may also be added to those suggested in Figure 2-17. Considering the scope of MM some other points of interactions between MM functions and other PM functions may be identified as it follows.

1) Dealing with handling of surplus and waste materials, MM has an interaction with *Byproducts Management* (i.e., *Surplus and Waste Management*).
2) Posting costs and paying for invoices, in materials purchasing, indicates an interaction between MM and *Accounting/Financial Management*.
3) Dealing with storage of materials, MM interacts with *Spatial/Plant Facility Management*.
4) Tracking of the materials stored in the storage units involves interaction with *Inventory Management*.
5) One part of *Change Management* activities relates to handling materials changes.
6) Materials usually come with safety issues, thus, interactions with *Safety Management* exists.
7) The defects handling during commissioning and occupancy (or operation) of a constructed facility involves an interaction with *Occupancy and Facility Management*.
8) MM involves acquiring, training, and evaluating human resources. This highlights a link with *Human Resource Management*.
9) Materials handling and transportation has a link to *Equipment Management*. A major part of the issues discussed in construction methods and equipment literature concerns the management of materials [Peurifoy 1985; Al-Hussein 1999].

The complexity of the resulting network of links between MM and other PM functions may be overwhelmingly increased as different possibilities of project organization are introduced to the network. Such a complexity would, of course, raise the challenge of modeling these functions and their data-sharing boundaries. The next section presents an overview of some research and development projects that their developed models and systems somehow relate to materials management.

2.6.5 A Review of Selected MM Modeling Research

2.6.5.1 Cheng and O'Connor [1996]

Cheng and O'Connor [1996] introduce an automated site layout system, called *ArcSite*, for temporary facility layout design in construction projects. ArcSite includes a GIS (Geographic Information System) integrated with a DBMS (holding database of temporary facilities as well as the GIS's data), and necessary algorithms. The system uses a heuristic approach to acquire the knowledge specific to construction site layout design and generate the potential sites for each temporary facility.

2.6.5.2 Cheng and Chen [2002]

Cheng and Chen [2002] describe the development of a prototype schedule monitoring system, called *ArcSched*, for precast building construction (lifting schedule of prefabricated structural elements).

The system is composed of a GIS integrated with a DBMS, MS Project, and a wireless barcoding system, and its programming environment is MS Visual Basic (VB). Each prefabricated unit is assigned a unique barcode number, which is used to refer to the unit for shop drawings development, schedule generation and control, inventory control, and so on in manufacturing, transportation, receiving, and installation.

The ArcSched system provides such functionality as erection schedule generation (in MS Project) and query (in VB's UI), video monitoring of installation process, status reporting of prefabricated elements (i.e., not delivered, delivered, installed), viewing of design drawing with erection sequence display, viewing and editing of the associated data (e.g., storage area, position, and erection sequence and schedule) of elements and viewing of the shop drawings associated with elements (in the Arc/Info GIS).

2.6.5.3 Echeverry and Beltran [1997]

Echeverry and Beltran [1997] describe a prototype system, which is an extended version of the system developed by Echeverry [1996] for collecting where-about information on field personnel using barcodes. The prototype system is implemented in MS Access and is integrated into a stand-alone barcode reader for the purpose of construction field personnel and materials control. The barcode reader records access/exit time of personnel and charge/discharge of materials (e.g., brick and steel rebar) to/from storage area. The data files created by the barcode reader are then downloaded into the prototype system for further information processing.

The system provides for detection of cost and time deviation through linkages among such entities as Activity, Cost item (with components of labor, equipment, and material), Labor (described through individuals having personal ID's and grouped into crews), Material (with attributes such as ID, unit cost, and supplier), and Material Supplier (with ID, name, address, contact, etc.). Deviations are determined by comparison of the planned and actual quantities. Issues encountered in implementing the system are listed as: high personnel turnover rate on construction site, requiring timely issuance of bar-coded ID cards and cost consequences; fraud issues; and wear resistance issue of barcodes.

2.6.5.4 Kong, Li, and Love [2001]

Comparing the traditional construction material procurement process with electronic procurement, Kong et al. [2001] describe the design and implementation of an e-commerce prototype system for construction materials procurement in China. They list a number of reasons for suggesting an "*online construction trading markets*": physical store elimination, convenience, more supply points choices, middlemen elimination, lower transaction cost and time, and win-win situation for parties.

Focusing on "on-line ordering with off-line delivery" in an "electronic market" setup (or "e-mall", as suggested by Timmers 2000, with possibility of presence of agents), they describe an internet-based e-commerce prototype system called *COME* (COnstruction Materials Exchange). The system includes five modules: *e-catalog* (providing information about materials and their suppliers, with the assistance received from a GIS system), *bidding* (a buyer request for desired materials, suppliers respond with their bids, and the buyer view and accept one bid), *requisition* (buyers send requisitions to suppliers, and suppliers view them) *quotation* (suppliers send quotations to buyers, and buyers view them), and *order* (buyers place orders directly or through the agent). The bidding and quotation modules, together, are designed to handle the quotation process. As described by Kong et al. [2001], the system's functionality may be summarized as the provision of information about materials and suppliers (provided by suppliers registered in the system) and handling the quotation and ordering processes through a centralized e-mall.

2.6.5.5 Pheng and Chuan [2001]

By the means of survey and interview with 40 precast-concrete contractors in Singapore, Pheng and Chun [2001] investigated the potentials of just-in-time (JIT) materials delivery (originated in the manufacturing sector) in construction. Overall, the study found that contractors were focused on price and have generally overlooked the bigger picture of total costs. As a motivation for JIT implementation in the industry, the study recommends the reimbursement of cost, sharing of the savings, and specific and precise contractual agreements, which spell out responsibilities, expectations, and scheduling of costs.

Due to the components' criticality to the construction process and in order to safeguard against delivery (and thus construction) delays, the contractors preferred to stockpile the components (usually between 1 day to 2 weeks) at the site as "stand-by-stocks", despite its inherent risks (e.g., space constraints, possible damage to components due to weather conditions and physical site conditions).

All contractors surveyed had had their site engineer or coordinator plan for deliveries, without exact timings, and precasters were found not being updated of the site's progress. Precaster's delivery vehicles had to wait for unloading for an average of 1.7 days for many reasons, such as site congestion, inadequate manpower for unloading, zero-double-handling (i.e., direct hoisting to installation, without storage) sought by contractors, busy cranes, and unavailability of site storage. Rejections due to deliveries (e.g., inadequate supply and wrong or poor quality items) were also quite common.

Expecting improvements in the problems observed, the study recommends JIT materials delivery; i.e., providing the right materials, in the right quantities and quality, just in time for production. The implementation of the JIT philosophy requires precise planning and coordination of the delivery process,

and it may help smooth the production process through the efficient materials handling in construction. Attaining such benefits, however, may not truly come into existence with current practice of relying on the *phone-and-fax* technology for coordination and communication of project information (as observed in the study), but with *integrated computer systems*.

2.6.5.6 Riley and Sanvido [1995 and 1997]

Riley and Sanvido [1995] focus on the use of space inside a building for interior work (e.g., mechanical work, drywall, and finishes) and enclosure work (e.g., masonry, and curtain wall) in multistory buildings (3-10 story) and propose a construction space model, which defines a collection of descriptive space types and typical patterns of space use in multistory building construction. This model (Figure 2-18) reflects an integration of product and process information as it relates to space planning.

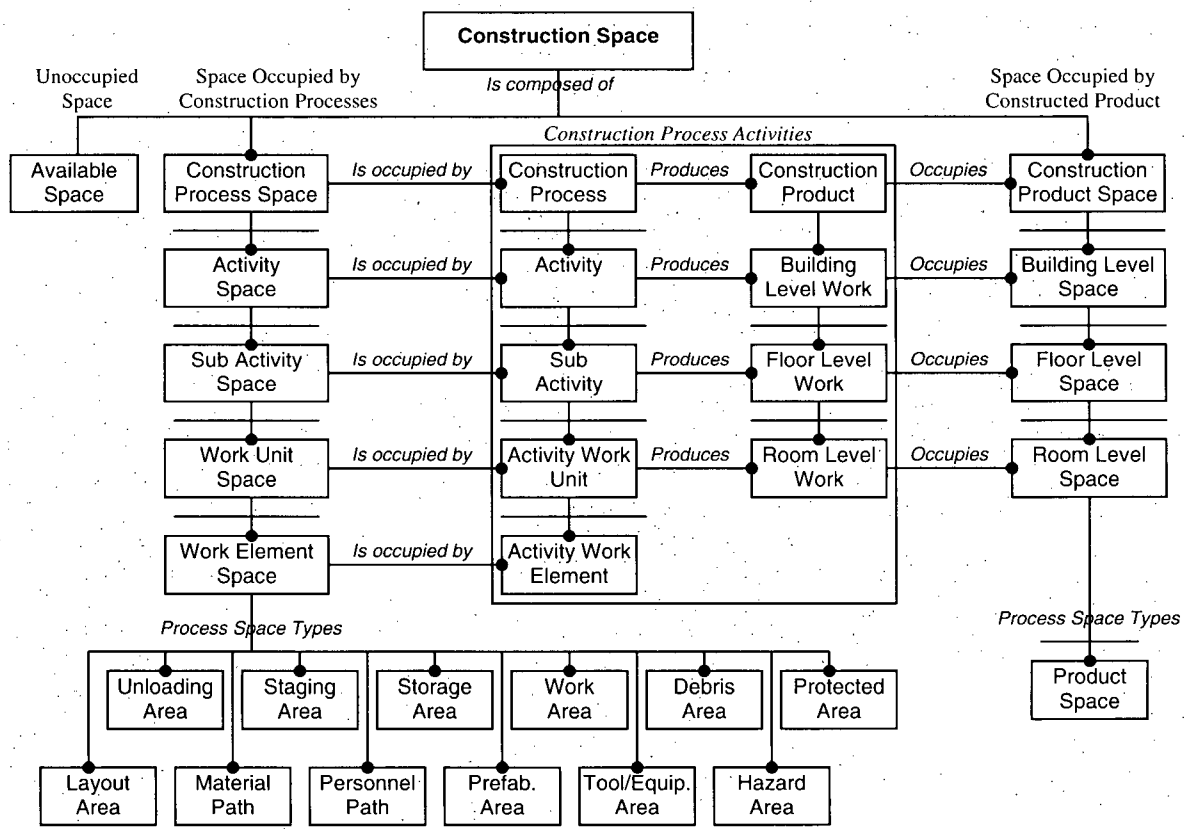


Figure 2–18: Construction-Space Decomposition

[Based on Riley and Sanvido, 1995]

Taking one step further to their previous studies [Riley and Sanvido 1995], Riley and Sanvido [1997] propose a *process model for space planning of multistory buildings*. Using $IDEF_0$ diagramming

technique (Appendix A), Figure 2-19 shows the high-level process of the model. The entire process breakdown structure is also summarized in a hierarchical formal in Figure 2-20.

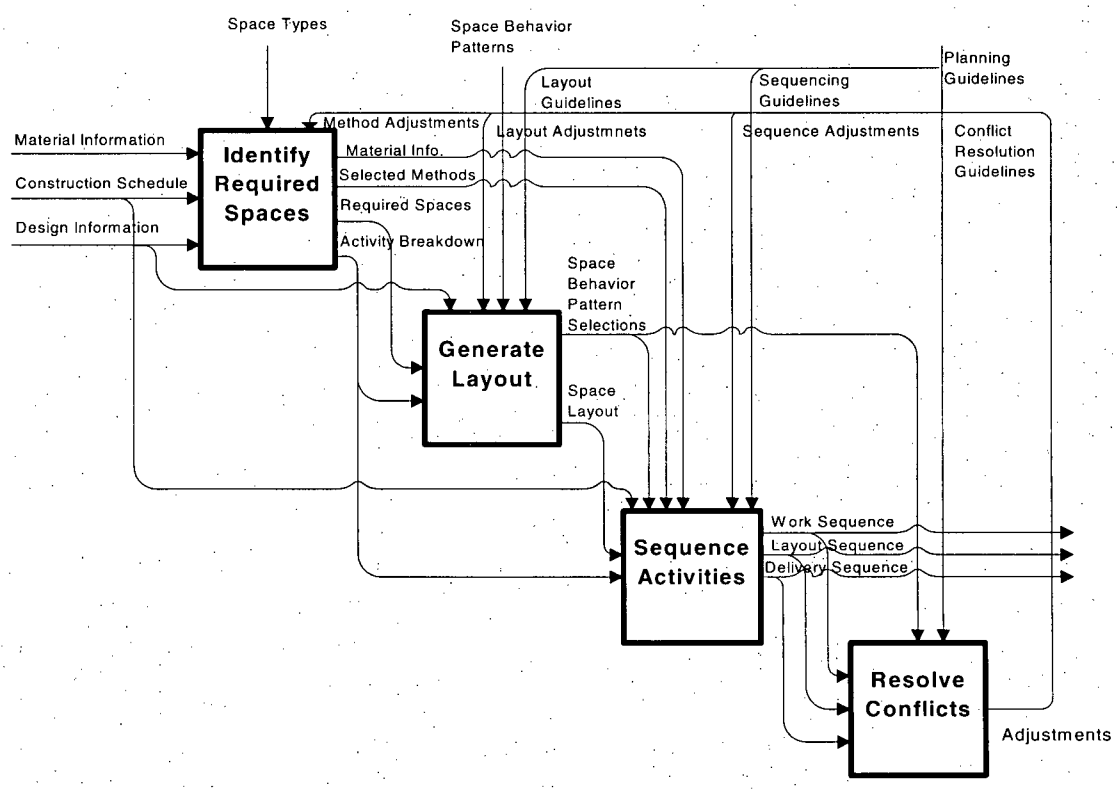


Figure 2–19: The High-Level Process (Create Construction Sequence) in the Space Planning Process Model

[Based on Riley and Sanvido 1997]

A0. Create Construction Sequence

- **A1. Identify Required Spaces**
  - **A11.** Identify Material Information: *Determine physical characteristic, spatial attributes, and availability.*
  - **A12.** Select Construction Methods: *Select means and methods needed for each work element.*
  - **A13.** Identify Work Activity Spaces: *Identify required work spaces based on selected methods.*
  - **A14.** Identify Material Spaces and Paths: *Identify spaces needed to access and support work areas.*
- **A2. Generate Layout**
  - **A21.** Assign Space Behavior Patterns: *Select a pattern in construction space model to characterize space needs.*
  - **A22.** Layout Room-Level Spaces: *Determine position of work areas and other room level spaces.*
  - **A23.** Layout Building-Level Spaces: *Determine position of unloading areas, vertical paths, and other building space.*
  - **A24.** Layout Floor-Level Spaces: *Determine position of storage area and other room-level spaces.*
  - **A25.** Create Space Layout: *Develop a graphical plan illustrating locations of needed spaces.*
- **A3. Sequence Activities**
  - **A31.** Identify Room-Level Sequence: *Determine work sequence based on hard logic and known dependencies.*
  - **A32.** Identify Building Sequence: *Determine the order each activity will work on floors, e.g., top-down/bottom-up.*
  - **A33.** Determine Floor Sequence: *Determine work direction for activities to work on floors, e.g., left to right.*
  - **A34.** Identify Material Delivery Sequence: *Identify milestone schedule dates when materials will be placed on floors.*
- **A4. Resolve Conflicts**
  - **A41.** Identify Interferences and Blockage: *Evaluate layout sequence for overlapping spaces.*
  - **A42.** Determine Activity to be Modified: *Identify which activity's space can be altered to prevent interferences.*
  - **A43.** Determine Conflict Resolution Method: *Select plan to adjust method, sequence, storage location, delivery date, etc.*
  - **A44.** Determine Conflict Resolution Action: *Take specific action to adjust plan to avoid interference.*

Figure 2–20: A Hierarchical Representation of Space Planning Processes

[Summarized and organized based on Riley and Sanvido 1997]

2.6.5.7 Sullivan [1989]

Sullivan [1989] describes an "integrated plant database system", which is used by Stone & Webster Engineering Corporation (SWEC). The system utilizes the 3-D solid modeling and relational database technologies in such application areas as plant engineering, construction sequencing, project control, document control, materials management, and facility management.

A mainframe-based CAD system (called CATIA) is used to produce 3-D solid modeling of plant facilities. Attached to solid models of physical components of a facility are user-defined attributes (e.g., a unique identifier). The components are linked to a relational database (DB-2), which holds geometrical information as well as other project information such as component ID, requisition number, purchase order number, specification number, delivery status, weights, design data, reference drawings, etc. The

data model represented in the database includes the information commonly used by the company's proprietary CM application systems, called COMANDS (Construction Management Display System).

The functionality of COMANDS is mostly based on the traditional planning and scheduling [Hendrickson and Au 1989] and association of the work and schedule information with the 3-D components of the project in the database. The components' 3- D solid model is used for estimating quantities. The installed quantities are later associated with the components. The work breakdown structure (WBS) of the project is defined, and scheduling is performed by identifying project activities and associating them with work packages and by estimating the activities' man-hours. The WBS of the project is used as a base for tracking installed quantities and physical progress reporting for accounting purposes. The integration of design, plan, and control quantities are achieved through the assignment of WBS accounts and activities' IDs as attributes of the 3-D components. The estimated and installed quantities can be updated (to measure physical progress) by selecting the components in the 3-D model.

The system can also visually display the construction sequence. This is accomplished by disassembling the engineering 3-D model into its base elements and grouping of them according to their related construction activities, and simulating their construction based on the activities' sequencing logic.

Based on the specifications and codes entered in the system in advance, materials requirements are automatically generated in the form of bill of materials (BOM) as the designer creates or modifies the 3-D solid model of a piping system, for instance. The user can then group the materials requirements into requisitions (represented in the database) for procurement. In order to identify what is remaining to be requisitioned, the system periodically compares the materials requirements (on a commodity basis) against previously requisitioned materials. The database also includes an inventory of purchase orders, with information about promised delivery dates, and received materials.

2.6.5.8 From Tommelein and Zouein [1993] To Riley and Tommelein [1996]

Explaining various construction-oriented space-time management models, Tommelein and Zouein [1993] focus specifically on dynamic layout planning of temporary facilities and equipment in construction. Considering a space as a resource, they emphasize the importance of management of spaces and site layout and suggest space plans be considered as a part of project plans. They also describe a PC-based prototype system, called *MovePlan*, which is an interactive object-oriented graphical program that assists field engineers in allocating spaces to resources over time.

Inputs to the system include dimensions (length and width) of the site, building, and resources (i.e., equipment, laydown areas, and other temporary facilities), the activity network (through the user or importing a Primavera data file), and a time interval for which to create site layouts. The system can then

generate resource-space histograms (very much like resource histograms in scheduling applications) as well as a two-dimensional (2-D) layout, which includes rectangular templates representing the building, the site, and the resources. MovePlan allows the user flexibly plan for any time interval (i.e., at any level of detail), but the evaluation of impacts of schedule change on layouts is impossible; once a site layout is saved, the schedule cannot be changed. User interpretations and judgments are central to its functionality.

MovePlan was later integrated with a knowledge-base reactive scheduler system, named ConRes (Conflict Resolver), to form a prototype decision support system, called *MovePlan-ConRes* [Tommelein et al. 1993], capable of heuristically allocating spaces to equipment and materials. In another extension, MovePlan was integrated with a surveying tool to form *MoveCapPlan* [Tommelein 1994], which assists real-time spatial positioning of elements on the site layout. Also, Zouein [1995] built on MovePlan and developed a system called *MoveSchedule*, which replaces the MovePlan's visual interference inspection with an automated interference checking.

Riley and Tommelein [1996] present a comparison of MovePlan and another space planning tool, called *CADPlan*, which integrates two off-the-shelf commercial tools (i.e., AutoCAD and Microsoft Project) using AutoCAD's AutoLisp programming language. The layering and blocking mechanisms of AutoCAD are used for representing space demands for temporary facilities/resources (i.e., areas and pathways used for the unloading, storage, and construction work, and material handling) and architectural plans. The status of space demand for selected time intervals is then graphically displayed by turning the layers on and off. The user detects space conflicts by visual inspection of the plan, and they are resolved through adjustments in the schedule. MovePlan is limited to 2-D space plan representation, while CADPlan is capable of space planning for multistory buildings and can potentially benefits the 3-D representation capability of AutoCAD through AutoLisp programming (though, it uses line-drawing and block methods).

2.6.5.9 Summary of MM Literature Review

This section highlighted the importance of MM in construction projects and presented a review and discussion of some MM literature and models. It demonstrated how computer application tools could assist PM functions, and more specifically MM functions (i.e., materials handling and construction space management) and improve process efficiency. Although these tools may be found intuitively simple (e.g., 2-D representation and missing the third dimension; shallow data modeling of central objects such as spaces; and heavily relying on the user's knowledge and input), they certainly represent a class of specialized software that can ultimately boost the productivity of the AEC/FM industry (e.g., 20% of construction time saving by MovePlan [Riley and Tommelein 1996]).

The review of such tools also highlights the next steps for improvements. First, due to the high importance of spaces and their management, especially during construction (e.g., for smooth processes with no congestions and disruptions), an explicit representation of spaces and the objects occupying such spaces (e.g., materials and equipment, temporary facilities, and even the construction crew and others) is needed. According to Tommelein et al. [1993], for example, in the MovePlan model, "*resources*" are interpreted "*in a broad sense*" to refer to "*physical objects (e.g., materials, equipment, trees, existing buildings) and spaces (e.g., laydown areas, finished building floors that can be used for laydown, excavation areas, or trenches).*" This raises this question: Does this interpretation satisfy other PM functions' views (than space planning)? Second, such tools can demonstrate the most potential when they are used in concert with other PM/MM tools. This potential, which requires interoperability between the tools, addresses a third opportunity for improvement: integration of PM/MM software tools. This is an area of research that is the subject of current CIC initiatives, and it is considered within the objective of this research. The next section describes the approach of this research to attaining this objective.

2.6.6 The Approach to Modeling MM Processes and Information Requirements

This section describes the approach (i.e., the scope and methodology) of the research to analysis and modeling the functional (i.e., processes) and information requirements of construction MM systems.

2.6.6.1 The Scope

In general, *construction MM processes and their information requirements* are considered as a core of discussions, analyses, and modeling in this chapter. Considering both operational-technical processes and managerial processes (section 8.1.1) within its scope, the research searches for a whole range of functions and information that may be of concern in MM systems. This modeling effort, which has mainly aimed at concretization of the conceptual ideas and frameworks of the research, has two main streams: business process modeling (e.g., uses case models and activity models) and object modeling (i.e., domain object models).

The research refers to MM as a *system* or a *process*, not as an organization. Thus, it tends more towards modeling materials-related processes and information, rather than dealing with organizational and human resource issues of the system (i.e., how the system is physically organized). Moreover, the research places an especial emphasis on MM from the perspective of project managers rather than others (e.g., manufacturers, supplier), although the ITPMS model (Chapter 7) used for representing MM models is generally capable of including the process and information requirements of all views.

There are processes that may be considered as *external* to a project's processes (from a PM point of view) but interacting with and, therefore, related to them. Examples of such processes include materials

prototyping and marketing (by manufacturer), purchase order processing (by supplier), responding to RFQs (Request For Quotes), shipping, invoicing, etc. Such processes are studied at a framework level, and their detailed analysis and modeling is not considered within the scope of the research. Moreover, moveable-construction equipment (e.g., loader, dozer, crane, etc.), which is the main focus of the equipment management function are not considered within the scope of this part of the research. Though, this function has a very close relationship with and may play a major role in MM, especially when materials handling becomes of concern.

In addition, the research does not intend to elaborate on theories, principles, and strategic aspects of MM that may be found elsewhere [e.g., Ammer 1983; Stukhart 1995]. However, it may refer to such aspects as they relate to the overall objectives and scope of the research (section 1.4.1), which are oriented towards AEC/FM process-and-information modeling.

2.6.6.2 The Methodology

The methodology applied for MM process and information modeling generally fits in the overall research methodology (section 1.4.3) and complements the PM process modeling activities and their results (section 2.5.3). In addition to the literature review presented in previous section, the methodology generally included three basic activities:

1) *Conceptual Analysis*: In a top-down, analytical approach, major groups of materials-related processes were identified.
2) *Detailed Analysis*: The CPMF model (section 6.2) and the two analytical techniques of two-dimensional tabling and hierarchical listing were used to list possible materials-related functions in a semi-structured fashion.
3) *Formalization*: Amalgamated with the results of bottom-up investigations (Chapter 3), the results of the above were fitted into the ITPMS model (Chapter 7) using the Streams CASE tool (Appendix A). This resulted in a set of MM business and information models.

The bottom-up investigations, which included the review of existing PM software, field observations and interviews, and the Process Modeling Practice Workshop, backed up the whole modeling process as verification checkpoints. Figures 2-21 and 2-22 present two activity diagrams depicting the workflows of MM process modeling and MM information modeling respectively. For simplicity, however, the diagrams do not show the feedback loops involved in the workflows.

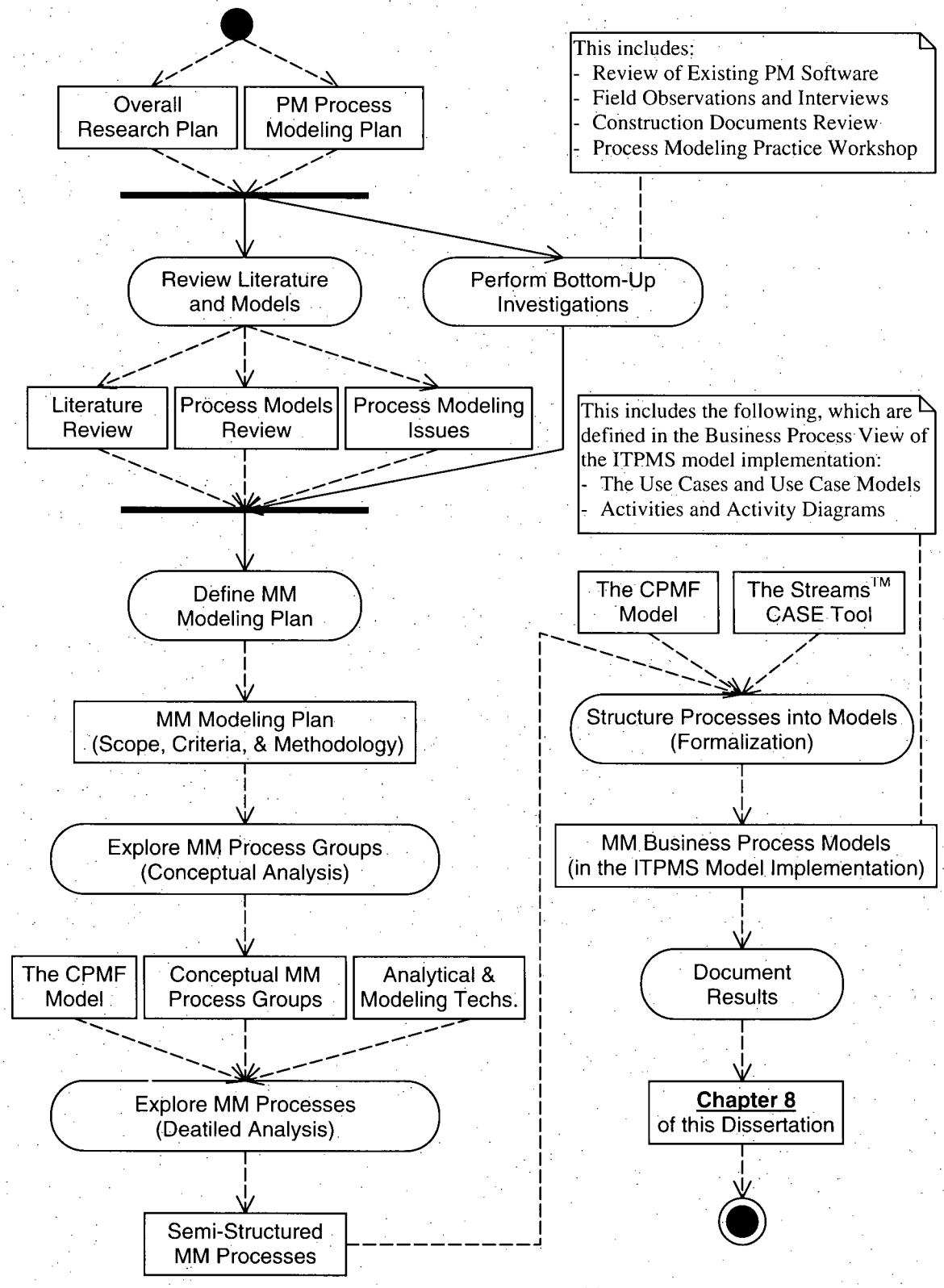


Figure 2–21: The Workflow Involved in MM Process Modeling

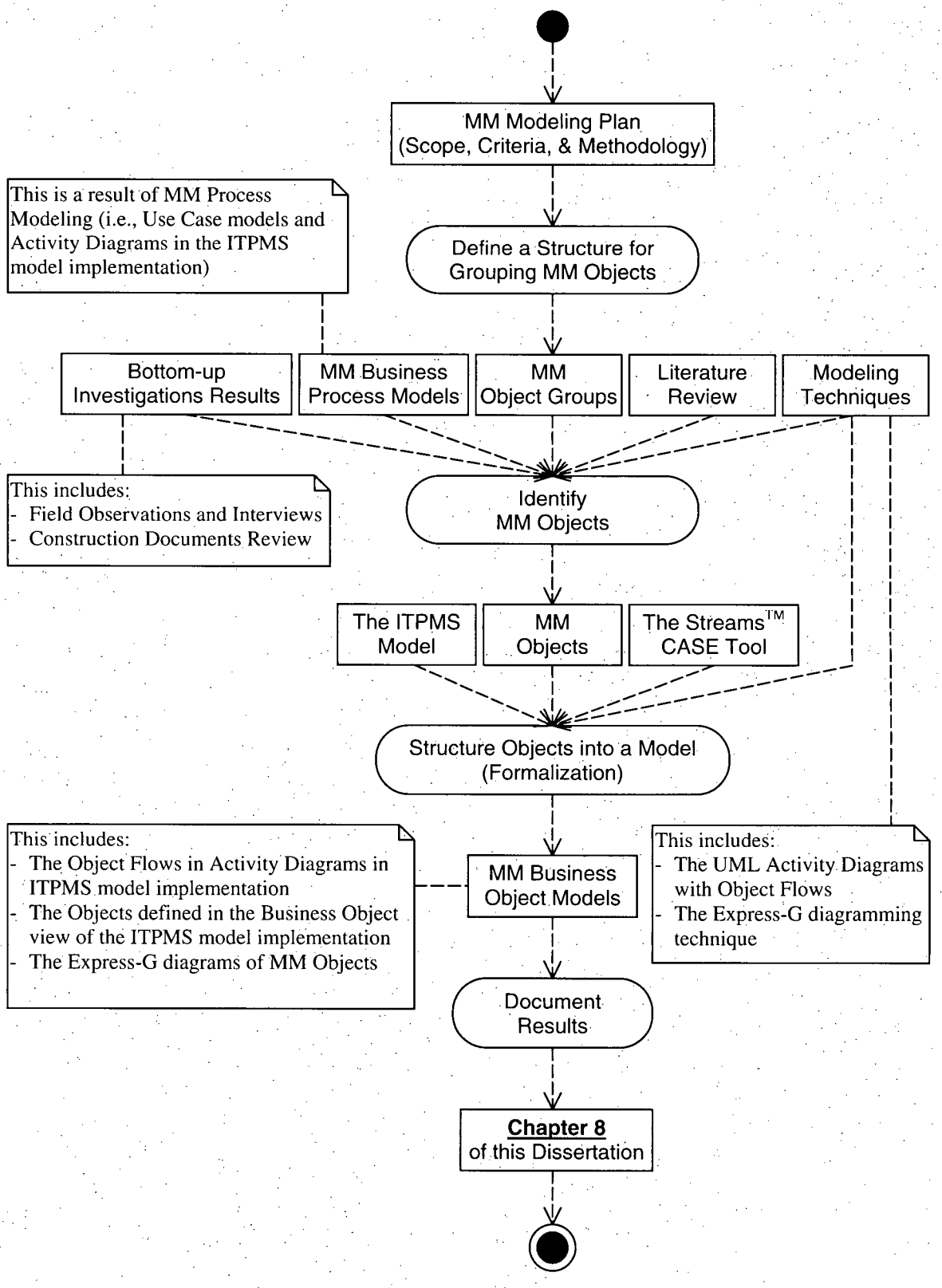


Figure 2–22: The Workflow Involved in MM Information Modeling

2.6.7 Summary of Materials Management Systems

This section concentrated on modeling MM systems in the context of CIC. It highlighted the importance of MM in construction projects and presented a review and discussion of MM literature and models. It also explained the scope and methodology of the research to modeling the functional and information requirements of MM systems. The presented materials are intended to serve as a background for the MM models developed in this research (presented in Chapter 8).

2.7 Chapter Conclusions

As a gate to the main body of this dissertation, this chapter elaborated on some of the knowledge areas of the research by explaining the related concepts, history, methods, techniques, and tools, and by presenting the research's views. Software development process and its workflows and artifacts (i.e., models) were described, and the emergence of the idea of integration of business process modeling into software development process was highlighted. It elaborated on the two major components of this integration (i.e., business process modeling and information modeling) in general and in the AEC/FM industry. It also presented an overview of major AEC/FM information and process models and MM literature and models. Reflecting on the issues associated with the models, the approach of this research to addressing them was then described.

The frameworks and models developed in this research (next chapters) reflect the research's journey through almost all workflows of the software development process in the context of construction PM with a special emphasis on the integration of business process modeling and software engineering. In complement to the critical literature and models reviews presented in this chapter, several bottom-up investigative activities were also performed. The next chapter elaborates on these activities and their results.

CHAPTER 3 Bottom-Up Investigations and Research Criteria

This chapter explains several bottom-up investigations that contributed to the understanding of the requirements for integrated project management systems in general and materials management in particular. "Bottom-up" in this context implies that the studies started with data collection activities relating to software systems and industry practice, and moved from these to more general analysis and design activities. These studies were critical in informing, providing the research criteria, and validating the later top-down activities that started with high-level conceptual models. The investigations include: 1) a review of current PM/CM software application systems, 2) software prototyping, 3) field investigations, and 4) a process modeling practice workshop. This chapter summarizes the studies and the conclusions drawn rather than detailing the work of each study itself.

3.1 A Review of Current PM/CM Software Application Systems

As a part of initial investigations, in March 2000, some of the commercial software application systems, specifically those intended to serve PM/CM functions, were reviewed mainly to appreciate the extent and quality of the coverage of PM functions by the systems. Many of these systems were examined by trying their full or demo versions, and others were reviewed through the information collected from their Internet sites or company brochures. Appendix B presents a list of some of these systems and their attributes. The major conclusions of this review were as follows:

- **PM/CM Functions Coverage Level**: The construction industry benefits from a variety of software application systems in handling projects. These systems range from general-purpose tools (e.g., word processing, image processing, accounting, drafting, and scheduling) to specialized tools (e.g., construction-oriented drafting and design, construction cost estimating and cost control packages). However, current systems fall short in supporting many other functions, including critical ones such as MM functions. With few exceptions for specific application areas (e.g., estimating materials cost), MM processes have not generally been supported in current systems. There is no software specifically designed to serve MM processes. Materials information is partially and separately modeled in various applications (e.g., materials cost in estimating and cost accounting software packages) with no data exchange possibility.
- **Software Integration**: Integration between current systems is generally minimal. Some computer programs can exchange data either with applications of their own type (e.g., using DXF file format in CAD programs) or with other specific applications (e.g., from CAD to estimating). In the latter case, the data is typically exchanged at a very low level; i.e., as basic numeric or textual data values rather than as semantic objects. The lack of software integration requires manual data transferring from one

application to the next, which makes the results prone to inaccuracy. To a large extent, this relates to the lack of an explicit semantic representation of information.

- **Explicitness of Information**: Most project information has not generally been represented explicitly. The product and material information, for instance, is represented by numbers and text (e.g., in specifications) and lines (e.g., in CAD systems) rather than as objects with any semantic meaning. The description of some material specified in a project specification document (e.g., in a word processing document) cannot be automatically and easily transferred to a quantity takeoff or estimating program and listed in a bill of materials. Likewise, the information cannot be later called and used in subsequent PM processes in an automatic, consistent manner. The information may not be understood as describing *material*, for instance, in materials ordering, receiving, storing, and testing applications. There is no consensus on project concepts and their relationships among the programs that use and manipulate them. In other words, there is no common language for communication among related applications. This necessitates human interpretations for data processing.

Although the examination of current software application systems did not generally add much to the list of general functions identified in the top-down investigations (Chapter 6)—due to the systems' lack of functional comprehensiveness, it highlighted the many areas of PM functions that are not supported by the systems; and thus, confirming the need for this research, which has attempted the formalization and standardization of information for the purpose of developing integrated total PM systems.

3.2 Software Prototyping

Software prototyping was used at the early stages of the research, prior to most of the information modeling work, in order to better understand and experiment with the nature of integrated materials management systems Appendix C provides a detailed description of the prototype system, Material Management System Tool (referred to as *MMS Tool* herein). The following presents a summary of the purpose and results of the prototype system development activity.

The MMS Tool was developed with two main purposes:

- To examine the data exchange capabilities of the IFC object model [IFC R2x, 2000a] for supporting MM functions in a distributed integrated PM systems environment, and
- To use the lessons learned from the prototype development as input for developing the research deliverables (i.e., as a bottom-up validation approach).

The result of developing the MMS Tool, which was based on the IFC object model, can be summarized as following:

- It generally validated the research hypothesis (i.e., the effectiveness of model-based PM integrated systems in supporting PM functions), and thus, it lead to the identification of the research objectives listed previously.
- It demonstrated that the IFC model falls short in supporting many MM functions, such as materials requesting, receiving, inspection and testing, storing, and so on. The major underlying reasons for this shortcoming was found to be:
  - The lack of representing the physical material items received, stored, and handled at the site (i.e., as viewed by construction trades and managers, for instance).
  - The lack of representing many materials-related transactions such as request for quote, request for materials, submittals, and so on.
- The complexity of MM processes was illustrated, specifically while designing the user interface elements and menu commands of the prototype; thus, showing the need for identifying and classifying the MM processes.

While the prototype system may not be treated as a major contribution of the research, it represents a significant research activity that acted as an evaluation/validation mechanism for the research's hypothesis and deliverables through identification of the modeling gaps in the IFC model and, thus, research needs.

3.3 Field Investigations

In order for the models of the research to be both realistic and useful, a study of actual field processes and interviews of field practitioners on a number of construction projects were included as part of the bottom-up investigations of the research. The results of these activities were used as input materials for forming and verifying the research's ideas, frameworks, and models. This section presents a brief description of the goal, objectives, and methodology of the field investigations, as well as an overview of the context and results of these activities.

3.3.1 The Goal and Objectives

To improve the pragmatic relevance of the work, the field investigations were performed with two basic objectives:

- To elicit process and information requirements of construction projects, in particular, as related to MM.
- To generalize the knowledge captured from specific field investigations into generic process and information models of the research.

3.3.2 The Methodology and Results

The field investigations took place from December 1999 to May 2001 on a number of construction projects carried out at the University of British Columbia (UBC) by the UBC Properties Trust. The projects were mainly intended to serve as educational, research, and housing facilities on UBC campus. In particular, the research focused on MM processes during the construction stage of a residential project (described in the next section).

Actual field processes and their links to other supporting processes (e.g., in the owner's organization: contracting, paying for invoices, etc.) were closely observed and documented. The data collected included photographs (about 1000 images), videos (about 7 hours), field documents (such as drawings, transactions, and materials tags and samples), observations of field meetings, and interviews with field practitioners (e.g., the project manager, superintendent, and subcontractors). As a result of these activities, a wide range of MM processes and their related information requirements were identified and studied. The processes studied included submittals, quoting, ordering (field purchasing and purchase agreements), receiving and storing, handling, processing (i.e., assembling and erecting), inspecting and testing (e.g., concrete, soil, and dampproofing), returning, reusing, disposing, commissioning, and occupation. Off-site processes (e.g., pre-and-post construction stages, materials manufacturing, off-site workshop work, and lab testing) were not within the scope of these field investigations; however, their links to the field processes were identified. For example, the sheet-metal subcontractor was interviewed to collect information about workshop-based steel duct activities.

Such processes were studied for a variety of materials, from the soil, sand, and gravels to the reinforcement, concrete, lumbers, masonry (i.e., bricks and concrete blocks), trusses, doors and windows, protecting membranes (e.g., air and vapor barriers, damp-and-water proofing, firestopes, and batt insulations), and mechanical and electrical materials and equipment. Many of these materials were observed from their ordering to their application and debris disposal.

The knowledge gained from the bottom-up field investigations was used in forming the research ideas, frameworks, and models. More specifically, the MM processes and their information requirements studied were generalized into the research models. Many of the documents (mostly project transactions and materials tags) were scanned and their images were used in the descriptions of the business objects defined in the MM models as a reference to the information contents of such documents (Chapter 8). The following sections describe the context and results of the observations in more detail.

3.3.3 The UBC Faculty and Staff Housing (Thunderbird) Project

The UBC faculty and staff housing (Thunderbird) project involved the construction of several residential buildings at the intersection of Thunderbird and West Mall, on UBC campus. Within its field investigation goals, the research conducted a close observation of the first phase of the project (*Thunderbird-1*), which involved the construction of two blocks of buildings (A and B) and their related site works (yards, walkways, streets and open parking, etc.). The study was performed from the excavation to the occupation of the buildings (January 2000 to April 2001). Table 3-1 presents a summary of some general information about the two buildings, and Figures 3-1 and 3-2 show their construction. The following sections briefly describe some of the aspects of the project as observed by the research.

Table 3–1: The General Characteristics of the Buildings of Thunderbird-1

Description	Building A	Building B
Construction Type	Woodframe townhouses, on slab-on-grade	Woodframe condominium, on one level of underground concrete parking structure
Number of Floors	2 (duplex units)	5 (4 residential plus 1 underground parking)
Number of Units	11	42
Approximate Area (Square Feet)	13,500	40,000 (excluding the underground parking)

3.3.4 CM/MM in the Thunderbird-1 Project

The Thunderbird-1 project was carried out by the UBC Properties Trust (the owner) using a (Construction Management) CM contracting approach, with Timberline Construction Group Western (TCGW) acting as the CM. An individual professional also played the role of owner's technical representative and control agent. The contract was based on a fixed management fee plus the superintendent cost and site operation costs (i.e., the labor and material costs incurred by non-managerial work performed by TCGW; e.g., miscellaneous works). TCGW produced a budget at the early stage of the construction phase and tried keeping construction costs as closely as possible to the budget.

TCGW played a major role in managing the project, which incorporated over 100 participants (i.e., architectural, engineering, and construction companies as well as materials and service suppliers and authorities such as the Ministry of Municipal Affairs), of which about 50 were major subcontractors (constructors and suppliers) who had been awarded major project's work packages. Coordinating with the owner, TCGW managed the selection and hiring of subcontractors most of whom got involved in the project through bidding and quoting processes. It was in direct contact with and managed all coordination among subcontractors, designers, control agencies, and the owner's representative and organization.

Figure 3–1: The Thunderbird Project, North East View of Phase 1, Building A (top left) and Building B (right), August 14, 2000.

Figure 3–2: The Thunderbird Project, North East View of Phase 1 (Building B, right) and Phase 2 (Building C, left), February 1, 2005.

Due to its CM role, TCGW appeared central to project MM. The owner generally assumed the procurement responsibility for most project materials, and subcontractors were mainly involved in the processing and installation of materials. On behalf of and in coordination with the owner, TCGW directly managed the procurement of major bulk materials (e.g., gravels, sand, concrete, bricks and concrete blocks, forming and structural framing lumbers, and drainage and sanitary systems materials), engineered materials (e.g., air conditioning unit), and fabricated materials (e.g., doors, windows, and trusses). It acquired and provided required materials based on the project schedule and the requests received from trade foremen (usually verbally). Other specialty materials, however, were included as a part of the subcontractors' contract (e.g., galvanized nails for the wood framing trade; glues, fasteners, and caulking materials for the finishing trade).

Project contracts and purchase orders were generally managed by TCGW. After bidding and quoting processes, most major contracts were produced by TCGW and were sent to the owner and the subcontractors, in sequence, for signing (e.g., Figure 3-3). The miscellaneous contracts (e.g., equipment rentals—heaters, electronic hammers, etc.—and fuel), on the other hand, were mostly generated and signed by related subcontractors and sent to TCGW for the owner's signature. Depending on their importance (i.e., criticality, volume, etc.), the contracts were coordinated and managed at various levels of TCGW's project organization. For instance, most equipment-rental contracts (e.g., heaters for drying external stud walls and meeting the 19% moisture content threshold for batt insulation) were produced by the rental company, after the verbal purchase orders placed by the superintendent or his assistant. At the payment stage, all construction invoices were received at the TCGW's main office, checked against deliveries by the office and the superintendent, and paid by the owner.

Except for drywall debris, all materials disposal was generally managed by TCGW. A disposal company had been contracted for delivering a disposal bin and disposing its content when filled with the various subcontractors' materials debris. All trades had to do their own cleanup and dump their wastes into the bin. The drywall contractor had to manage disposing gypsum debris himself. Materials debris disposal was one of the items discussed in most site coordination meetings.

The project's centralized owner-based procurement mode had both advantages and implications. For example, it provided potentials for cost saving (e.g., though elimination of middle persons) and time saving (e.g., by reducing the volume of submittals and related approval and coordination processes). However, it introduced more risks to the owner side, since any shortcoming (e.g., unavailability of the right material at the right time and location) could be blamed on the owner. Therefore, it required a more rigorous MM organization and procedures to be established by TCGW.

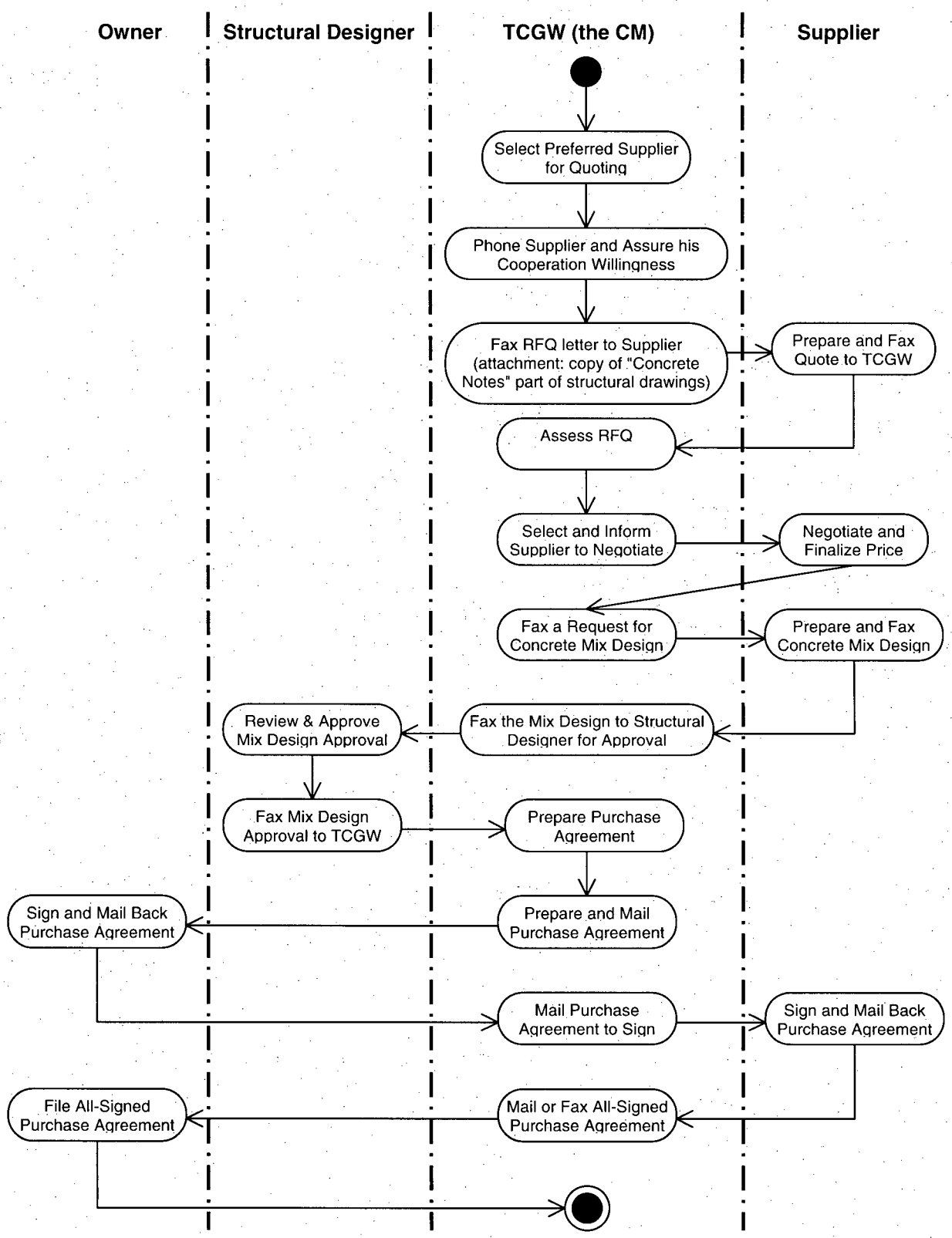


Figure 3–3: Concrete Supplier Selection and Contracting Processs in Thunderbird-1

3.3.5 An Overall Assessment of the Thunderbird Project

Like any other construction project, the Thunderbird-1 project exhibited both positive and negative characteristics, some of which are summarized as follows:

Teamwork Relationship: A good teamwork relationship was observed within and between various organizations involved in the project: the owner, TCGW, and subcontractors. In order to keep the project within the budget and schedule, TCGW had to be very proactive in several ways. One of the very obvious areas that positively impacted the project was the close, trusting, cooperating relationship established between TCGW and the subcontractors. For example, it happened many times that TCGW's personnel helped in unloading subcontractor's materials or equipment. To some extent, this relationship was found to have roots into previous work experience of TCGW with many of the subcontractors on previous projects. While the cooperating relationship between project participants was a positive driver for productivity improvement during construction, it may not address or replace the level of formality that may be required during the construction and post-construction stages of project.

Materials Receiving: The materials receiving function was not generally sophisticated enough to provide for effective receiving MM processes. TCGW was more concerned with owner-procured materials, and did not really get involved in materials processes of the subtrades that had to supply their trade materials. An example of a loose receiving observed on the site related to a load of reinforcement: The reinforcement delivery truck arrived at the site. The superintendent assistant directed the truck to an open area for unloading. He drove a Bobcat lifter to the area, unloaded the load, and the truck left the site without going through any formalities (Figure 3-4). Returning the wrong doors to the supplier was also observed on the site. The finishing subcontractor discovered the problem at installation time. The doors had, in fact, been procured, received, and stored by the CM's personnel at the site. The loose receiving function is a major contributor to ineffective inventory tracking and control, and thus, poor project planning and scheduling and wastes generation.

Materials Labeling and Storage: Many materials such as doors and windows had usefully been labeled by their suppliers to include construction information (in addition to manufacturing information) such as their types and installation locations, so that they could be easily identified at the time of unloading, storing, and installation (e.g., Figure 3-5). This arrangement prevented unnecessary materials handling during construction. The nature of the bulk materials such as sand, gravels, bricks, and concrete blocks (which normally loose their identity in the final product) might not allow for their identification as such, but this labeling limitation had been overcome in some instances. Packaging materials for lumber, for instance, had been labeled at the time of ordering by supplier, based on the location (i.e., building) they were intended to be used, so that they could be unloaded as closely as possible to their final use (e.g.,

Figure 3-6). Though their exact unloading location was decided at the receiving time. Identifying and tracking materials with such construction labels or barcodes provides for MM improvements in productivity (e.g., through saving time), budget (e.g., through saving resources), and quality (e.g., through prevention of possible damages resulting from multiple movements of materials).

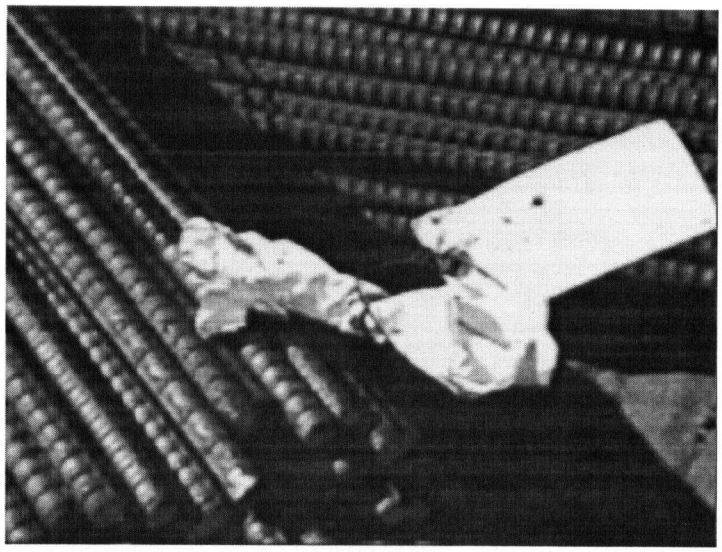


Figure 3–4: Reinforcement Delivery Ticket Strapped and Left in the Open Laydown Area

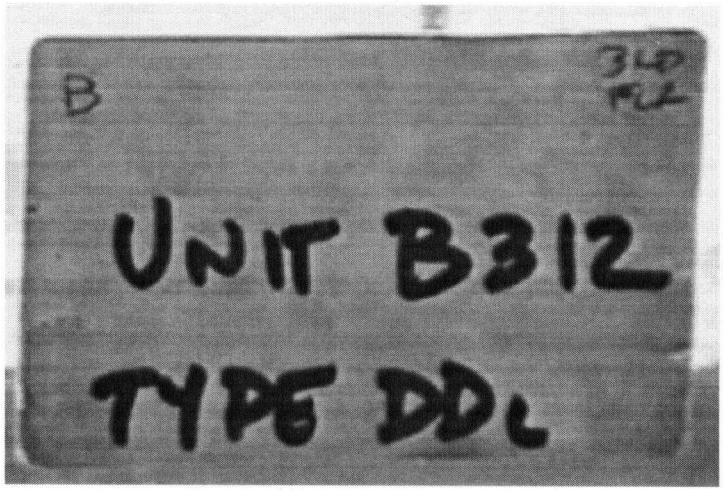


Figure 3–5: An Example of a Label Stuck on Window's Glass (in the 3rd Floor, Unit B312)

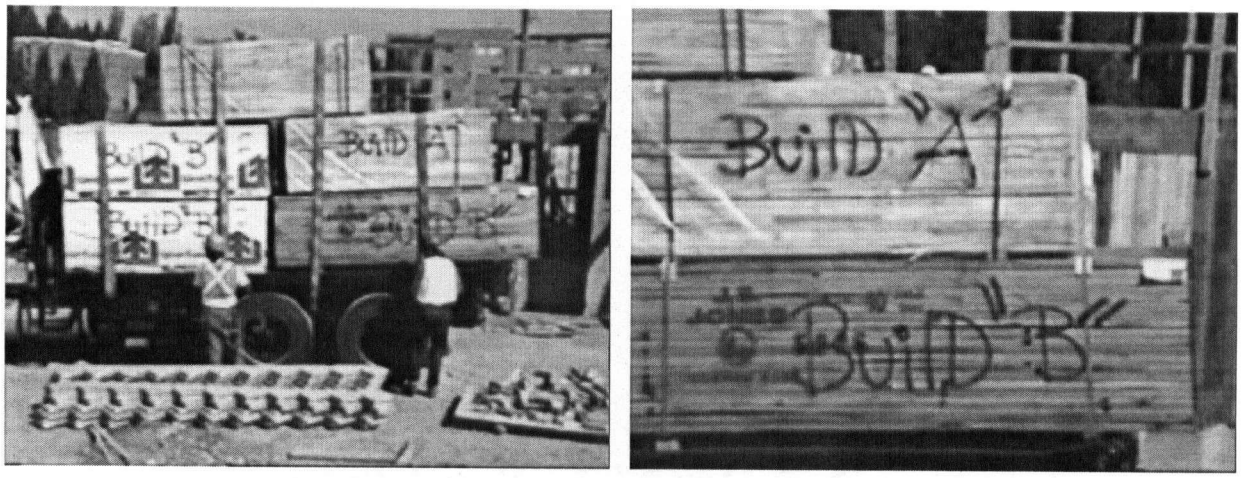


Figure 3–6: Unloading Labeled Lumbers

Construction Rework: There were a number of instances of rework generated as a consequence of lack of necessary coordination of various design disciplines. Figure 3-7 illustrates one of the examples of such deficiencies, which involved other preceding processes such as Request-For-Information (RFI) handling and resulted in an extension to the project budget and schedule.

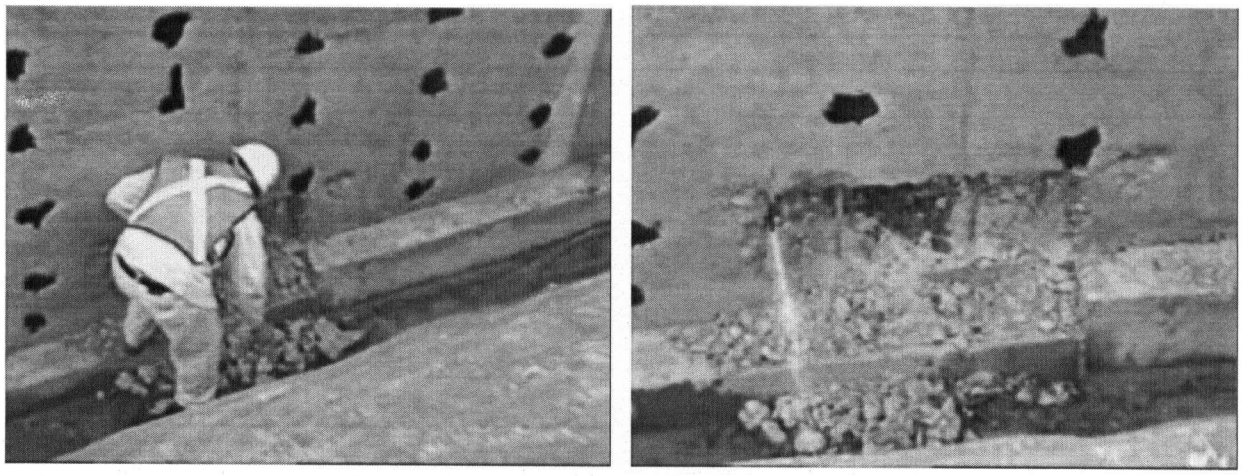


Figure 3–7: Cutting the Foundation Wall to Run Electrical and Communication Cables

Change Orders: Project change orders (CO's) ranged from site works to the wood structure framing (e.g., resizing all 102 indoor door openings of building B, ordered after completion of the structure), doors and their hardware, windows, light and bath fixtures, lockers, painting, and so on. They affected 37 (out of about 50) major contractors and suppliers of the project. The number of companies affected by individual CO's varied from one to six, with the mechanical subcontractor affected more frequently than others (i.e., by 19 CO's). Most CO's involved materials changes.

Among the 53 CO's generated during the project (excluding change orders contemplated but not executed), only 7 involved cost savings (a total of about 1.0% of the budgeted cost), and the rest

originated from additions, upgrades, and design deficiencies. While the overall direct cost effect of CO's (e.g., materials and labor costs increase) might not have been overly significant (about 3.8% of the budgeted cost), the volume of coordination efforts and paperwork as well as the time spent in handling the CO's was generally considerable. The schedule extension from nine months to fourteen months may be an indicative of inefficiencies and rework, including those resulted from the CO's.

Overall, the construction of Thunderbird-1 finished with no major budget overrun, accidents, or litigation. To a large extent, this could be related to the open and trusting working environment created by the project's management team (especially the CM). However, the level of formality (e.g., in terms of organization, procedures, and total project control) was not found to be at the level required to thoroughly satisfy all project objectives (e.g., time, quality, and productivity). Greater formality could have been achieved through the use of integrated PM systems.

3.3.6 The Use of Information Technology

The projects studied, especially Thunderbird-1, did not generally make good use of current Information Technology (IT). The use of IT was mainly limited to CAD systems for 2D drawings (by the architectural and engineering firms involved); word processing and spreadsheet applications for correspondences, specifications preparations, and data recording and analysis; scheduling packages (e.g., MS Project) for project schedule presentation (not more than a bar chart); and accounting packages for cost accounting and control purposes. The exchange of information were mainly paper-and verbal based, and communications were largely based on using telephone, fax, and couriers. Email messages (usually on a one-to-one basis) were also being used occasionally, due to the lack of trust to their formality. The use of IT was generally minimal, especially on the site, where telephone and fax were almost the only media for communication and transferring information (e.g., Figure 3-3). As a result, except for design documents (i.e., drawings, specifications, etc.), project information was scattered throughout organizations in the form of paper-based transactions, hand-written fax transmittals, the superintendent's diary or notes, verbal communications, and so on.

The fragmented processes and paper-based documentation and communication of information may have adversely impacted the performance of the projects observed, especially Thunderbird-1. The poor documentation of project events and transactions together with improper document handling and filing could impose the risk of unavailability of information in the next stages of project life cycle. An informal receiving process (Figure 3-4), for example, could hinder many processes such as materials inventory management (as to know what is where), invoice processing (i.e., checking invoices against receiving), and possible litigation resolutions (e.g., between the owner, contractor, and supplier). As another example, relying on human interpretation of data and manual transferring of data from one

document to another (e.g., Figure 3-8), possibly with improper documents handling (e.g., Figure 3-9), increases the risk of inaccuracy and loss of data and time and cost escalation in the project. Moreover, the owner's operation and maintenance body may not receive all information required for post-construction processes. Such risks could have been reduced with integrated PM/MM systems.

Figure 3–8: Concrete Tester Transferring Information from a Delivery Ticket to his Concrete Test Report

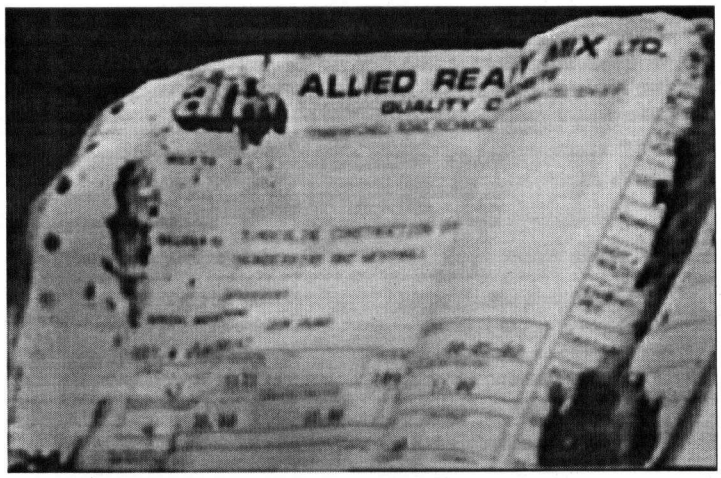


Figure 3–9: Concrete Delivery Ticket Covered with Patches of Concrete

3.3.7 Major Outcomes and Lessons Learned

The field investigations provided values for the research in many ways. Many MM functions and their related information requirements were closely observed by walking through processes and attending field meetings. This resulted in a body of knowledge recorded in the form of movies, images, transactions

copies, diagrams, and notes. The follow-up interviews with field practitioners further resulted in clarification and verification of the knowledge.

One of the most important lessons affirmed in the field investigations was the fact that MM processes observed reflected the various project's contractual, organizational, procedural, and technological constraints. More specifically, process sequencing was generally found as being context dependent. For example, not all materials go through the same sequence of processes. A batch of concrete received at the site, for instance, requires its sampling and testing shortly after its receipt and before its pouring. The process, as practiced, involves a direct input of information from the delivery ticket to the concrete test report by the tester. Door frames, on the other hand, may not go through such a process; their delivery is usually carried out independent of their installation inspection. In concrete testing, the information about the supply-end (e.g., the supplier, batch and location information) could automatically be linked and retrieved into the test report, instead of manual input, as is usually practiced.

Therefore, it was learned that, in the business process modeling activities of this research, it would be tremendously difficult (if not impossible) to define all types of PM processes for various project contexts in detail; however, it would be possible and even more useful to define independent high-level generic processes that could be sequenced as dictated by contextual and managerial constraints. Among the solutions found for defining such generic processes was an approach of ignoring the "technology" (e.g., the tools and media—such as fax, phone, email, etc.—used for communicating and exchanging information) and eliminating "actors" (i.e., role players: owner, supplier, superintendent, etc.) in process models and focusing on the processes (i.e., actions) themselves. This is elaborated more in next chapters.

Finally, the study of filed processes generally revealed that great potential exist for productivity and quality improvements, and thus, time and cost savings (though not quantified, due to the scope) in construction projects. As suggested in the preceeding sections, such improvements could be achieved through the integration of MM processes, especially using computers. The confirmation of such potential further highlighted the significance of the research deliverables aimed towards the improvements.

3.3.8 Summary of Field Investigations

The bottom-up investigation of construction fields was performed in order to improve the pragmatic relevance of the research deliverables (specifically MM process and information models), and to formalize PM knowledge for the purpose of development of integrated total PM systems. This section presented a brief description of the goal, objectives, and methodology of the field investigations, as well an overview of the context and results of these activities.

Overall, the investigations confirmed many of the issues associated with construction projects; e.g., the low level of IT use, reliance on verbal communication and paper-based documentation and exchange of information, and fragmentation of processes and information. Moreover, great potentials for improvements through computer-based integration of processes were also realized. Central to these improvements was found as being the integration of MM functions.

The investigations also resulted in a collection of images, video movies, field documents (such as drawings, transactions, and materials tags and samples), and diagrams and notes describing processes and their information requirements in the projects. The knowledge gained in the investigations was later generalized into the research models, in which the collected images and documents were also usefully utilized (next chapters).

3.4 The Process Modeling Practice Workshop

As a part of the bottom-up investigations of the research, the potential uses of $IDEF_0$ [$IDEF_0$ 1981; Marca and McGowan 1993; Feldman 1998] for modeling MM processes was examined in the form of a Process Modeling Practice Workshop (PMPW). The workshop was held in a graduate course ("*Project Management for Engineers*"), offered by Dr. Thomas Froese of the Construction Engineering and Management Group of the Department of Civil Engineering, UBC [CIVL-523, UBC 2002]). The lessons drawn from this workshop were used as input materials to forming and verifying the research's ideas, frameworks, and models. This section briefly describes the goal, objectives, and methodology and results of the workshop. More detailed information and background knowledge related to this study are presented in sections 2.2, 2.3, and in Ghanbari and Froese [2004].

3.4.1 The Goal and Objectives

The goal of the workshop was to experiment with a comparative approach to business process modeling in the context of construction MM. Within this goal, two basic objectives were assumed:

- To experiment with capabilities of the $IDEF_0$ technique and observe how consistently different domain experts would model MM processes using this technique.
- To apply the lessons learned from the experiment in forming and verifying the research's ideas, frameworks, and models.

3.4.2 The Methodology and Participants

A total of twenty-nine graduate students (called *participants* or *modelers* hereafter) were grouped into ten modeling groups based on their education, knowledge, and experience (in the AEC/FM industry and in software development), which were captured using questionnaires. Overall, the levels of theoretical knowledge and practical experience of the participants were found reasonably high, and many of them had already taken some courses in PM and in computer application development (in the AEC/FM and other domains), which usually include process modeling as a part of system analysis in their syllabus.

Prior to the modeling session, an instructional session was arranged with two main purposes: 1) refreshing the participants on the basics of the $IDEF_0$ and node tree techniques; 2) providing a general description of the modeling task to the participants. Participants were then asked to draw a breakdown and sequencing of the processes involved in the MM functions of construction projects in the standard $IDEF_0$ diagram forms provided to the groups. The modeling session, which took over an hour, resulted in ten different models.

The models were reviewed and analyzed by capturing their quantitative and qualitative characteristics into a spreadsheet format. A description of the models produced in the workshop along with a discussion of the models and their associated modeling issues is presented in the next section.

3.4.3 Results and Lessons Learned

Table 3-2 presents a summary of some quantitative characteristics of the models, while Table 3-3 provides some information about the contents of the models. In the latter table, the general format of "P1 (P11, P12, ...); P2 (P21, P22, ...); etc." has been used to represent the sub-processes defined at a given level of process hierarchy of a model. The following briefly explains some of the aspects of the models and discusses some of the modeling issues associated with these specific models and with the $IDEF_0$ technique in general.

A detailed discussion of the models is presented in Ghanbari and Froese [2004], however, a brief explanation of the major issues observed follows. First, despite the given instructions, not all groups used $IDEF_0$ as their technique for producing their models. Among other techniques used were the *node tree* technique (representing quick process-breakdown hierarchies) [Feldman 1998], and the *flat, box-and-arrow flowchart* technique (representing workflows in one or more independent drawings). Apparently, the modelers had found the flowchart technique more suitable for modeling.

Second, closely related to the modeling technique issue (i.e., the flowchart technique preference) was the $IDEF_0$'s lack of full support for representing *conditionality* (decision points) and *concurrency* of processes. It is not uncommon, especially in AEC/FM processes, that the start of an activity is subjected to the satisfaction of a condition or criteria (e.g., order materials when the inventory falls below a certain level; approve document if it satisfied the requirements; etc.). Moreover, $IDEF_0$ does not offer any mechanism for modeling concurrency of processes belonging to different parents (i.e., diagrams).

Third, *inconsistency of process sequencing* was observed in models. Most models suggested dissimilar sequencing of processes (e.g., *request-order-receive-distribute* sequencing versus *order-receive-store-request-distribute* sequence). Moreover, a majority of the models (more than 90%) considered a straightforward sequencing of processes in their $IDEF_0$ diagrams and did not include any feedback loops between processes. Others were not consistent in using such feedback loops (in terms of the involved processes and their levels).

Fourth, the issue of sequencing was found to have a close link to the *viewpoint* issue. Almost all of the produced models generally represented a project manager's viewpoint, however not all diagrams of a model reflected this viewpoint thoroughly and consistently. In order to model MM processes more comprehensively, most models had to shift from one viewpoint to another (e.g., supplier, contractor,

project manager, manufacturer, etc.) so that they could include a wider range of material-related processes (e.g., the *Quote*, *Contract*, and *Manufacture* processes in one model, indicating a shift from *Project Manager* to *Manufacturer* viewpoint). This aspect of the models (i.e., a model reflecting multiple viewpoints) was, of course, in disagreement with the recommendation made to restrict an $IDEF_0$ model to only one specific viewpoint [Marca and McGowan 1993; Feldman 1998]. Having produced separate models to reflect the various viewpoints involved in construction MM, the linkage among such models would still remain to be an issue. $IDEF_0$ does not offer any mechanism to support integration of multiple viewpoints.

Fifth, a number of models were found to repeat one process (e.g., *Record/Document/Update*, *Request*, and *Approve*) in different parts of a model. The *repetition of processes* within and over models (of different viewpoints) is very common, but it is not supported by the $IDEF_0$ technique. This technique is strictly hierarchical in nature, and every process is uniquely identified by its id (or name) and cannot be repeated in several diagrams. In other words, the technique does not provide for *reusability of processes* in a model and over multiple models.

Finally, the models mainly focused around MM's common chain of processes (i.e., requisition, purchasing, receiving, and storage), and many other MM process areas (such as materials planning, inventory management, inspection and testing, and expediting) were left unexplored. The duration of the workshop was a major reason for this outcome.

The above listed some of the major modeling issues observed in the PMPW. The observation confirmed some of the $IDEF_0$'s potentials and limitations that are elaborated in the literature [e.g., Noran 2004]. The collection of the technical issues identified in the workshop lead the research to search for potentials of other alternatives (such as the UML) [UML 1.4, 2001] as well. Moreover, the extent of coverage of MM processes by the models produced raised a research question as how a full range of MM processes can be identified and modeled.

3.4.4 Summary of the PMPW

Like any other modeling technique, the $IDEF_0$ technique has it own applications, advantages and limitations. $IDEF_0$ has been widely used by software development projects for modeling business processes. In comparison with the flat, box-and-arrow flowchart technique, $IDEF_0$ allows modeling of more dimensions of processes (e.g., inputs, constraints, outputs, and resources). As a diagramming technique, it also allows for dealing with complexity through hierarchical breakdown and visualization of processes as well as the viewpoint mechanism.

This section briefly described the goal, objectives, methodology, and a discussion of the models produced in the PMPW, which was held as a part of the early bottom-up investigations of this research to examine the potentials of using the technique for construction MM process modeling. Overall, the results of the workshop confirmed some of the potentials and limitations associated with the $IDEF_0$ technique. The technical limitations observed include lack of support for conditionality, concurrency, synchronization, integration of models of multiple viewpoints, and reusability of processes. To these limitations may also add the difficulty of integration of $IDEF_0$ process models with data models (prepared in EXPRESS or EXPRESS-G, for example). While the latter limits the traceability and maintainability of the two types of models, the combination of such limitations may adversely impact the effectiveness of the process models produced for the purpose of integrated total PM systems development. The results of the workshop helped the research in two major ways. The $IDEF_0$'s technical issues lead the research to search for other alternative modeling techniques such as the UML, which offers solutions to such limitations (Appendix A) and better satisfy the research's modeling criteria (sections 2.5.2 and 3.5). Moreover, the extent of coverage of MM processes by the models produced raised a research question as how a full range of MM processes can be identified and modeled.

Table 3–2: The Quantitative Characteristics of the Models

Group No.	Technique used			No. of Pages	Levels of Breakdown	No. of Sub-Processes of a Process		Domain	
	NT	PS	OT			Min	Max	General	Specific
G1	1	1		3	2	2	3	1	
G2			1	5	1	N/A	N/A		1
G3	1	1		5	2	3	5	1	
G4		1		4	2	2	5	1	
G5	1		1	2	3	2	4	1	
G6		1		5	2	3	4	1	
G7		1		6	2	3	5	1	
G8		1		2	1	4	4	1	
G9	1	1		3	3	3	4	1	
G10	1		1	3	2	2	5	1	
All Results	5	7	3	4	2	2	5	9	1
	Total	Total	Total	Average	Average	Minimum	Maximum	Total	Total

Notes: NT = Node Tree, PS = $IDEF_0$ Process Sequencing, OT = Other Techniques

Table 3–3: Process Breakdowns in the Produced Models

Group No.	1st-Level Processes	2nd-Level Processes	3rd-Level Processes	Elements (ICOM)
G1	Analyze, Procure, Consume	Analyze (Identify, quantify, Schedule); Procure (Marketing, Transport); Consume (Store, Count, Order).		Drawing, Schedule, Info, Material, Price, Quantity, Vehicle.
G2	Procure, Deliver, Store, Receive, Record, Distribute, Utilize, Monitor Waste, Monitor Productivity			Material, Labor, Equipment, Cement, Water, Availability, Quantity, Price, Hand Cart, Fork Lift, Un-Used Cement.
G3	Supply, Storage, Quality Control, Distribution, Financial	Supply (Purchase, Receive, Document); Storage (Receive, Unload, Store, Document); Quality Control (Check Quality, Test, Document); Distribution (Request, Distribute, Transport); Financial (Request Payment, Check Quantity, Pay).		Storage, Equipment, Staff, Lab.
G4	Procure, Quality Checking, Store, Control Inventory, Utilize	Procure (Search Products, Compare Prices, Place Orders, Purchase); Store (Determine Storage Requirements, Store); Control Inventory (Prepare Catalog).		Performers: Buyer, QA Dept., Warehouse Manager, Contractor, Administrator.
G5	Procure, Store	Procure (Determine Requirements, Order); Store (Inspect, Record, Deliver to Site).	Determine Requirements (Locate, Plan Delivery, Quantify Requirements, Determine Specs & Standards).	Information, Material, Computerized Systems.
G6	Estimate Quantity, Evaluate Quality, Procure, Issue	Estimate (Obtain Drawings & Specs, Take-off Dimensions, Calculate Quantities); Evaluate (Obtain Samples, Test Samples, Select); Procure (Order, Deliver, Inspect, Store); Issue (Check Schedule, Determine Daily/Weekly Usage, Determine Delivery Method, Update Records).		Drawing, Specs, Assignment, Estimator.
G7	Plan, Procure, Receive, Store, Utilize	Plan (Review Drawings & Specs, Prepare BOM, Submit BOM for Approval, Document); Procure (Select Supplier, Sign Contract, Deliver); Receive (Inspect, Test, Sign Receipt, Record); Store (Classify, Warehouse Register, Label, Protect); Utilize (Approve for Use, Utilize, Return Unused, Record).		
G8	Estimate Quantity, Order, Receive, Monitor Usage			Design, Work Scope, Quantity, Project Schedule, Bulk Material, Stored Material, Rate Of Usage, Productivity, Inventory, Quantity Surveyor, Supplier, Shipping Company.
G9	Plan, Order, Store, Use	Plan (Design, Take-off Quantity, Approve, Procure)	Procure (Get Quote, Contract, Manufacture)	
G10	Plan, Quantity Takeoff, Quote for Price, Purchase, Deliver, Consume	Plan (Determine Timing, Determine Delivery); Quantity Takeoff (Find materials from Specs, Find Quantities from Drawings); Quote for Price (Review Alternative Materials & Prices, Get Quotes); Purchase (Purchase, Delivery, Store); Consume (Review Quantities, Review & Update Schedule).		

3.5 Setting Research Criteria

The results of the bottom-up investigations (previous sections) and the literature and models review (Chapter 2) helped set evaluation criteria for both selecting modeling techniques and tools (section 1.4.3.2) as well as shaping and evaluating the contents and functionality of the frameworks and models of the research. The following generalizes the modeling criteria established for the research.

3.5.1 General Modeling Criteria

The research deliverables were generally sought to have several characteristics:

1) **Scope and Objective**: The scope and objective of each model should fit within the overall scope and objective of the research, and should reflect the notion of "integrated total PM systems".
2) **Comprehensiveness**: Each deliverable should include a wide range of the elements and views concerned in its scope and objective.
3) **Integration**: The elements of each model should be integrated.
4) **Neutrality**: Each model should be generic enough to be adapted to any situation within its scope. The model should be independent of any specific context (e.g., contracting strategy, organization setup, and construction method, implementation method).
5) **Realism and Usefulness**: The models should portray the real practice and should ultimately be useful for AEC/FM application systems development. The bottom-up investigations were generally intended to enhance this characteristic of the models. The easing of model management should be one of the major consideration of the models.
6) **Modularity and Reusability**: For the sake of efficiency and effectiveness, each model should incorporate the notion of *reusability* and avoid repetition and redundancy to its possible extent.
7) **Flexibility and Openness**: The models should be evolutionary; i.e., they should openly and flexibly accommodate future changes.

3.5.2 Flexibility as a Basic Requirement

Models of AEC/FM processes and information should be flexible enough to accommodate complexities and future changes for a number of reasons. First, no single model may be envisioned by project functional bodies (section 8.1.2) or by different modelers, mainly due to the diversity of interests, views, and personal experiences (section 3.4.3). Second, AEC/FM processes are not handled uniformly and constantly at all times mainly due to contextual factors such as environmental, social, and economical constraints, contacting strategies, management practices, as well as technological factors (e.g., handling a process based on available resources and tools). The introduction of new technologies, especially in

materials and methods of construction, is a major factor that impacts the various dimensions of construction projects. Specialization of tasks and formation of new trades are examples of such impacts.

Models of information systems should be *generic* enough to accommodate possible *variations* in business practices. Such a requirement may be explained in the context of the materials planning process, where the procurement, logistics, and field control responsibilities of the owner, contractors, and subcontractors are defined. "It is not unusual for the owner and engineer to assume the responsibility for procuring major engineered equipment (identified by a unique number on design drawings). Occasionally, however, materials responsibilities are divided by management function, that is, the owner may wish to purchase all materials and select the vendors, while assigning the contractor the responsibility for takeoff, expediting, and field control" [Bell 1989, p. 455].

The process of transferring goods to the customer may be used as another example in this regard. In some instances, an *invoice process* can later evolve into the adoption of a *shipping notice process*; i.e., a *shipping notice* may play both roles of *noticing* and *invoicing*. A business model, for instance, must be flexible enough to accommodate such a *process evolution*; i.e., it should offer the choice of one of the two processes. A process model should be flexible to any future changes such as adding and modifying processes and, more importantly, detailing processes down into more specific processes.

Flexibility is required to avoid major reworks resulting from possible changes and evolutions, but it should not outweigh the importance of formalization and standardization, which should be kept as the ultimate purpose of process and information modeling. Therefore, the challenge remains in having both flexibility and standardization satisfied.

3.5.3 Summary of Research Criteria

This section presented an overview of the major criteria set for the research deliverables. In addition to the aforementioned general criteria, the research analyses identified some more specific requirements—e.g., for AEC/FM process modeling (section 2.5.2) and for AEC/FM information modeling (Chapter 4). The general criteria constrained both the research models and the selection of modeling techniques and tools used in the research. The techniques and tools had generally to provide for efficient and effective communication, documentation, and specification of various models, including the research models. Section 1.4.3.2 explains how this objective was achieved. Chapter 9 futher presents an evaluation of the research models against the general modeling criteria and the specific requirements.

3.6 Chapter Conclusions

Several bottom-up investigations were performed that supplemented the literature search to provide a solid understanding of the issues addressed in this research and industry data with which to formulate solutions. These investigations include: 1) a review of current PM/CM software application systems, 2) software prototyping, 3) field investigations, and 4) a process modeling practice workshop.

The results of all these activities helped to identify AEC/FM modeling requirements and to set the research criteria. They also helped the research better focus on real-world problems as well as better shape the research's ideas, frameworks, and models. This chapter presented an overview of the bottom-up investigations and the research criteria. In complement to these materials, the next chapter presents top-down analysis of AEC/FM project information modeling requirements.

CHAPTER 4 AEC/FM Project Information Modeling Requirements

This chapter examines the level of support for PM functions by current AEC/FM information models and lays out the requirements of an AEC/FM project information model. It addresses the analytical part of the research question (section 1.4.1): *What are the information requirements of PM functions*? Focusing on product modeling, the first section formulates product information, discusses how such information is viewed, shared, and used by PM functions, and compares how current AEC/FM product models encompass such information requirements. Stepping beyond product modeling, the second section discusses a broader range of project information modeling issues and requirements and their level of support by current models.

In general, this research argues that each of the functions of PM involves a distinct view of project information, yet it is possible to bring these views together into a unified project model encompassing both product and process information. The top-down analysis presented in this chapter (i.e., working from a collection of basic concepts and issues towards specific model requirements) complements the bottom-up analyses described in previous chapter.

4.1 AEC/FM Product Information Modeling

This section explores product information, how it is modeled in existing AEC/FM product models, and how it relates to non-product information.

4.1.1 Product-Related Concepts: Target Product, Space, and Material

Product, *material*, and *space* are three integral concepts in AEC/FM product models. Products are made of materials. The concept of *material* is, therefore, an immediate concern in designing and developing a product. A product is usually thought of as a physical, tangible thing. For buildings, this concept can be refined to distinguish between physical (*tangible*) products and *spaces* defined or occupied by the tangible products (e.g., a set of rooms containing some shared walls, columns, and doors). Spatial information is at the core of the architectural design process, which involves translating required functions into spaces bounded by and containing physical components. Target products (e.g., a building, a wall, or a road), spaces (e.g., a zone, a floor, or a room), and materials (e.g., bricks, concrete, lumbers, or gravels) are all physical things whose relationships may be conceptually represented as in Figure 4-1.

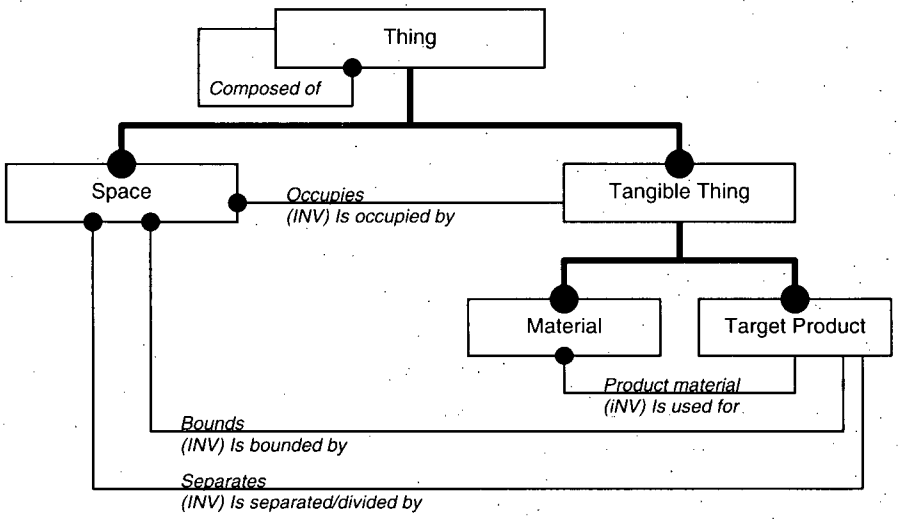


Figure 4–1: A Conceptual Representation of Product-Related Concepts

4.1.2 Form, Function, and Behavior/Performance

For any product (e.g., a building), might want to know: 1) *What* it is, and what it looks like; 2) *Why* it exists (referring to its role, purpose, and function); 3) *How* it functions (i.e., the way it performs its role). These three types of interrelated product information may be expressed as three basic aspects of the product: Form, Function, and Behavior (or Performance); which are referred to as the FFB/P framework in this dissertation (Figure 4-2).

1) **Form:** the geometric information (i.e., dimensionality—size, height, length, etc.) and topological information (i.e., the physical structure and configuration—inclusion, intersection, overlapping, adjacency, and composition concepts).

2) **Function:** the *activity, purpose,* or *use* of the product [Oxford Dictionary 1989]. This relates to the *design intent.* Examples are a building designed to function as a *school*, a wall with a *load-bearing* function, or a wall *separating* two spaces.

3) **Behavior:** a "way of acting or functioning", essentially synonomous with *performance*: "process or manner of performing (i.e., working or functioning)" [Oxford Dictionary 1989]. Behavior or performance of a product refers to the *way* the product performs its functions.

There is a strong relationship among the three aspects. Examples are: room temperature required for a specific activity (i.e., behavior-function relationship), size and shape of a beam and its structural strength characteristics (i.e., form-behavior relationship), a circular-shape room suitable for a meeting (i.e., form-function relationship). However, in many instances, the study of such relationships may not be possible without consideration of the *environment* in which the product is situated.

4.1.3 Products and their Environment: The Concept of Agent

A major part of decision-making processes in AEC/FM projects involves analyzing the relationships between products and their environment (e.g., the behavior or performance aspect). A product, in general, is in an action-reaction relationship with its external environment. This relationship is normally established between the product and some *agents* that exist in the environment. An agent could be anything that surrounds and impacts, and/or is impacted by, the product and is relevant to the situation (Figure 4-2). Agents may be classified into four major groups:

1) **Natural Agents:** This type of agents can include earthquake, wind, rain, sun, moisture, etc.
2) **Product Agents:** Products themselves can act as agents. For example, the structural analysis of a column may require considering the loads imposed by its connected beams. Another example is the impact of the weight, volume and/or moisture content of a bulk of material on its storage or container.
3) **Resource Agents:** Any process-oriented object involved in the development of a product can be considered within this category. Examples are project participants (e.g., project manger, designer, and contractor), equipment, and tools. The selection of construction equipment with consideration of site and spatial conditions is an example of the impacts of products and resource agent on each other.
4) **Other Agents:** Other types of environmental agents may also be considered. Examples include social and economical factors considered in a development process.

The need for inclusion of *environmental agents* in an AEC/FM core model is suggested by Gielingh, but it has been restricted to the effects of *natural agents* (e.g., wind, rain, load, earthquake, insects, etc.) on products [Gielingh 1988]. The incorporation of the concept of *agent*, as defined and classified above, in an AEC/FM core model would make the model useful to all PM functions. The modeling of agents is also associated with the issue of *views*.

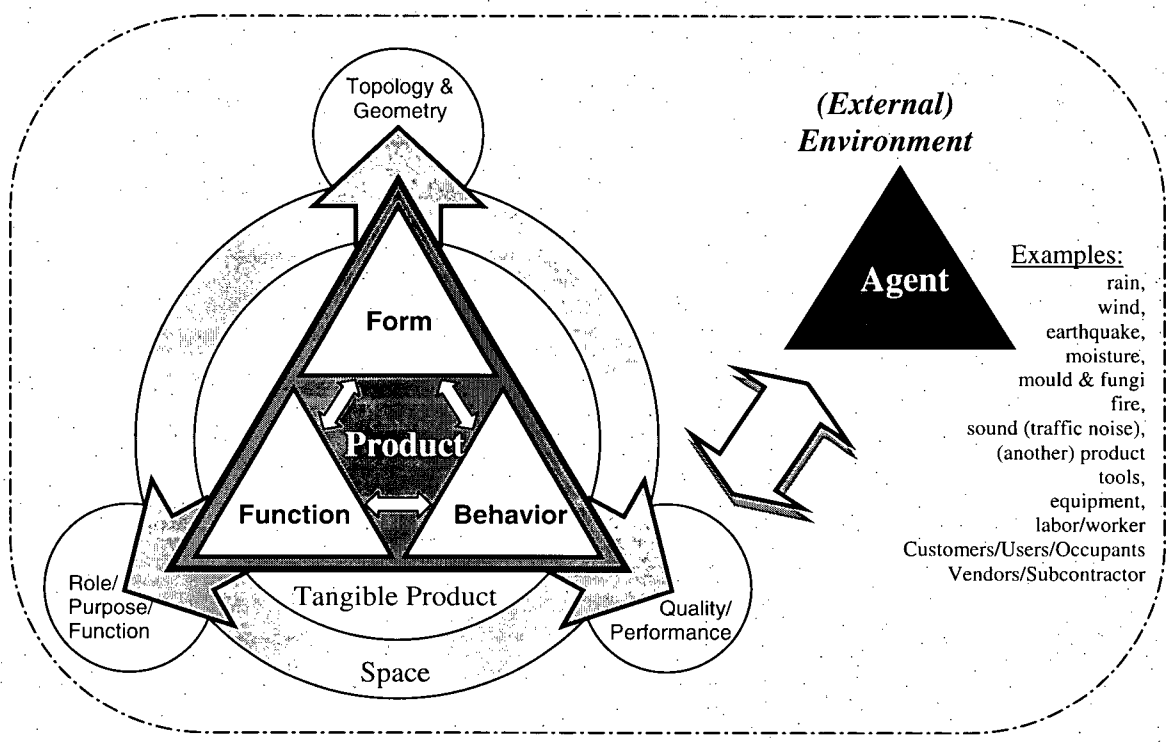


Figure 4–2: The Product, its Aspects, and its Environment

4.1.4 Views

Every project participant looks at and defines a project differently depending on his or her function (role and responsibility). For example, a concrete column may be defined as a *product* to the structural designer and constructor, but it could be considered as an *agent* for an electrical engineer and roofing sub-contractor. Thus, in an AEC/FM core model, product-oriented elements (i.e., spaces, target products and materials) should be defined in a manner that allows them to be flexibly used and viewed as a product or an agent, depending on the interest of the application area of concern.

Another example is the concept of *material*, which has been modeled both as a type of *resource* and as a *property* of other products [Froese 1994; Luiten et al. 1993; IFC R2x, 2000a]. In reality, however, an object (e.g., a door) may be considered as a product to one project participant (e.g., the manufacturer or the carpenter), while it is treated as a material resource to another participant (e.g., the constructor). This research generally refers to the concept of *material* as a *physical thing* that is acquired and used by CM and FM functions. Section 4.2.3 further describes a solution to the *view* issue.

4.1.5 Product Models and PM Functions: Examining Existing Product Models

As a step towards identifying the requirements of an AEC/FM core model, this section focuses on *target product* and *space* concepts and their related FFB/P aspects. It discusses how product information

(encompassing FFB/P aspects) is viewed and used by PM functions, and how it relates to process information. This is done by the means of examples and reviews of current AEC/FM models.

4.1.5.1 Target-Product and Space Concepts in Models

Spatial and physical component information is modeled differently in existing product models. **AP225** [ISO 1995a] treats *space* as a type of *building element* object; however, shape representation of space is considered to be different from that of other types of building elements (Figure 2-7). A similar approach is taken by **BCCM** [ISO 1997] (Figure 2-6). **COMBINE** [1995] relates form, function, and behavior of space (discussed under function and performance in sections 4.1.5.3 and 4.1.5.4).

The **IFC** object model [IFC R2x, 2000a], introduces the concept of *element* as a generalization of all types of target products, and relates this closely with the concept of *space* (through various objectified relationships). Figure 4-3 shows the product-oriented objects defined in the *IfcProductExtension* schema of the IFC object model and their related objects in other schemas. At the highest level, the *IfcProduct* object is specialized into three objects: *IfcElement* (components that make up an AEC/FM product), *IfcSpacialStructureElement* (spatial elements used to define a spatial structure), and *IfcGrid* (a planar design grid—e.g., a rectangular, circular, or triangular grid—defined in 3D space to aid in locating structural and design elements). The *IfcElement* object is classified into seven subtypes including *IfcBuildingElement* (the "primary parts of a building"), which is specialized in the *IfcSharedBuildingElements* schema into such objects as *IfcWall*, *IfcColumn*, *IfcBeam*, *IfcSlab*, *IfcRoof*, *IfcDoor*, *IfcWindow*, *IfcStair*, etc. The *IfcSpacialStructureElement* object has also been classified and related to the subtypes of *IfcElement* through objectified relationship.

In order to facilitate exchange of information about any type of object that is not otherwise semantically defined in the IFC model, an object named *IfcProxy* is also defined as a subtype of *IfcProduct* in the Kernel module of the model (Figure 4-4). The *IfcProxy* object may have a geometric representation in the project model (inherited from the *IfcProduct*) only if it represents a product.

In summary, the IFC's model *space* and *target product* as two closely related objects that encompass the idea of *product*, while it deals with *material* separately (section 4.1.6). The declaration of the *IfcGrid* (which is actually a "graphical presentation object") as a subtype of *IfcProduct* is also noticeable. Some of the challenges that can arise with respect to modeling product information become apparent when considering the FFB/P aspects, as elaborated in the following sections.

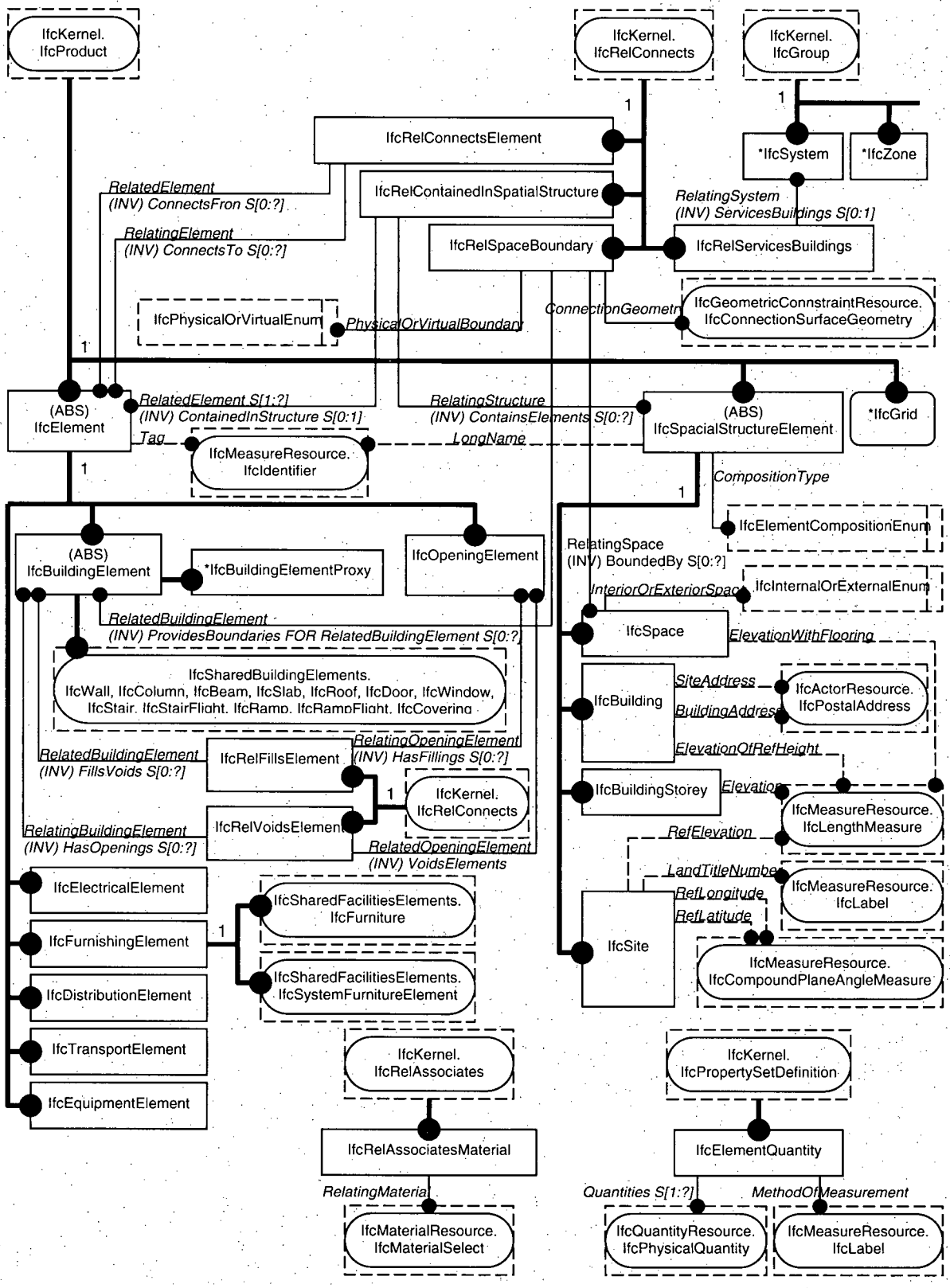


Figure 4–3: Product-related Objects in the IfcProductExtension Schema of IFC R2x

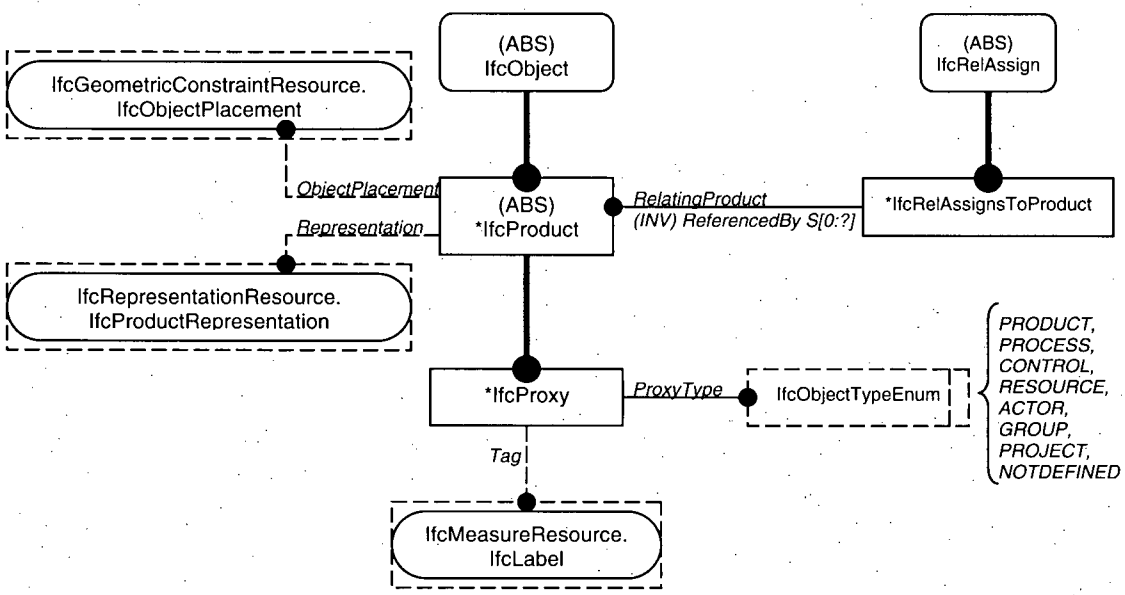


Figure 4–4: The IfcProxy Object in the IfcKernel Schema of IFC R2x

4.1.5.2 Form Information

Form-related product information is some of the most important data regularly used by AEC/FM processes, and thus, it makes up a large portion of most product models. **AP225** [ISO 1995a] defines shape-related building product information as needed in the detailed design stage. It specifically deals with shape-related information of "*building elements*". **COMBINE** [1995], on the other hand, extensively models spatial topological information in the context of building energy and HVAC design. At the highest level, it defines four topological relationships as attributes of *physical_object* (product and space): *includes*, *adjacent_to*, *composed_of*, and *bounded_by* (Figure 2-8), but they are exclusively used for spaces.

The **IFC** object model [IFC R2x, 2000a] specifically deals with the geometry and topology characteristics of products (i.e., the *IfcProduct* object, representing a target product, space, or grid). For this purpose, it defines a number of basic schemas in the resource layer of the model (section 2.4.4.7.2). The geometry and topology characteristics can be defined through the *IfcProduct* object's two attributes of *Representation* and *ObjectPlacement* (Figure 4-4).

Geometry and topology may be equally attributed to any physical thing: tangible built elements (e.g., a building), spatial elements (e.g., a room), and materials (e.g., bricks). For example, *composed-of* and *includes* relationships can be used to define the relationship between concrete and its constituent material elements water, cement, sand, gravel, air bubble, etc. The definition of topological relationships among spatial elements and tangible elements (i.e., products and materials) can help in sequencing and

scheduling of construction activities [Echeverry et al. 1991]. Other examples include the use of geometric information in developing work plans and analyzing productivity—Thomas et al. [1990] report on the impact of slab depth on concrete placement productivity; Hinze and Parker [1988] explain how the number of corners may impact the productivity of forming, stripping, rebar, and concrete pouring operations; Al-Hussein [1999] shows construction simulations involving the geometric and topological representation of both construction products and equipment (cranes); and Riley and Sanvido [1995] address space management for activity scheduling and material handling based on the geometric space requirements for both the building and the work crew activities.

The product's form data becomes input data in the evaluation and analysis of PM process elements. A typical example is the use of *size*, *quantities*, and *location* of product items used for *quantity takeoff* and *estimating* as well as *construction engineering* purposes (e.g., *formwork design*); i.e., a relationship between the product's form characteristics and the resources (e.g., formwork) used for its construction. A construction *progress report* is another example in which CM functions make references to the geometrical product information. The *area of walls* constructed during a particular day, the *length of road* paved, or the *volume of concrete* placed in *foundation walls* are examples where geometrical product data play an important role for the definition of the concepts used in CM processes.

In summary, PM functions use products' form information as a primary input to their processes. Explicit representation of *form* and its association with any physical object (e.g., products, spaces, and materials, equipment, and personnel resources) provides a mechanism to model PM functions applications more effectively. In the IFC model, the form properties could have been associated with the *IfcObject* class (instead of *IfcProduct*). Inclusion of such a capability, backed by the position of the geometry resources within the modular, layered architecture of the IFC model, can provide support for many PM application areas such as simulation of construction processes.

4.1.5.3 Functional Information

Functional information is an important ingredient to PM/CM decision-making processes. Any facility project starts with the recognition of a *need* for the physical facility that satisfies certain functions specified as functional requirements. The requirements represent just one type of functional information, for which various representational approaches have been proposed [Wittenoom et al. 1997].

Existing project data models have dealt with functional information differently. The **BCCM** [ISO 1995b] introduces a *Function Object,* defined as "a description of the function or role played by parts in systems." However, the class is specialized into six functional *physical product* entities (*Distribution*, *Structural*, *Space*, *Furnishing*, and *Separation* objects); i.e., the BCCM does not represent functions.

In **COMBINE** [1995], *function* has been associated with the *Space* object alone, and only for its *planned usage* (Figure 4-5). *Space_function*, as a sub-type of the root entity *idm_object*, is decomposed exactly in the same manner as the *space* object. This means that functional requirements are modeled at the same level of detail as the *Space* definitions (Figure 2-8). Defining *usage* as "an activity of a user in a space", the model describes *space function* as a set of *space_usages*, which is related to the occupancy and technical requirements (e.g., indoor air quality, lighting, and ventilation). The **IFC** object model [IFC R2x, 2000a], however, does not deal with functional information.

CM functions may not require access to the same spatial functional requirements as designers may (e.g., a room designed to function as a living room). However, they may involve manipulating other spatial, functional information (e.g., assigning a room to construction materials storage, temporary construction offices, etc. [Riley and Sanvido 1995]). In this instance, the COMBINE model (Figure 4-5) could perhaps be used with an additional attribute "*temporary_space_usage*" for the *space_function* entity. Functional information, however, may not be restricted to spaces, and it may also be attributed to target products (e.g., a *load-bearing* wall versus a *partition* wall), materials (e.g., a piece of batt insulation functioning as an *insulating* material), and other process-related concepts such as equipment and tools.

Construction contracts and specifications, for example, may specify functional requirements, leaving the selection of specific products and materials that meet the requirements to the contractor. As another example, the planning of construction methods and resources may be expressed by first defining the functions required to complete a task, and then selecting the resources capable of performing these functions: the functional requirements of placing concrete or loading soil into trucks can be fulfilled by the resources of a certain crew or a front-end loader, respectively. A resource may generally have many functions, and a function may be performed by different resources.

The existing AEC/FM product models do not deal with product functions thoroughly, if at all. Functional information about a product does not usually get beyond the designer's mind (i.e., in the form of design intent), and it is not electronically communicated to the construction, operation and maintenance stages of the product. Explicit representation of *function* and its association with product, resource, and process concepts provides a mechanism to model PM functions applications more efficiently.

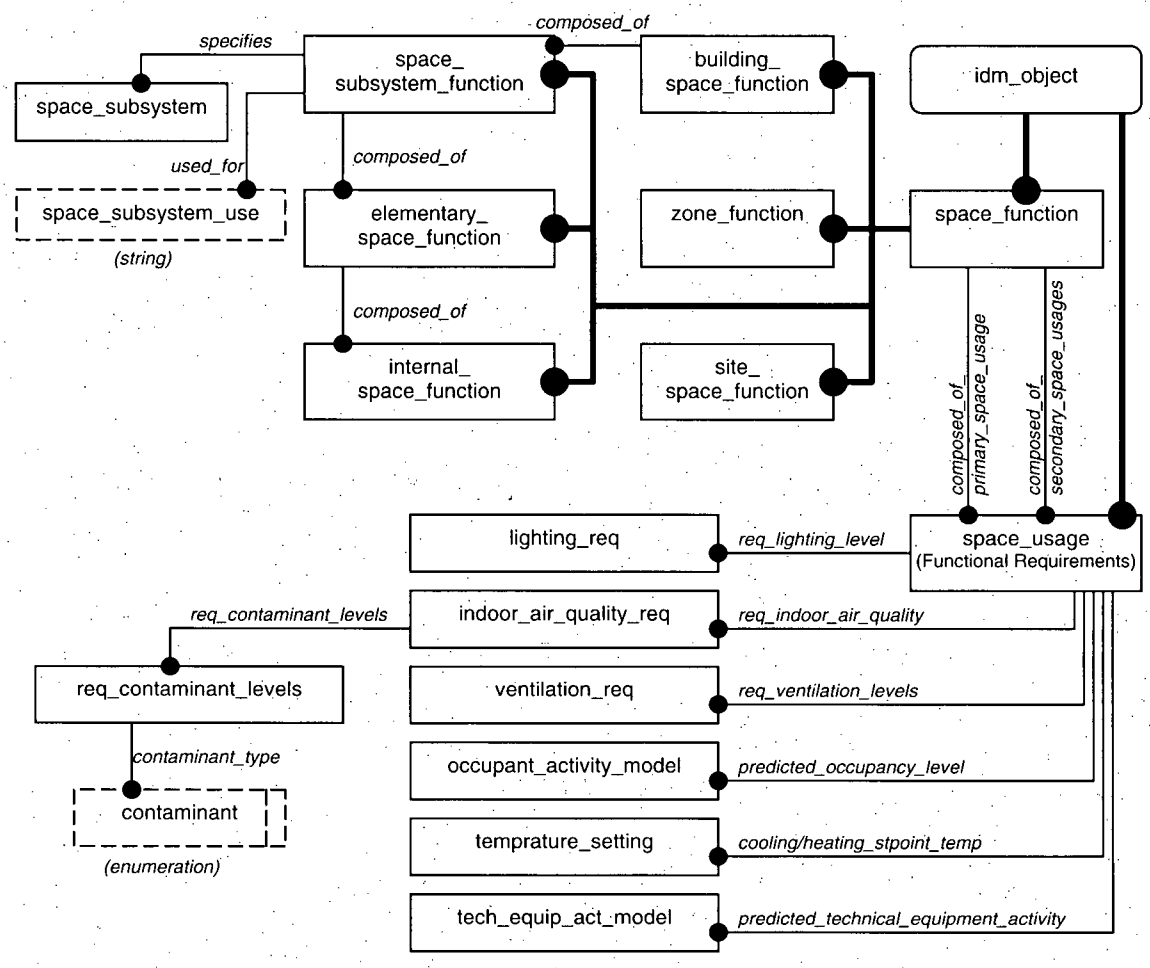


Figure 4–5: A Partial, Simplified Model of Functional View in COMBINE

4.1.5.4 Behavior/Performance Information

Product performance data plays a major role in AEC/FM products analysis and evaluation processes and in PM functions integration. **COMBINE** [1995] views performance as it relates to spaces and physical building elements for a number of energy and HVAC Design Tools (DT). At the highest level, an abstract entity of *performance* is defined with the relation of *is_related_to* to the abstract *physical_object* (Figure 2-8). Every root of a DT schema (intended to store the data created by the tool) is then considered as a subtype of performance. Material has not been associated directly to the performance entity, but if needed, DTs can use material behavioral properties (e.g., thermal conductivity) through an association between a physical building element and *material* or *glass* entity.

The **BCCM** [ISO 1995b] defines the object, *BC_Performance,* as "behavior related to use", with only one attribute of *performance_rating*, defined as "a rating or measure of behavior related to use." Despite its potential association with any type of object, it has been associated only with

BC_SpecificationClause through a one-to-many relationship. The model also introduces two quality objects (though omitted in the 1997 version of the model): 1) *BC_Quality,* defined as the "totality of properties that bear on ability to satisfy needs", with one attribute of *quality_document_name*, and 2) *QualityClause,* defined as "a particular property that bears on ability to satisfy needs", with two attributes of *quality_clause_description*, and *quality_clause_number*. BC_Quality has a one-to-many relationship with QualityClause. However, no relationship has been considered between performance and quality.

The **IFC** object model [IFC R2x, 2000a] does not explicitly refer to the term *performance*, though it defines geometric and material properties and provides a few mechanisms for associating externally-defined properties with objects. A review of the approach of the IFC model to modeling properties of object is presented in section 4.2.1.2.

Performance data is central to the analysis and evaluation of AEC/FM products. Examples of performance analysis made during design are resistance of a column under load of a slab, durability of a roofing membrane under existing weather condition, and visual effects of interiors and exteriors of a building. Some of this performance data is later referred to in CM tasks, particularly in QM/QC processes [Battikha 2000]; e.g., the *required slump of concrete* transferred from design tasks to concrete procurement, placement, and testing tasks.

Performance, however, may not be restricted to products. The notion of the performance of a piece of construction equipment, for example, is similar whether the equipment is used as a resource or a product. *Production rates* and *productivity* are examples of performance in construction. They are used as measures of efficiency of construction resources and effectiveness of the construction processes that utilize the resources [Hendrickson and Au 1989].

Quality is a key concept in construction. It refers to a measure of performance: "degree of goodness or worth (e.g., a material with a very poor quality)" [Oxford Dictionary 1989]. Much of the QC tasks involve checking and testing products' performance. The quality of a product is, in essence, a measure of goodness of performance and is derived by comparing two versions of performance characteristics (e.g., as-built and as-required values).

The existing models do not model performance as is needed by CM functions. Performance can be attributed to both product-and process-related objects. Concerning product-process information integration, the concept of *performance* can be introduced at the high level of an AEC/FM core model to be used for the definition of the behavioral characteristics of both product and process oriented objects.

4.1.5.5 Change and Versioning

AEC/FM products evolve through an iterative development process. Increasing levels of details during design, continuous knowledge input into the project from different disciplines, organizational changes, and unforeseen conditions all contribute to the iteration of processes, resulting in revisions and versioning of products and, thus, process elements.

AP225 [ISO 1995a] addresses change and versioning of the *shape* aspect of building elements alone. It defines two entities: *approval* (the acknowledgement or verification of a building element shape) and *change* (a modification, request for a change, or documentation of a change incorporated into the shape of a building element), with a one-to-many relationship between them (Figure 4-6).

The **IFC model** [IFC R2x, 2000a] addresses versioning and change in three places:

1) In *IfcKernel* schema: four attributes (*OwnerHistory*, *GlobalId*, *Name*, and *Description*) are defined for the *IfcRoot* object, from which other objects are specialized (Figure 2-10).
2) In *IfcSharedMgmtElements* schema: the *IfcChangeOrder* object is defined as a subtype of *IfcProjectOrder* (Figure 4-7). This object can reference any non-control object, as well as project documents, including the change order itself.
3) In *IfcApprovalResource* schema: the *IfcApproval* object is defined as "information about approval processes for a plan, a design, a proposal, a change order, etc." (Figure 4-8). It captures information such as the involved actors and timing and status of the approval. Being referenced by *IfcApprovalUsage* (in *IfcControlExtension* schema), it can be related to any subtype of *IfcObject* through appropriate objectified relationship declared in the *IfcKernel* schema.

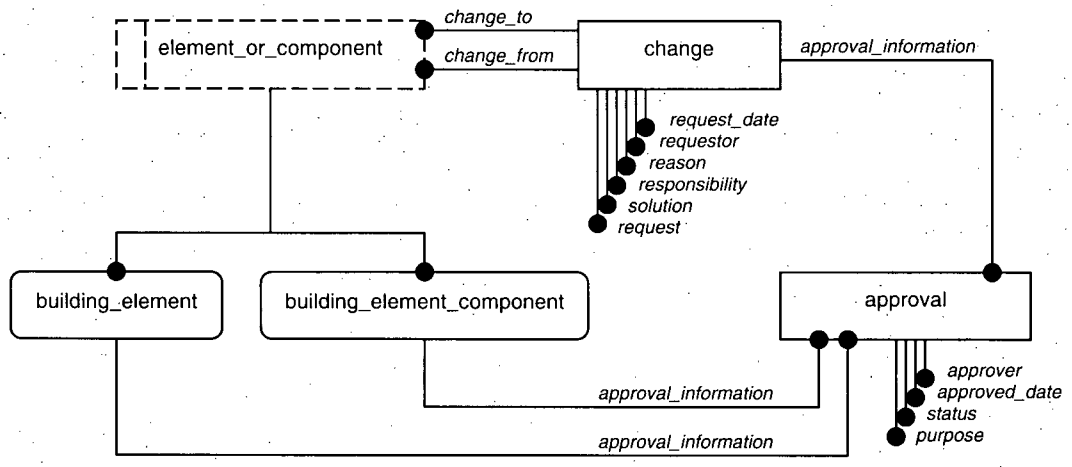


Figure 4–6: Versioning and Change in AP225

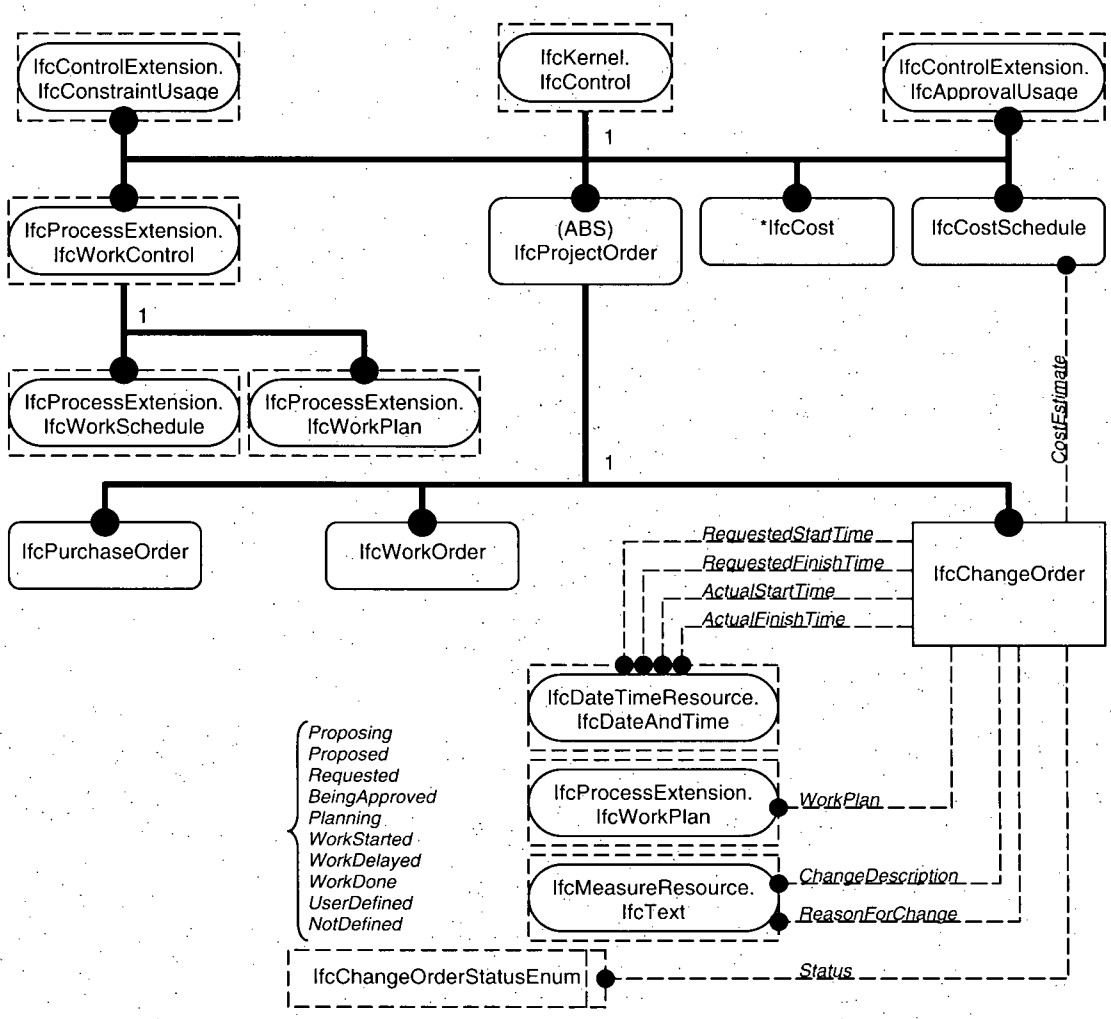


Figure 4–7: Change Order in the IfcSharedMgmtElements Schema of IFC R2x

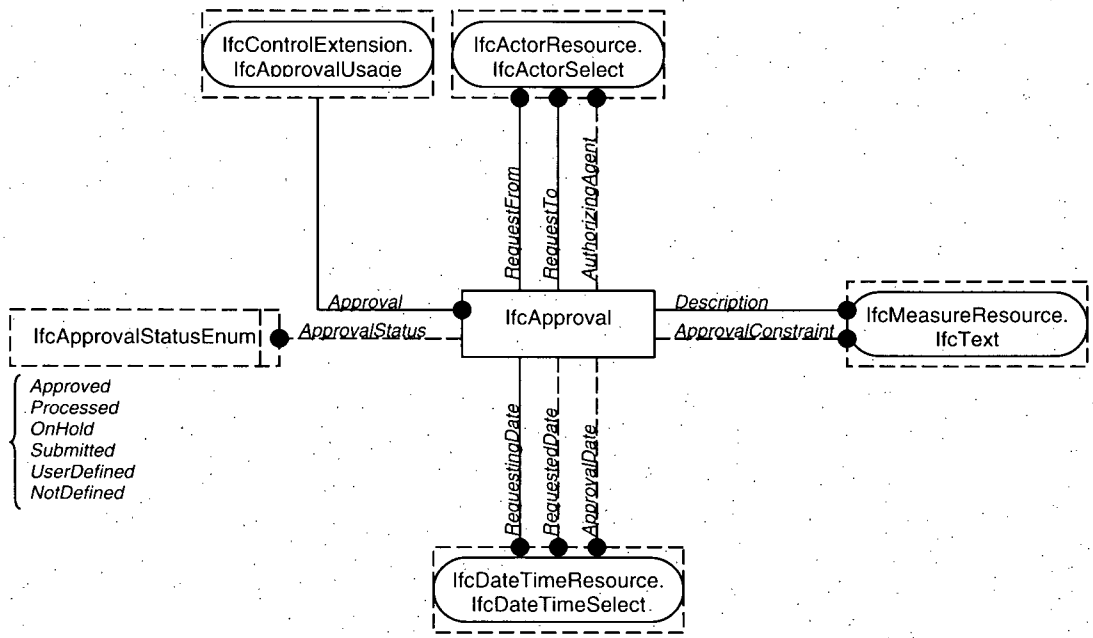


Figure 4–8: Approval in the IfcApproval Resource Schema of IFC R2x

Compared with AP225, the IFC model addresses change and approval concepts more comprehensively. It provides a mechanism to associate the concepts with the various types of occurrence objects defined in the model. Nevertheless, the objects useful for *change management* are scattered at different layers of the IFC model. While the *IfcChangeOrder* and *IfcApproval* objects represent transactions, they are treated differently (e.g., their locations in the model). Moreover, it may be reasonably argued that not all things (e.g., a design, a submittal, etc.) that go through an approval process may receive approval. Rejection, request for revision and resubmitting, and conditional approval (e.g., approval with some modification) are usually among possibilities in the process, but this is not foreseen in the models.

The FFB/P aspects may change throughout the life of a product. For example, performance characteristics are usually *required* by the *owner/user* of the product (i.e., *user requirements*), *standards*, and/or authorities. They are later analyzed and expressed as *expected* physical and structural characteristics of the product by designers Later, they are *measured* and *controlled* by CM and FM functions and authorities in the form of *quality* of the product (e.g., performance of an installed furnace and finishing quality and structural strength of a wall, and air tightness of an envelope). A comparison of *as-required* and *as-realized* versions of air-tightness of an envelope, with some acceptable limits, would give a measure of quality of the constructed envelope and, thus, result in an *accepted* or *rejected* envelop.

Functions of a product may also be subjected to versioning. A building designed as a school, for example, might change its intended function, at some other time, into a medical center. Another example is the assignment of temporary functions to spaces (e.g., material storage) during construction.

Versioning may not be limited to products. For example, an insulation material shipped to and stored on the site may be subjected to a change order by the designer to be replaced with another one. Moreover, a change in product specifications may require modification of documents (e.g., drawings) and construction processes (i.e., methods and sequencing, resources, organization, etc. Another example is unforeseen (natural, social-cultural, and economic) conditions, which often target construction processes and subject them to changes. Delays and claims analyses would then be examples of the upcoming processes, which need information input on versions of both product and process elements.

The concept of versioning is common to all PM functions in all types of projects, but it is not fully addressed in existing models. The gradual evolution of product information, from conceptualization to specification and realization together with subsequent changes in the processes requires especial consideration in AEC/FM project modeling. Concerning integrated total PM systems, however, the concept of *versioning* must be considered for both *product* and *process* information in an AEC/FM core model. Section 4.2.4 discusses a potential solution to the issue of versioning.

4.1.6 Material Information Modeling

Continuing on the requirements of an AEC/FM core model, this section focuses on material information from a PM function perspective.

4.1.6.1 Material: Evolution and Changing Roles

A facility is made up of materials, and thus, designing, constructing, and maintaining the facility requires information about the constituent materials and their characteristics. The concept of *material* encompasses evolution and changing of roles of a *thing* in different project views. Materials are produced and sold as *products* by manufacturers and suppliers. They are selected and specified in design documents as *parts of the planned target product* (e.g., lumber used for a stud wall). They are acquired, stored, handled, tested, and installed as *material resources* by construction crews. Becoming a part of the target product, later, their quality and performance may be questioned, and inspected by regulatory agents, managers, supervisors, occupants, and inspectors, within the context of the *realized target product* (i.e., having no role of *resource* what so ever). This process is comparable to the use of materials for assembling construction *resources* (e.g., lumber for formwork).

Although the procured construction materials may be intended to be consumed (wholly or partially), and therefore become a part or whole of some final target product, the concept of *material* and its link to the target product would continue to exist after its use. For example, the face bricks delivered on pallets and stored at the site would finally be installed to form one of the components (and thus become a part) of an external cavity wall. Nevertheless, later in the construction or operation, an inspector may determine that the *wall* fails to pass the QC because of the poor performance of the *bricks*. Consequently, a complaint may be placed against the brick supplier, with a reference to the brick (i.e., material resource) and its related information.

Another example is the concrete placed in a structural component of a building such as a column. Loosing its natural identity (i.e., a loose, bulky material), the concrete would become a part of the component, but its related information (e.g., its supplier, load number, etc.) should not be discarded from the inventory of project information. If the concrete fails to demonstrate its required performance (e.g., low strength appeared in the lab-tests results of the samples collected from the batch of the delivered concrete), follow-up would require reference to the same loose, bulk of concrete delivered to the site. In fact, the component (i.e., target product) must continue to carry information about its constituent materials just as it would with other types of information such as its contractor, time of operations, testing and inspecting agency, and so on.

The above discussions may be applied to all types of materials (i.e., engineered, fabricated, and bulk materials; section 2.6.3.1). Unlike a bulk material (e.g., concrete), an engineered material (e.g., a fan, a furnace, etc.) may not loose its identity after installation. It is the same object, but its *roles* and *status* have changed over time. It can be tagged with a barcode, for example, and identified, tracked, and monitored at any time of its life cycle. Although it may not be referred to as a material resource (but rather as a building component) after installation, it would still carry all its related manufacturing information (e.g., product information and manufacturer), procurement information (e.g., its supplier and its receiving, handling, and storing conditions), and installation information. This information, which is needed for the performance of PM/CM/FM functions, is a part of the history of the equipment, and it is usually distributed in the material's manufacturing documents, project design and specifications, as well as CM documents (inspection reports, memos, etc.).

Material information is closely related to other types of project information, especially products and spaces. A typical example is the thickness of a cavity wall, which is the sum of the thicknesses of the layers of the consumed materials (i.e., brick, cavity, and so on). Another example is the shape and volume of a loose material (e.g., a bulk of cement, a pile of excavated soil, etc.) that is very much dependent on the shape of the *container* of the material (e.g., a loader's bucket, a truck's container, a laydown area, a

room, a box, etc.); i.e., the shape of the *space* provided by the *container* dictates the shape of the contained material. Such physical characteristics of the space as well as those of the material (e.g., degree of looseness, stack-ability, strength and load-bearing capabilities) are among the information that many MM functions such as space planning and material packaging and storing depend on.

The *product* and *material* concepts are closely related. They both refer to physical *things*, while they are, in fact, some *roles* given to the things, by various actors and at different stages of their journey, from their origin (e.g., nature) to the processing or manufacturing unit (e.g., mine, factory, etc.) and to the construction site and the constructed facility. Consideration of the intimacy of these concepts is a major requirement of an AEC/FM core model intended to serve as a backbone of integrated total PM systems. The next section elaborates on the breadth of coverage of such information in existing AEC/FM models and opportunities for improvement.

4.1.6.2 Material Information In AEC/FM Information Models

Existing integrated models have treated material information in several different ways. **COMBINE** [1995] defines two basic material-related entities—*material* (subtype of *idm_object*) and its subtype *glass*—in one schema in the IDM core model. The properties defined for the *material* entity can be assigned to any one of the seventeen *material types* (carpet, insulation, wood, etc.) (Figure 4-9). Almost the same approach is taken for defining the *glass* entity and its attributes.

This model shows a designer's perspective of materials information (mostly *performance* and partly *form* characteristics). The types of information included in the model are just those needed for the required analysis for envelope performance, lighting, etc. in a building. Material is viewed in the form of some *material properties* associated with the (materialized) layers of an enclosing element, finishing, and joint (Figure 2-8), rather than the *physical* bulk of material packaged, received, and stored at the site.

Like COMBINE, **AP225** [ISO 1995a] focuses on the behavioral aspect of materials, and more explicitly uses the term *material property* to represent such information (Figure 4-10): "a descriptively-named characteristic of the material from which a Building_element or Building_element_component is made" [ISO 1995a]. Inherited from its supertype *Property*, this entity has been associated with *building_level, Building_element,* and *Building_element_component* with many-to-many relationships.

As such, by adding some other subtypes and attributes, the COMBINE and AP225 models might serve some other design disciplines than they are intended to, but they may not serve many processes of CM functions, especially those involved in managing materials. For example, the damages identified by a receiving function need to be associated with specific material occurrences received (i.e., labeling as "defected"); such information is not represented by the entities defined in the models.

BCCM [ISO 1997] suggests a *bc_material* entity, whose only textual attribute (i.e., *material*) represents the material's name. Considering its definition ("substance that can be used to form products and resources"), this entity is apparently intended to define a physical construction material. Considering its associated entities, however, the entity seems to define some topological relationships between layers of a building element (e.g., an envelope) and some geometrical attributes (such as length, area, and volume) of the layers. Nevertheless, not all materials are layered in nature (e.g., concrete). Moreover, no explicit representation of all aspects (i.e., the FFB/P) of all types of materials exists in this model.

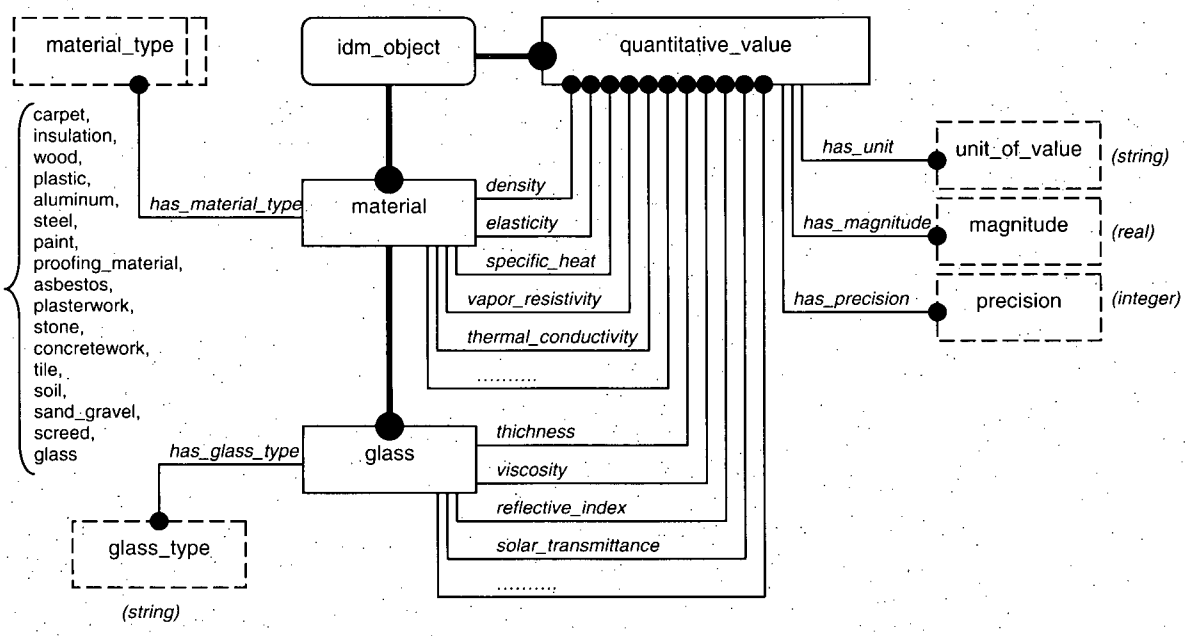


Figure 4–9: A Simplified Model of Material View, with Selected Attributes, in COMBINE

(Note: The attributes of *material* and *glass* are partially shown, and dashed boxes are *defined data types*.)

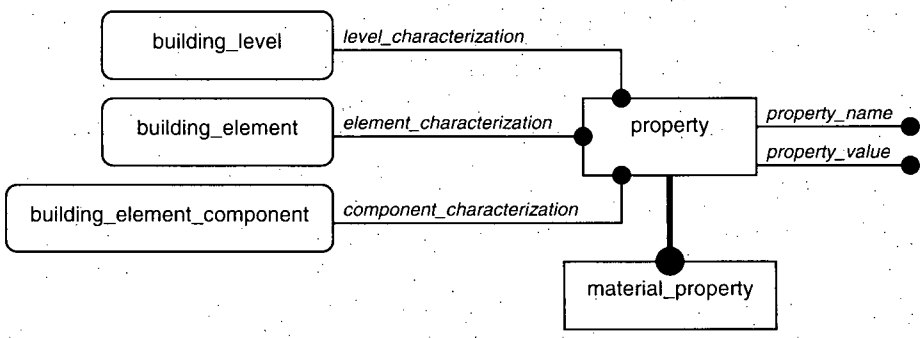


Figure 4–10: A Model of Material (Property) View in AP225

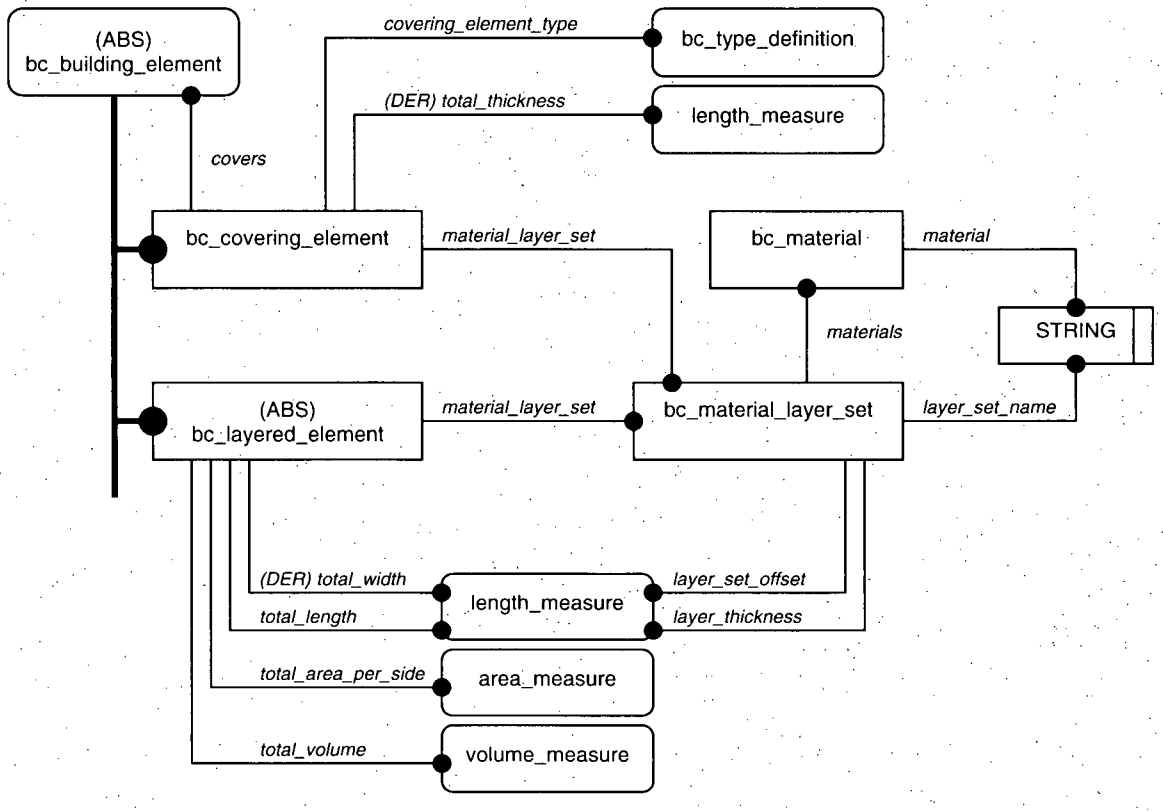


Figure 4–11: A Model of Material View and its Related Entities in BCCM

4.1.6.3 Material Information in the IFC Model

The **IFC** model [IFC R2x, 2000a] views material from two perspectives: *construction material resource* and *design material* (definition and properties), as described and discussed in the following.

Construction Material Resource: Figure 4-12 shows the resource-related objects and their relationships defined in the *IfcConstructionMgmtDomain* schema together with other related objects defined elsewhere in the IFC object model. The abstract *IfcResource* object (representing "use of things", i.e., a project's resource type used in a specific process/task) is specialized into six subtypes, including *IfcConstructionProductResource* and *IfcConstructionMaterialResource*. The IFC model emphasizes the former as an object whose instances should have a product counterpart, while such a condition is considered as optional for an instance of the latter object. Moreover, the object does not reflect whether the resource is a material or equipment, for example, whereas the latter identifies the resource as being a *material type* ("e.g., 'gravel', but not product substances, e.g., '5 tons of gravel'" [IFC R2x, 2000a]).

Both of these objects can only represent a *resource type*, as opposed to an *occurrence resource*—the subject of type and occurrence objects are further discussed in section 4.2.2. They may be simply

thought of like a *line item* in a project transaction (e.g., *formwork of type A* in an estimate or *gravel of size X* in a purchase order). They are assigned as an item to a task through *IfcRelUsesResource*.

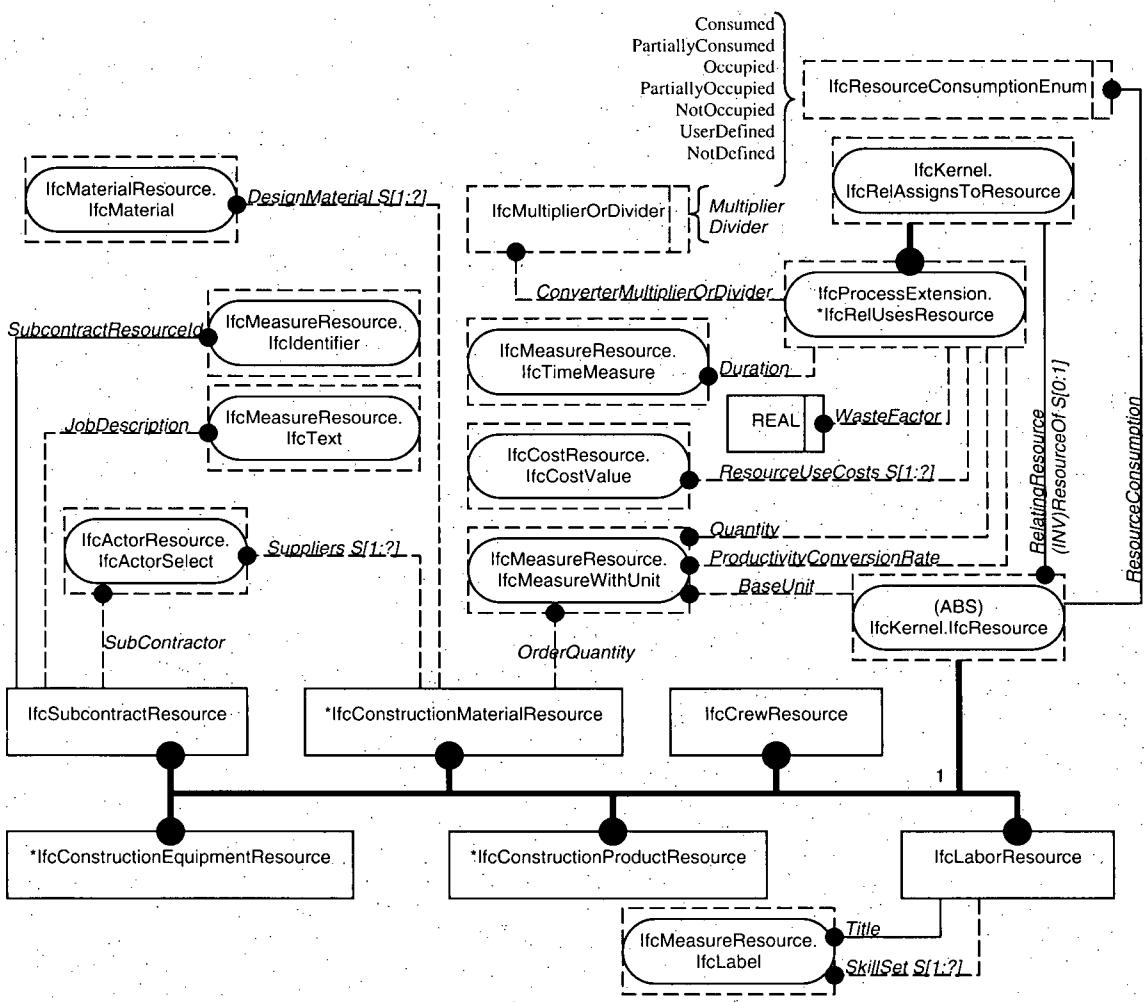


Figure 4–12: The Resource Objects in the IfcConstructionMgmtDomain Schema of IFC R2x

Design Material (Material Definition): Figure 4-13 shows the object model of the *IfcMaterialResource* schema (in the Resource Layer) and some other material-related objects from other schemas of the IFC object model. The *IfcMaterialSelect* type allows an actual material definition instance to be one of the following three types:

- *IfcMaterial* (a single homogeneous substance in a building elements),
- *IfcMaterialList* (a multiple/composite material element, whose precise structure is not specified), or
- *IfcMaterialLayerSetUsage* (a layered element, whose structure is specified).

A material definition instance (selected through *IfcMaterialSelect*) may be associated with an instance of any one of the subtypes of the *IfcElement* object (excluding *IfcOpeningElement*) through the

IfcRelAssociatesMaterial objectified relationship (Figure 4-14). Moreover, the *IfcConstructionMaterialResource* object (explained above) has an optional relationship (i.e., *DesignMaterial*) with the *IfcMaterial* object. While the former typically captures information about a *type of bulk material* (e.g., sand and gravels) used in a *process*, the latter captures only the *name* and *classification* of a *substance* used in a *building element* or a *task*. Thus, none of them can represent an occurrence of construction material on the site.

Design Material (Material Properties): "*Material properties*" and their assignment to *IfcMaterial* are captured in the *IfcMaterialPropertyResource* schema (shown in part in Figure 4-15).

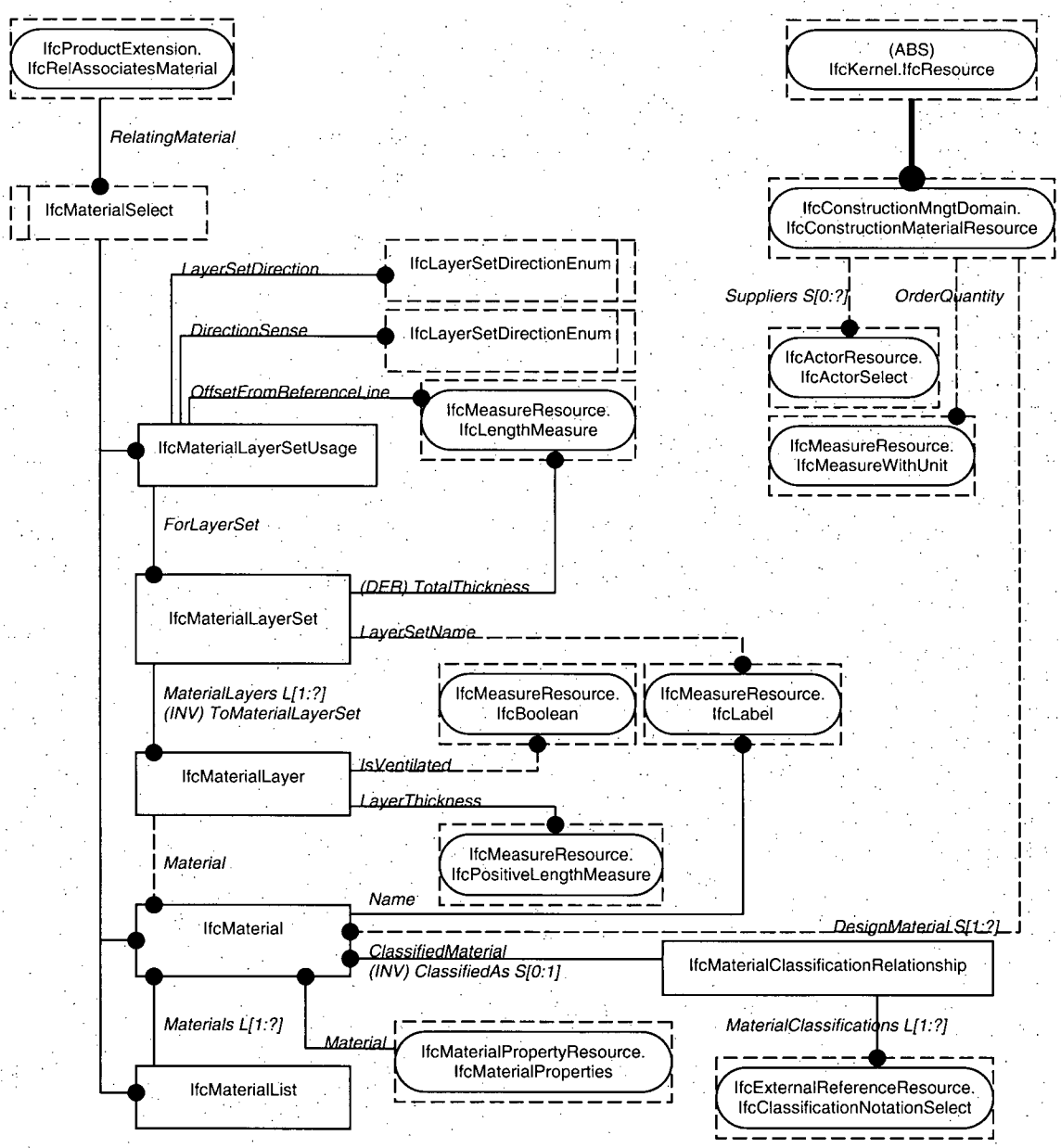


Figure 4–13: Material-related Objects in the IfcMaterialResource Schema of IFC R2x

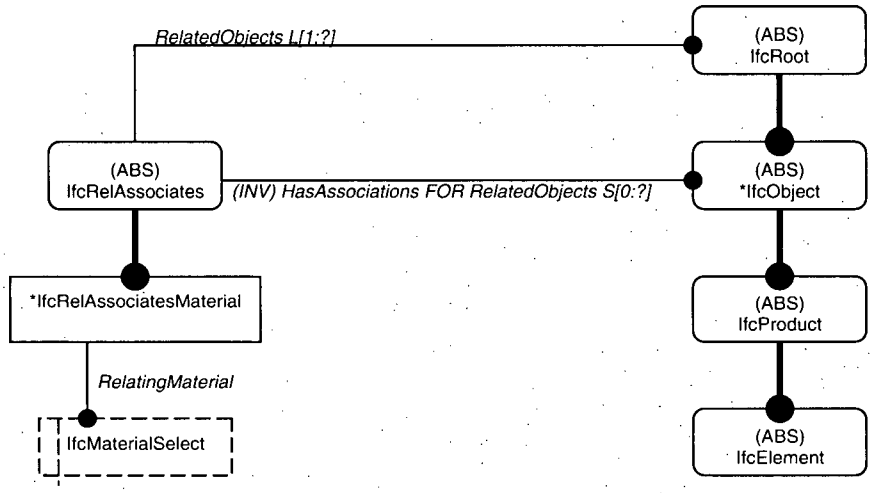


Figure 4–14: Association of Building Elements and Material Definitions

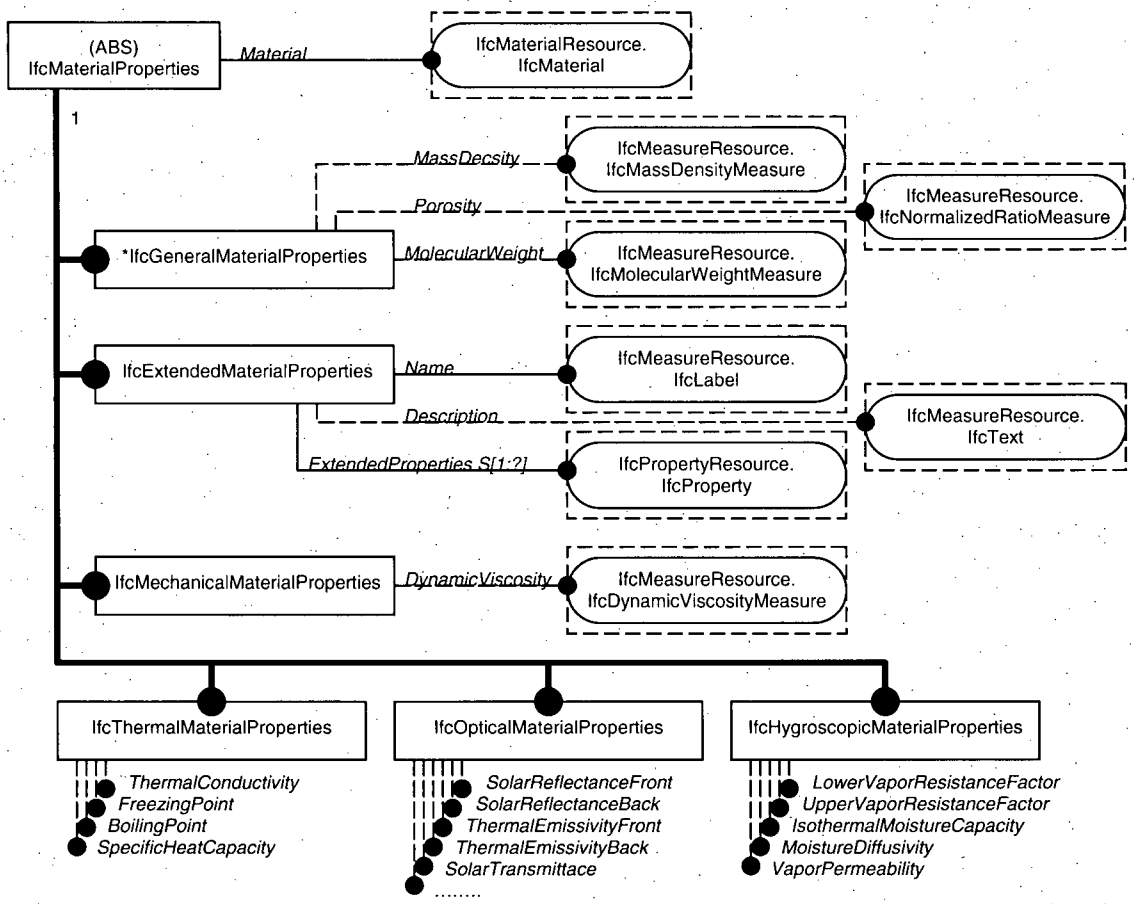


Figure 4–15: Material Properties in the IfcMaterialPropertyResource Schema of IFC R2x

Places for Improvement in the IFC's: Compared with other AEC/FM models, the IFC model deals with material information more comprehensively, though places for improvement do exist. The IFC model touches upon the *performance* and, partially, the *form* characteristics of materials. It also reflects some CM views on material information. The *IfcMaterialResource* and *IfcMaterialPropertyResource* schemas are the main parts of the IFC model that provide such information in the form of *material property types* definitions. The *IfcKernel* schema, which should (by definition) provide the basic modeling constructs for representing things, models material information more like a *material line item* in a transaction (e.g., an estimate), rather than as an *occurrence of a material* in a project. There is a need for treating and defining material as a *thing* (e.g., a bulk/package of material)—similar to a building element (e.g., a wall)—in the *IfcKernel* schema. The other schemas (i.e., Extension, Shared Elements, and Domain modules, as well as Material Resource and Material Property Resource modules) could then extend upon or be used by this basic construct(s) as needed—according to the referencing constraints set between layers of the IFC model architecture (section 2.4.4.7.2).

4.1.6.4 Summary of Material Information Modeling

The concept of *material* has not been comprehensively addressed in most existing models. Material is not usually modeled as a *physical thing* (e.g., a real bulk of material, mechanical equipment, etc.), but as *material properties* (i.e., behavioral aspect of material) and *material types*; which may satisfy the information requirements of engineering analysis and design. This presents a barrier for the electronic communication of material information between project disciplines and phases. An AEC/FM core model, intended to serve all PM functions, must explicitly represent *material* as a physical object—just like any other physical object and with its own explicit FFB/P characteristics—that is used to make *target product* and/or *resources*.

4.1.7 Summary of AEC/FM Product Information Modeling

This section discussed issues and requirements of an AEC/FM core model as they relate to product and material information modeling in the context of integrated total PM systems. The FFB/P aspects of products were discussed, and it was shown how PM functions view and use FFB/P data, and how product information may be shared among these functions. Reviewing some of the existing models, it can be generally concluded that none of the existing product models capture all aspects of products. Most of them model a designer's view of product information and may not fully support a full range of PM functions. It was also suggested that these aspects should be generalized at the highest level of an AEC/FM core model so that they could be used to define both product and process-related objects. The necessity of supporting for versioning of such objects in the core model was also addressed.

4.2 Beyond Product Information: Modeling Project Information

This section addresses information modeling that goes beyond the specifics of AEC/FM products to consider more general issues that are important for an integrated AEC/FM core model.

4.2.1 Properties and Relationships of Objects

An object is described with its *properties*, including *relationships* with other objects (section 2.1.1.3). This section discusses issues relating to modeling object properties and relationships, as well as requirements of an AEC/FM core model in this regard.

4.2.1.1 Some Modeling Issues on Properties and Relationships

1) ***The Context Issue (Contextual Properties versus Non-Contextual Properties)*:** Objects' properties may be broadly classified into two groups: *contextual properties* (those that are dependent on the context in which the objects exist) and *non-contextual properties* (those that are independent of the context). Objects get their contextual information from their relationships with other objects. For example, a person may maintain his/her name and identification number (e.g., social insurance number) throughout any different context (e.g., when ordering, requesting, or sending an item). However, other information such as the number of orders or requests previously placed or the roles played by the person (e.g., buyer, supplier, requester, or designer) is generally context dependent and is most directly associated with the relationship between the related objects.

2) ***Internal Data versus External Data*:** Software applications are built upon their underlying data model, which reflects the data that is *internal* to the scope of the application. In a collaborating environment, however, software applications often use each other's data, which may be considered *external* relative to the application's own data. It may not be possible or necessary to explicitly model all possible object properties in the internal data model of a software application. For example, the information released by a manufacturer about a type of door may not be directly modeled in the project data models of various collaborating applications (e.g., materials selection and specification). Thus, provision of a mechanism for exploiting such external information is an important requirement of an AEC/FM core model serving such systems.

3) ***Optional Properties or Relationships*:** Along with the issues explained above is the issue of *optionality* of data that may be explained in an example: The manufacturing and prefabrication of construction materials usually take place irrespective of the location they would be used (e.g., manufacturing doors and engineered materials, section 2.6.3.1); i.e., no need for association between the material and the place concepts. However, in some circumstances—especially for the materials fabricated for a specific project (e.g., cutting and forming steel rods for concrete reinforcement and

assembling roof trusses) and in integrated MM systems incorporating materials workshoping and planning systems—such a link would be necessary. The link, for instance, provides for an efficient shipping and transporting of the fabricated materials from the workshop to the point of use.

An AEC/FM core model must *flexibly* accommodate object properties and relationships as required by various application areas (i.e., context considerations) in a manner to avoid *complexity*. The next section discusses how the IFC model approaches this matter.

4.2.1.2 Properties and Relationships in the IFC Model

The IFC object model [IFC R2x, 2000a] is the leading AEC/FM model dealing with properties and relationships of objects. The model uses several mechanisms to define object properties (Figure 4-16):

1) ***Properties as Attributes*:** These are directly attached to objects (e.g., ID, name, and description). Some are defined as *optional* (as opposed to obligatory) to accommodate contextual matters.
2) ***The IFC's Explicitly-Defined (or Typed) Properties*:** Some properties are explicitly defined (i.e., with semantics) and can be attached to products using objectified relationships. These include geometric properties and material properties (section 4.1.6.3).
3) ***Properties Not Defined in the IFC Model*:** The model provides two basic mechanisms to facilitate exchange of information that is not explicitly included within the IFC model:
   a) <u>*The Property Definition mechanism*</u>: allows developers and users of the model to define and use data-driven information. This mechanism, which encompasses the *IfcPropertyDefinition* object and its subtypes as well as the *IfcProperty* object (Figure 4-16), is described in section 4.2.2.3.
   b) <u>*The Association Relationship mechanism*</u>: introduces a concrete objectified relationship object called *IfcRelAssociates* as a subtype of *IfcRelationship* in *IfcKernel* schema (Figure 4-16). This object is defined to establish unidirectional associations (i.e., "links") from occurrence objects or property definitions to external sources of information (e.g., a classification, library, or document). Despite its role of "*external-source definition*" provider, however, one of its subtypes (*IfcRelAssociatesMaterial*) is defined for association between building elements and "material definitions", which are defined *inside* the IFC model (section 4.1.6.3).

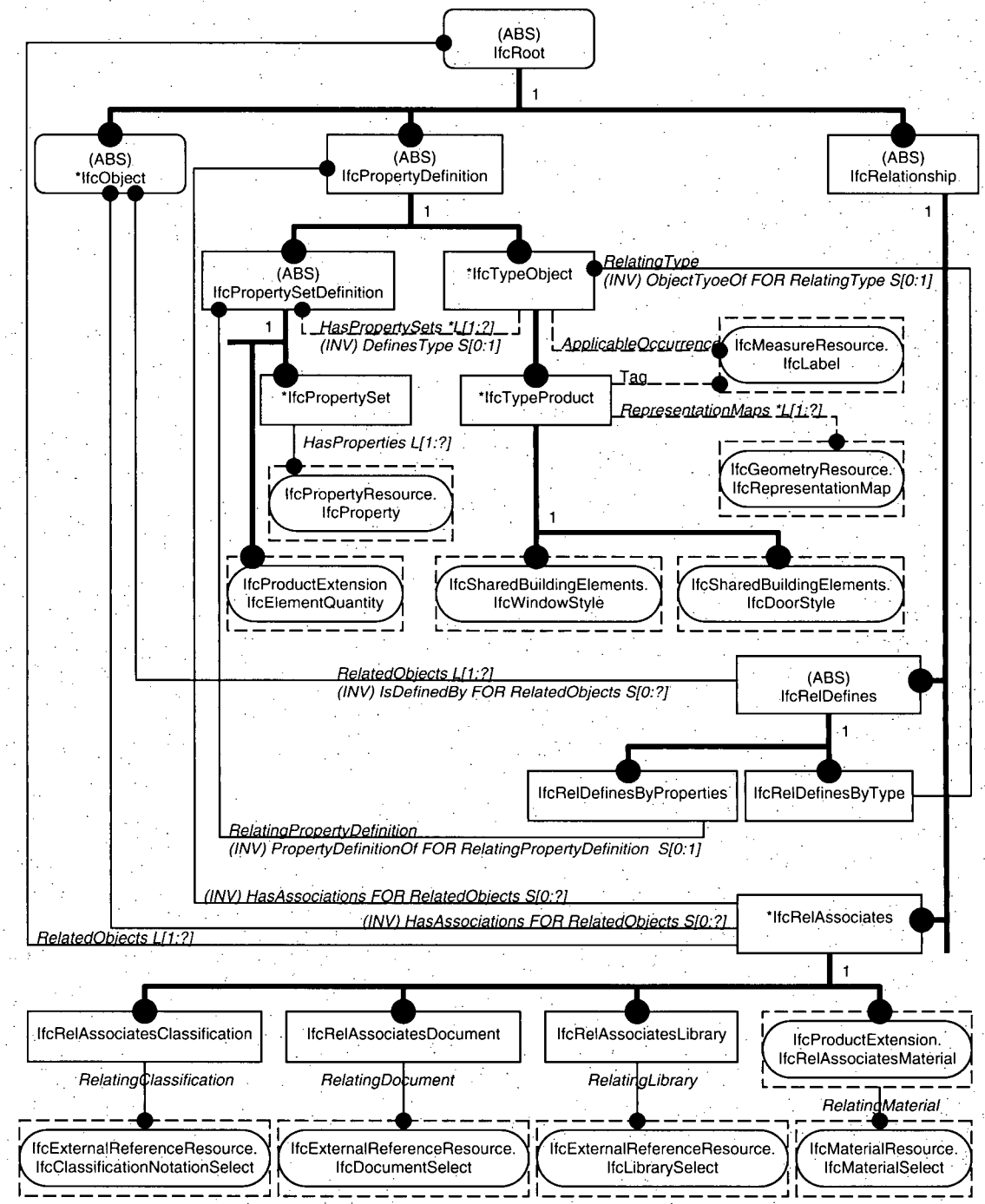


Figure 4–16: Properties in the IFC Object Model

A review of the evolution of the IFC model from one release to the next shows a dramatic shift from using *direct (including optional) relationships* to *objectified (or indirect) relationships*. The objectified relationships defined in the model (section 2.4.4.7.4) generally have a number of characteristics:

1) They are classes in their own rights (i.e., having properties and objects), capturing relationship-specific properties of objects (i.e., contextual properties, as opposed to non-contextual ones) and reserving the possibility to handle relationship- specific behavior later.
2) They are used to refine many-to-many relationships into one-to-many or one-to-one relationships.
3) They are suggested as preferred candidates to relate objects within or between schemas.

Although the sum of the objectified relationships defined in the IFC model might not suffice to address the various possible contextual situations in AEC/FM projects (section 4.2.2.4), the use of such relationships has generally had a great effect on reducing the complexity of the model. Compared with direct relationships, the use of objectified relationships between objects offers several advantages: they can reduce the *complexity* from and add *flexibility* to a core model. It can provide the user of the model with the means of associating objects as needed. Considering the various views involved in handling AEC/FM processes, provision of such flexibility appears as an essential requirement. However, several precautionary measures need to be considered. First, such relationships must cover a full range of the relationships required between various types of objects in different AEC/FM domains. Second, the relationships must be consistently applied throughout the model.

In conclusion, as a leading AEC/FM core model, the IFC's provide a number of mechanisms that help define properties or exploit externally-defined properties, in addition to the properties explicitly defined in the model. All these are supported by the various forms of relationships, especially the objectified relationships. However, explicit representation of the basic properties of objects (i.e., the FFB/P aspects, section 4.1.2) and relating them to all types of objects can considerably improve the effectiveness of the model in satisfying various information requirements of AEC/FM systems. Such an improvement seems necessary for the IFC model, which is intended to serve as a common core model to exchange information among AEC/FM application systems.

4.2.1.3 Summary of Properties and Relationships

This section presented a discussion on modeling object properties and relationships as it relates to an AEC/FM core model. It outlined a number of modeling issues concerning properties and relationships, and it discussed the IFC's approach to addressing them, as well as places for improvement. The major suggestions made in this section include the necessity for generalization of the FFB/P concepts to define properties of objects, and handling objects relationships in a manner to provide flexibility while avoiding complexity (e.g., defining contextual and external properties). The next section elaborates on levels of specificity of objects and their associated properties and relationships.

4.2.2 Levels of Specificity of Objects: Type and Occurrence Information

PM processes involve manipulating two basic categories of information (*type* and *occurrence*), which are central to the design and implementation of computerized systems supporting the processes. This section elaborates on the definition, use, and modeling of these two categories of information, whose consideration is a part of the requirements of an AEC/FM core model.

4.2.2.1 Type and Occurrence Information

Models are abstractions of reality, and data models describe selected *real world* objects and their selected characteristics; i.e., a subset of information about real objects that is of interest. Both physical and conceptual objects (a door and a person versus privacy and cost) can be the subjects of modeling.

Figure 4-17 illustrates the relationships between real world objects and data model objects, at three levels of abstractions: *generic*, *type*, and *occurrence*; which may be mapped to those defined in the GARM for Product Definition Units (generic, specific, and occurrence) [Gielingh 1988] (section 2.4.4.1). At the *generic* level, an object is described generically. A door, for instance, may be described as an object having width, length, price, etc. The *generic door* is like a parameterized description of a door, which may be instantiated into a *door type* (e.g., WD-02; wooden door type 02)—containing the information that in usually described in product catalogs and manuals published by manufacturers, parts schedules (e.g., a door schedule), detail drawings (e.g., a shop drawing of a door installation detail), and business transactions (e.g., a purchase order). As such, an instance of a type object may have actual values for many of its attributes (e.g., values specified for length, width, and height of a WD-02-type door).

As shown in Figure 4-17, a *door type* provides a description of some individual real doors (e.g., RD-208 and RD-308, representing the doors of rooms 208 and 308 respectively), which are referred to as *occurrence doors*. Such occurrences, for instance, may be referred to in a keying schedule of a building, in an inventory of doors labeled for installation, and so on.

In summary, a *type object* is a grouping of some properties that is used to describe zero or more real *occurrence objects*, which can be identified and tracked individually, e.g., by serial numbers, rather than by quantity. The type object, in turn, can also be described by another *type object* (e.g., a manufacturer-typed door). The next sections explain how such information is and should be modeled.

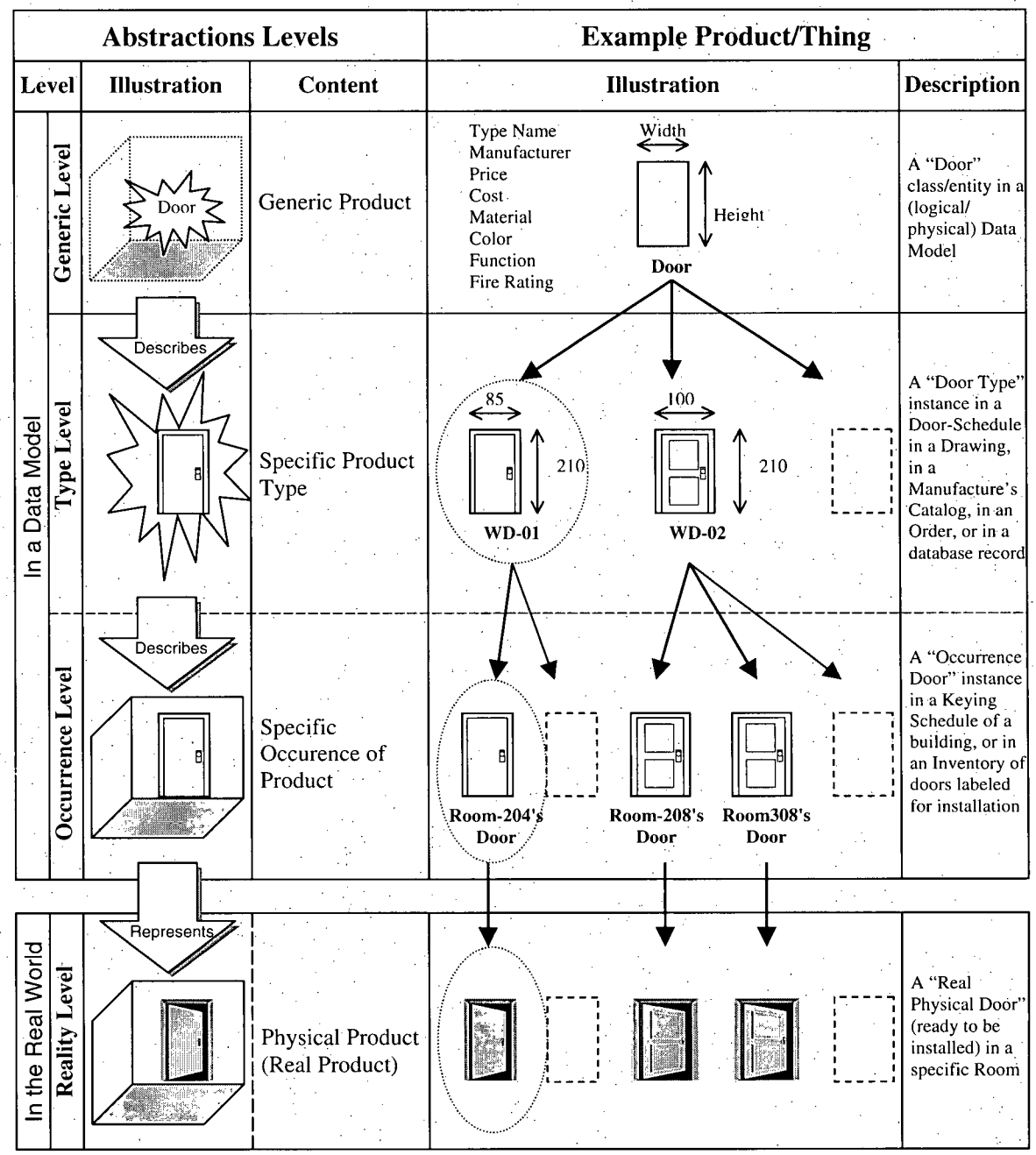


Figure 4–17: Levels of Abstractions (Generic, Type, and Occurence Information)

4.2.2.2 Type and Occurrence Information in Models

The distinction between type and occurrence objects has been recognized by many object-oriented methods and by some AEC/FM data models. Fowler [1997] uses the terms "*knowledge level*" and "*operational level*" to refer to two abstract layers of an object model including the type and occurrence objects respectively. Also, considering these two types of objects as two basic ingredients of

all object models, Coad et al. [1999] refer to them as "*Description*" and "*Party, Place, or Thing*" respectively. As such, they limit occurrence objects to *physical* objects.

Most standard and research AEC/FM data models portray the data structure of occurrence objects; however, the necessity for inclusion of type objects in core models has also been recognized by some others varying in views and scopes [Bjork 1992; Rankin 2000; Hassanain 2002]. Among the models including the type and occurrence information are GARM [Gielingh 1988], TOPS (Total-Project Systems) [Froese et al. 1997a and 1997b], and the IFC model [IFC 2.x, 2000a]. However, none of the models have dealt with the issue profoundly; i.e., a lack of comprehensiveness, in terms of coverage of various types of both physical and conceptual objects and their relationships.

Proposing an instance model (as opposed to a type model, section 2.4.1.4), the TOPS project, uses the terms "*Type Object*" and "*Instance Object*" (i.e., occurrence), at an application-model level: modeling construction methods [Froese and Rankin 1998] and computer-assisted planning [Froese et al. 1997b; Rankin 2000]. It proposes a data model with six basic entities for modeling *Products*, *Processes*, and *Resources* information: three for Types and three for Instances (Figure 4-18). The type information is modeled in aggregation and specialization hierarchies. However, instance information is structured using only aggregation hierarchies (because it is not meaningful to talk about sub-type of a specific occurrence of an object; e.g., sub-type of the UBC Main Library). The model uses the term "*construction method*" (represented by the *Process Type* object) as an equivalent to other terms such as method, work technique, and construction technology. Obviously, the model is intended to serve a specific PM view rather than as an AEC/FM core model, though it challenges the ways the type and occurrence information related.

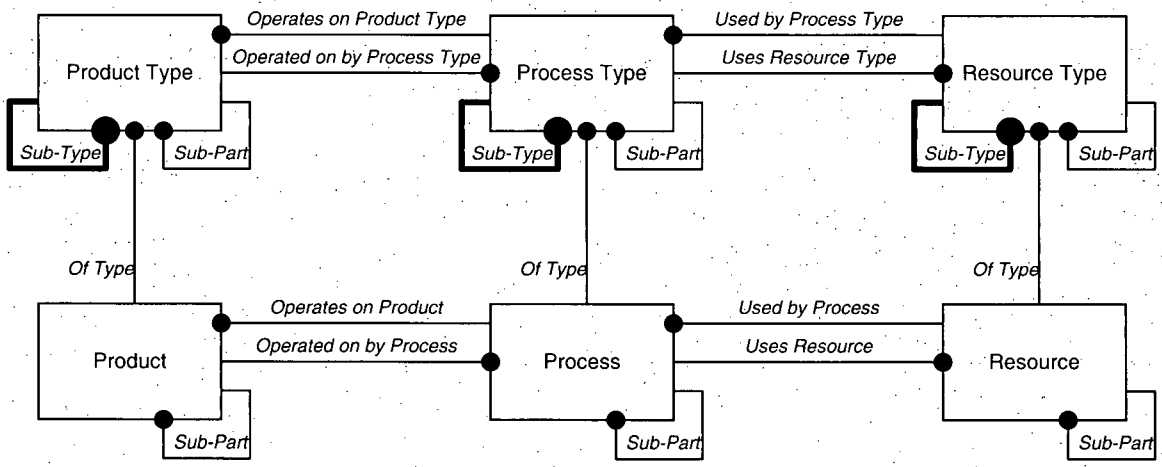


Figure 4–18: Type and Instance (i.e., Occurrence) Objects in TOPS

[Based on Froese et al. 1997b]

4.2.2.3 Type and Occurrence Information in the IFC Model

At the highest level of its *IfcKernel* schema, the IFC model [IFC 2.x, 2000a] defines two objects: *IfcObject* (an occurrence object) and *IfcPropertyDefinition* (properties and descriptions of an occurrence object). The latter object is specialized into *IfcTypeObject* (representing "*type objects*") and *IfcPropertySetDefinition* ("*partial type information*", i.e., a set of individual properties); which may be linked to occurrence objects using the two objectified relationships *IfcRelDefinesByType* and *IfcRelDefinesByProperties* respectively (Figure 4-16).

Despite the clear distinction between the two types of information, the definitions are not consistently applied throughout the model. For instance, among the seven subtypes of *IfcObject* (Figure 2-10), two objects (*IfcProduct* and *IfcActor)* are explicitly defined to represent occurrence objects, and the definitions and usages suggested for the *IfcResource* object and its subtypes recommend the objects to represent a type object (e.g., a material or equipment type and its required quantity for a specific task defined in an estimate) rather than an occurrence object. Moreover, as Figure 4-16 shows, the type objects defined in the current IFC's are limited to window and door styles (subtypes of *IfcTypeProduct*).

4.2.2.4 How Type and Occurrence Information are Used and Related

There is a bilateral relationship between *type* and *occurrence* information. AEC/FM processes generally involve processing of both types of information. There are situations in which information is processed in a top-down manner (i.e., from type to occurrence; from top to down in Figure 4-17); e.g., a project's door-types schedule is first designed, then individual doors (occurrences of these defined types) are assigned to individual rooms. Alternatively, information may be created in bottom-up processing (i.e., from occurrence to type; from down to top in Figure 4-17); e.g., the individual doors defined for a building are grouped to define the door schedule for the building. The extent of variation of approaches to the design of built elements is elaborated in literature [e.g., Wittenoom et al. 1997].

Overall, in project information processing, six types of relationships can be established between type and occurrence objects, as listed with examples in the following:

1) *Relationship between Occurrence and Type Objects of the Same Kind*:
   a) *Product or material specification*: relationship between a door and a door type (explained above).
   b) *Document specification*: a reference made from a specific physical document (e.g., a report) to a document type (e.g., report type).
2) *Relationship between Occurrence and Type Objects of Different Kinds*:
   a) *Product specification*: specifying the type of materials used in a product (e.g., concrete 3500 psi used for a slab on grade) in the drawings or specifications of a project.

b) *Resource allocation/planning*: assignment of resource types (e.g., crew type: C-8) to occurrences of processes (e.g., the task concrete slabs of the building).

c) *Procurement planning*: defining types of materials (e.g., 2" gravels) that need to be procured by a specific procurement activity (procure gravels).

3) *Relationship between two Type Objects of the Same Kind*:

a) *Project specifications*: a reference made from door type specified in a project's door schedule to that of a manufacture's catalog or to a standard.

b) *Project specifications*: assignment of tool-resource types to labor-resource types.

4) *Relationship between two Type Objects of Different Kinds*:

a) *Product specification*: a reference made from a foundation type to its constituent material types (e.g., in a reinforcing schedule of a construction drawing).

b) *Resource library creation*: assignment of operator types to an equipment type.

5) *Relationship between two Occurrence Objects of the Same Kind*:

a) *Structural Analysis*: Wall #13 in the first floor bearing the load of Wall #23 in the second floor.

b) *Scheduling*: Successor and predecessor relationship (e.g., the procure gravels preceding the grading activity of a project).

6) *Relationship between two Occurrence Objects of Different Kinds*:

a) *Material delivery*: a delivery ticket generated for a load of concrete sent to the site.

b) *Construction material or product testing*: relating a material test (e.g., concrete test) and a load of concrete, or a column in which the concrete is poured.

c) *Product, material, or equipment inspection*: relating an inspection to a specific room of a building, a segment of a road, or a piece of mechanical equipment (e.g., a furnace).

An AEC/FM core model should facilitate both top-down and bottom-up processing of project information, as well as the various possible relationships among type and occurrence objects.

4.2.2.5 Summary of Type and Occurrence Information

This section elaborated on the definitions, importance, and use of type and occurrence information in managing a construction project, and thus in modeling integrated PM systems. The various ways in which AEC/FM processes may use or relate type and occurrence objects were also categorized. Current models have not addressed the two types of information and their relationships profoundly. The distinction between type and occurrence information is an essential part of data models of most AEC/FM applications and must be considered as an integral part of an AEC/FM core model. The core model should

extend the *type* and *occurrence* concepts to both physical objects (e.g., products and materials) and conceptual objects (e.g., processes and transactions). It should also provide a flexible mechanism for relating the two types of objects as used in AEC/FM processes. The next section focuses on modeling *roles* of objects.

4.2.3 Roles of Objects

Objects play roles in business processes. This section discusses some issues and approaches to modeling roles of objects and requirements of an AEC/FM core model in this regard. It explains how the role concept may be used as a means of representing various views of PM functions on a same object.

4.2.3.1 Role Players and Roles

A *thing* (i.e., an occurrence object, section 4.2.2.1) may play a variety of roles in different contexts, and the properties expected from the thing depend on the roles. The thing (or *role player*) can be an actor (e.g., a person and/or organization), an artifact (e.g., formwork), or a natural element (e.g., a tree). A person, for example, may play roles such as employee, customer, laborer (i.e., a resource), superintendent, father, etc., depending on the context or process of concern. An individual person acting as a superintendent may attain a different set of properties than if the individual was fulfilling the role of an employee. A role may simply be defined as a way of participation by a thing in a business event or process. An object's roles are the different hats that the object wears in different situations.

The concept of "*role*" and approaches to its modeling have been elaborated in the literature. Fowler [1997], Coad et al. [1997], and Booch et al [1999] discuss the concept as it relates to actors (i.e., persons and organizations). The Coad's DNC model [Coad et al. 1999] (Appendix A) further extends the concept of role beyond actors (which they call "party—i.e., person or organization") and relate it to physical artifacts (which they call "place" and "thing") as well. It explicitly models the concept of *role* but not *role type*. It models many objects such as *Product*, and *Material Resource* as a *Thing* (i.e., an occurrence of a physical thing). The next section searches the IFC model for the role concept.

4.2.3.2 Roles in the IFC Model

Except for roles of actors (i.e., persons and organizations), the IFC model [IFC R2x, 2000a] does not explicitly model the concept of *role*. Figure 4-19 shows all actor-related information defined in the *IfcActorResource* schema and other related schema of the model. The *IfcActorRole* object can be used to define roles such as *supplier*, *manufacturer*, *contractor*, *architect*, etc. for three types of actors: a person, an organization, and person-organization (i.e., a person identified within an organization context). The *IfcRelAssignsToActor* object handles the assignment of objects (i.e., subtypes of *IfcObject*) and roles to an

actor (represented by *IfcActor*). It captures information about the actor's roles through a relation with the *IfcActorRole* object.

The *IfcActorResource* schema supports relating people to organizations and relationships between organizations (e.g., a hierarchical organization structure); however, its information content is not sufficient to exchange detailed information about human resources (HR). For example, concepts such as *position* and *responsibility* may need to be modeled and linked to the existing information, which is central to HR management processes.

To a large extent, the model flexibly allows the definition of non-contextual and contextual actors and roles (i.e., without or within a project context).

However, it may not support all possible AEC/FM business processes, including the assignment of *non-contextual roles* to *occurrence objects* other than actors. Examples scenarios include:

1) *Project Specifications*: are prepared mostly based on *generic* and *type* information, often with an association with occurrence objects other than project actors. For instance, most project specification *sections* include clauses on Submittals, Adjusting/Caring and Cleanup, which mostly define the responsibilities of the *role types* (e.g., Contractor, Structural Engineer, Installers, Applicators, and Manufacturer/Distributor, etc.) involved in the *work items* of the *sections*; i.e., assignment of *responsibilities* to *role types* and association between *role types* and specific sections and/or *work items* (with no indication of the role player).
2) Project Planning: long before the real work starts and real actors are identified, the organizational and operational plans are usually prepared by identifying the basic *role types* (e.g., *project manager*, *superintendent*, *general contractor*, *subcontractor*, *surveyor*, etc.) required for the project. After preparation of a work breakdown structure, subcontractors of work packages are generically identified (e.g., assigning *tasks* to *role types*). Responsibilities and communication requirements may further be defined as they relate to the involved roles.

In order to add the above feature, the *IfcRelAssignsToActor* object could be renamed and redefined (or a new IfcRelAssignsActorRole relationship be added) so that it could be used for assignment of an instance of *IfcActorRole* to one or more instances of *IfcObject*, independent of the *IfcActor* object. The model does not support such a requirement, since the obligatory relationship from *IfcRelAssignsToActor* to *IfcActor* (Figure 4-19) enforces indication of a role player in any event. Despite the various usage scenarios supported by the model, as a common AEC/FM project model, the IFC model does not provide sufficient flexibility. The many ways in which AEC/FM processes may occur may well justify such a requirement. Moreover, the model does not go beyond modeling organizational roles.

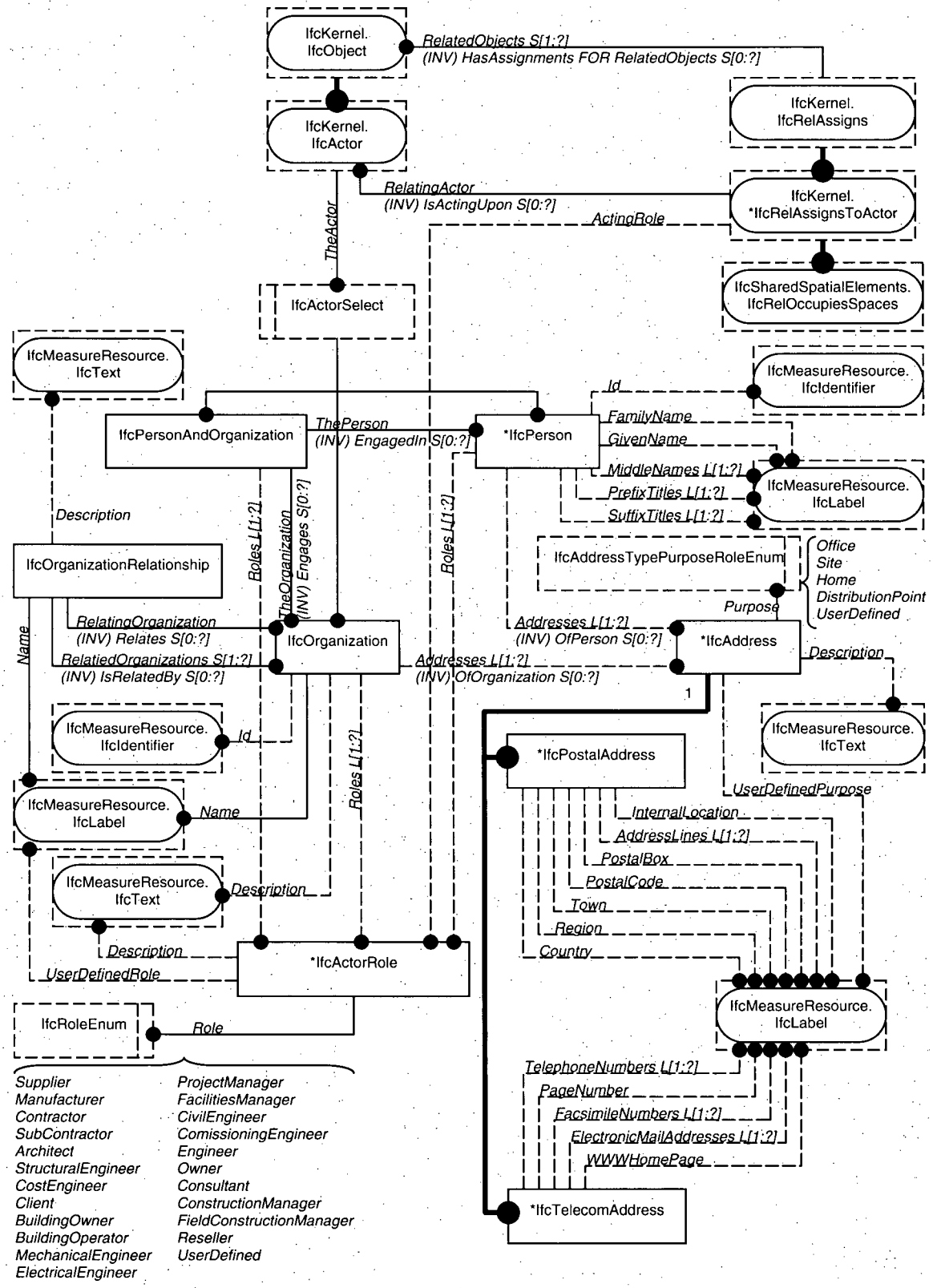


Figure 4–19: Actor-Related Objects in Various Schemas of IFC R2x (Base Schema: IfcActorResource)

4.2.3.3 Beyond Organizational Roles

Organizational roles and role players may well illustrate an approach for modeling object roles in general. Figure 4-20, which includes a summarized version of the actor-role view of the IFC model (originally presented in Figure 4-19), demonstrates a synopsis of the views of this research to modeling *roles*. The figure categorizes the objects into two groups of *contextual* and *non-conceptual* objects. The *IfcRelAssignsToActor* object enables contextual role definition through a relationship with a context (i.e., *IfcObject*), the contextual role player (i.e., the *IfcActor* object), and a non-contextual role (i.e., the *IfcActorRole* object). Except for the *IfcActorRole*, which represents type information (section 4.2.2.1) about organizational roles, all other objects capture information about occurrence objects. *IfcActorRole* is a role type object, and it acts as a description that can be attached to both non-contextual actors (i.e., *IfcPerson*, *IfcOrganization*, and *IfcActorPersonAndOrganization*) and contextual actors (i.e., *IfcActor*).

The *role* concept provides a good mechanism for modeling the way objects appear in different views of the real world (section 4.1.4). The IFC's objectified relationships may reasonably be used to represent information about *contextual* roles (i.e., information about the way a role or role type is used in a context), but not to represent *non-contextual* roles (e.g., an *estimator type* or an *estimator position*). In other words, as intended, they are useful for *assignment* of roles and role types to an object. Therefore, this research suggests a separation of the two concepts of *role* and *relationship*.

Although the separation of the two levels of abstraction (i.e., contextual and non-contextual) may add to the complexity of a model, it offers several advantages. Most importantly, this approach reflects and supports the way business processes occur. For example, project managers commonly maintain a contact list of potential actors to be selected as needed (e.g., assigning a supplier to a purchase order or request for quotes); i.e., selecting from non-contextual objects and assigning them to some context of concern. Moreover, this approach provides for readability and modularization (and thus, flexibility) of the model; for instance, changes to objects of any level may be performed without affecting other objects. The next section suggests a set of guidelines for modeling roles in an AEC/FM core model.

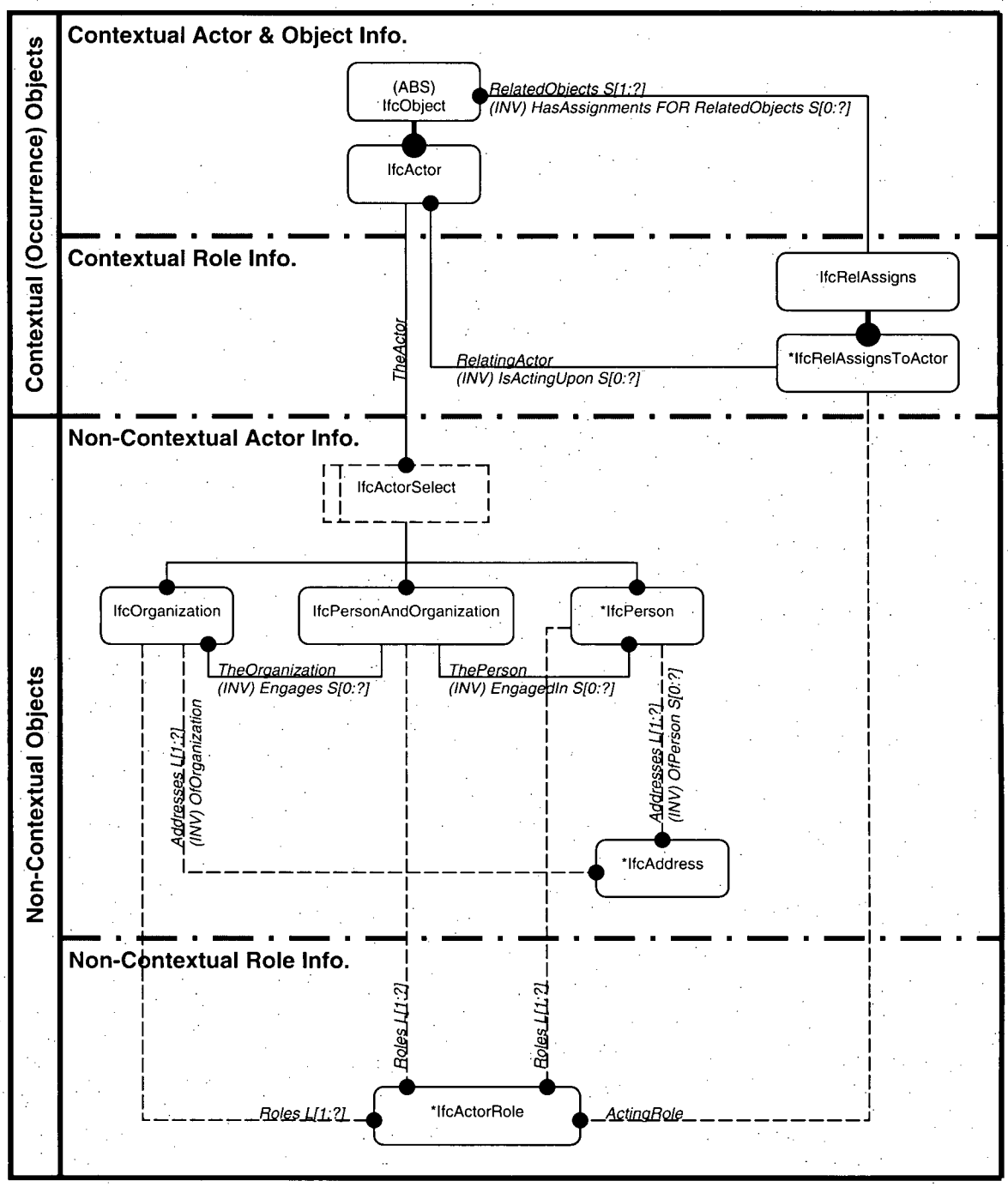


Figure 4–20: Contextual and Non-Contextual Actors and Roles in IFC R2x

4.2.3.4 Some Guidelines for Modeling Roles

Considering the discussions presented in previous sections, the following generalizations may be made regarding roles and role players in AEC/FM projects:

1) The notion of *role* can be extended beyond organizational roles and be applied to any *thing* (i.e., actors, artifacts, and natural elements).

2) Things may play zero to many roles in different situations; i.e., roles are dynamic. The following are some examples:

   a) A *person* may appear in a building project as an *engineer*, an *estimator*, and a *subcontractor* in different projects or stages of a construction project.

   b) A piece of *formwork* can be considered as a *product* of the activity *build formwork*, while it appears as a *resource* of the activity *build walls*.

   c) A *drawing* is a *product* for a draftsman who produces it. As soon as the drawing becomes available to the contractor, it may be considered as an information *resource*.

   d) A *roof* may be considered as a *product* in a roof design or construction process, an external entity (or *agent*) in a column design or lighting-fixture design, and an *asset* in an asset management or maintenance process.

3) Things may play more than one roles at the same time. For example, a *person* may act both as a *crewmember* and as a *foreman* in a construction process.

4) Roles may be of concern, and thus be modeled, in two major ways:

   a) *Non-contextual role*: is when no specific business context is of concern for a role; e.g., a list of organizations, which can potentially act as actors in projects, may include company ABC, which is a *supplier* of doors.

   b) *Contextual role*: is when the role of an object is considered in a specific business context; e.g., company ABC acting as *supplier* of the specific doors used in project 123. In this case, information about the role (e.g., supplier) is associated with the context (i.e., project 123's doors).

5) The same distinction as the *role* may be considered for the *role player*. Examples of non-contextual and contextual things include an occurrence of a building in abstract versus the building as a subject of a rent transaction, and a piece of equipment waiting on the site versus the equipment working in a construction task.

4.2.3.5 Summary of Roles of Objects

This section presented a discussion on modeling roles of objects as a solution to the view issue (section 4.1.4) in an AEC/FM core model. It searched for the notion of role in the IFC model and suggested the generalization of organizational roles into a higher level to model roles of any other objects than actors. Some guidelines were also suggested in this regard. The next section elaborates on *transactions* in an AEC/FM core model.

4.2.4 Transactions in Construction Projects

Elaborating on the importance of transactions in communicating project information, this section explains how transactions may be modeled in an AEC/FM core model as a means of recording the historical data about project processes and their elements, and thus managing changes.

4.2.4.1 Process-Centered Construction

Processes have traditionally been the focal point in managing construction projects. PM software (e.g., estimating and scheduling) have built all project information around processes; i.e., explicit representation of processes attached with other project information (product, resource, cost, time, etc.). In scheduling, for instance, activities are defined as a central information unit with their temporal information (start, finish, float, etc.). Then, the requirements (labor, material, money, etc.) for their fulfillment are expressed for planning and controlling of the requirements [Hendrickson and Au 1989].

Construction projects are heavily process based, and historical information plays an important role in their management. Controlling functions, to a large extent, rely on the data collected during execution and monitoring. As-built data is compared against as-planned data in order to modify and redirect current plans for future [Ahuja 1984]. In claim analysis, a contractor may request for compensation from the owner for the cost incurred as a consequence of a delay in responding to a Request For Information (RFI), or for an unexpected change in soil conditions during an excavation task. An efficient processing of such claims without detailed information about the incidents would be nearly impossible [Jergeas 1989]. A review of traditional book-keeping and filing cabinets reveals the importance of the concept of *transaction* in creation of historical records of business activities, in the form of binders of purchase orders, delivery slips, sales-receipts, rentals, memos, and so on.

4.2.4.2 The Need for Standardization of Transactions

Information is communicated in processes through transactions; thus, standardization of transactions is a key to AEC/FM information standardization and software interoperability. The term *transaction* is defined as "transacting (i.e., conducting or carrying out business, especially between two people)" and "piece of business transacted (e.g., transactions on the stock exchange)" [Oxford 1989]. Transactions are usually named by a *noun* (e.g., a *sale* transaction, corresponding to a *selling* process). Examples of transactions are *Memo*, *Notification, Purchase Order*, and *Request For Test*. Considering their types of information attached, the *process* (section 2.1.1.2) and *transaction* concepts share some basic characteristics. For example, both concepts carry the notion of time and progression, have a responsible party (i.e., performer), and use resources.

Despite the fact that most PM software is modeled around processes, much of the AEC/FM modeling effort (Chapter 2) has concentrated on the standardization of information and less on the processes in which the information flows [Eastman 1999]. Although processes may seem to be unique to each project (given the contractual, environmental, and managerial context of the project), there are some generic processes that are common to all projects. For example, projects usually go through processes such as bidding, solicitation, procurement, cost control, etc. There are usually some typical processes involved in managing an undertaking whether it is a design, construction, or facility management task. There is a need for the standardization of processes and their related information.

Identification of generic AEC/FM processes and their flows helps identify process elements such as documents (i.e., the physical means for business transactions) and their related information contents. For example, a generic *RFI* process usually involves filling an *RFI transaction* by a requester (e.g., a subcontractor) to inquire information about an object. The RFI, which may be generated in different ways and contexts (e.g., automatically by a computer program or manually by a sub/general contractor or a project manager), would reach a receiver (e.g., the object's owner) and would be desirably followed by a response on the requested information. A *Request For Material* process is principally similar to an RFI process, but with the subject of *material* instead of *information*. Defining the whole process of *Acquire Material* would help in identification of a chain of such business transactions as *Request For Material*, *Acknowledgement*, *Request For Quote*, *Quote*, *Purchase Order*, *Pending Order*, etc. Such transactions and their information contents are commonly used and referred to by many processes throughout the project life cycle [Stukhart 1995].

The ultimate goal of most CIC projects is to enable interoperability among AEC/FM software application systems. With the opportunities offered by the Internet and XML (www.w3.org/XML) for e-business [Timmers 2000; Trus and Collica 1995; Turban et al. 2000], the exploitation of the content of the IFC object model (section 2.4.4.7) into the physical XML file format has recently been added to the scope of the IAI project [IAI-NA 2005]. This has resulted in such initiatives as the aecXML (section 2.3.5), which requires standard AEC/FM business transactions. The concept of transaction, however, has not been explicitly defined in the IFC model, which is intended to serve as a common project information model.

Transactions convey the history of projects; thus, standardization of AEC/FM transactions is essential to the CIC movement. It is through such transactions that a project's objects are created or their related data are changed, recorded, used, and transferred. Therefore, standardization of transactions is a step towards solving the change/versioning issue discussed in section 4.1.5.5. The concept of transaction should be defined in an AEC/FM core model, which is intended to be used as a means of integration and

interoperability among AEC/FM software application systems. An object's historical information (i.e., changes) can then be captured through a link between the object and the transactions in which the object has been involved. The next section explores the IFC model for the notion of transaction.

4.2.4.3 Transaction Information in the IFC Model

The IFC model [IFC R2x, 2000a] does not refer to the term *transaction*; however, it suggests a number of objects that may fit into the definition of a transaction. The first group of objects falls under the specialization hierarchy of the abstract object *IfcControl* (a thing that controls or constrains product or process objects within an AEC/FM project), which is defined in the *IfcKernel* schema. Figure 4-21 shows this hierarchy (based on the *IfcMgmtElements* schema), together with details of a few of the objects, in order to describe the extent of information captured by such objects.

The *IfcApprovalResource* schema is another source of transactional information. In this schema (Figure 4-8), the *IfcApproval* object is declared as "information about approval processes for a plan, a design, a proposal, a change order, etc, in a construction or facilities management project." Being referenced by *IfcApprovalUsage* in *IfcControlExtension* schema (Figure 4-21), it can be related to the subtypes of *IfcObject* through appropriate objectified relationship declared in *IfcKernel* schema.

The IFC object model falls short in specifying all AEC/FM transactions. Even for the part that has been dealt with, a more consistent approach would be required. For instance, despite the fact that the *IfcApproval* object, like other transactional objects (e.g., *IfcChangeOrder* or *IfcPurchaseOrder)* capture information such as time, involved actors, and status, it is treated differently than other transactions (e.g., in terms of its location in the layers and specialization hierarchies of the model). The various types of transactions should be studied and modeled under a single concept of *transaction*. The next section suggests some guidelines for modeling transactions and their associated information.

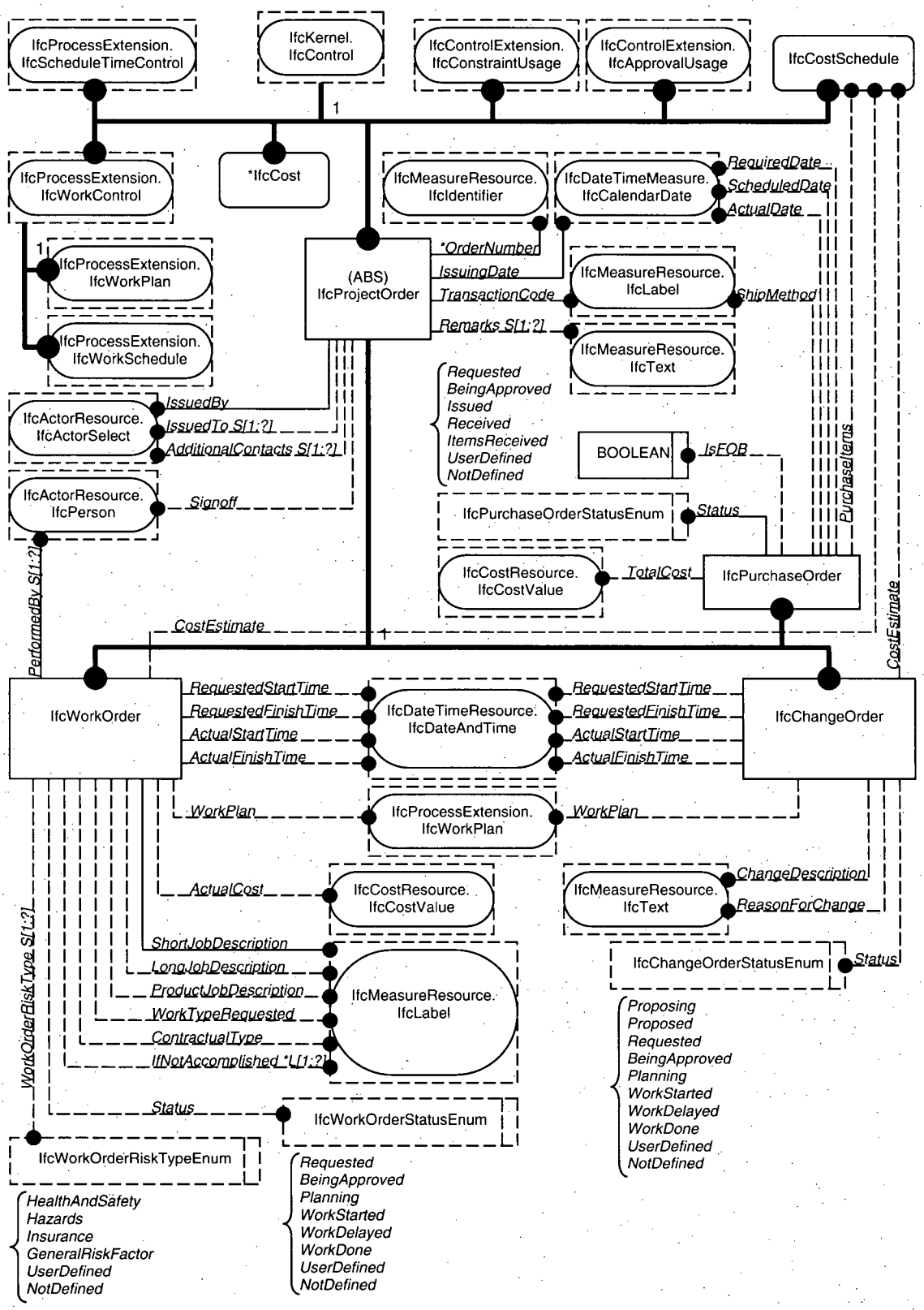


Figure 4–21: The Specialization Hierarchy of IfcControl in IFC R2x

4.2.4.4 Some Guidelines for Modeling Transactions

As a synthesis of the discussions presented above, this section presents some guidelines that represent a synopsis of the views and the approach of this research to modeling transactions. Transactions are created throughout processes and share some *basic characteristics*, as summarized in the following:

1) The concept of transaction can be considered at two occurrence and type levels (section 4.2.2).
2) An occurrence of a transaction is usually uniquely identified (e.g., a *purchase order number* for a *purchase order*), and this is critical to the performance of project processes (i.e., *shipments*, *deliveries*, and other transactions may reference the purchase order through the identity number).
3) A transaction has some *intent* or *purpose* (e.g., a *request for material* sent to *attain some material*).
4) A transaction usually has one or more *line items* as its subjects. Each *line item* may reference an occurrence object (e.g., a *specific column* referenced in an *inspection report* or in an *order for correction* sent to a constructor) or a type object (e.g., a *material type* in a *purchase order*).
5) A transaction is initiated by an actor/role in the project (e.g., a *request for material* by a *requester*—i.e., a foreman, a superintendent, etc.).
6) A transaction is usually exchanged between two or more parties; i.e., it is sent by a *sender* to a *receiver* through a medium.
7) A transaction might result in one or many subsequent actions or transactions (e.g., a *request for material* resulting in a *rejection* or in a *purchase order*; an *order* resulting in one or many *shipments* and *deliveries*, etc.).
8) A transaction can be the result of many other transactions (e.g., many *invoices* paid in one *payment*).
9) A transaction can be planned (e.g., a material *delivery* is usually planned or promised to happen at a specific time) and controlled. A *planned* (or promised) transaction can have zero or many *actual* transactions (e.g., a *planned delivery* might not happen on time).
10) Considering the above, a transaction can have many versions. This versioning may be as a consequence of a change in the subject or other properties of the transaction (e.g., reduction of the suggested quantity of a *purchase order*'s items after its approval, or a change in *time* of a *delivery*).

In addition, the following is a list of some other considerations for modeling transactions.

1) <u>*Relation With Other Basic Concepts*</u>: There exist relationships between the concepts of *transaction*, *process*, and *product* (i.e., the thing). For example, there are many ways to model information about the *built year* of a *Building* object. To name a few, the information may be directly attached to the *Building* object (i.e., as an attribute). It may alternatively be derived from the temporal information attached to the project's processes (e.g., the *finish time* of the *Construct Building* process), or from the

project's transactions (e.g., the *date* of the transactions created at the closeout stage of the project; i.e., *Certificate of Substantial Completion*, *Certificate of Occupancy*, etc. [Mincks and Johnston 1998, pp. 384-389]). The latter approach would highlight the importance of transactions as a central repository of information for other project objects.

2) <u>*Generalization of the Transaction and Process Concepts*</u>: Considering the similarities of transaction and process information (section 4.2.4.2), the two concepts may be generalized into the concept of *event* (i.e., something that happens and is important to remember [Oxford 1989]).

3) <u>*Discrete versus Continuous*</u>: Depending on their usages, transactions may be viewed as either a discrete *event* (i.e., moment-oriented; e.g., a sale, an order, etc.) or a *continuous event* (i.e., within a time interval, i.e., with start and end; e.g., a lease, a rent contract, a reservation, etc.). For the same reason of usage, however, a transaction may be arbitrarily moved from one category to another. For instance, for performance assessments purposes, a sale may be even placed in the second category.

4) <u>*Levels of Abstractions*</u>: Transactions may be of concern at different levels of abstractions. Different data (e.g., Material = lumber 2x4, Quantity = 500, Unit = pieces) are attached to a *logical transaction* (e.g., a *logical Purchase Order*), which is cast into a *physical transaction* (e.g., a *Purchase Order File Document* in the form of an EDI document or an XML file). As a human-interpretable means, the content of such a file may be presented in a *physical transaction medium* (e.g., a physical, human-interpretable, *Purchase Order Document or Form*).

Based to their characteristics, project transactions may be identified, categorized, and modeled as the basic ingredients of an AEC/FM core model. This research focuses on identification and definition of *logical transactions* and their *information contents* through the study of *business processes* and their involved *physical transaction mediums* (i.e., *documents*).

4.2.4.5 Summary of Processes and Transactions

This section presented a discussion of transaction information as a basic ingredient of an AEC/FM core model and as a means of managing projects' historical information and changes (section 4.1.5.5). Overall, this research argues that formalization and standardization of project transactions is an important step in AEC/FM project information standardization, and this may be better facilitated through the study and modeling of AEC/FM project processes. Current AEC/FM models (e.g., the IFC model) do not deal with modeling transactions profoundly. The modeling guidelines presented in this section are used as one of the bases for developing the core model suggested by the research in chapters V. Moreover, chapters 6 through 8 describe the approach of this research to identification and specification of project transactions through studying and modeling project processes. As a wrapper to the analytical

materials presented so far in this chapter, the next section presents an examination of the basic data structure of the IFC model.

4.2.5 An Examination of the IFC Object Model Using Coad's DNC Model

This section examines the consistency of object definitions at the highest level of the IFC model [IFC R2x, 2000a] from the perspective of Coad's DNC model [Coad et al. 1999] (Appendix A). For this purpose, Figure 4-22 uses a mix of EXPRESS-G notation and UML's stereotype notation (Appendix A) representing the archetypes suggested by the DNC model to distinguish the basic objects defined at the highest level of the *IfcKernel* schema of the IFC model. The following summarizes some of the results of this examination.

1) Considering the definitions suggested in the IFC model, some objects are difficult to be categorized under only one of Coad's archetypes. The sterotype notation "«?????»" is used to denote such cases.
2) Except for organizational roles (which are in the Resource Layer of the IFC model and are not shown here), the model does not explicitly model the concept of *role* (section 4.2.3.2). In other words, no «role» Archetype (yellow) is observed in the Kernel schema of the model.
3) Not all seven subtypes of *IfcObject* (an occurrence of "any semantically treated thing or process", i.e., a «thing» or «moment-interval» archetype) represent an occurrence object (section 4.2.2.3). This contradicts Coad's suggestion ("Challenge Multi-Colored Inheritance"):

   "*Challenge any subclasses that belong to a class archetype different to that of their superclass. Inheritance is an extension mechanism. An object of a sub-class is the same sort of thing as an object of the superclass. It is nonsense to have a subclass belong to a different archetype than the class it extends*" [CoadLetter #77, 2001].

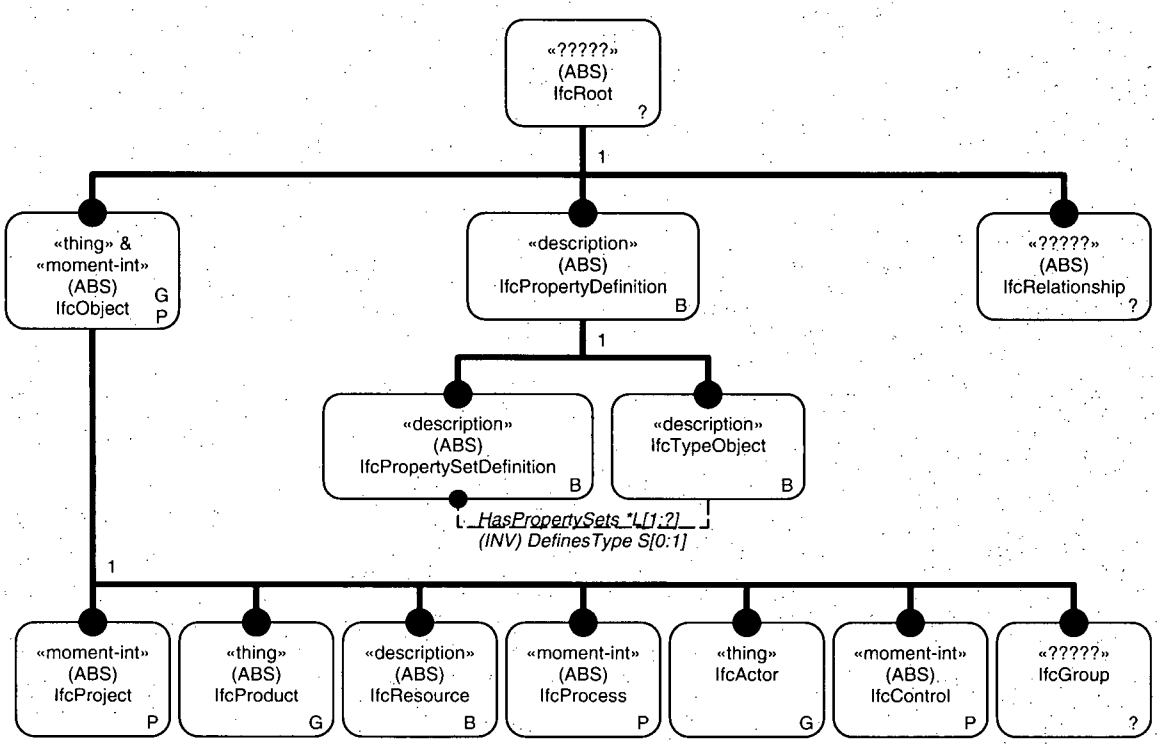


Figure 4–22: A Partial View of the High-Level Objects in the IFC Model

4.2.6 Summary of Modeling Beyond Product Information

This section stepped beyond information modeling specific to AEC/FM products and discussed some more generic information modeling issues that are critical to modeling various types of AEC/FM project information, including objects properties and relationships, type and occurrence information, roles of objects, and transactions. The discussions were coupled with an analysis of the IFC's approach to modeling such information as well as suggesting a set of guidelines for addressing the issues. The results of the discussions and analyses highlighted a need for a conceptual AEC/FM project information core model that could address all the identified issues by applying the advantages offered by existing models and the guidelines presented throughout this section. Such a model is presented in the next chapter.

4.3 Chapter Conclusions

Overall, this research argues that while every PM function involves a distinct view of project information, it is possible to bring all views together into a unified AEC/FM project model, which integrates both product and process information to provide interoperability among PM systems. Building on the background information presented in previous chapters and by the means of exemplification and references to existing models, this chapter addressed some of the major information requirements of AEC/FM processes that need to be reflected by an AEC/FM core model.

Focusing on product modeling, it first formulated product information within the FFB/P aspects and the environment within which products are of concern. It also showed how FFB/P data is viewed, shared, and used by PM functions, and whether this is supported in current AEC/FM product models. Stepping beyond product modeling, a broader range of project information modeling issues and requirements (e.g., objects properties and relationships, type and occurrence information, roles of objects, and transactions) and the IFC's approach to addressing such issues were discussed, and some guidelines for resolving the issues were suggested. Altogether, the analyses and discussions lead to this conclusion that none of the existing product models capture all aspects of objects. Most models portray a designer's view and, thus, may not fully support a full range of PM functions. This confirmed a need for an AEC/FM project core model that could address all the identified issues by applying the guidelines presented throughout this chapter while benefiting from the advantages offered by existing models.

Among specific suggestions made in this chapter regarding the requirements of an AEC/FM core model are: 1) Generalization of the FFB/P aspects to define properties of both product and process-related objects; 2) Handling objects relationships in a manner to provide flexibility while avoiding complexity (e.g., defining contextual and external properties); 3) Extension of the type and occurrence concepts from products to process-related objects; 4) Representation of the role concept as a means of representing various views of PM functions on project information; and 5) Formalization and standardization of project transactions through the study and modeling of AEC/FM project processes, and representation of the transaction concept as a means of recording and managing objects' historical data and changes.

The concepts suggested in this chapter may not introduce new technologies, per se, but rather encompass an extension of the concepts presented by the well-established ones for the purpose of improving their shortcomings in responding to the problems identified in the investigation processes of the research. For instance, while applying the FFB/P concepts for definition of product information is not revolutionary by itself, the extension of the concepts for modeling both *product* and *process* information is novel. Another example is the concept of role, which has commonly been used in models for representation of persons' roles but has not been applied in other contexts (e.g., roles of physical objects such as buildings, equipment, and so on). The identification of such concepts and their potentials to represent AEC/FM project information is considered as a major outcome of this chapter. Altogether, the materials presented in this chapter are envisioned to provide a body of knowledge for standardization of AEC/FM information for development of *model-based integrated total PM systems*. The next chapter describes a conceptual AEC/FM Project Core Model (APCM) that is built on the findings of this chapter.

CHAPTER 5 A Conceptual AEC/FM Project Core Model (APCM)

This chapter addresses the second part of the research question (section 1.4.1): *How could the information requirements of PM functions be formalized into a conceptual AEC/FM project information model*? It represents a synthesis of the analytical materials presented in the previous chapter. In order to address the AEC/FM project information modeling issues and requirements identified, a conceptual project information model, called the AEC/FM Project Core Model (APCM) was developed in this research. This chapter describes the scope, structure, and characteristics of the APCM.

5.1 The Scope of the Model

The scope of the APCM generally follows the overall scope of the research (section 1.4.2). The APCM is intended to serve as a conceptual product-process-integrated core model defining the core concepts of AEC/FM projects. It expresses the information required for managing a typical AEC/FM project across all segments of the industry, not for a specific type of project. For example, it defines concepts such as *project*, *product*, and *process* but not *building* and *stair*, which are specific to the building industry. The model focuses mainly on the high-level business objects (section 2.2.2) shared among various AEC/FM application systems, though it is open and extensible to adding lower-level objects and their attributes.

5.2 The Basic Structure of the Model

Using the EXPRESS-G notation (Appendix A), Figure 5-1 depicts the high-level objects of the APCM together with selected conceptual relationships. At the highest level, an abstract object called ***AECFM Object*** is defined as a generalization of any kind of AEC/FM-project-related information (i.e., anything that exists, real or imagined). As the root of the specialization hierarchy of the model, it is specialized into four major objects:

1) ***Occurrence Object:*** a generalization of anything that occurs (exists) and for which useful information exists. It can be a physical or logical thing, an event, or a role. It is uniquely identified (e.g., with a serial number), recorded, and tracked (section 4.2.2.1) for business or legal reasons.
2) ***Type Object***: a catalog-entry-like description that holds some information that can be applied to a collection of objects (e.g., a *door type* describing a number of individual occurrences of physical *doors*). The objects in the associated collection can be of type *Occurrence Object* or *Type Object*.
3) ***Property Definition:*** the generalization of characteristics of any type of AEC/FM object.
4) ***Relationship:*** a generalization of information about how two objects may be related.

The following sections describe the subtypes of these four primary classes.

5.2.1 Subtypes of Occurrence Object

The *Occurrence Object* is specialized into three basic objects:

1) ***Thing:*** the generalization of anything that physically or conceptually exists, is uniquely identified (e.g., with a serial number), and may play various roles in different situations. It can be a physical object (e.g., a building, a person, a tree, a space), a conceptual object (e.g., a job, an account), or an arbitrary group of things that is conceptually defined (e.g., an inventory of items, an asset).

2) ***Role:*** the way that a *Thing* (or one of its subtypes) participates in an *Event* (defined below). A *Space*, for example, may play the role *Product* in a design process, while it may appear as a *Resource* in a material storage process. Therefore, a *Role* is usually defined in relation with a *Thing* and/or an *Event*. Roles are dynamic, and roles of things can change dynamically from one event to another. Each *Thing* can have zero or more *Roles*, and each *Role* is defined for zero or one *Thing*. Also, each *Role* can occur in zero or more *Event*, and each *Event* can have zero or more *Role*. The major characteristic of a *Role* (e.g., a *product role*) is that it is not sizeable (i.e., has no geometry, topology or quantification), while a *Thing* (e.g., a wall) that plays that role can potentially have all FFB/P aspects (including *form* characteristics). On the other hand, the role is *uniquely identified* (like the thing), and it may have *status* (e.g., the product is "designed").

3) ***Event:*** the generalization of any *happening*; i.e., something that takes place (at one moment or within a time period) and is important and interesting enough to be recorded and tracked for business or legal reasons (i.e., an occurrence of some importance). *Events* may be of concern in various businesses, natural, social, or technical contexts. Moreover, an *Event* usually entails involvement of an *AECFM Object* (often a *Thing*). Examples include a work task resulting in a product, a contract resulting in construction of a building, a purchase order resulting in some materials, and a meeting of project parties resulting in a meeting report or a memo. Considering the definition and these examples, *time* and *function* may be regarded as two major aspects of this object. The *Event* object has a central role in the integration of all other objects since it creates the context within which *Things* play *Roles*.

5.2.2 Subtypes of Type Object

The *Type Object* is specialized into four objects:

1) ***Thing Type:*** information that describes a collection of *Thing* objects; e.g., a *door type* object describing a collection of *door* occurrences (e.g., more specifically, some specific *sliding steel door* occurrences described by a *door type* object).

2) ***Role Type:*** information that describes a collection of *Role* objects; e.g., a *designer type* object describing a collection of *designer* occurrences (e.g., more specifically, a collection of *structural designer* roles described by a *designer type* object); note that here, a *designer* is a *role* rather than a tangible actor or person.

3) ***Event Type:*** information that describes a collection of *Event* objects; e.g., a *process type* object or a *transaction type* object describing a collection of *process* or *transaction* occurrences, respectively.

4) ***Generic Type Object:*** information that describes a collection of *Type Objects* and/or *Occurrence Objects*; e.g., a standard *generic material type* (versus a *brand-name material*) that may be referred to in a project's specifications to be used for one or more *building elements* of the project.

5.2.3 Subtypes of Property Definition

Considering the levels of abstractions dealt with in modeling properties, the abstract object *Property Definition* is specialized into three objects:

1) ***Aspect:*** a set of properties that describes a particular view of an object. It denotes the three fundamental categories of objects properties: *Form*, *Function*, and *Behavior/Performance* (section 4.1.2). An *Aspect* can include one or more properties.

2) ***Property:*** a specific characteristic that can be used to describe an object (e.g., length, count, name, and so on). A *Property* may be associated with an object directly or through an *Aspect*.

3) ***Property Measure:*** an abstract generalization of the measures used to quantify or qualify characteristics of objects (represented by the *Property* object). Examples of such measures are *values* and *units* (e.g., a value of *2.70* with the unit *meter* for the length characteristic). The *Property Measure* object may be used directly by attributes of objects or through the objects *Property* and *Aspect*. The assignment of a *Property Measure* to a *Property* is established through a unilateral link from the latter to the former.

5.2.4 Subtypes of Relationship

The *Relationship* object is specialized into six basic relationship objects, which capture information about various types of relationships between non-relationship objects:

1) ***Aggregation Relationship:*** captures (physical or logical) whole-part relationship between occurrence objects or type objects (i.e., *occurrence-to-occurrence* or *type-to-type*).

2) ***Assignment Relationship:*** relates two occurrence objects (*occurrence-to-occurrence* relationship) or a type object to an occurrence object (*occurrence-to-type* relationship). This relationship represents a

simple link between two objects, and it does not imply or establish any constraint between them (in contrast with the *Connection Relationship*).

3) ***Association Relationship:*** captures a unilateral reference from an occurrence, type, or aspect object to an external source of information such as a classification, standards, and so on (i.e., an *occurrence/type/aspect-to-external source* relationship). The referencing object would then be dependent on the referenced object.

4) ***Description Relationship:*** describes an occurrence or type object in terms of a type object (*occurrence-to-type* or *type-to-type* relationship).

5) ***Definition Relationship:*** relates an occurrence or type object or an aspect to an aspect (*occurrence/type/aspect-to-aspect* relationship).

6) ***Connection Relationship:*** represents a physical or logical relationship between occurrence objects or between type objects (i.e., *occurrence-to-occurrence* or *type-to-type* relationship) under some *criteria* or *constraint*. The objects that are related using this relationship may be of the same or different types (e.g., adjacency of two buildings versus supporting of a beam by a column). The definition of the criteria plays an important role in defining the semantics of the relationship it represents. The subtypes of this relationship should define the criteria. Examples of various types of such relationships are *Embedding* (Embeds/Embedded-In, e.g., a concrete foundation and its reinforcement, a stud or concrete wall and its embedded electrical conduit); *Coverage* (Covers/Covered-By, e.g., a wall covered by paint and a graded road covered by paving); *Supporting* (Supports/Supported-By); *Containment/Inclusion* (Contains/Contained-In, e.g., a material stored in a space); *Protection* (Protects/Protected-By/Protected-From, e.g., a sheet of polyethylene protecting a façade from the rain); *Filling* (Fills/Filled-With, e.g., a caulking filling a window's air gap); *Joining* (Joins/Joined-By, e.g., an expansion joint between two blocks of concrete slab); *Adjacency* (Constrains/Is-Constrained by Adjacency, e.g., a neighbor building adjacent to a construction site); *Sequencing* (Precedes/Proceeds, e.g., sequencing of tasks in scheduling), and so on.

5.3 Lower-Level Objects of the Model

The basic objects of the APCM defined above are specialized in lower levels of the model, as explained in the following sections.

5.3.1 Subtypes of Thing

Based on its materialization status, three subtypes are defined for the *Thing* object (Figure 5-1):

1) ***Physical Thing:*** a generalization of any occurrence object that physically exists (i.e., *materialized*). Examples include a building and a space. Objects of this type can potentially be described by their

form, in addition to other properties. This is a major aspect that distinguishes a *Physical Thing* object from a *Non-Physical Thing* object.

2) ***Non-Physical Thing:*** a generalization of any conceptual thing, which may not be materialized and is not sizable; i.e., it may not be capable of being perceived by human senses (e.g., sight, touch, hearing, smell, and taste); however, it is uniquely identified and tracked in AEC/FM processes. A *Non-Physical Thing* is not sizeable; therefore, no *form* characteristic may be imagined for it. Examples of possible subtypes for this object are *Project*, *Account*, *Responsibility*, *Position*, *Job*, *Cost Center*, *Address*, and so on.

3) ***Thing Group:*** a set of objects that are arbitrarily selected and logically grouped (i.e., to be treated as a whole) for a specific purpose. A *Thing Group* can include objects of different types, and membership is not exclusive (i.e., one object can appear in more than one group). A *Thing Group* can include other groups (i.e., groups can be nested). Examples of possible subtypes for this object include *Crew*, *System*, *Zone*, *Inventory Items*, *Asset*, and so on.

5.3.2 Subtypes of Physical Thing

Based on its tangibility, the *Physical Thing* is specialized into two subtypes (Figure 5-2):

1) ***Tangible Thing:*** an occurrence of a physical thing that is capable of being perceived by the sense of site or touch (i.e., materialized visually). Of the major characteristics of this object is that it has *form* aspects (i.e., shape and topology), and it occupies space. Examples of such objects include a building, a person, a tree, and so on. Based on their nature, the subtypes of the *Tangible Thing* include (Figure 5-2):

   a) ***Actor:*** an individual person and/or organization involved in AEC/FM processes.

   b) ***Artifact:*** a generalization of any physical, tangible thing that is made or created by an artificial process (i.e., by human or machine) to serve a particular purpose. It can be a part or the whole of a fixed or moveable constructed facility (e.g., a building, a road, a mobile house, or a tent), construction material, construction equipment, temporary equipment structure (e.g., formwork, scaffolding structure, and so on), physical document, and so on.

   c) ***Natural Thing:*** an observable fact or occurrence that may be perceived by senses in nature. Examples of the objects that may be considered under this object are: *Land*, *Plant*, *River*, *Rain*, *Snow*, *Flood*, *Earthquake*, and so on (Figure 5-2). This object is concerned with the *physical thing* (e.g., the rain, the flood, etc.) not the related *event* (e.g., the event of raining, flooding, etc.), which can be represented with the *Event* object.

d) ***Byproduct:*** a generalization of any substance produced (usually unwontedly) during the making of something else (i.e., an *Artifact*), which had been the main purpose of the production process. A *Byproduct* may play various roles (such as a disposal, or a raw material) in different processes.

2) ***Space:*** a physical area or volume bounded actually (i.e., by some tangible artifacts) or theoretically (i.e., by convention) and can be occupied by some tangible things. A space is usually defined in terms of its boundary, which may be defined by some tangible artifacts, and can be decomposed. Examples include a room or a floor in a building, an open area on the site, and an air space in a cavity wall.

5.3.3 Subtypes of Role

Some of the key subtypes of the *Role* object (shown in Figure 5-1) are as follows:

1) ***Actor Role:*** a role considered for an *Actor* object (i.e., for a real person and/or organization). Examples of such roles include employee, employer, salesperson, supplier, and designer. An *Actor* may play one or more *Actor Roles*, and an *Actor Roles* may be played by one or more *Actors*.

2) ***Product Role:*** the role of a *Physical Thing* (usually an *Artifact*) as being produced. The role is used for the ultimate output thing of a process. For example, a road may be considered as a product of a construction contract, while a set of drawings may be considered as a product of a design contract.

3) ***Place Role:*** the generalization of a role played by a *Constructed Facility* or a *Space* (i.e., a place). The role played by a place is directly related to its *use*; i.e., the *function* expected from or served by the place. For example, a building may play the role of a *house*, a *hospital*, a *school*, a *storage facility*, and so on. Similarly, a space (e.g., a site) may serve as a *parking lot*, a *material laydown area*, and so on.

4) ***Resource Role:*** a role used for a thing as being acquired, used, or consumed (wholly or partially) in the process of developing a product thing. Examples include process-aid resource, human resource, equipment resource, material resource, money/capital/fund resource, computer resource, crew resource, and so on. A *Resource Role* may be considered as an aggregation of other *Resource Roles*. For example, if *human resource* is of concern, it may be considered as an aggregation of some *Actor Roles*. Similarly, a *crew resource* may be considered as an aggregation of some *Actor Roles* and *Resource Roles* (e.g., equipment resource). A *Resource Role* is usually defined relative to a process and/or thing (i.e., the role player), thus it can be associated with the process and/or thing.

5) ***External Agent:*** a role used for something that affects, constrains or is capable of affecting something else. Both the affecting and affected objects may be of type *Thing* (e.g., an *Actor*, an *Artifact*, or a *Natural Thing*) or *Event*.

6) ***Other Roles:*** serves as a placeholder for other types of roles. Examples include *input*, *output*, *sink*, *source*, and so on.

5.3.4 Subtypes of Event

Based on its context, four basic subtypes are defined for the *Event* object (Figure 5-1):

1) ***Process:*** the generalization of information about an occurrence of an action or series of actions or operations performed to do, make, or achieve something. This handles the idea of work being carried out over a period of time. The acquisition (e.g., renting, leasing, buying) and development of things are among the major common forms of processes in construction projects.

2) ***Transaction:*** the generalization of information about an occurrence of a *business event*. The business event may range from managerial to operational events, and it may involve planning, dealing, ordering, record keeping, announcing, evaluating, etc. of a thing. Examples include a request, a purchase order, an acknowledgement, a change order, a work schedule, and a meeting. The *Transaction* object is different from a *Document* object. While the former captures the semantics of the information about a business event, the latter represents the physical means of communication and exchange of the information. The *Transaction* object is a means of capturing historical information about various objects in a project. A *Transaction* may be nested. For example, a submittals collection being a composition of product data, materials sample, and shop drawings.

3) ***Phenomenon:*** the generalization of information about any observable occurrence of a natural, social, economical, and/or technological event. Examples of a natural phenomenon include earthquake, wind, rain, or flood. Traffic, electric shock, and traffic noise are examples of socio-technical events. A phenomenon such as a raining or flooding *event* is different from the physical occurrence of the rain, or the flood involved in the event. The latter ones can be represented with the *Thing* object.

4) ***Event Item:*** a part of an *Event* that captures information about the occurrence or type item involved in that part of the event. Therefore, it does not exist by itself and is always attached to an *Event*. In fact, the *Event Item* acts as a mechanism for associating *Things*, *Roles*, and *Type Objects* with an *Event*. Each *Event* has one or more *Event Item*, and each *Event Item* belongs to one and only one *Event*. An example is an event item of a purchase order that defines the type, quantity, and required date of a material (i.e., a *Type Object*) that is ordered. The item itself can include various types of information (i.e., an *Occurrence Object*, a *Type Object*, an *Aspect*, *Property*, and/or a *Property Measure)*. The information about the item included in an *Event Item* is captured through links to the related information. In the case of a purchase order, for instance, a link can be established to a *Type Object* (e.g., a material type).

5.3.5 Subtypes of Process

Subtypes of the *Process* object (Figure 5-3) include:

1) ***Task:*** an occurrence of a work item. A task may be production-oriented (e.g., a construction operation such as pouring concrete) or service-oriented (i.e., non-production, e.g., purchasing, delivery, and handling materials, renting equipment, cleaning windows, repairs, and so on). Tasks may be hierarchically nested for various purposes such as scheduling.
2) ***Impact:*** an occurrence of the effect of something on something else. Examples of impacts include impacts of external agents on project objects, impacts of weather conditions on processes, and impacts of material delivery delay on the project's progress.

5.3.6 Subtypes of Transaction

The bottom-up investigations of the research involving the collection and analysis of construction projects documents (Chapter 3) resulted in the identification of many transactions that are of concern in construction projects (section 8.4). Considering their characteristics, these transactions were grouped into thirteen categories, which constitute the abstract subtypes of the *Transaction* object (Figure 5-4):

1) ***Request:*** the action of asking for something formally. Examples include *Request For Information*, *Request For Material*, *Application For Payment*, *Request For Quotation*, and *Request For Proposal.*
2) ***Proposal Or Suggestion:*** the generalization of something that is suggested, offered, or put forward for consideration. Therefore, a *Proposal Or Suggestion* is normally subjected to an evaluation (e.g., review), and at some point in time, it can leads to an *Order Or Agreement* (i.e., its agreed or approved version would be referenced by the *Order Or Agreement*). Examples include *Bid Proposal*, *Quotation*, *Submittal*, and *Change Order Proposal.*
3) ***Order Or Agreement:*** certain terms or conditions of a business arrangement that are agreed upon, instructed, or commanded to come into effect. Examples include *Building Permit*, *Change Order*, *Work Order*, *Purchase Order*, *Confirming Order*, *Blanket Order*, *Pending Order*, *Contract*, *Purchase Agreement*, *Buyback Agreement*, *Addendum*, *Rent*, and *Sale.*
4) ***Plan:*** the generalization of any scheme, roadmap, or program for something that is intended to be done. A *Plan* describes *what* and/or *how* something is intended to be done to achieve some pre-defined objectives. Examples include *Design*, *Specification*, *Work Plan/Method*, *Work Schedule*, *Equipment Maintenance Schedule*, and *Material Delivery Schedule.*
5) ***Evaluation Or Judgment:*** the generalization of forming an opinion on one or more aspects (size, quantity, value, worth, cost, performance, nature, etc.) of something or somebody; i.e., quantification

and/or qualification. Examples include *Quantity Estimate*, *Cost Estimate*, *Appraisal*, *Approval*, *Design Review*, *Submittal Review*, *Field Inspection*, *Shop Inspection*, *Test*, and *Test Result*.

6) ***Certificate:*** the generalization of any statement that formally or officially declares a thing or an event as true, accurate, according to some requirement, etc. Examples include *Certificate Of Compliance*, *Certificate Of Occupancy*, and *Certificate Of Substantial Completion*.

7) ***Allocation:*** the generalization of setting apart or assigning some specific things (e.g., money, materials, space, etc.), as allowance, to somebody or something for a specific purpose. Concerning *Resource Allocation*, an *Allocation* may be of type *planned*, *committed*, or *trial* [Stukhart 1995].

8) ***Movement Or Transfer:*** the generalization of information about any transaction involving the movement or transferring of physical or financial matters (e.g., materials, credits, and so on). Examples include *Delivery*, *Shipment*, *Transmittal*, *Payment*, *Progress Payment*, and *Refund*.

9) ***Gathering:*** the generalization of information about any purposeful coming-together of people (e.g., project participants) to discuss or decide on project matters. Thus, a *Gathering* usually has a *purpose* and some *attendees* and *end results*. Examples include *Meeting*, *Conference*, *Seminar*, and *Congress*.

10) ***Notification:*** the generalization of any formal announcement or warning imparting required or pertinent information. A *Notification* may be informative and/or instructive. Examples include *Acknowledgement*, *Memo*, *Notice Of Acceptance*, *Shipping Notice*, *Approval/Rejection Notification*, *Safety Notice*, and *Stop Work Notice*.

11) ***Report:*** the generalization of any formal statement that provides information on the status and/or results of an event (e.g., an inspection). A *Report* is usually informative, and it may not explicitly require any further action. A *Report* may be used as an attachment to another transaction (e.g., to an *Approval Notification*) to provide details, reasons, and/or background for the content of the transaction. Examples include Receiving Report, *Materials Test Report*, *Daily Site Report*, *Bid Analysis Summary*, *Meeting Minutes*, and *Weekly/Monthly Progress Report*.

12) ***List Or Label:*** the generalization of any statement listing or describing a set of things (e.g., in terms of their nature, name, size, quantity, owner, destination, and so on) for a specific purpose. The purpose dictates the types of information that a *List Or Label* object captures. Examples include *Waybill*, *Packing Slip/List*, *Shipping List*, *Supplier/Vendor List*, *Contractor List*, *Participants List*, *Punch List*, *Spare Part List*, and *Equipment List*.

13) ***Log:*** the generalization of any official recording of events during a particular period of time. Examples of the transactions that may be considered in this category are *Submittal Log*, *Change Order Log*, *Correspondence Log*, *Materials Storage Log*, and *Demurrage Log*.

5.3.7 Subtypes of Aspect

At a very high level of abstraction, the fundamental categories of objects properties (section 4.1.2) are used as a basis for defining the subtypes of *Aspect* (Figure 5-1):

1) ***Form:*** an abstraction of the topological characteristics (i.e., physical structure and configuration, e.g., adjacency and overlapping) and geometrical characteristics (i.e., dimensionality, e.g., size, height, and length) of a thing. Objects of type *Physical Thing* (section 5.2.1) can have such properties.
2) ***Function:*** an abstraction of the activity, purpose, or use of a thing. For example, the main function associated with a house is to *shelter* its occupants, while the main function of a truck is to *haul* a load.
3) ***Performance:*** an abstraction of the behavioral characteristics of a thing with respect to the imposed impacts from its environment. Examples of building element performance are aesthetic/pleasance, cost, price, safety, lighting, durability, ventilation, indoor air quality, heat conductivity, and so on.
4) ***Aspect Group:*** a set of properties arbitrarily selected and grouped to be associated with and to describe an object. It can include any combination of *Form*, *Function*, and/or *Performance* properties. An *Aspect Group* can also include another *Aspect Group* (i.e., nesting of a set of properties).

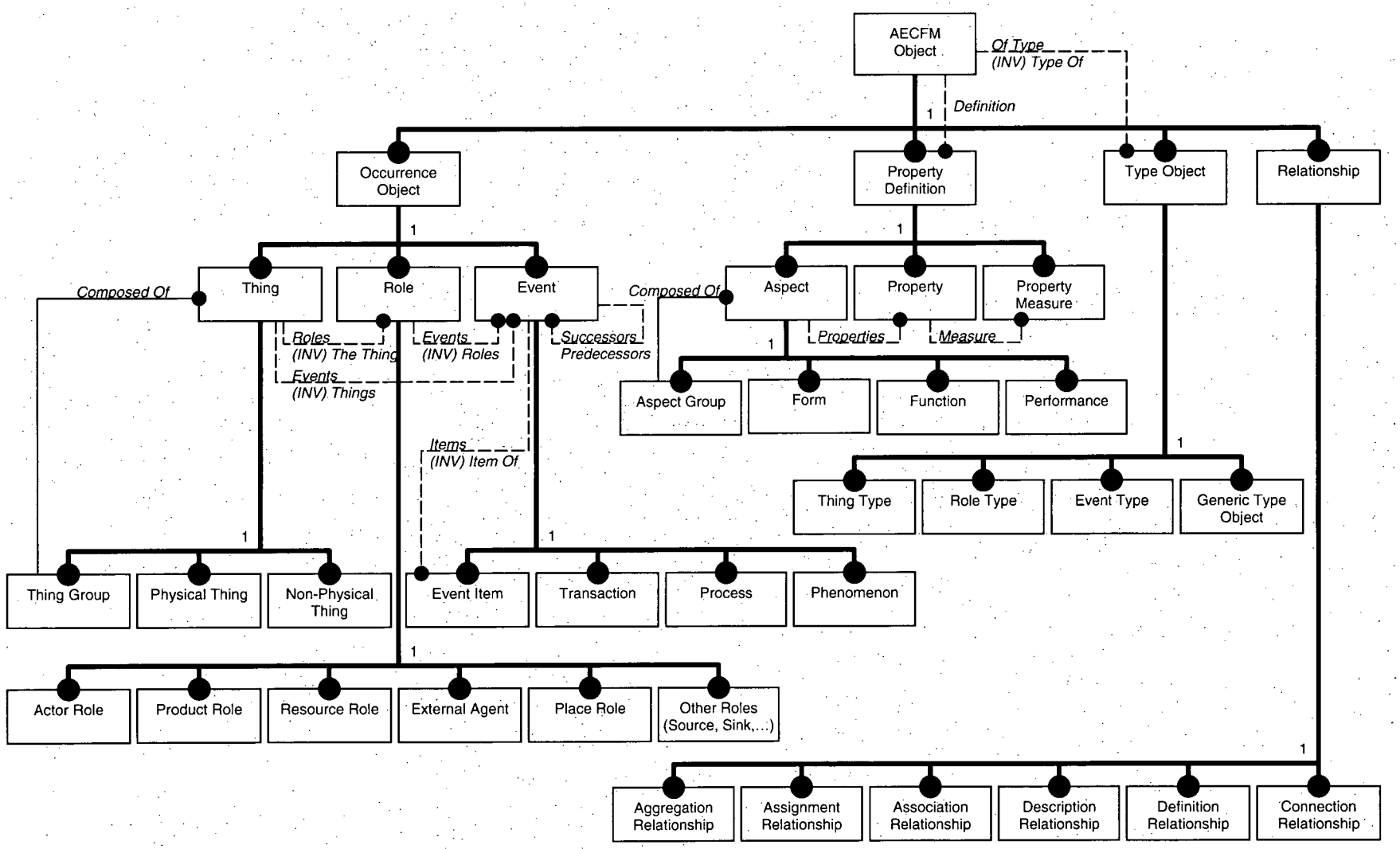


Figure 5–1: The Basic Entities of the Proposed AEC/FM Project Core Model (APCM)

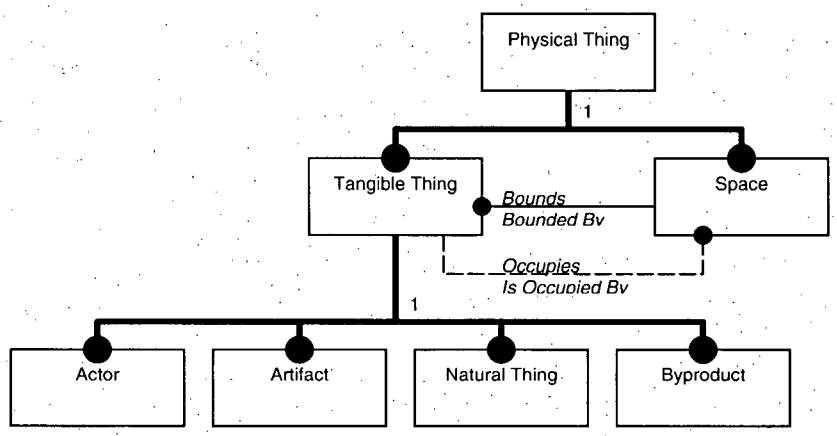


Figure 5–2: The Specialization Hierarchy of the Physical Thing

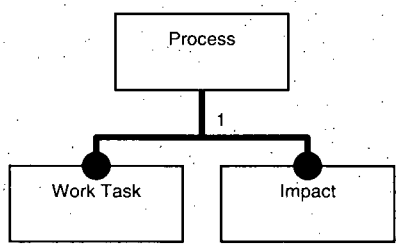


Figure 5–3: Subtypes of the Process

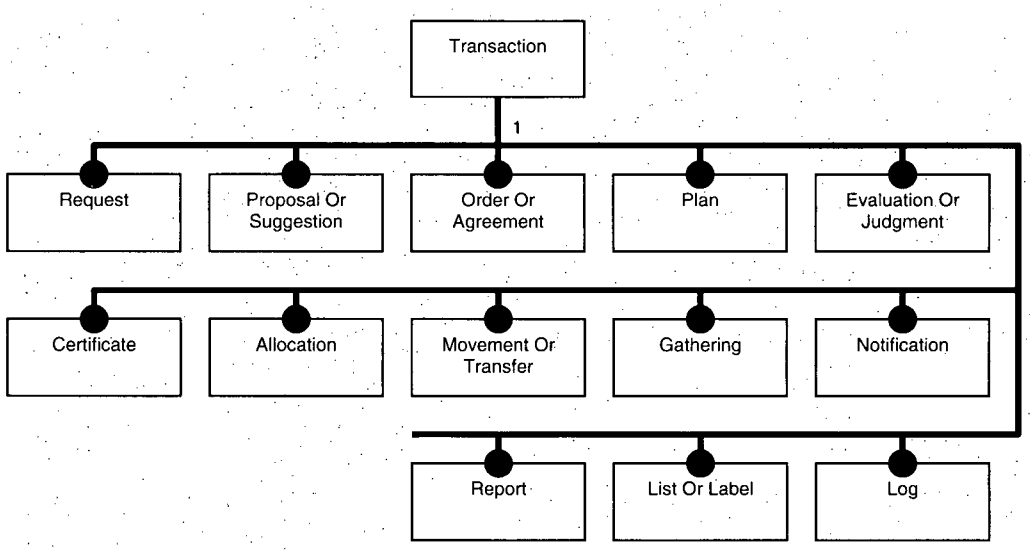


Figure 5–4: Subtypes of the Transaction

5.4 Chapter Conclusions

This chapter concentrated on one of the dimensions of this research: *integration of product and process information* into a *unified project model* for the purpose of development of *integrated total PM systems*. It described the scope, structure, and characteristics of a conceptual project information model, called the AEC/FM Project Core Model (APCM), which is a synthesis of the analyses presented in the previous chapter on AEC/FM project information modeling issues and requirements.

In general, this research argues that each of the PM functions involves a distinct view of project information; yet it is possible to bring these views together into a unified project model, which integrates both product and process information to provide interoperability among AEC/FM software application systems. The APCM, which may be viewed more as an "IFC-inspired" conceptual core model, represents such a model that incorporates the advantageous features of current AEC/FM models (e.g., IFC's) as well object-oriented methodologies. The concepts represented in the model may not introduce new technologies, per se, but rather encompass an extension of those presented by well-established models for the purpose of improving their shortcomings, which were identified in the investigation processes of the research. Examples include (but not limited to) extending the *FFB/P* concepts for modeling both product and process information, extending the *role* concept to represent roles of not only persons but also other objects (e.g., roles of physical objects such as buildings, equipment, and so on) to manage varying PM views into project information, and representing *transactions* as a means of recording and managing objects' historical data and changes. The collection of such extensions, compared to existing models, are considered as a major output of the research and are envisioned to provide a body of knowledge for AEC/FM information standardization for the purpose of development of *integrated total PM systems*.

The model was not intended to define all, but rather the most fundamental concepts encountered in the management of AEC/FM projects at a conceptual level. Chapter 9 describes an evaluation of the research results, including the APCM. The model's objects are powerful enough to be used as building blocks of lower-level object models to support a variety PM functions' information requirements. This requires a further step: identification and modeling of these functions (or processes) and their information requirements in more details (next chapters). Whereas this chapter focused on information modeling, the next chapter concentrates on process modeling and the integration of these two processes.

CHAPTER 6 Project Management Functions Classification

As a first step towards formalizing PM processes, the development of an analytical framework (called Conceptual PM Functions Framework, CPMF) within which a fairly comprehensive list of PM functions could be explored was sought in the research. Building on the background discussions presented in Chapter 2, and in complement to the bottom-up investigations of the research (Chapter 3), this chapter takes a top-down approach to modeling PM functions PM functions (that is, starting with very general concepts and refining these to add specific detail). It focuses on the process modeling part of the software development process as it relates to *model-based integrated total project management (PM) systems*. It addresses the research question (section 1.4.1): *What are the full range of PM functions, and how could they be classified*?

The first section lays down the theoretical bases of the framework by formulating the various dimensions of PM. The second section introduces the CPMF model. The third section describes how the CPMF model was used to further identify and classify PM functions, considering the criteria listed in sections 2.5.2 and 3.5.

6.1 Dimensions of Project Management

The management of construction projects involves three basic concepts: *objectives* (i.e., purposes), *processes*, and the *elements* contributing to or affecting the processes. These concepts are commonly referred to in various definitions of the terms project, project management (PM), and construction management (CM) [Kavanagh et al. 1978; Oberlender 1993; PMI 1996] (section 1.2.1). Considering the research objective "classification of PM processes" and the top-down approach to this classification, these three concepts are considered as three basic dimensions for PM and named as *project objectives*, *basic PM functions*, and *project elements* respectively.

6.1.1 Project Objectives

Every construction project is initially sought to be completed with the right scope, within a reasonable time, cost, and quality, no (or minimal) injury and accident, and minimum risk. Thus, at the very highest level of importance, the objectives of management of a project are summarized into control of *scope, time, cost, quality, safety,* and *risk*. These objectives, essentially, govern all the processes involved in the development of the project.

6.1.2 Basic PM Functions

Regardless of types or phases of projects, there are a number of basic functions that are performed in managing a project. Kavanagh et al. [1978, p. 12], list five basic functions from which scientific management is comprised: *planning*, *control*, *organization*, *coordination*, and *direction*. In a very similar approach, Oberlender [1993, pp. 9-10] defines five basic functions of management: *planning* (i.e., development of project plan), *organizing* (i.e., arrangement of resources), *staffing* (i.e., the selection of individuals and project team development), *directing* (i.e., the guidance of the work), and *controlling* (i.e., measure, report, and forecast deviations in the project scope, budget, and schedule).

The Project Management Institute (PMI), in another approach, groups PM processes into five "process groups": *initiating*, *planning*, *executing*, *controlling*, and *closing* [PMI 1996, p. 28]. Section 2.5.1.6 of this dissertation may be referred to for a detailed description of this approach.

With some little differences, there are major overlaps and similarities among the proposed classifications of the functions. Benefiting from them, and considering its objectives, the basic PM functions may be classified into seven groups:

1) *Initiating*: recognition of a need to a project and commitment to doing the project.
2) *Planning*: developing and maintaining a project plan.
3) *Organizing*: selecting and arranging of project resources.
4) *Executing*: coordinating resources and implementing the project plan.
5) *Controlling*: tracking of execution of project plan to ensure project objectives are met.
6) *Closing-out*: formalizing acceptance of the product of the project and closing the project.
7) *Following-up and Redirecting*: providing warranties services and learning from project processes.

The last function has not been considered as basic functions in the above references, although it is a part of almost all projects. Projects may literally be considered as finished after close-out process; nonetheless, as far as liability is of concern, the subsequent issues related to the facility might call for review of the previous processes and, thus, their performers, for resolution of the incurred problems. Examples could include design-related problems during construction calling for the designer; operation problems calling for the contractor; etc. This would require a *follow-up* process to resolve liability-related issues. Another process, which is of strategic importance, is the application of lessons learned for future processes. Although a project is defined as a "temporary undertaking to create a unique product" [PMI 1996], in construction projects, there is a common practice to use some of the lessons learned from previous projects on new ones. The application of lessons learned on future projects is, in fact, a *redirection* of the processes in a company and, thus, a strategic function of PM. The follow-up function

encompass the sum of those activities that complete and give an end to other PM functions within one project, while the redirection function is usually of concern at an enterprise level and has a close link with the planning function. Although these functions may not traditionally be considered as a part of a project's life cycle, they do exist as a part of PM functions and play important roles in an enterprise level.

Moreover, *coordination* is another major common function that is administrative in nature and has an important role in the success of a project. An example of coordination activities is notifying project participants of some events in a project. A notification, for instance, may be sent to a subtrade to announce the completion of its task's prerequisites; i.e., implying a request for an action towards fulfillment of its task. Another example is where a superintendent informs a testing agency of a planned operation (e.g., pouring concrete in foundations) and requests the agency to arrive on the site to perform the relevant tests. However, coordination activities may widely vary based on types of processes. Considering this variation and for the sake of simplicity, coordination has not been listed above, and it is considered as an integral part of other functions (particularly *executing*).

The *basic PM functions* are those generic and conceptual activities that are usually performed in the progress of a typical project. These functions are generic in the sense that they may be of concern for any project view and at any level of detail. For example, they may be considered within one specific phase (e.g., design, construction, operation and maintenance) or the whole life cycle of a project (i.e., from inception to operation and demolition of the facility). These functions may also be considered for the study of processes within any project viewpoint (e.g., a project manager's view, a contractor's view, etc.). Besides, they may be of concern at different levels of project organization (i.e., managerial, technical and operational).

6.1.3 Project Elements

PM has traditionally been defined as the art of managing the "4Ms", namely *Materials*, *Machines*, *Manpower*, and *Money*, to produce a product. The 4Ms are, in fact, the resources used in the process of crating a final product. However, management of a construction project involves using some resources to create a targeted product in an environment. The resources, the product, and the environment may be referred to as *project elements*. Considering the objectives of PM [Hendrickson and Au 1989; PMI 1996], functions of controlling time, cost, and quality are exerted as they relate to the project elements that are within the scope of the project. These elements and their management are extensively elaborated in the literature [Walker 1984; Hendrickson and Au 1989; Clough and Sears 1991].

Applying $IDEF_0$ notations (Appendix A), Figure 6-1 schematically shows some of the major potential elements involved in a typical PM process (in the middle). Although the process is labeled as

"Manage Project", it may represent any level of process breakdown structure. As this figure shows, management of a project may not be limited just to the management of the "4Ms".

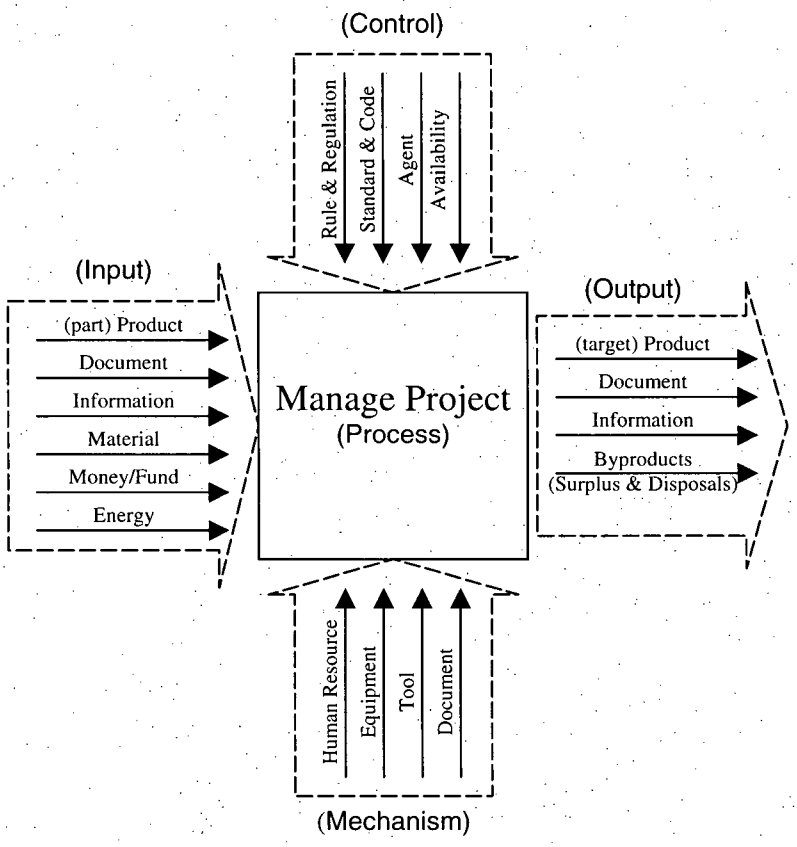


Figure 6–1: A Typical Project Process and Its Elements

The elements shown in Figure 6-1 may be categorized into ten major groups:

1) *(Target) Product*: is the main deliverable of the process. It can include such physical objects as construction products (e.g., a building, a road) and the site.

2) *Materials*: represent anything that is partially or wholly used in a process: raw or bulk materials (e.g., brick, concrete, cement, water, etc.), prefabricated materials (e.g., a roof truss, a precasted slab or wall, etc.), and engineered materials (e.g., water pump).

3) *Human Resources*: represent the human part of the resources used in a process.

4) *Equipment and Tools*: represent non-human, manufactured objects that are used in a process as helpers to produce the product, and it does not become a part of the product. It can include construction equipment (e.g., loader, truck, etc.), hand tools (e.g., drill, nail gun, pliers, etc.), computer hardware and software, etc.

5) *Money/Funds*: represent the funds that are incorporated into a process.

6) *Energy*: represents any form of energy that is used in a process. Examples include electricity, water (e.g., used for consumption of occupants of a building or used for drinking purpose on the construction site), etc.

7) *Information*: represents both the information about AEC/FM objects and the physical documents that contain such information. A document may appear in any format (e.g., digital, paper, or others such as a material sample). Information and documents may be of concern in processes as an input (e.g., an unapproved purchase order in an approval process), output (e.g., drawings produced by engineers), or mechanism (e.g., using construction drawings in a construction process).

8) *Byproducts*: represents the surplus and waste part of the physical resources used in the process. Examples include the material, equipment, money, or energy leftover and scraps in construction operations, the garbage produced by the occupants of a building, etc. Various leftovers of processes may be treated differently depending on their potential usefulness to other processes.

9) *Spatial/Plant Facility*: represents any physical-spatial object that facilitates the process and is used or occupied during the process. A material storage is an example of a spatial facility.

10) *Context and External*: represent any object or process that is external to and constrains a process. Examples include access roads, adjacent facilities, vendors, customers, competitors, natural environment, weather conditions, permits, etc.

The key characteristic of these elements is that they represent some physical or information elements on which the progress of a project is somehow dependent. Moreover, many project concepts such as cost, quality, production rate, productivity, availability, progress, etc. are usually defined in terms of these elements. Examples are cost of equipment, production rate of a crew (i.e., human and equipment), productivity in a process, quality of materials, etc. However, the above is a list of major classes of project elements and may be extended to include other elements as well.

6.2 A Conceptual PM Functions Framework (CPMF)

A Conceptual PM Functions Framework (CPMF) was devised to identify PM functions. The CPMF is based on a matrix format. It may be illustrated as a three-dimensional rectilinear grid that represents the three major conceptual dimensions of PM (section 6.1) on its three axes, graphically represented in Figure 6-2. The elements represented on the *project elements* dimension are the major ones and may be extended to some other elements. The same could be true for *project objectives*.

Each two-dimensional grid coordinate (i.e., a cell in a coordinate plane) would identify a function, holding a number of processes, relating to the two corresponding dimensions concepts (e.g., *cost*

control for *cost* and *controlling*; extending vertically throughout project elements). Each three-dimensional grid coordinate (i.e., a cube), on the other hand would represent a finer conceptual process, holding a number of sub-processes, originating from the three corresponding dimensions concepts (e.g., *materials cost control* at intersection of *material*, *cost*, and *controlling*).

Figure 6-3 further illustrates how the three dimensional rectilinear grid can be used to identify processes, as relate to *cost* and *basic PM functions*, and how *materials management* may interact with *cost management*. This framework is flexible, and the coordinates can help to list any possible functions of PM from different perspectives.

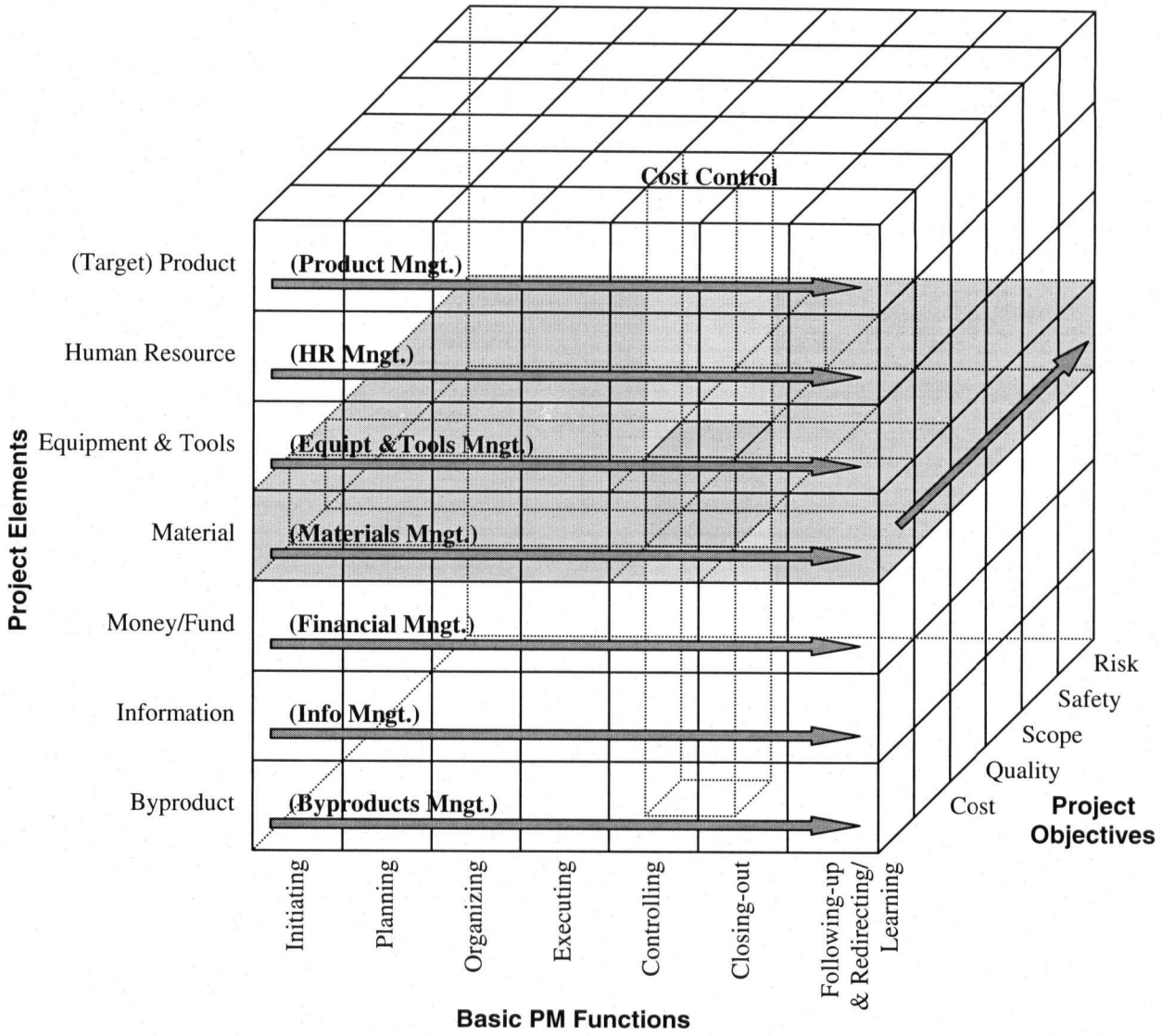


Figure 6–2: Dimensions of Project Management

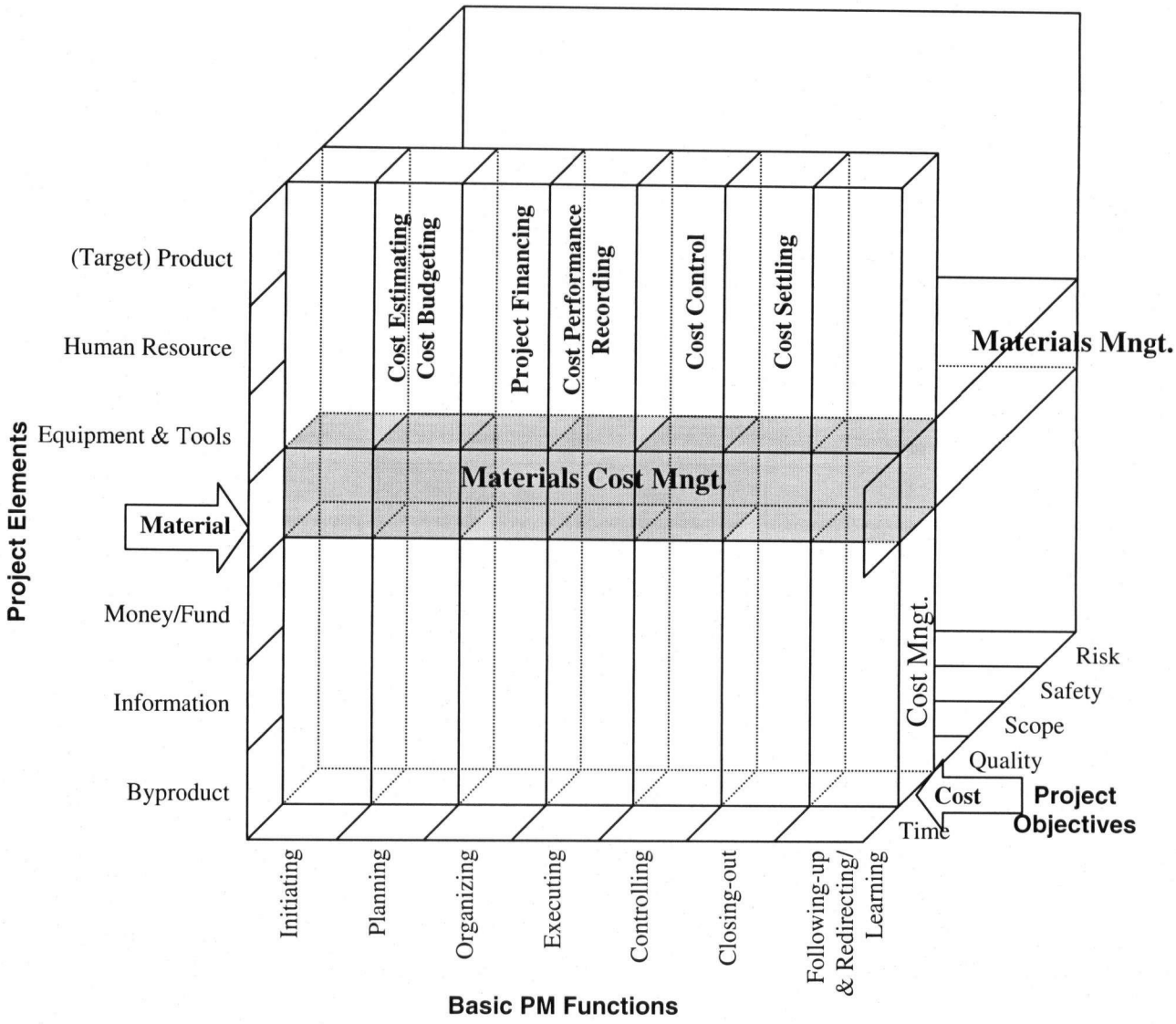


Figure 6–3: PM Functions relating to Cost and Material

6.3 Further Exploration of PM Functions

The CPMF model was used in two ways to explore PM processes. First, a two-dimensional *tabling (matrix) format* was used for finding the intersections and interactions of different types of dimensions of PM and listing of possible PM functions and processes. Next, the resulting processes were organized into a *tree-format, textual hierarchy* of processes. The following sections explain and discuss these activities.

6.3.1 A Tabular Listing of PM Functions

Applying the CPMF model, a tabling (matrix) technique was adopted to come up with a wide range of possible PM functions. As suggested by the CPMF model, each one of the three PM dimensions

was examined against the other two and against itself in separate two-dimensional tables. For example, Table 6-1 illustrates some of possible functions that can be captured by intersecting *project elements* and *basic PM functions*. Table 6-2, on the other hand, helps in identifying the functions by intersecting the selected *project elements* concepts with each other. The first table helps identify generic processes, while the second one may identify more detailed processes.

A process identified in a cell may not exclusively relate to a specific stage of a project or any specific type of contracting strategy (i.e., the *neutrality* criteria). For instance, "material purchasing" may occur in pre-construction phase and could extend, as an ongoing process, over the next phases as well. This characteristic of the processes is also considered as one step towards satisfaction of the *reusability* criteria (sections 2.5.2 and 3.5); i.e., a process definition capable of being used in various contexts (e.g., stages or phases).

Table 6–1: Project Elements versus Basic PM Functions

(Continued on the next three pages)

Project Elements ↓	Basic PM Functions							
	Initiating	**Planning**	**Organizing**	**Executing**	**Controlling**	**Closing-out**	**Following-up**	**Redirecting**
Human Resources	+Initiating HR Mngt.	+Org/Workforce *Planning* (Staff & Crew) +*Job* analysis and design +*Recruiting Plan* Development +Workforce *Training Prog.* Development +*HR Performance Plan* Development +*Reward/Penalty* Mngt. +*HR. Info.* Mngt.	+Project Team Development +Workforce *Acquisition* (Hiring) +Workforce *Training*	+*Payroll* Mngt. (a part of Financial Mngt.) +*Performance* Recording +*Absenteeism* Mgt.	+Org. Control +*Recruiting* Plan Control +*HP Performance* Control +HR *Safety* Control	+Workforce *Firing* +Workforce *Reassignment*	+Honor/Acknowledge/Follow/Respond to *Liabilities*	+Project HRM *Evaluation* +HRM-Lessons-learned *Recording*
Equipment & Tools	+Initiating Equipment Mngt.	+Equip. *Acquisition/Procurement* Strategy +Equip. *Maintenance* Strategy +Equip. *Selection* (Method/ Tech) +Equip. Cost *Estimating* +Equip. *Scheduling* +Equip. *Control Planning* (quality, quantity, cost, safety, etc.)	+Equip. *Procurement* (Bid, Purchase/ Rent, Handling & Storage)	+Equip. *Performance* Recording (status, production, etc.) +Equip.	+Equip. *Performance* control (*production, safety, etc.*) +Equip. *Cost* Control +Equip. *Maintenance* +Equip. *Replacement*	+Equip. *Reassignment*	+Honor & Respond to *Liabilities*	+Project EM *Evaluation* +EM-Lessons-learned *Recording*

Project Elements ↓	Basic PM Functions							
	Initiating	**Planning**	**Organizing**	**Executing**	**Controlling**	**Closing-out**	**Following-up**	**Redirecting**
Target Product	+Initiating Product Mngt.	+Scope Planning +Quantity *Takeoff* +*Scheduling* +Project *Bidding* +Const. *Permits* Planning Planning	+Building *Permits* Acquisition & Renewal	+*Progress Recording* +Progress *Reporting*	+Scope *Change Control* +Product *Quality/ Performance* Control (Inspection) +Product *Cost* Control +*Progress* Control	+Facility *Turnover*	+Prod. *Inspect.* +Prod. *Evaluation* +Prod. *Repair/ Change* +*Facility Mngt.*	+Project PM *Evaluation* +PM-Lessons-learned *Recording* +Product *Sale Mngt.*
Material	+Initiating Materials Mngt.	+Mat. *Procurement Strategy Selection* +Mat. *Control Strategy Selection* +Mat. *Selection* (Method/ Tech) +Mat. Quantity *Takeoff* +Materials Cost *Estimating* +Materials *Scheduling* +Mat. *Control Planning* (quality, quantity, cost, safety, etc.)	+Materials *Procurement* (Bid, Purchase, Handling & Storage)	+Mat. *Distribution* +Mat. *Processing* +Mat. *Installation* +Mat. *Disposal* (Waste, Surplus)	+Mat. *Quality* Control +Mat. *Cost* Control +Mat. *Inventory* Control +Mat. *Schedule* Control	+Mat. *Disposal* (Waste, Surplus)		+Project MM *Evaluation* +Lessons-learned *Recording*
Money/Fund	+Initiating Financial Mngt.	+Strategic Project *Planning* (Partnering, Bidding, etc.) +*Cost Estimating* +*Econ. Analysis* (Cash-flow) +*Risk Analysis* +Cost *Budgeting*	+*Financing/ Bounding*	+ *Payables* (e.g., Payroll, Invoice) +*Receiveables*	+*Cost Control*			+Project FinM *Evaluation* +FinM Lessons-learned *Recording*
Information	+Initiating Information Mngt. +Initiating Document Mngt.	+Communication *Planning* +*Documents* Mngt. +Communication Facilities &Equip. + Communication *Cost* Estimating	+Info *Procurement*	+*Info Flow* Mngt. +*Communication Facility* Mngt.				+Project IM *Evaluation* +IM Lessons-learned *Recording*

Project Elements ↓	Basic PM Functions							
	Initiating	**Planning**	**Organizing**	**Executing**	**Controlling**	**Closing-out**	**Following-up**	**Redirecting**
Context & Externals	+Initiating the Project	+*Marketing & Sales* Mngt. +*Permits* Planning (e.g., for demolishing, construction, etc.) +*Contracting* Strategy +*Supplier* qualification +*Security* Planning	+*Permits* Acquisition +*Contract* Development	+*Site Condition* Recording & Reporting	+*Permits* Renewal +*Contract* Change Control +*Supplier/Contractor Performance* Evaluation +*Security* Control +*Progress* Control	+*Contract* Closeout	+*Claims* Mngt.	+Project CEM *Evaluation* +CEM Lessons-learned *Recording*
Plant/ Facility	+Initiating Facility Mngt.	+Plant Mobilization Planning +Materials Storage Planning	+Plant Mobilization + Materials Storage Acquisition	+Maintenance Mngt. +Occupancy Mngt.				

Table 6–2: Project Elements versus Project Elements

Project Elements	Human Resources	Equipment & Tools	Physical Product	Material	Money/Fund	Information & Documents	Context & Externals
Human Resources	+HR Performance evaluation +Rewarding & Penalizing	X	X	X	X	X	X
Equipment & Tools	*+HR's Equip.* Assignment	+Equip. Selection *+Fleet Design*	X	X	X	X	X
Physical (Target) Product	+HR *Roles & Responsibilities* Assignment	*+Equip. Selection* (Method/ Tech)	+Product *Standardization* +Product *Specification* +Product Aspects Identification & Control (FFB/P)	X	X	X	X
Material	+Crew Design (Method/ Tech) +Materials Procurement (responsibilities) +Specialties Mgt.	+Hauling/ Handling unit selection (Transportation Tech)	+Mat. Selection +Material Processing	+Material Standardization & Specification +Materials-Use Sequencing (Method/ Tech)	X	X	X
Money/Fund	+Internal Monetary Transactions (Loan, Payroll)	+Equipment Econ. Analysis (Cash-flow, Replacement) +Equipment Cost Analysis (Cash-flow anal) Cost Estimation Cost Control, Cost Recording Purchasing	+Product *Cost/Quality Tradeoff* +Product Cost Control	+Mat. *Cost/Quality Tradeoff* +Mat. Cost Control		X	X
Information & Documents	+Info *Provider/ User Identification* +IT HR Estimation	+IT Equip. Estimation				+Document Mngt.	X
Context & Externals	+External *Contact & Reporting* *+Labor Relations* Handling	+Equipment *Sourcing* (Make/Reuse/Purchase/Rent) +Supplier qualification	+Building/ Demolition *Permits* +Demolishing/ cutting trees +Product *Marketing*	+Equipment *Sourcing* (Make/Reuse/Purchase/Rent) +Supplier qualification +Material Purchasing	+External Monetary Transactions (Invoice, Tax, etc.)	Codes & Regulations	+Consumer Selection (Marketing) +Supplier Selection

6.3.2 A Hierarchical Classification of PM Functions

Considering the three PM dimensions suggested by the CPMF model, a tree-format textual *hierarchy* was used to organize high-level PM functions into a hierarchical representation. In a fashion similar to the PMBOK's approach (though with modifications), processes were organized hierarchically into groups of generic functions (Figure 6-4). Considering the two PM dimensions of *project objectives* and *project elements*, first, the most high-level PM functions (so called first-level functions) were identified. Examples include *Time Management*, *Cost Management*, *Quality Management*, *Human Resource Management*, and *Materials Management*. Considering its nature and the third dimension of PM (*basic management functions*, section 6.1.2), each function was then narrowed down into more low-level processes.

One of the challenges in this approach was to abide to the criteria set for process models of the research (e.g., non-redundant processes), especially for low-level processes. This was due to the fact that some processes could potentially be included under more than one process groups. In order to avoid duplications, each process is placed under the process groups to which it is more closely related. For instance, *Marketing* and *Sales Management* could be related to both *Product* and *Context and Externals* categories. However, due to their closer relationship, they are considered once under the first category.

Like the tables of functions (in the previous section), the functions identified in the hierarchy of functions are generic in the sense that they do not concern any specific type of contracting strategy or phase of a project (i.e., the *neutrality* criteria; sections 2.5.2 and 3.5). They may be further refined into *corporate-level* and *project-level* functions. For example, *payables* can be interpreted as project-level payables (e.g., paying for materials purchased or human resources employed for a specific project) and corporate-level payables (e.g., paying for a bulk of material purchased for a company to be stored for future needs of any of the projects in the company, or office employees). Moreover, each process may relate to any of three levels of *strategic* (i.e., management-oriented), *tactical* (i.e., technical), and *operational* (i.e., physical assembling and installation of the product) focus. Therefore, the process breakdown may be extended to include other processes by considering *organizational levels* as well as the *basic management functions*. Following this procedure, some generic processes (such as forecasting, and tracking) that may be equally applicable to many project elements (e.g., sale of product, cost of resources, etc.) can be taken into consideration.

Figure 6–4: A Simplified Portion of the Hierarchy of PM Functions

(Continued on the next three pages)

Proj. Objectives & Proj. Elements	1st-Level PM Functions / 2nd-Level PM Functions / 3rd-Level PM Functions
Time	**Time Management** [Work (HR, Equip. Mat.) and Product] • Planning Work Definition (i.e., process scope) Sourcing of Resources and Services (on-time availability) • Scheduling (i.e., Schedule Development) Activity/Work Scheduling (Definition, Sequencing, Timing, etc.) Resource Scheduling (HR, Material, Equipment, Fund, etc.) Resource Leveling Schedule Control
Cost	**Cost Management** [Work (HR, Equip. Mat.), Environment, and Use] • Cost Estimating • Cost Budgeting • Cost Planning • Cost Control • Historical Cost Data Collection
Quality	**Quality Management** [Work (HR, Equip. Mat.), Environment, and Use] • Quality Planning • Quality Assurance • Quality Control
Scope	**Scope Management** [Work (HR, Equip. Mat.) and Product] • Scope Planning • Scope Control • Scope Change Mngt.
Safety	**Safety & Security Management** [Work (HR, Equip. Mat.), Environment, Product Use] • Safety Planning • Safety Control • Security Planning • Security Control
Risk	**Risk Management** • Risk Identification • Risk Quantification • Risk Response Development • Risk Response Control
Human Resource	**Human Resource (HR) Management** • HR Procurement/Acquisition (Recruiting/Hiring) • HR Appointing (Assignment, Reassignment, Firing, etc.) • HR Training • HR Appraisal (i.e., Performance Evaluation) • HR Rewarding • HR Recording • HR Safety Mngt.

Proj. Objectives & Proj. Elements	1st-Level PM Functions 2nd-Level PM Functions 3rd-Level PM Functions
Equipment & Tool	**Equipment Management** • Equipment Selection • Equipment Procurement Mngt. (Renting, Leasing, Buying, etc.) • Equipment-in-Process/Operation Mngt. • Equipment Maintenance Mngt. (Storage, Inspection, Repair, etc.)
Target Product	**Product Management** • Site Selection • Product Design • Product Specification and Technology Selection • Product Marketing (Seeking potential Customers, etc.) • Product Sales Mngt. (e.g., forecasting, tracking, etc.) • Product Turnover Mngt. (interacts with Occupancy Mngt. in FM)
Material	**Materials Management** • Materials Procurement Mngt. • Field Materials Mngt. • Materials-Fabrication & Installation • Material Surplus Mngt. • Material Disposal Mngt.
Money/Fund	**Financial Management** • Investment Mngt. (seeking potential Investors, Partnership) • Financing (Loans arrangement and commitment, Cashflow, etc.) • Accounting Mngt. (Payable, Receivable, General Ledger, Payroll, etc.) Accounts Payable (Vendor Invoice) Accounts Receivable (Customer Invoice) General Ledger (receiving report from and update to all other) Payroll
Energy	**Energy Management** • Energy Procurement Mngt. • Energy Consumption Mngt. • Energy Saving
Information	**Information Management** • Document Mngt. • Communication Mngt. • Information Mngt. (distribution, security, etc.)
Byproduct	**Byproducts Management** • Surplus Mngt. • Disposal Mngt. (Disposing, Recycling, etc.) • Waste Mngt.

Proj. Objectives & Proj. Elements	1st-Level PM Functions 2nd-Level PM Functions 3rd-Level PM Functions
Plant/Facility	**Plant Facility Management** • Facility Mobilization Mngt. (preparing the physical grounds for operations; e.g., *temporary facilities, utilities, etc.* for initialization of construction) • Facility Maintenance Mngt. (Alteration, Repair, etc.) • Facility Operation Mngt. • Occupancy Mngt. (interacts with Turnover Mngt. in Product Mngt.) Space Use Mngt. Tenancy and Lease Mngt. (seeking potential tenants, leasing, etc.) Moving Mngt.
Context & External	**Context Management** • Contract Mngt. (Bidding & Tendering, Administration, Changes, etc.) • Claims and Disputes Mngt. (i.e., Legal issues) • Permits Mngt. (Planning, Acquisition, Renewal, etc.) • Environment Study and Treatment
Resource	**Resource Management** (in general; it may include material, equipment, HR, facility, land, etc.) • Resource Procurement Mngt. (i.e., Resource Acquisition Mngt.) Purchasing Renting/Leasing Supplier Qualification & Selection Supplier Appraisal (i.e., Performance Monitoring) • Resource Inventory Mngt. • Resource Productivity Mngt. • Resource Conflict Mngt. • Asset Mngt. (may include Property Mngt.)
Process	**Process Management** • Process Engineering • Process Productivity Mngt. • Progress Reporting

6.4 Chapter Conclusions

Building on the background discussions made in Chapter 2, this chapter took one step further towards the principle goal of the research (i.e., exploring *model-based integrated total PM systems*). It concentrated on the process modeling part of the software development process and suggested a solution, i.e., an analytical tool called Conceptual PM Functions Framework (CPMF), and demonstrated how it was used in the research to identify and classify a wide range of PM functions. The hierarchical representation of processes was generally found useful in understanding the overall process areas in the research, while the matrix (table) representation provided a listing of lower-level processes. The next chapter describes further application of the CPMF model to the design of a model that is capable of capturing both process and information models of business and computer environments, and Chapter 8 explains how the identified processes were formalized into process models.

CHAPTER 7 The Integrated Total Project Management Systems (ITPMS) Model

In the past, information modeling has tended to focus on the data and processes (in a business and software environments) in isolation. Most information modeling efforts involve three major activities: 1) *Business process modeling*: This involves listing and analysis of the business processes involved in the domain of concern; which is usually done using $IDEF_0$ diagramming and similar techniques. 2) *Software process modeling*: This activity includes an analysis of the software processes using different diagramming and descriptive techniques (e.g., $IDEF_0$ and usage scenarios, used in the IAI). 3) *Data modeling*: The graphical and textual representation of the information required by the processes is developed in this activity (e.g., using EXPRESS data modeling language).

In practice, however, a *loose link* is established between these three activities. For example, most AEC/FM information modeling projects (e.g., STEP and IAI) use $IDEF_0$ diagrams only to show the context and scope of a modeling project. No explicit link is provided between such business process models, software process models, and object models. In a broader context, there is no model that could capture the processes and information requirements of both the AEC/FM business and software environments comprehensively in an *integrated* manner (Chapter 2). This chapter describes the research solution suggested to address this gap. It addresses the research question (section 1.4.1): *How could models of AEC/FM systems (i.e., process and information models of both business and software systems) be integrated into a single framework to support the integrated total PM systems development process*?

Considering the basic goal of the research (i.e., exploring model-based integrated total project management systems) and the predefined research criteria (sections 2.5.2 and 3.5, the development of a model called Integrated Total Project Management Systems (ITPMS) was sought. The CPMF model (section 6.2) was used, as a principle guide, together with the UML modeling constructs [UML 1.4, 2001] (Appendix A), to cast the basic structure of the ITPMS model. This chapter describes the architecture and underlying meta-model of the ITPMS model. It also presents an overall description of its application.

7.1 The ITPMS Model Architecture

The ITPMS model architecture reflects an integration of business process modeling and software engineering. It includes five views, portraying the overall process of computer application development (section 2.1.3) prior to the deployment and testing of the system. Figure 7-1 illustrates these views, the related activities (i.e., analysis, design, and implementation), and their artifacts. The term *analysis* is broadly used here in the context of both business and computer environments. Therefore, it encompasses business (process and object) modeling as well as software analysis.

The *business process view* captures information about PM functions and their related processes. The *business process view* and *business object view*, which constitute the business modeling and account for a major bulk of the model, are essentially at the analysis level. Their artifacts (i.e., business process models and domain object modes) are highly neutral in nature; meaning that they do not reflect any technical (e.g., software environment), contextual (e.g., contractual and organizational pattern of project), and other constraints. They are intended to be useful for being adopted as components for different computer applications as desired; thus satisfying the *neutrality* criteria set for the research models.

On the other hand, the *software process view* and *software logical/class view* are more specific to computer application systems. The software process view sits at the edge of analysis and design levels. This view, with its artifacts (i.e., software process models), serves to define the functionality required by such systems. Applying UML techniques, for instance, such models can be captured in software use cases, use case models, and activity diagrams. This view can also include the user interface requirement of such systems (captured in user interface models) at the design level. This view, however, may further contribute to the definition and modeling of business objects in the business object view.

In relation with this view, at the design level, the *software logical/class view* serves to define software classes (i.e., entity, boundary, and control classes) and their relationships (e.g., in the form of class diagrams). The *entity classes*, which constitute a major portion of software classes handling business functions, would have a direct mapping to the business objects defined within the business object view. Its contents (i.e., classes and class diagrams) are based on the results of the previous views (i.e., software process and business object views). The artifacts of this view help form those of the *software component view*, which is termed as deployment view elsewhere [Booch et al. 1999], at a lower level (i.e., implementation level).

The *software component view* selects software classes (defined by the software logical view) and assembles them into functional components that, in concert, provide for the functionalities desired by the software application system. Such functionalities would be the ones defined by the software process view at the analysis level. This implies a close relationship between the two views. The level of neutrality decreases downward, from the business process view to software component view (Figure 7-1).

As explained in the following section, due to the scope of this research (section 1.4.2), the research (and thus Figure 7-1) emphasizes the pre-deployment activities of the software development process. The ITPMS architecture, however, allows for incorporation of other views such as testing and deployment views to integrate the whole workflows of the process.

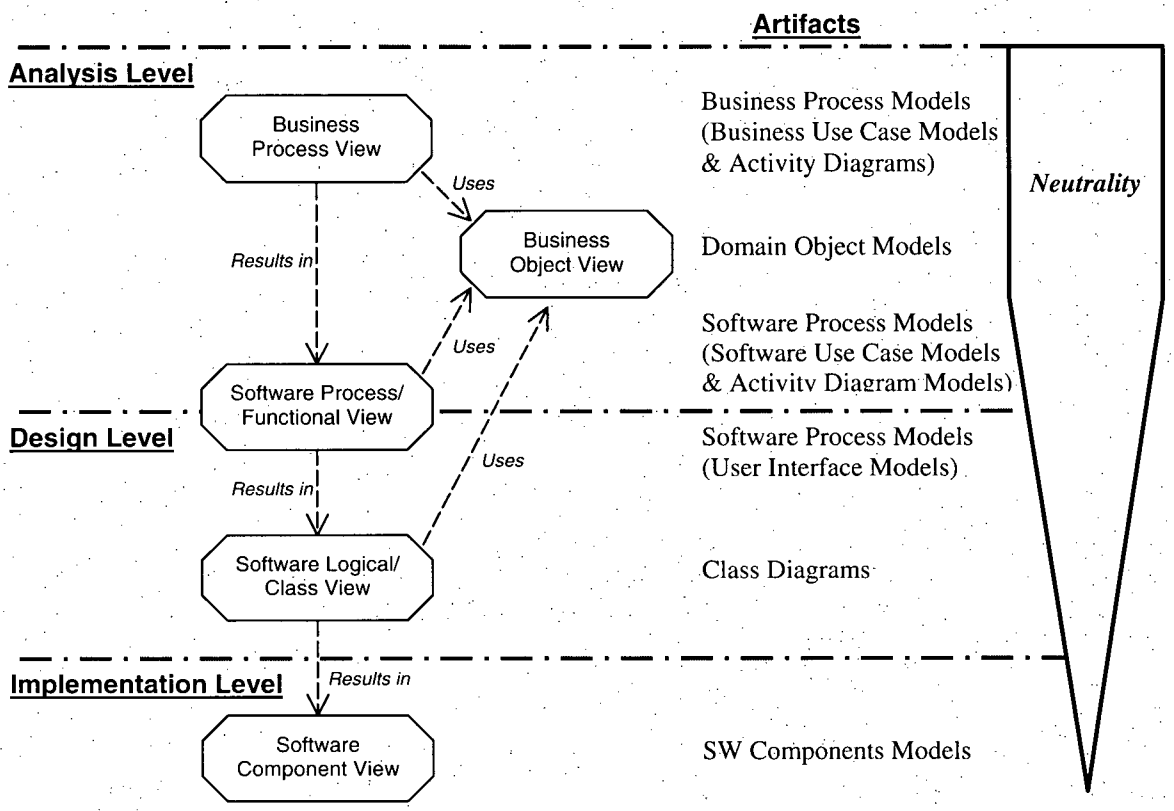


Figure 7–1: The Views and their Dependency Relationships and Artifacts in the ITPMS Model

7.2 The Meta-Model of the ITPMS Model

The ITPMS model is based on an application of UML notations (Appendix A) and the CPMF model (Chapter 6). Using EXPRESS-G notation [ISO 1991] (Appendix A), Figure 7-2 shows the meta-model of the five views incorporated into the ITPMS model (as explained in the previous section). Figure 7-3 further illustrates the meta-model of the elements of the first four views of the ITPMS model with examples (underlined). At the highest level, based on groupings of *project objectives* and *project elements* (section 6.1), *PM Topic Areas* are modeled using UML package notation. They include such areas as *time management*, *cost management*, *quality management*, *materials management (MM)*, etc. Each of these packages may be further divided into sub-topic area packages. Such packages can be created according to the *basic management functions* (section 6.1.2). For example, the MM function, which is represented as a topic area, may include such sub-topic areas as *MM initiating*, *MM planning*, *MM controlling*, etc.

Each topic area can include one or more *Business Process Groups*, which are represented by UML packages and can include sub-groups themselves. *MM execution*, for example, may include *request processing*, *procurement*, *purchasing*, etc. The process group of *purchasing* may potentially be broken down to *order processing*, and *invoice processing*.

In a next level, *Business Processes* of each function are modeled using UML use cases. They represent *processes*. For instance, *request material* and *order material* may be considered as two processes in the *request processing* and *order processing* components of MM. The *workflow* (i.e., flow of works and events) of each business process is captured in the form of one or more UML activity diagrams (attached to the use case representing the process). The UML activity notation is used to represent the *work* (e.g., *add order*, *get phone number*). Depending on the level of abstraction of interest, each activity may be further broken down into some sub-activities.

In a further modeling of the processes, flow of business objects is captured in the activity diagrams. The identified business objects that are mainly the documents and transactions objects are grouped in the *business object view* of the model, as described in the model architecture. Such objects are the potential candidates in the domain object model of concern.

Transparent to the model elements of business process and business object views are the elements of software process and software class views. *Business Processes* and *Business Objects* can have their counterparts in a computer environment as *Software Processes* and *Software Classes* respectively (Figure 7-3).

As a summary for the model elements of the business-related views, it may be concluded that: 1) Topic/sub-topic areas and business process groups/sub-groups are, in fact, logical components (represented by packages) that can hold one or more of other logical components. They represent PM functions. 2) A business process group/subgroup, however, can include processes too. 3) The decision on the number of breakdown levels of topic areas and business process groups is highly dependent on the subject of concern as is the case for activities. Similar conclusions may be drawn for the model elements of the software-related views.

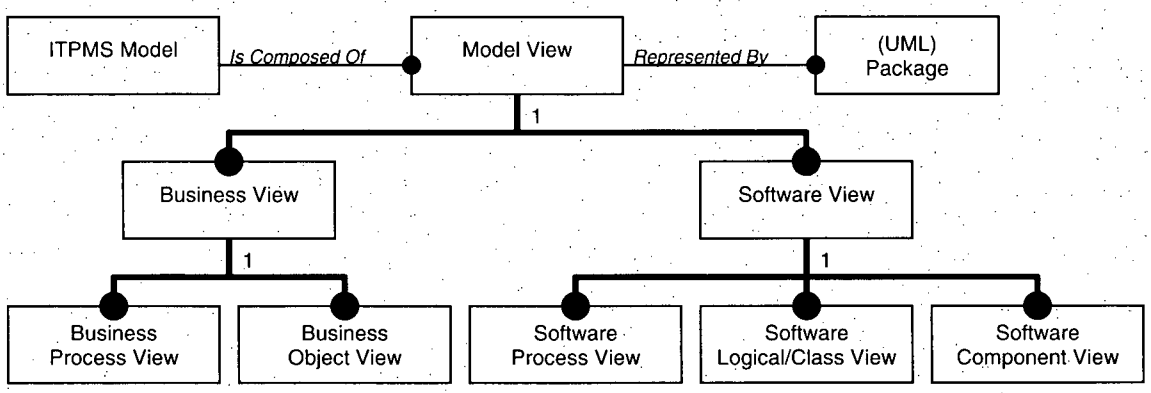


Figure 7–2: The Meta-Model of the Incorporated Views of the ITPMS Model

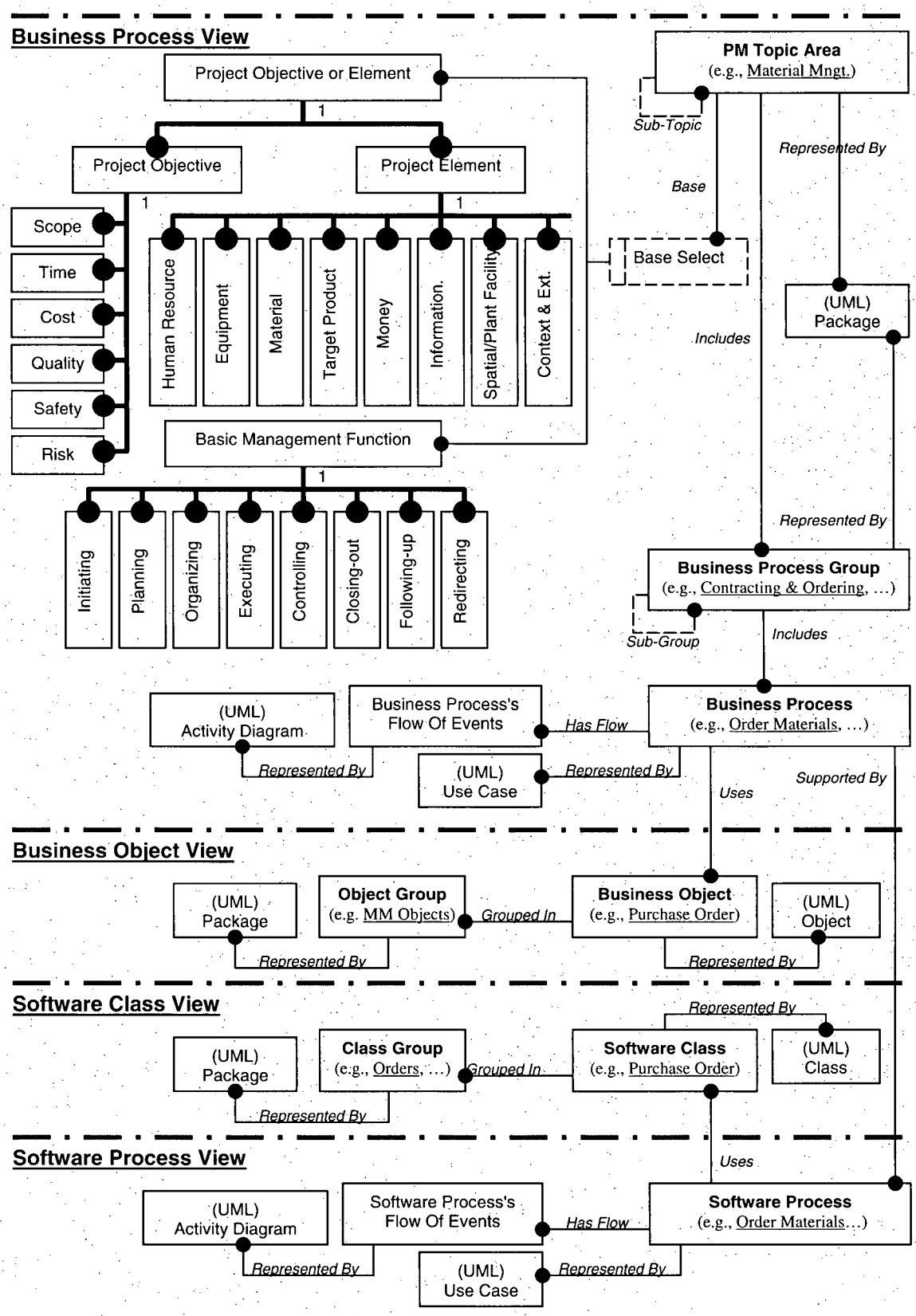


Figure 7–3: The Meta-Model of the ITPMS Model

7.3 An Application of the ITPMS Model

Following the architecture and the meta-model of the ITPMS, the research sought to implement the model using a CASE (Computer-Aided Software/System Engineering) tool. The AEC/FM industry encompasses a very large, complex set of non-standardized processes and object model requirements. This makes the management of the models of such processes and objects even more difficult. The use of proper CASE tools can considerably facilitate this modeling process [Kendall and Kendall 1995]. Such tools, for example, usually provide for a small change to an object to be instantly propagated and reflected throughout the model. A discussion of CASE tools and their advantages was given in section 2.3.4. Considering these facts, an application of a CASE tool that would facilitate incorporation of the architecture of the ITPMS model appeared as a necessity. Therefore, after reviewing some of the existing CASE tools, Ensemble Streams™ R3.2 (Streams) [2000] was selected as a tool for business process modeling and for a part of software modeling of the research.

In the model created using Streams, each one of the five views is represented by a UML package notation. Moreover, for the sake of clarity, every model element is assigned a specific *stereotype*. Generic business processes, which are defined in the *business process view*, are modeled by the UML use case notation and are stereotyped as «business use case». The stereotype «business activity» is used for the activities presented in activity diagrams, which present process workflows. The business objects, which are identified in activity diagrams, are grouped and organized in the *business object view*. They are represented by the UML object notation stereotyped as «business object». For distinguishing purposes, a similar stereotyping approach is also used for use cases, activities, and other model elements of the software-related views.

Streams is intended to serve as a business process modeling tool, and it provides very limited capabilities for software object modeling. For example, it provides the basic tools for UML activity diagramming and use case notations and documentation of model elements, but it does not offer any means for class diagrams; though it is possible to represent a class, using UML class notation, on an activity diagram. However, Streams offers an import/export capability with Rational Rose™ [Rational 2002] for software systems design and implementation (Appendix A may be referred to for more explanation). Therefore, only the analysis level and a part of the design level of the ITPMS model architecture can be modeled in the Streams tool, and the object models and class diagrams are prepared outside the Streams environment in this research. Chapter 8 extensively elaborates the MM component of the ITPMS model.

7.4 Chapter Conclusions

Building on the background discussions made in chapters 2 and 6, this chapter took one step further towards the principal goal of the research (i.e., exploring *model-based integrated total PM systems*). It described the structure of the ITPMS model, which is an application of the CPMF model and the UML constructs and is suggested by the research to address the integration of the process and information modeling workflows in integrated AEC/FM application systems development. It also presented an overall description of the application of the model.

While the issues associated with the ITPMS model are important, they are seen to be outweighed by the many advantages offered by the model (section 9.4). Although the model is intended to serve the main goal of the research, the architecture of the model is generic enough to be usefully adapted to non-AEC/FM domains as well. The next chapter demonstrates different dimensions and capabilities of the ITPMS model by describing its application to modeling the functional and information requirements of MM system in a CASE tool.

CHAPTER 8 The Materials Management Process and Information Modeling

Following the background materials presented in section 2.6, and using the research results presented in previous chapters, this chapter addresses the last research question (section 1.4.1): *Given a specific PM context (i.e., the MM function), how could the proposed models be implemented (i.e., using the frameworks to model the functional and information requirements of MM systems)*? It is structured based on the methodology described in section 2.6.6.2. The first section presents the analytical approach of the research to classification of MM processes. The second and third sections describe the MM business process models (functional requirements) represented in the ITPMS model structure. The earlier presents the "business process groups" defined in the model, while the latter describes the business use case modeling part. The fourth section describes the MM domain object models (information requirements). The MM process and object models suggested in this chapter are a synthesis of the top-down analysis (the first section) and the knowledge attained in the bottom-up field investigations (Chapter 3). They are the main building blocks for modeling software processes and classes required for development of integrated total PM systems.

8.1 Classification of MM Functions

This section presents a top-down analytical approach (from general concepts to detailed models) to explore the ways in which MM processes could be classified into definable groupings.

8.1.1 Managerial Processes versus Operational-Technical Processes

MM processes may be divided into two major groups: *operational-technical* and *managerial* processes. The first group concerns the *subject* of the work (i.e., the *material*) and gives an operational view, while the second one relates to the *work* and the *organization* (in which materials-related activities are handled) and reflects a *managerial* view.

Operational-technical processes are those activities involved in acquiring, distribution, dispatching, and applying materials. Estimating, quoting, receiving, inspecting, storing, processing (e.g., fabrication), and testing of materials are some examples of operational-technical processes. The product of MM systems (i.e., an ordered, received, stored, fabricated, or installed material) is usually the main focus of operational processes. *Managerial* processes, on the other hand, concern the management of MM systems (i.e., directing and administering the processes). They deal with managerial issues such as how to run MM systems, what is needed in terms of resources (i.e., MM resource requirement identification), how well the system is performing, and so on. Examples of managerial processes include planning,

scheduling, monitoring, and controlling the system. The terms "*direct functions*" and "*support functions*" suggested by Michael Porter [Andersen, Flaatten, et al. 1992] may be used as synonymous terms for these two groups of processes respectively.

8.1.2 Functional Bodies Involved in MM processes

Processes take place in physical or logical locations (e.g., a factory, a site, an organization, etc.), associated with certain functional bodies (i.e., organizational units or actors). For example, a *workshop* functional body carries out an *assembling parts* process, while a *standards organization* functional body may deal with a *standardization of the parts* process. The functional bodies involved in materials-related processes can be typically grouped into three major categories (each of which may be further detailed into lower-level bodies): *regulatory*, *production-and-supply*, and *construction* (Figure 8-1).

Regulatory bodies deal with processes such as establishing standards and monitoring materials processes in terms of codes and regulations. Standards organizations and municipal governments are example organizations in this body. *Production-and-supply* bodies involve the production of materials and supply of products to the customer (i.e., the consumer). Finally, *construction* bodies represent the point of consumption for materials. They include the home office (managing materials at the corporate and project levels), corporate or project warehouses or storerooms (storing, inventorying, and distributing materials), workshops (assembling and fabricating parts), and the construction site (i.e., the combination of the superintendent and subcontractors organizations; handling and installing materials on the site).

Each functional body has it own processes and, thus, its own view of materials (due to diverse practices and project organizations, although overlaps do exist). For example, the processes of *receiving*, *storing*, and *distributing* materials are central processes for the *warehouse* functional body. However, such processes may also take place in the *factory*, in the *supply center*, and at the *construction site*; though, with different scopes and degrees of complexity.

Although this research has aimed at modeling generic, context-independent MM processes (e.g., independent of the type of organization or person in charge of the processes), consideration of the basic functional bodies involved in material-related processes helped in attaining a more realistic and comprehensive MM model. The grouping of MM processes according to the associated functional bodies was found to be a useful way of organizing the processes and to provide a checklist to verify the comprehensiveness of the model.

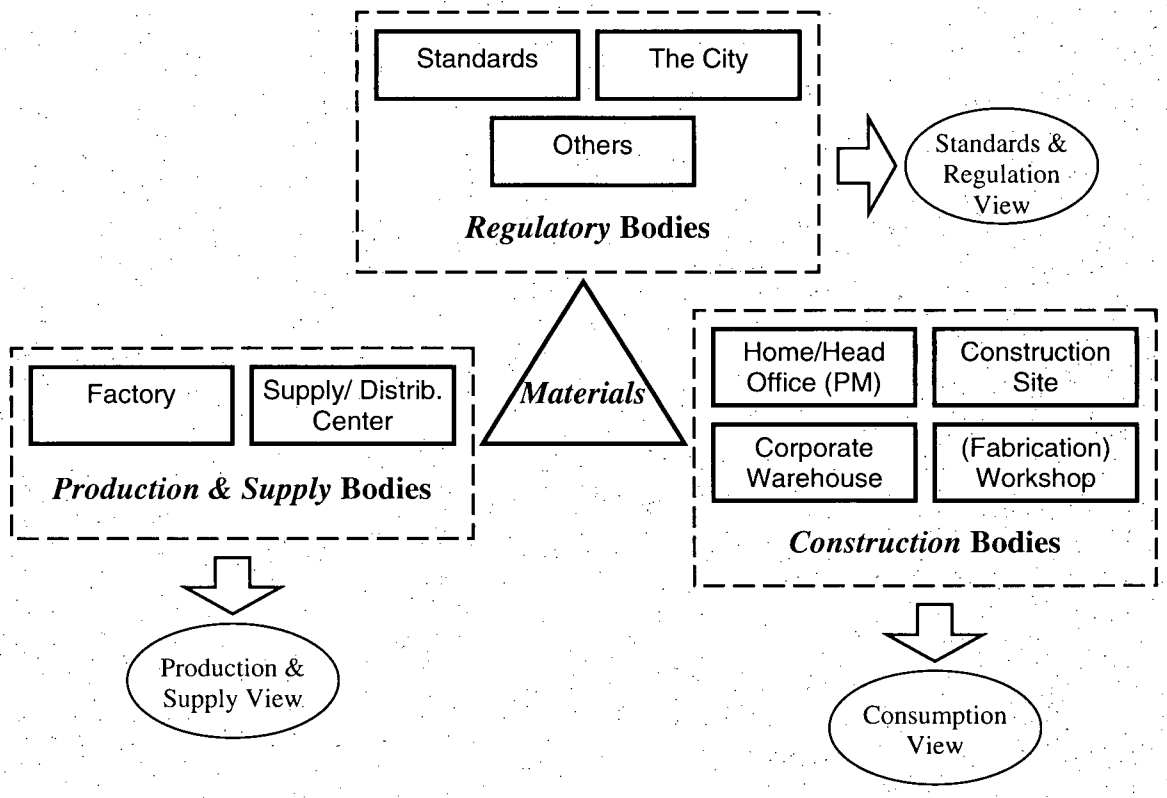


Figure 8–1: Major Functional Bodies Involved in MM Processes

8.1.3 Major Categories of MM Processes

As an initial exploration, materials-related processes may be categorized into the two aforementioned top-level groups of *managerial* and *operational-technical* processes (section 8.1.1), as illustrated in Figure 8-2. Managerial processes generally relate to the MM system (or organization), while operational-technical processes deal with materials. On the left side of Figure 8-2, the basic management functions (suggested by the CPMF model, section 6.1.2) are used as a basis for listing the managerial processes of MM systems. The right side of the figure, on the other hand, considers the functional bodies involved in materials-related processes (section 8.1.2) and the overall life cycle of materials (e.g., standardization, manufacturing, supply, planning, procurement, usage/construction, and so on) as a basis for listing of *operational-technical* processes.

Notwithstanding these suggested grouping, all of these processes can be very much interrelated. The performance of a process of one type may be dependent on the information (or output) produced in one or more processes of other types. A typical example is the estimating and scheduling of materials that are usually performed with consideration of organizational elements such as the equipment, the human resources, and the space required to handle the materials. Another example is the monitoring and control of participants' performance, which is usually measured in terms of response time and quality of response

of a participant to his/her predefined tasks and responsibilities. In the case of suppliers, for instance, performance measures may be derived from the volume of back orders, number of in-time deliveries, quality of the delivered products, buy-backs, and so on. Moreover, the two types of processes may equally be of concern at both project and corporate levels.

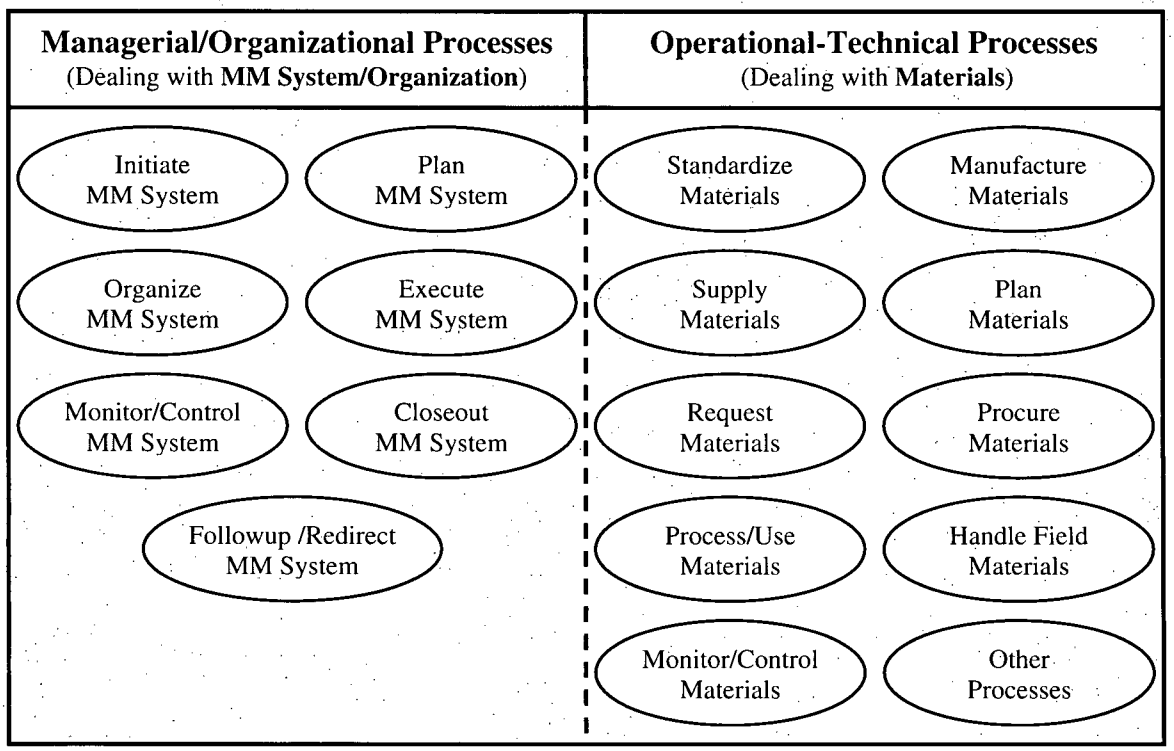


Figure 8–2: The Major Groups of Materials Management Processes

8.1.4 A Hierarchical Classification of MM Processes

This section describes how a hierarchical representation technique was used in the research for further identification of MM processes. Figures 8-3 through 8-7 show how the major groups of MM processes identified in the previous section (Figure 8-2) were detailed into lower-level processes. At this stage of analysis, repetition of processes was not considered as a major restriction in listing of processes; i.e., it was allowed. Although this approach may not have resulted in a very detailed comprehensive list of MM processes, it helped to freely explore and identify a wide spectrum of MM processes from various viewpoints.

The selection of material life cycle as a basis for categorizing the second group of processes (i.e., operational-technical) revealed the advantages and implications listed in section 2.5.2.4. In particular, it helped to identify a wide range of scenarios in which materials are of concern. On the other hand, repetition of processes was observed. For example, *purchasing of materials* may take place by different project participants (e.g., owner, general contractor, subcontractor, and so on) in different situations

before and/or during construction. The same is true for *transporting, receiving, storing, and distributing of materials* that may take place under procurement activities, field MM activities, and even workshop activities. The process breakdown of field MM, in fact, generally follows the same pattern as that of procurement. To overcome this shortcoming (i.e., process repetition), the identified operational-technical processes were later grouped based on the basic management functions, similar to the managerial processes. Consequently, the results became closer to the ITPMS model architecture (Chapter 7), which is based on the "*reusability*" criteria (sections 2.5.2 and 3.5). The following section describes how the results of this analysis were fit into the ITPMS model.

A0. **MM System Processes**

- **A1.** **Manage MM System**
  - **A11.** ***Initiate*** MM System
    - **A111.** Define ***Goals & Objectives***
    - **A112.** Define ***Overall Plan***/Methodology
    - **A113.** Establish ***Initial MM Team*** (relating to PM team)
    - **A114.** ***Formalize*** MM System
  - **A12.** ***Plan*** MM System
    - **A121.** ***Develop Materials Plan***
      - **A1211.** Identify & Define Processes (Work Scope) of MM system (from major to specific)
      - **A1212.** Identify & Define Process-Handling Methods (major to specific)
      - **A1213.** Plan MM Organization
      - **A1214.** Plan MM Communications (overall to specific)
      - **A1215.** Plan MM Transportation (overall to specific)
      - **A1216.** Plan MM Quality (QA/QC strategies and procedures)
    - **A122.** ***Prepare Materials (Functional) Procedures***
    - **A123.** ***Plan Cost*** of MM System
    - **A124.** ***Schedule*** MM System (processes, resources, cost, etc.)
  - **A13.** ***Organize*** MM System
    - **A131.** ***Develop MM Team*** & *Assign Responsibilities*
    - **A132.** ***Acquire MM Resources and Facilities***
      - **A1321.** Acquire/Hire MM Workforce
      - **A1322.** Procure MM Equip & Materials
      - **A1323.** Procure MM Info/Docs
      - **A1324.** Acquire/Obtain MM Facilities (e.g., store)
      - **A1325.** Finance MM
    - **A133.** ***Train MM Workforce***
    - **A134.** ***Acquire MM Permits***
  - **A14.** ***Execute*** MM System
    - **A141.** ***Record & Report*** HR & Equip. Resource ***Performance***
    - **A142.** ***Coordinate*** Processes
  - **A15.** ***Monitor & Control*** MM System (Corporate & Project)
    - **A151.** ***Control Overall MM System Performance***
    - **A152.** ***Control MM Resource Performance*** (e.g., Equipment)
    - **A153.** ***Control Participants Performance***
    - **A154.** ***Control Schedule***
    - **A155.** ***Maintain Materials Plan***
    - **A156.** ***Maintain Materials Procedures***
  - **A16.** ***Closeout*** MM System
    - **A161.** ***Fire/Reassign MM-HR*** (i.e., a part of administrative closure)
    - **A162.** ***Closeout Contracts***
  - **A17.** ***Follow-up & Redirect*** MM System
    - **A171.** ***Acknowledge & Respond to Liabilities***
    - **A172.** ***Evaluate MM Systems***
    - **A173.** ***Record Lessons Learned*** for Future

Figure 8–3: Overall Process Breakdown of Managerial Processes ("Manage MM System")

A0. MM System Processes

- **A2.** **Manage Materials**
  - **A21.** ***Standardize & Manufacture*** Materials
    - **A211.** ***Prototype*** Materials
    - **A212.** ***Market*** Materials
    - **A213.** ***Advertise*** Materials
    - **A214.** ***Test & Standardize*** Materials
    - **A215.** ***Manufacture*** Materials
  - **A22.** ***Supply/Distribute*** Materials (to customers)
    - **A221.** ***Plan Supply/Distribution System*** for Materials
    - **A222.** ***Advertise*** Materials
    - **A223.** ***Sell*** Materials (includes Order Processing & Invoicing)
    - **A224.** ***Ship &Deliver*** Materials
    - **A225.** ***Manage Inventory***
  - **A23.** ***Plan*** Materials (types, at project & corporate levels)
    - **A231.** ***Collect & Maintain*** Materials Information
    - **A232.** ***Select & Specify*** Materials Requirements (i.e., specification)
    - **A233.** ***Handle Submittals***
    - **A234.** ***Identify & Quantify*** Materials Requirements
    - **A235.** ***Plan Materials Cost***
    - **A236.** ***Schedule*** Materials
    - **A237.** ***Plan Materials Acquisition***
    - **A238.** ***Plan Materials Safety***
    - **A239.** ***Plan Materials Fabrication***
  - **A24.** ***Request*** Materials
    - **A241.** ***Identify/List*** materials
    - **A242.** ***Suggest*** (possible) Suppliers
    - **A243.** ***Send Request*** to Procure
    - **A244.** ***Prepare & Submit a Revision of Materials Requisition***
  - **A25.** ***Procure*** Materials
    - **A251.** ***Receive & Process Materials Request*** (from PM/CM or Site)
    - **A252.** ***Purchase*** Materials (including suppliers selection, negotiation, and contracting)
    - **A253.** ***Transport/Deliver*** Materials
    - **A254.** ***Receive & House/Store*** Materials
    - **A255.** ***Handle Non-Conforming*** Materials (NCMs)
    - **A256.** ***Respond to Changes in Materials Requirement***
    - **A257.** ***Expedite*** (i.e., Quicken) Processes
    - **A258.** Manage ***After-Sale Services*** (from Supplier)
    - **A259.** ***Manage/Handle Disposal and Surplus*** Materials
  - **A26.** ***Handle/Manage*** Field Materials
    - **A261.** ***Request*** Materials (from Procurement body)
    - **A262.** ***Receive & Process Materials Request*** (from Operational bodies)
    - **A263.** ***Field-Purchase*** Materials
    - **A264.** ***Receive & Store/House*** Materials (from Procurement, Suppliers, or Common Carries)
    - **A265.** ***Handle Non-Conforming*** Materials
    - **A266.** ***Distribute & Transport/Handle*** Materials
    - **A267.** ***Manage/Handle Disposal & Surplus*** Materials
    - **A268.** ***Supervise Crafts' Material Operations***
    - **A269.** ***Manage Materials Space*** Materials
  - **A27.** ***Process/Use*** Materials (at the Site, Workshop, etc.)
    - **A271.** ***Fabricate*** Materials
    - **A272.** ***Install*** Materials

Figure 8–4: Overall Process Breakdown of Operational Processes ("Manage Materials") (p. 1/2)

A28. ***Monitor & Control*** Materials (Project, Site/Field, & Corporate)
- **A281.** Control Materials ***Changes***
- **A282.** Control Materials ***Quality*** (before, during, and after processing)
- **A283.** Control Materials ***Cost***
- **A284.** Control Materials ***Inventory*** (e.g., tracking materials)
- **A285.** Control Materials ***Schedule***
- **A286.** Control Materials ***Waste, Surplus, and Disposal***
- **A287.** ***Expedite*** (i.e., Quicken) Materials Processes

Figure 8-4: Overall Process Breakdown of Operational Processes ("Manage Materials") (p. 2/2)

A2. **Manage Materials**

A23. ***Plan*** Materials (at project & corporate levels)

- **A231.** ***Collect & Maintain*** Materials-Related Information
  - **A2311.** Collect & Maintain Materials Information (e.g., cataloging and standards)
  - **A2312.** Collect & Maintain Materials Samples (i.e., samples collection)
- **A232.** ***Select & Specify*** Materials Requirements
  - **A2321.** Search & Find Materials (i.e., browsing the market)
  - **A2322.** Evaluate & Select Materials (Type)
  - **A2323.** Specify Materials (Preliminary & Detailed Specs of Materials Types)
- **A233.** ***Handle Submittals***
  - **A2331.** Prepare and Submit Submittals
  - **A2332.** Review Submittals
  - **A2333.** Revise and Resubmit Submittals
- **A234.** ***Identify & Quantify*** Materials Requirements (i.e., Identification, Quantification and Consolidation of materials requirement)
  - **A2341.** Identify Type of Estimate (e.g., detailed/elemental cost estimate)
  - **A2342.** Identify Scope Of Estimate (e.g., a Product or a Process)
  - **A2343.** Identify Scope Objects and their Materials Elements (e.g., walls and their related materials)
  - **A2344.** Quantify Objects and their Materials Elements
  - **A2345.** Consolidate Quantities (in the form of Bill Of Materials)
  - **A2346.** Forecast Materials Usage Level (i.e., quantity over time, from production schedule)
  - **A2347.** Determine Materials Safety Stocks
- **A235.** ***Estimate Cost*** of Materials
  - **A2351.** Estimate Materials Cost
  - **A2352.** Budget Materials Cost
- **A236.** ***Schedule*** Materials (i.e., its Order, Delivery, Use, etc.)
  - **A2361.** Define Materials Schedule
  - **A2362.** Evaluate Materials Schedule
  - **A2363.** Modify Materials Schedule
- **A237.** ***Plan Materials Acquisition*** (e.g., Contracting Strategy Selection)
- **A238.** ***Plan Materials Safety*** (e.g., preparation of plans and procedures for Hazardous Materials)
- **A239.** ***Plan Materials Fabrication*** (e.g., Request For Jobsite Dimensions)

Figure 8–5: A Partial Process Breakdown of "Plan Materials"

A2. **Manage Materials**

- **A25.** ***Procure*** Materials
  - **A251.** ***Receive & Process Materials Request*** (from PM/CM or Site)
    - **A2511.** Receive & Group Requests
    - **A2512.** Prioritize Requests (i.e., Reject or Approve; zero-priority means *rejected.*)
    - **A2513.** Check Availability of Materials
    - **A2514.** Assign/Allocate Materials to the Requester
  - **A252.** ***Purchase*** Materials
    - **A2521.** Select Suppliers (i.e., manufacturers, distributors, etc.)
      - **A25211.** Select Purchasing Strategy (e.g., selecting between buying directly from manufacturer or from distributors as well as selecting an RFQ/IFQ strategy)
      - **A25212.** Select Potential Suppliers
      - **A25213.** Request For Quote (RFQ)
      - **A25214.** Invite For Quote (IFQ)
      - **A25215.** Receive & Process RFQ/IFQ (concluding negotiations)
      - **A25216.** Select target Suppliers (concluding negotiations)
    - **A2522.** Contract with Suppliers
    - **A2523.** Order/Reorder materials
    - **A2524.** Receive Order Acknowledgement
    - **A2525.** Receive & Process Invoice
    - **A2526.** Confirm/Facilitate for Payment (to the supplier/vendor)
  - **A253.** ***Transport & Deliver*** Materials (Traffic or Logistic function)
    - **A2531.** Identify Receiver/Destination (e.g., storage, site, operational bodies, etc.)
    - **A2532.** Identify Delivery Mode (e.g., FOB, C&F, CIF, etc.)
    - **A2533.** Identify Delivery Terms (e.g., destination, obligations and liabilities, etc.)
    - **A2534.** Determine Delivery Technique (i.e., selection of handling equipment)
    - **A2535.** Assign Delivery Crew (own crew or external services)
    - **A2536.** Inform Receiver of the Shipment (PO ref, line items, quantities, due time, etc.)
    - **A2537.** Transport/Ship Materials to Destination (e.g., from supplier/storage to storage/site/operational bodies or from site storage to the point of usage)
    - **A2538.** Deliver Materials
  - **A254.** ***Receive & House/Store*** Materials and ***Handle Non-Conforming*** Materials (NCMs)
    - **A2541.** ***Receive*** Materials
      - **A25411.** Check-in (usually in the check-in gate at the yard): The gate guard verifies the identity of the truck and the shipment, and checks them for security.
      - **A25412.** Identify Received Materials (RMs); by the receiving clerk
      - **A25413.** Inspect RMs & Identify, Record, & Report Non-Conforming Materials (NCMs) [i.e., Defective & Shortage/Overage/Wrong-Item]; by the quality control inspector
      - **A25414.** Record & Report RMs
    - **A2542.** Store/House Received Materials (in Warehouses &Laydown Areas)
      - **A25421.** Assign Space
      - **A25422.** Assign Storing Crew
      - **A25423.** Inform Crew & Store Materials
    - **A2543.** Handle Non-Conforming Materials (NCMs); e.g., defectives, wrong items, etc.
      - **A25431.** Identify Type of Non-Conformance
      - **A25432.** Determine Type of Treatment of NCMs (i.e., Store/Repair/Return/Sell/Dispose/Recycle)
      - **A25433.** Treat NCMs (Store/Repair/Return/Sell/Dispose/Recycle)
  - **A255.** ***Distribute*** Procured Materials (Distribution function)
    - **A2551.** Receive & Process Materials Request [see A251 above]
    - **A2552.** Deliver Materials [see A253 above]

Figure 8–6: A Partial Process Breakdown of "Procure Materials" (p. 1/2)

A256. ***Respond to Changes in Materials Requirement***
- **A2561.** Receive Materials Change Notification (e.g., a changed materials request)
- **A2562.** Change Materials Purchase Order

A257. ***Expedite*** Processes (i.e., Quicken proactively or reactively)
- **A2571.** Expedite Purchase Order
- **A2572.** Expedite Delivery

A258. ***Manage After-Sale Services*** (from Supplier)
- **A2581.** Identify Required Services
- **A2582.** Acquire Services
- **A2583.** Report After-Sale Service Problem

A259. ***Manage/Handle Disposal, Surplus and Waste*** Materials
- **A2591.** ***Manage/Handle Disposal*** Materials
  - **A25911.** Identify Disposal Materials
  - **A25912.** Determine Type of Disposal Materials
  - **A25913.** Treat Disposal Materials (Salvage/Recycle/Dispose…)
  - **A25914.** Monitor Disposals Activities
- **A2592.** ***Manage/Handle Surplus*** Materials
  - **A25921.** Identify Surplus Materials
  - **A25922.** Determine Type of Surplus
  - **A25923.** Determine Type of Treatment (considering Surplus Mngt. Strategies and Plans)
  - **A25924.** Treat Surplus Materials (Return/Sell/Store-for-future/Give-away), according to Surplus Plans and Procedures, Contracts, and Agreements (e.g., Buy-Back)
- **A2593.** ***Implement Materials Waste Plans***
  - **A25931.** Observe Wastes
  - **A25932.** Identify Waste Measures
  - **A25933.** Measure Wastes
  - **A25934.** Identify Sources of Wastes
  - **A25935.** Record and Report Wastes
  - **A25936.** Prevent Wastes

Figure 8-6: A Partial Process Breakdown of "Procure Materials" (p. 2/2)

A2. **Manage Materials**

- **A26.** ***Handle/Manage*** Field Materials
  - **A261.** ***Request*** Materials (from Procurement body) [see A24 in Figure 8-4]
  - **A262.** ***Receive & Process Materials Request*** (from Operational bodies)
    - **A2621.** Receive & Group Requests
    - **A2622.** Prioritize Requests
    - **A2623.** Check Availability of Materials
    - **A2624.** Assign/Allocate Materials to the Requester
  - **A263.** ***Field-Purchase*** Materials
  - **A264.** ***Receive*** Shipped Materials (from Procurement dept., Suppliers or Common Carriers)
    - **A2641.** Check-in (usually in the check-in gate at the yard): The gate guard verifies the identity of the truck and the shipment, and checks them for security.
    - **A2642.** Identify Received Materials (RMs), by the receiving clerk
      - **A26421.** Remove the packing slip of the container
      - **A26422.** Mark PO no. on the container, if not done so
      - **A26423.** Verify shipped items against POs/ shipping notice, specs, and drawings
    - **A2643.** Inspect RMs & Identify, Record, & Report Non-Conforming Materials (NCMs) [Defective & Shortage/Overage/Wrong-Item], by the quality control inspector
    - **A2644.** Record & Report RMs
    - **A2645.** Update Materials Inventory
    - **A2646.** Report Receiving
      - **A26461.** Create & Distribute the "Receiving Report"—usually to Purchasing, Accounting, & Storage
      - **A26462.** Report defectives, shortages, & overages to vendor
  - **A265.** ***Store/House*** Received Materials (in Warehouses & Laydown Areas)
    - **A2651.** Assign Space ("in Warehouse—by craft, or in Laydown areas—for large bulks, engineered equipment, fabricated"), "done by storage supervisor".
      - **A26511.** Get list of potential, available places/spaces
      - **A26512.** Consider usage constraints (e.g., accessibility)
      - **A26513.** Consider place's technical constraints (e.g., strength of a slab from concrete tests; may need to call & ask the testing agency)
      - **A26514.** Consider material's technical constraints (
    - **A2652.** Assign Storing Crew
    - **A2653.** Inform Crew & Store Materials
  - **A266.** ***Handle Non-Conforming*** Materials (NCMs); e.g., defectives, wrong items, etc.
    - **A2661.** Identify Type of Non-Conformance
    - **A2662.** Determine Type of Treatment of NCMs (i.e., Store/Repair/Sell/Return/Dispose/Recycle)
    - **A2663.** Treat NCMs (Store/Repair/Return/Sell/Dispose/Recycle)
      - **A26631.** Store/Keep NCMs (for repair and/or later use, e.g., in other projects)
      - **A26632.** Repair NCMs
      - **A26633.** Return NCMs (to Procurement Dept. or Supplier); e.g., Buy-back process
      - **A26634.** Sell NCMs
      - **A26635.** Dispose NCMs
      - **A26636.** Recycle NCMs
  - **A267.** ***Distribute & Transport*** Materials (to Operational Crafts/Crews)
    - **A2671.** Inform Receiver of the Shipment (PO ref, line items, quantities, due time, etc.)
    - **A2672.** Assign Delivery Crew
    - **A2673.** Deliver Materials to Customer (from site storage to point of process)
  - **A268.** ***Manage Disposal*** Materials
    - **A2681.** Coordinate with Disposal Management
  - **A269.** ***Manage/Handle Surplus*** Materials
    - **A2691.** Handle Buy-Back Procedure (BB Agreement)
    - **A2692.** Coordinate with Surplus Management

Figure 8–7: A Partial Process Breakdown of "Handle Materials"

8.2 MM Business Process Modeling in the ITPMS Model Structure

This section describes how the results of the top-down PM/MM process analyses (Chapter 6 and the previous section) and bottom-up investigations (Chapter 3) were formalized into the ITPMS model structure using the Streams CASE tool. Sections 2.5.3 and 2.6.6.2 and Figure 2-21 show how this section relates to the rest of the dissertation.

More specifically, applying the CPMF model suggested for modeling PM functions (section 6.2) and the ITPMS model architecture, the initial process breakdown of MM processes presented in the previous section was further modified to better satisfy the process modeling criteria (section 2.5.2). The common, reusable processes were factored out and modeled with UML *packages* and *use cases*. The relationships between use cases were modeled with UML *use case models*, while the general flow of information and actions for each use case were modeled with UML *activity diagrams*. The following presents an overview of the model and an explanation of its included processes (in a package hierarchy), while the detailed, lower-level process models (use case and activity models) are described in section 8.3.

8.2.1 An Overview of the Model and MM Processes

8.2.1.1 The High-Level Components of the Model

Figure 8-8 shows the organization of the model elements defined at the highest level of the ITPMS implementation model. The model includes two top-level packages: *PM Business Models*, and *PM Software Systems Models*. As its name implies, the first package includes those model elements that are defined to reflect the PM business view; thus it is stereotyped as a «business view». The second package, which is stereotyped as a «software view», includes models of the software systems that support the business.

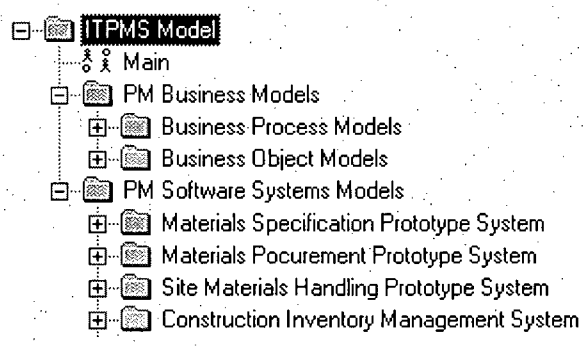


Figure 8–8: The Top-Level Model Eelements of the ITPMS Implementation Model

The *PM Business Models* package has two child packages named *Business Process Models* and *Business Object Models*, stereotyped as «business process view» and «business object view» respectively. The first package includes models of processes (i.e., PM functions and their related business processes), while the second one holds a repository of the business objects defined in the various PM topic areas.

The *Business Process Models* package includes eighteen packages representing the major PM topic areas; thus stereotyped as «topic area». Figure 8-9 illustrates these topic-area packages, one of which is the MM package, which includes definitions of all material-related processes occurring in the context of construction projects. The last package (*Other Potential Functions*) is a placeholder for functions that may be included in the model in the future.

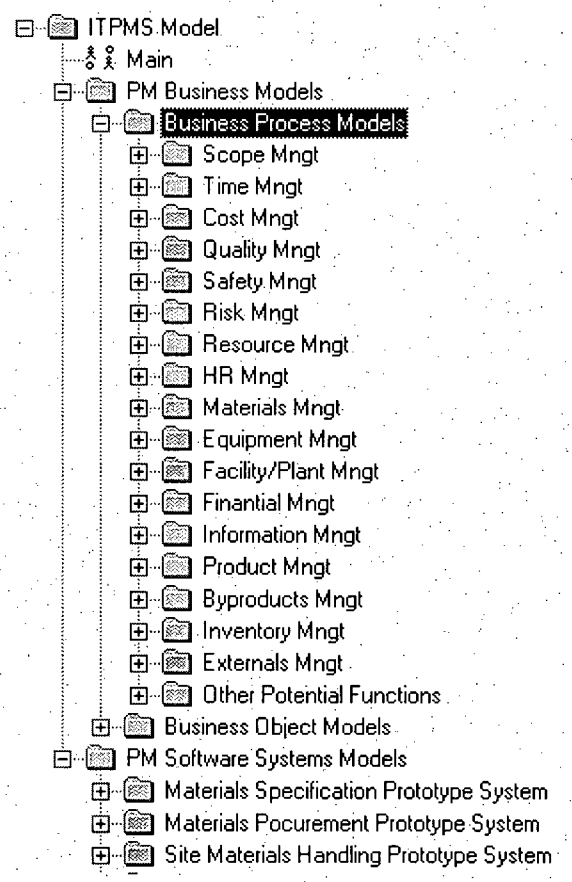


Figure 8–9: The Basic PM Topic Areas Defined in the ITPMS Implementation Model

8.2.1.2 The MM Package at a Glance

The MM function package is initially divided into seven packages (stereotyped as «topic area»): *MM Initiation*, *MM Planning*, *MM Organizing*, *MM Execution*, *MM Controlling*, *MM Closeout*, and *MM Follow-up & Redirection* (Figure 8-10). Each of these packages is further divided into lower-level packages, which include other topic areas as well as business process groups and business processes of

MM systems. At the lowest level, real business processes are modeled by use cases, which are stereotyped as «business use case». The following subsections describe these packages and their included elements, while Figure 8-11 illustrates the overall structure of the MM package and some of its various included elements.

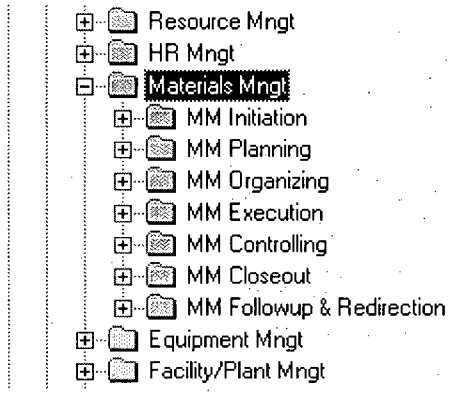


Figure 8–10: The First-Level of MM Business Processes in the ITPMS Model

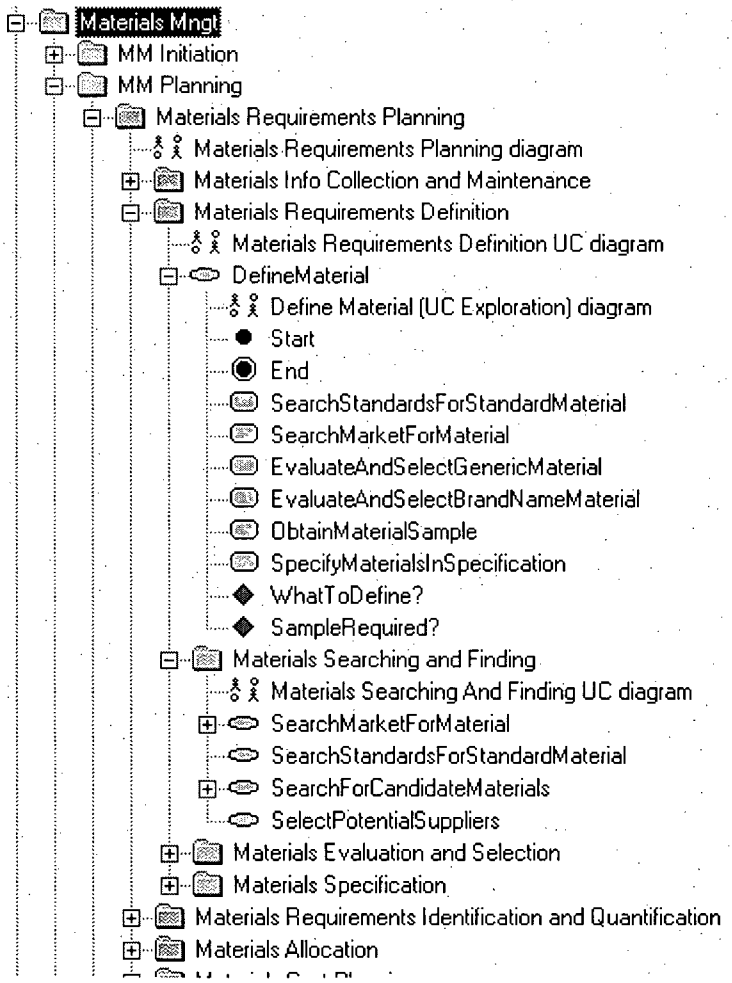


Figure 8–11: The Overall Structure for MM Process Modeling in the ITPMS Model

8.2.1.3 The Assignment of Stereotypes to Model Elements

The assignment of stereotypes to various packages and their included elements is performed according to the meta-model of the ITPMS model (section 7.2). This assignment is facilitated through an interface provided within the CASE tool (i.e., Streams). Figure 8-12 shows an example interface through which such assignments are performed. The figure also shows a list of stereotypes used for packages.

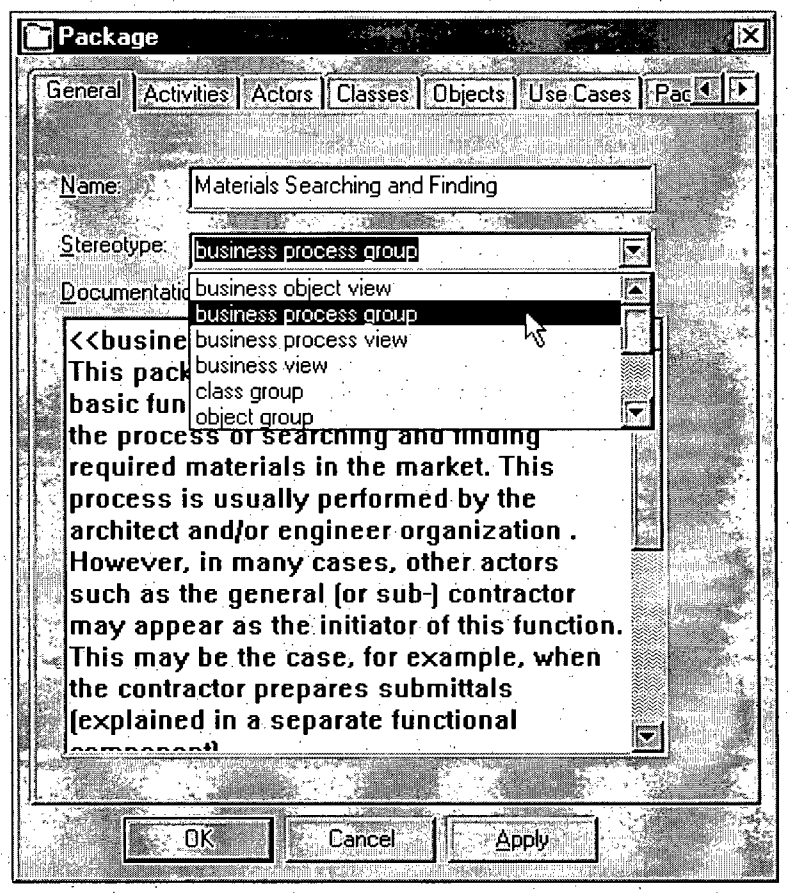


Figure 8–12: An Example of the Assignment of Stereotypes to Packages

8.2.2 MM Initiation

The MM Initiation package (of type «topic area») includes models of the business processes involved in the initiation of the MM system; i.e., bringing the MM system into effect (e.g., committing the organization to carry out MM tasks). The processes mostly portray an organizational view of MM; i.e., when MM functions are a recognized area of responsibility of an organizational unit (e.g., a MM Department, Team, etc.). The processes defined in this package include *MM Goal and Objective Definition*, *Initial Team Establishment*, and *System Formalization*. Many of the use cases representing MM processes in this package and its sub-packages are reused (mostly through specialization) from those defined in the *Scope Management* package.

8.2.3 MM Planning

MM Planning encompasses both *developing and maintaining plans in the MM system.* Here, the term *planning* is used very broadly to mean all planning activities that relate in some way to materials. MM planning may relate to various elements of the MM system and it may include such areas as construction operations planning, resource planning, maintenance planning, and organizational planning within MM systems. However, central to MM planning activities is the identification and definition of materials requirements in terms of *what*, *how much*, *when*, and *where* materials are needed and *how* materials are to be acquired and handled.

In the ITPMS model, the planning component of MM is represented by a UML package notation labeled *MM Planning*, with the stereotype «topic area». This package is decomposed into two separate, yet interrelating, functions (with the same stereotype as their parent): *Materials Requirements Planning* and *MM System Planning*. The first function deals directly with the *materials* incorporated into the constructed facility, while the latter concerns the whole MM system (MMS), including the MM organization within which materials are planned and handled.

Since the output of planning functions are usually used as a basis for control functions, there is a close relationship between the activities and objects of these two functions. For example, the result of a *Materials Cost Estimating* process is used in a *Materials Cost Control* process (represented in the *MM Controlling* package). The functions included in the *MM Planning* package are explained in the following.

8.2.3.1 Materials Requirements Planning

According to Stukhart [1995], the Materials Requirements Planning (MRP) function encompasses the activities associated with the definition, quantification, and procurement timing of project materials. It spans the organizational groups of owner, engineer, and constructor. In the ITPMS model, however, the MRP package focuses on the product of MM systems (i.e., the *material*). It comprises all processes relating to the planning of the materials (rather than the MM system's required resources, for instance); i.e., what, how much, when, and where materials are required for a project. Thus, the processes included in the MRP package may encompass the *identification*, *definition*, *modification*, *quantification*, *costing*, and *timing* of the materials requirements (e.g., materials selection, takeoff, estimating, and scheduling).

Figure 8-13 illustrates the basic functions defined in the MRP package. These functions (stereotyped as «business process group») and their related processes are further described as follows.

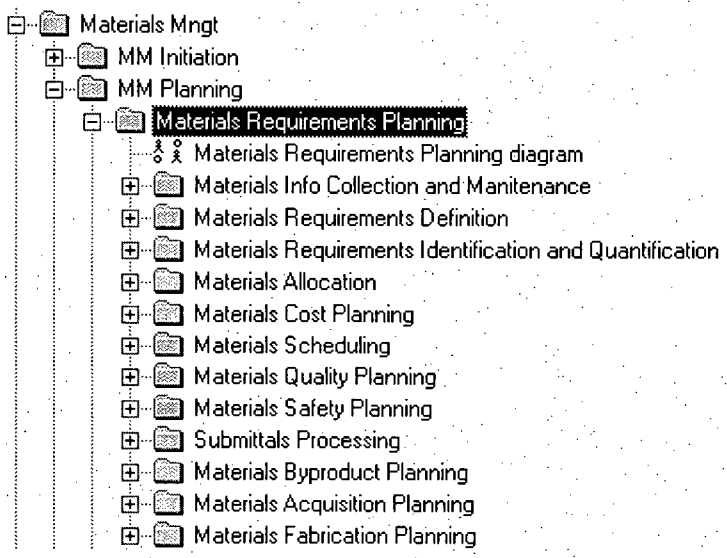


Figure 8–13: The "Materials Requirements Planning" Package in the ITPMS Model

- ***Materials Information Collection and Maintenance***: this package includes all processes encompassing the collection and maintenance of materials information. Examples of such processes are *Request Material Information* or *Obtain Material Sample*. The information may range from materials properties (e.g., the product data published by manufacturers) to standards information (e.g., materials safety and engineering standards).
- ***Materials Requirements Definition***: this package includes all processes involved in the definition of a project's materials (as opposed to standard materials). Such processes are grouped into three packages: *Materials Searching and Finding*, *Materials Evaluation and Selection*, and *Materials Specification*.
- ***Materials Requirements Identification and Quantification***: this package includes all processes concerning the materials quantification as it relates to the planning of MM systems (i.e., the types and quantity of materials); Examples include estimating quantities in a *Materials Quantity Estimate* and consolidation of the quantities into a *Bill Of Materials (BOM)*, and *Inventory Plan* development. Such processes may take place for a variety of purposes. For example, a BOM may be created for the purpose of materials cost estimating, tendering, requisition, ordering, or expediting purposes. However, of all these, only the *quantification* and *consolidation* of the required materials (which are already selected and specified, e.g., in specifications) in the BOM is within the scope of this package.
- ***Materials Allocation***: this package includes definitions of various business processes involved in assigning specific materials to be used for a specific assembly. Stukhart [1995, p. 284] explains various types of materials allocation. Examples are *committed allocation* (i.e., setting the materials

aside in the inventory documents to be used for a specific assembly), *trial allocation* (i.e., allocation of materials to work packages for the purpose of work scheduling by checking shortages against the schedule and scheduled delivery), and *physical allocation* (i.e., setting the materials aside and bundling or tagging it for its issue and ultimate use).

- ***Materials Cost Planning***: this package includes all processes concerning the cost planning of materials. The term *cost planning*, here, is used in a very broad sense; i.e., economic analysis and planning (e.g., cash-flow analysis, cost/benefit analysis, cost estimating, value analysis/engineering, and so on). A materials cost estimate may be developed with different levels of details and accuracy (e.g., an order-of-magnitude cost estimate versus an elemental or detailed cost estimate) and for various purposes (e.g., for tendering, budgeting, and so on). In this package, it is assumed that Materials Requirements are already identified and known. In other words, this package uses the results of the processes defined in the *Materials Requirements Identification and Quantification* package.
- ***Materials Scheduling***: this package includes all processes concerning the definition, evaluation, and modification of a materials schedule. Here, the term *materials schedule* generally refers to the definition of when, where, and for what and whom materials are needed and the definition of the chain of activities involved in the provision of such materials.
- ***Submittals Processing***: this package includes all processes related to submittals; i.e., the submission of product (i.e., material) data, samples, and shop drawings by the contractor to the architect and engineer for approval and the verification of the right products. The submittals process may equally be required for products that are unspecified, partially specified, or even fully specified. In many cases (e.g., for fast delivery of the project such as in fast track contracts), products are not fully (if at all) specified in the drawings and specifications. Thus, the final selection of appropriate materials by the architect is done during the submittals process [Minks and Johnston 1998, pp. 56-57]. For the evaluation of the materials (associated with a submittal), this package uses the process definitions suggested in the *Materials Evaluation and Selection* package.
- ***Other Possible Functions***: in addition to the functions listed above, some other functions may be included in the Materials Requirements Planning package. Examples include (but are not limited to) *Materials Quality Planning* (e.g., preparation of materials quality plan, and its communication with related project participants), *Materials Safety Planning* (e.g., identification of hazardous materials and their related safety rules and standards as well as disposal planning of such materials), *Materials Acquisition Planning* (e.g., contracting strategy selection, negotiated procurement versus competitive bidding procurement—i.e., Request For Quotation versus Invitation For Quotation), *Materials*

Fabrication Planning (e.g., requesting jobsite dimensions by the fabricator), *Materials Byproduct (e.g., Scrap, Surplus, and Disposal) Planning* (e.g., identification of potential surplus, required actions for monitoring the trends and patterns of such potential surplus, and various means of disposal, alternative or required actions for surplus—e.g., substitution, return, sale, transfer, etc.- and communication of the plans).

8.2.3.2 MM System Planning

The MM System Planning package includes the planning processes related to the selection of materials acquisition and handling methods. It concerns the whole MM system (MMS), including the MM organization. The processes included in this package range from preparation of *materials plan* and *procedures* to the planning of the *resources* required to handle materials. The functions included in this package are illustrated in Figure 8-14 and are explained in more detail in the following.

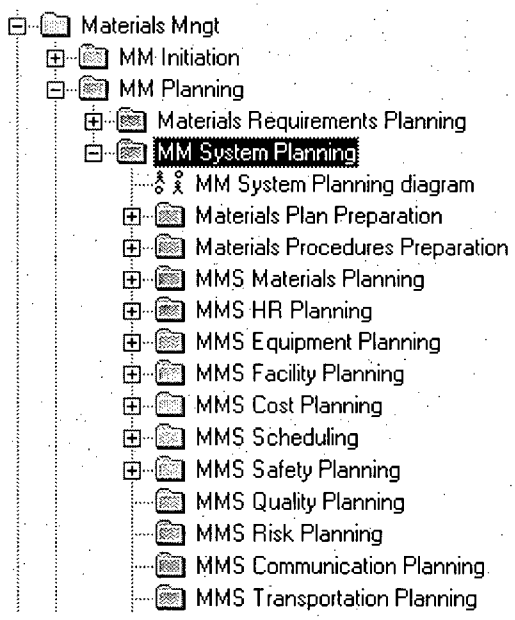


Figure 8–14: The "MM System Planning" Package in the ITPMS Model

- ***Materials Plan Preparation***: this package includes the process of preparing the *Materials Plan*, whose primary purpose is to develop the scope, parameters, and interfaces of MM functions. The definition and attributes of the materials plan [Stukhart 1995, p. 295] suggests a close relationship between this package and other MM planning packages, especially those defined under *Materials Requirements Planning* (section 8.2.3.1). While the *Materials Plan Preparation* package focuses on the overall planning of materials (e.g., methods of transportation, i.e., a managerial perspective), the *Materials Requirements Planning* package encompasses other planning activities (e.g., identification, specification and so on) specific to the *materials* themselves (i.e., a technical-operational view). The

Materials Plan Preparation package is a major user of the results of some other packages such as *Materials Requirements Planning*, *MMS Facility Planning*, and *MMS Equipment Planning*. The identification and definition of materials-related requirements is considered to be within the scope of the latter packages.

- ***Materials Procedures Preparation***: this package includes the process of preparing *Materials Procedures*. These materials procedures, which are much more detailed than the materials plan, describe the functional relationships (e.g., communication and reporting links) within the MM system and between the system and the project organization. These functional procedures are intended to serve as "hands-on, detailed instructions on how to execute each function" [Stukhart 1995, p. 157]. They are also used as a ready reference in personnel training.
- ***MMS Facility Planning***: this package includes the processes involved in planning the facility structures used for materials storage. Examples of such processes are identification and quantification of types of storage, selection of storage acquisition strategy (e.g., rent/own/buy-service), and so on. The facility may be of concern as it relates to both construction materials and the MM system. There is a very close relationship between this package and the *Facility Management* package (Figure 8-8) as it may reuse the definitions suggested by that package.
- ***MMS Materials Planning***: this package includes the processes that relate to planning the materials required for the MM system itself (e.g., office supplies and so on). Many process definitions in the Materials Requirements Planning package (section 8.2.3.1) under the MM Planning package may be reused within this package.
- ***MMS Equipment Planning***: this package includes the processes involved in planning the equipment used in handling materials. Examples of such processes are selection of materials handling equipment, selection of rent/lease/own strategy, and so on.
- ***MMS HR Planning***: this package addresses human resources-related planning processes of MM systems (e.g., staffing, training, and so on).
- ***MMS Cost Planning***: Similar to the Materials Cost Planning package (section 8.2.3.1), this package includes definitions of cost planning processes (including estimating and budgeting), though with the scope of the whole MM system in which materials processes occur. This addresses the cost of various elements of the system such as equipment, supplies, HR resources, and so on.
- ***MMS Scheduling***: this package includes all processes concerning the definition, evaluation, and modification of an MM system schedule. Here, the term *MM system schedule* refers to a chain of activities involved in the MM system. The schedule may include any managerial, technical, and/or

operational activity in the MM system. This package is closely related to the *Materials Scheduling* package (section 8.2.3.1).

- ***Other Possible Functions***: *Functions:* in addition to the functions listed above, some other functions may also be included in the *MM System Planning* package. Examples are those functions related to the planning of *Safety*, *Quality*, *Risk*, *Communication* and *Transportation* in the MM system.

8.2.4 MM Organizing

This package refers to the pre-execution (though non-planning) processes. It is mainly concerned with the MM system itself, and it encompasses the function of arranging the organization (i.e., the MM system team and so on) required for MM activities. The included processes in this package are the *acquisition of required human resources*, *equipment*, *supplies*, and *spatial facility* (e.g., storage), as well as provision of other things that may be considered as preconditions (grounds) for real execution of MM plans (e.g., possible *permits*). The identification and quantification of such requirements, however, are within the scope of the *MM System Planning* package (section 8.2.3.2).

8.2.5 MM Execution

This package includes business processes and their related activities as they relate to the *execution* of the MM system (i.e., coordinating resources and implementing the MM plans). Figure 8-15 illustrates the process groups defined under this package. Some of these packages may seem to be more relevant to other topic-area packages; however, in order to avoid complexity and considering the research scope, they have been included here under MM. Examples are *Selling*, *Shipping*, *Distributing*, and *Storing* packages, which may be relevant to both *MM* and *Product Management*. Such packages may be included under one of the topic areas, and their contents be reused in others requiring such definitions. The process areas defined under MM Execution are explained in the following.

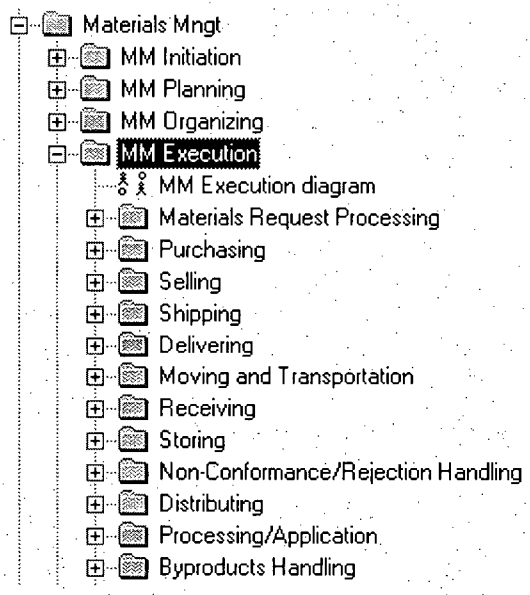


Figure 8–15: A Partial View of the "MM Execution" Package in the ITPMS Model

- ***Materials Request Processing***: this package contains definitions of the business processes that relate to the *requisition* of materials. It includes both the *requisition* tasks and the *processing of the request.*
- ***Materials Purchasing***: this package includes definitions of business processes relating to materials *purchasing*. It mostly reflects a purchaser's viewpoint. The major processes incorporated in this package are *contracting and ordering* (i.e., supplier selection, quoting, and ordering), *invoice processing*, and *field purchasing*.
- ***Materials Selling***: this package encompasses the processes of the *selling* function; i.e., a seller's viewpoint, as opposed to the purchaser's point of view (which is reflected in the *Purchasing* package under *MM Execution* package). The major processes incorporated in this package are *invoicing*, *bidding/quoting* (i.e., responding to RFQ/IFQ), and *order processing* (e.g., order acknowledgement).
- ***Materials Shipping***: this package includes definitions of business processes relating to the *shipping* (i.e., sending) of materials by the supplier to the buyer. Examples of the processes are the *shipping notification of the buyer/receiver*, and *physical shipping*.
- ***Materials Delivering***: this package encompasses the *delivery* process from a supplier's point of view; in contrast with the *receiving* function, which reflects a buyer's point of view. It is comprised of processes such as *physical delivery of materials* and *delivery problem handling*. The delivery process usually acts as a link between the *shipping* and *receiving* processes. A delivery also often involves *moving and transportation*. Thus, a close relationship may be assumed between the processes defined under the *Delivering* package and those defined in the *Shipping*, *Receiving*, and *Moving and Transportation* packages.

- ***Materials Moving and Transportation***: this package includes all business processes related to (*off-site and on-site*) *materials handling of materials*. Examples include *transporting materials to the buyer's storage or site* (e.g., directly by the buyer or through an intermediate transportation agency), *moving materials within the site* (e.g., from storage to the point of use).
- ***Materials Receiving***: this package represents the *receiving* function, which reflects a buyer's point of view; in contrast with the *delivering* function, which reflects a supplier's point of view. It includes definitions of processes such as *handling shipment at the gate*, *shipment validation*, and *shipment checking and inspection*.
- ***Materials Storing***: this package encompasses the function of *storing materials* (e.g., in warehouses or laydown areas). Examples of the processes included in this package are *identification of available spaces*, *allocation of spaces to materials*, *assignment of crew resource*, and *physically storing of materials*. The subject of storage may include not only the received materials (e.g., from the supplier) but also surplus and disposal of materials.
- ***Materials Non-Conformance/Rejection Handling***: this package includes definition of the processes that involve handling of the materials that are put aside (e.g., by the receiving function, inspection function, etc.) and considered as non-conforming or rejected materials. The package concerns only the activities that take place after the identification of non-conforming materials. Examples of these processes are *determining the treatment method for rejected or non-conforming materials* (e.g., store, repair, return, etc.), *labeling materials as rejected*, and *repairing, returning (e.g., to the supplier for credit), and disposing rejects* [Ammer 1976, p. 139]. The disposal and storing processes are generally studied in the *Materials Waste & Disposal/Surplus Handling* and *Materials Storing* packages respectively, but any possible specialization of these processes (where they relate to defected/rejected materials) is studied in this package.
- ***Materials Distribution***: This package represents the *distribution* function. It generally portrays the distribution of materials to users from any point of view (e.g., supplier, PM/CM, etc.). On the construction site, for instance, the distribution of materials to various trades may be addressed within this package.
- ***Materials Processing/Application***: this package focuses on the *application (i.e., usage)* of materials. It defines the processes involved in the *fabrication*, *erection*, and *installation* of materials. An example of such processes is the *fabrication of concrete reinforcing* (e.g., in a workshop), in which the reinforcing bars are cut, shaped and assembled in bulk to be moved to and placed in forms for further processes (e.g., placing conduits and pouring concrete). Functions such as requesting,

receiving, storing (i.e., non-fabrication operations) are not considered within the scope of this function and have been dealt with separately (described earlier in this section).

- ***Materials Byproduct Handling***: this package encompasses the treatment of the leftover materials of construction processes. It includes definition of the processes involved in handling scrap and surplus materials as well as disposal of materials. Examples of these processes are returning surplus materials to the original supplier (e.g., using a buy-back agreement [Stukhart 1995, p. 285]), transferring surplus materials from one project to another, scrap collection and segregation (e.g., using a color-code system), quantifying byproduct materials, processing scraps (e.g., drying and cleaning for selling, recycling, etc.), selling surplus and scraps (e.g., to scrap brokers or dealers) [Ammer 1974, pp. 428-431], and disposing of scraps and disposals.

8.2.6 MM Controlling

This package encompasses the function of ensuring that MM objectives are met. This function is usually performed by monitoring (i.e., observing, measuring, recording, etc.) the progress of MM plan execution, identifying deviations from the plans and suggesting and taking corrective actions. This function may be divided into two parts: materials controlling and MM system controlling. The former concerns the control of materials while the latter focuses on the system within which materials flow. Correspondingly, this package includes two subpackages: *Materials Controlling* and *MM System Controlling*. Nevertheless, the processes included in these packages are closely related.

8.2.6.1 Materials Controlling

This package concerns the control of materials, rather than the system within which materials are managed; i.e., the materials and their flows in a project or company rather than the MM organization and system. Such controlling processes [Stukhart 1995, p. 177] include materials *schedule control*, *cost control*, *quality control*, *safety control*, *byproducts control*, *inventory control*, and *expediting*.

8.2.6.2 MM System Controlling

This package concerns the control of the system within which materials are managed. It mostly involves *monitoring (i.e., measuring and recording) and controlling of the performance* of the MM system; i.e., how *efficiently* and *effectively* the system performs. The system's *resources* (i.e., people, equipment, information systems, etc.) and the involved *parties* are the most important subjects of this monitoring and control process. The process areas studied in this package include *monitoring and assessment of performance of MM human resources* (e.g., *labor productivity* related to materials supports), *suppliers*, (e.g., delivery and service responses) *an MM organizational unit* (e.g., procurement

cycle time for engineered materials—i.e., time from purchase approval until the last major item is procured [Stukhart 1995, p. 167]—or cycle time for receiving process of a specific type of bulk material), or the *overall MM system* (e.g., *assessment of the integration level of materials information systems*—e.g., use of integrated database concepts throughout the home office and project site [CII, 1993 and Stukhart 1995, p. 57]), *assessment of efficiency in MM resource utilization* (e.g., percentage of the time a resource is utilized, i.e., not being idle), and so on.

8.2.7 MM Closeout

This package includes models of closeout processes of the MM system; i.e., those materials-related processes contributing in the formalization of the acceptance of the product of the project and closing of the project [Minks and Johnston 1998, p. 382]. Such processes include *punch list generation* (e.g., by the architect and owner's maintenance personnel), *completion*, and *approval*, materials *contracts closeout* (e.g., settlement and resolution of any open items) and *MM administration closeout*.

8.2.8 MM Follow-up & Redirection

This package includes models of the materials-related *follow-up* processes (i.e., involving the accomplishment of after-closeout *commitments* and resolution of *liability* issues) and *redirection* processes (i.e., the strategic, enterprise-level processes aiming at *learning* from past experiences by taking the advantage of historical data for future plans). The latter group of processes has a close link with the planning and control functions (sections 8.2.3 and 8.2.6). The processes included in this package are *resolving materials-related problems* (after project completion), *requesting and accomplishing warranty services* (e.g., after-sale services), *appraisal of the MM system* and *redirection of MM strategies*.

8.3 Business Use Case Modeling in the ITPMS Model Structure

This section further describes the lower-level elements (e.g., use case models) defined in the business process models of the ITPMS model implementation to represent specific functional requirements of PM systems.

8.3.1 An Overview of the Business Use Case Models

In the ITPMS model implementation, functional requirements of PM systems are captured using UML use case models and activity diagrams (Appendix A), through which business objects are identified as information requirements of the systems. These model elements, which specify the business view of PM functions, are distinguished from their counterparts in the computer environment by their stereotypes. Examples of such stereotypes are «business use case» and «business activity» versus «software use case» and «software activity» respectively. All elements may have their textual descriptions attached to them.

At the lowest level of the hierarchy of business-process-view packages (section 8.2), each package (stereotyped as «business process group») includes one or more use case diagrams, activity diagrams, and use cases. Each use case is defined in two major complementary ways:

1) ***Formal Structured Textual Description***: Each use case is formally defined with a structured textual description [Booch et al. 1999, p. 224], which includes several fields:
   a) *The Scope*: an overall statement of the purpose and type of the process the use case involves.
   b) *Preconditions*: a list of conditions that must be true before the use case starts.
   c) *Postconditions*: a list of conditions that must be true after the use case ends, regardless of which scenario (of actions) occurs.
   d) *Main Flow of Events (MFE)*: the primary flow of the events that occur when the use case takes place. A use case may have only one MFE.
   e) *Exceptional Flow of Events (EFE)*: a description of the events that may occur in some special circumstances. A use case may have zero, one or more EFEs.
2) ***Activity Diagrams***: As a complementary technique to the textual description, the workflow of each use case is defined in the model with one or more UML activity diagrams, including their related modeling elements such as *Start State*, *Activity*, *Object*, *Note*, *End State*, etc. (Appendix A). The next section presents a number of such diagrams produced within the Streams CASE tool.

The following section describes the uses case models defined in the *Materials Request Processing* package (section 8.2.5).

8.3.2 An Example: Materials Request Processing

A materials request may undergo a series of processes such as creation, filling, source suggestion, approval, revising, posting/submission, filing, prioritization, legitimacy verification, materials allocation (i.e., sourcing), closing, and so on. Therefore, the *Materials Request Processing* package (section 8.2.5) includes a total of six use cases: *Request Materials*, *Process Materials Request*, *Revise Materials Request*, *Specify Materials Source*, *Approve Materials Request*, and *Post Request Rejection*. Figure 8-16 shows a use case diagram presenting these use cases and their relationships as well as three other use cases (*Check Materials Availability*, *Allocate Materials By Commitment*, and *Allocate Receiving Materials*), which are reused from other packages. Next to these reused use cases, their owning packages are shown. Two of the key use cases, *Request Materials* and *Process Materials Request,* are described in detail in the following sections.

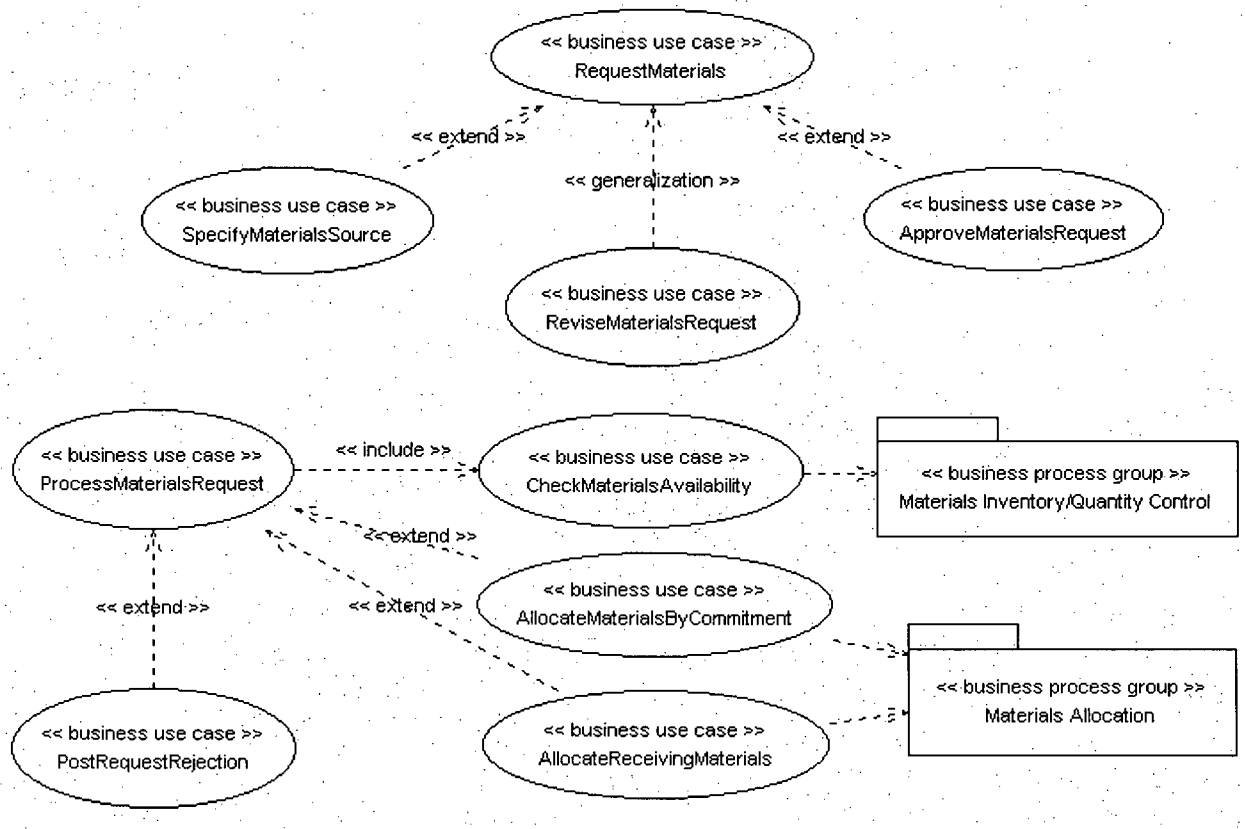


Figure 8–16: "Materials Request Processing" Use Case Digram

8.3.2.1 Use Case: "Request Materials"

8.3.2.1.1 The Scope

This use case generically specifies the process of materials requisition; i.e., a requester placing a new request for some specific materials. Various configuration settings (in terms of the involved actor, the means of communication, etc.) may be assumed depending on organizational and procedural practices. The following, for instance, lists a few of the possible configurations.

1) A foreman or craftsperson requesting materials from the superintendent or the procurement body.
2) The superintendent at the site requesting materials from the procurement department.
3) An assembly workshop requesting materials from the superintendent or the procurement body.
4) A subsidiary storeroom or warehouse requesting materials from its parent storeroom.
5) A computerized system (representing one of the above types of requesters) automatically placing a request, considering the materials inventory and plans.

Before a materials request is posted for fulfillment, it may undergo a series of some other processes, such as material source identification and approval; however, such processes are considered as

extensions to the main process of generation and posting of the request. Therefore, the details of such processes are not considered within the scope of this use case. Moreover, unlike a purchase order, a materials request may not primarily involve an exchange of money for the requested materials. It is usually requested on the basis of some kind of agreement and procedure.

8.3.2.1.2 Preconditions

1) The need for a specific material(s) is known.
2) The requester is registered (e.g., an person or organizational entity authorized to place a request).

8.3.2.1.3 Postconditions

1) A materials request (listing fully specified materials) is submitted for fulfillment.
2) The request is tagged as a pending request and is filed in the requester's system.

8.3.2.1.4 Main Flow of Events (MFE)

1) The use case starts when the need for a material(s) is realized by the requester.
2) The requester invokes the requesting system, which may involve a telephone call, filling a paper or computer form, or other means of communication.
3) The requester expresses the need; i.e., a list of needed materials, in terms of what, for what, by when, etc.
4) The request is submitted (or transferred) to the request-processing agent for fulfillment.
5) The request is tagged as a pending request, and its request is filed in the requester's system.
6) The use case ends.

8.3.2.1.5 Exceptional Flows of Events (EFE)

1) The requester wishes to suggest some sources (e.g., preferred suppliers) for acquisition of the requested materials; however, the details of such a process are not within the scope of this use case.
2) The request undergoes an approval process, whose details are not within the scope of this use case. A request might be partially or totally rejected in this process.

8.3.2.1.6 Activity Diagrams

Figure 8-17 shows the activity diagram representing the workflow involved in the use case *Request Materials*.

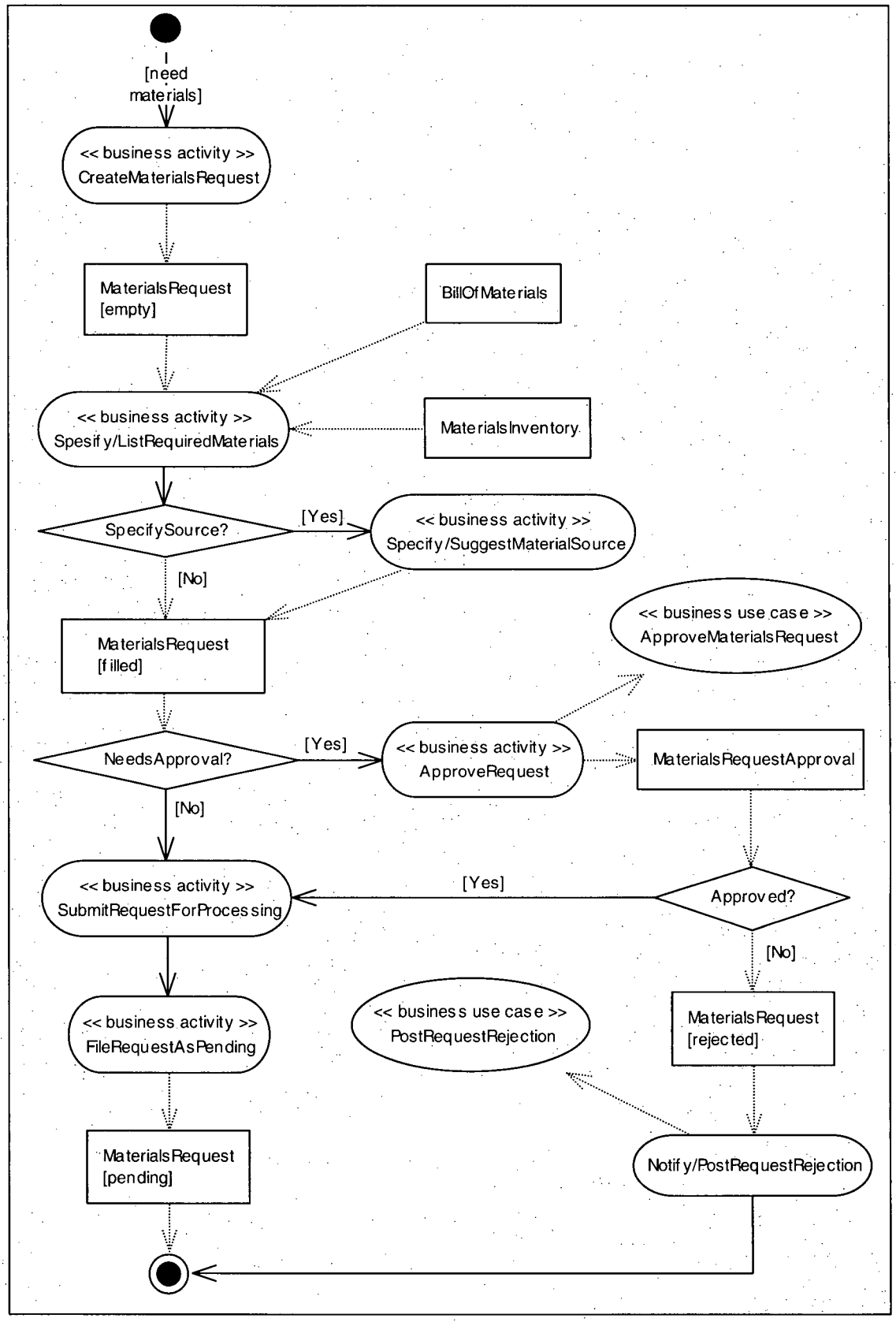


Figure 8–17: "Request Materials" Activity Diagram

8.3.2.2 Use Case: "Process Materials Request"

8.3.2.2.1 The Scope

This use case generically specifies the workflow involved in handling a materials request, after submission of the request for fulfillment. Prioritization, sourcing, and allocation of materials to the request are the main focus of this workflow. Similar to the *Request Material* use case (section 8.3.2.1), various configuration settings (e.g., in terms of the involved actor and the means of communication) may be assumed for this use case.

8.3.2.2.2 Preconditions

1) The legitimacy of the request is proved; hence, it does not require verification.
2) Materials inventory is known.

8.3.2.2.3 Postconditions

1) The prioritized, sourced request is passed for fulfillment.
2) A purchase requisition is generated and sent to the purchasing system, provided the on-stock and scheduled deliveries do not collectively provide for the requested materials.

8.3.2.2.4 Main Flow of Events (MFE)

1) The use case starts when the request-processing agent (RPA) is notified of a material request receipt.
2) The RPA invokes the system.
3) The RPA acknowledges the receipt of the request.
4) The RPA prioritizes the request.
5) The RPA locates available materials.
6) The RPA allocates the available materials to the request (i.e., commitment).
7) The RPA submits prioritized, sourced request to the fulfilling agent (e.g., storeroom).
8) The use case ends.

8.3.2.2.5 Exceptional Flows of Events (EFE)

1) The request is rejected (a priority of zero), and a rejection notification is sent to the requesting agent.
2) The committed material available in-stock does not fill the total requested material, so the materials delivery schedule is searched, and the material scheduled to be received is allocated to the request.

3) Neither the available material nor the material scheduled to be received (individually or in combination) satisfies the required material quantity, a *purchase requisition* is prepared for the material and sent to the purchasing system.

8.3.2.2.6 Activity Diagrams

Figures 8-18 and 8-19 show two activity diagrams representing the workflow involved in the use case *Process Materials Request*. While the first diagram shows the overall flow of the work and its related business objects, the latter focuses on the sourcing and materials allocation activities. Moreover, there are a number of other use cases (e.g., *Check Materials Availability*, *Allocate Receiving Material*, and *Allocate Material By Commitment*) that are referenced in these diagrams. These use cases, which are mainly located in other packages of the model, are in an extend or include relationship with the use case *Process Materials Request* (Figure 8-16); i.e., their definitions are reused here.

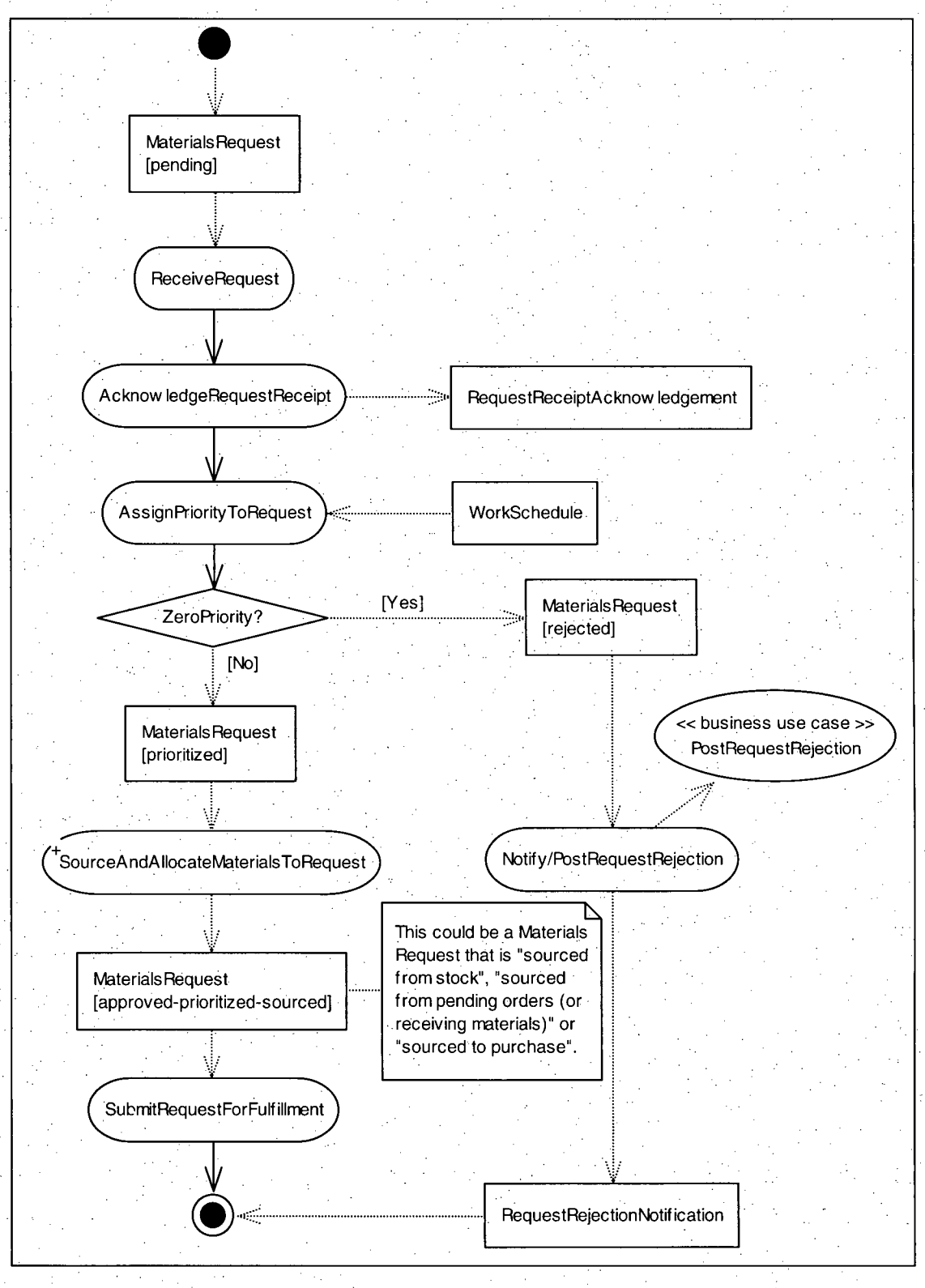


Figure 8–18: "Process Materials Request" Activity Diagram

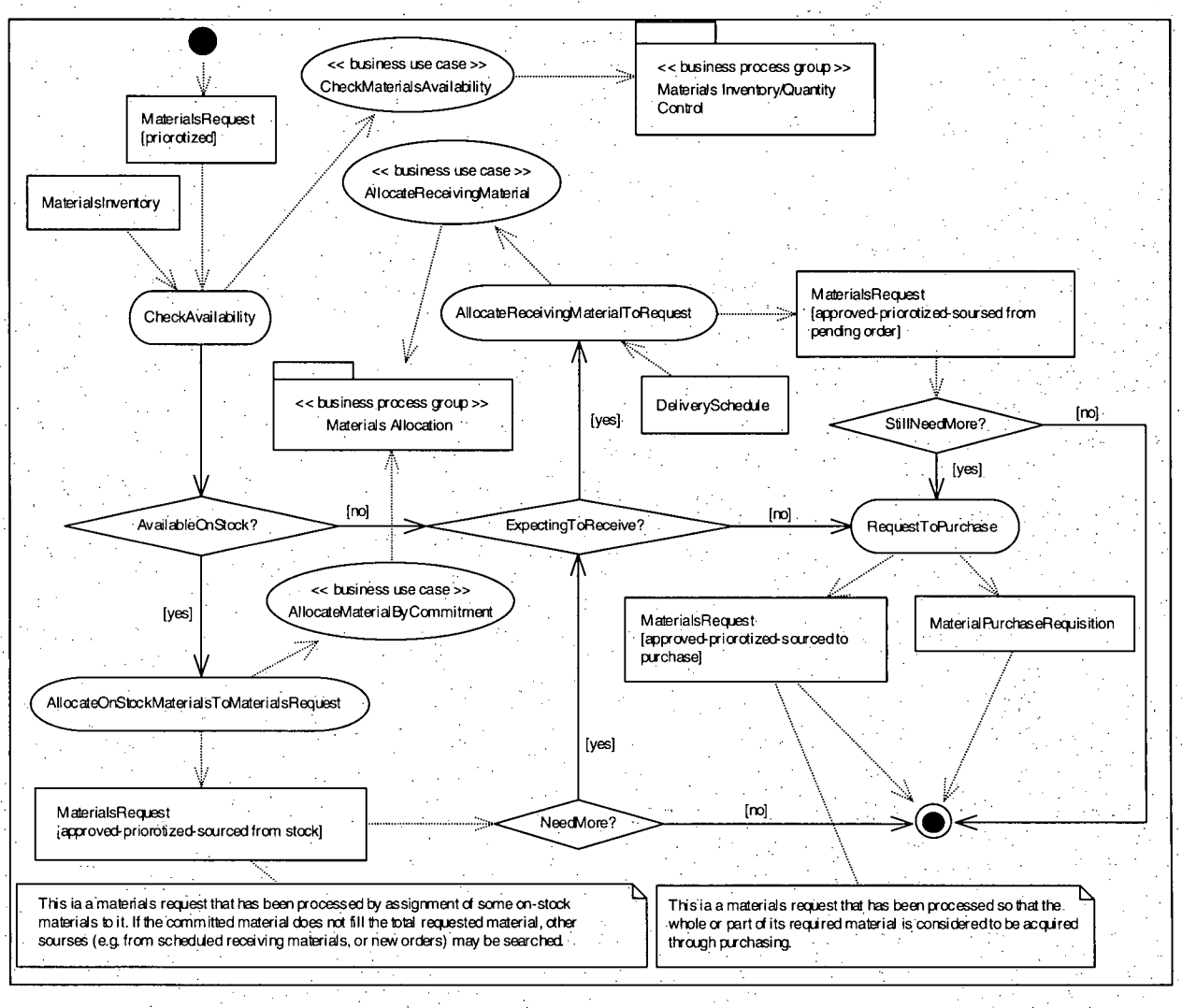


Figure 8–19: "Source And Allocate Materials To Request" Activity Diagram

8.4 MM Business Object Modeling in the ITPMS Model Structure

Business objects are found at the overlap of business and software models (section 2.2.4), and thus, they play an important role in the software development process (section 2.1.3). The identification and grouping of objects and their allocation to process-domain packages is one of the keys to a successful systems analysis [Coad and Yourdon 1991a; Jacobson et al. 1992].

This section describes the *business object view* of the ITMPS model implementation, which is one of the main outcomes of the process modeling activity explained in the previous sections and represents the information requirements of MM systems. The information gathered in the research's bottom-up field investigations (Chapter 3) was a major input to object definitions included in the model.

8.4.1 An Overview and the Approach to Business Object Modeling

In the implementation of the ITPMS model, the *Business Object Models* package is defined under the *PM Business Models* package to reflect the information view of PM business systems. This package, which is stereotyped as a «business object view», holds the business objects defined in various PM topic areas (Figure 8-20). Two packages are defined under the *Business Object Models* package: *MM Business Objects* and *Other Business Objects*; all stereotyped as «object group». The former includes the object models relating to MM functions, and the latter is a placeholder for other object packages encompassing information requirements of other PM functions, whose detailed study were not within the scope of this part of the research. The following describes the *Business Object Models* package and its constituents in more detail, and Appendix D presents a list of these groups and their included objects with their definitions.

8.4.2 Business Objects and Domain Packages

In the ITMPS model implementation, business objects are identified and allocated in two major ways. The majority of these objects are identified through the study and analysis of business processes; i.e., through their role in activity diagrams representing business processes (section 8.3). Such objects are simply allocated to their related processes by reference. However, the model also includes some other objects, which are not used in the activity diagrams of the model. For example, *Material Safety Data Sheet (MSDS)* is an object that is defined in the *Material Safety* package. This object has not been found through process modeling, but rather through the literature review and site visits (Chapter 3). The main reason for inclusion of such objects in the model is to define areas where the future extensions are warranted. The *MSDS* object, for instance, is central to materials safety planning and control but is not specifically within the scope of this research.

Business objects are grouped in packages based on their relevance to various MM domains. Often, a very close mapping may be found between the names of objects' packages (i.e., containers) and their related processes, which are modeled in process packages. For instance, the «object group» package of *Receiving* contains objects (such as *Receiving Report*, *Receiving Issue*, etc.) relating to the receiving function, which is modeled with the «business process group» package of *Receiving* (section 8.2.5).

While most *business objects* defined in the model may ultimately have a software *class* counterpart, some of them have a potential to be mapped into *attributes* or *relationships* between classes. The *PM Software Systems Models* package (Figure 8-20) is intended to include such software model elements, though its elaboration was not within the scope of this research.

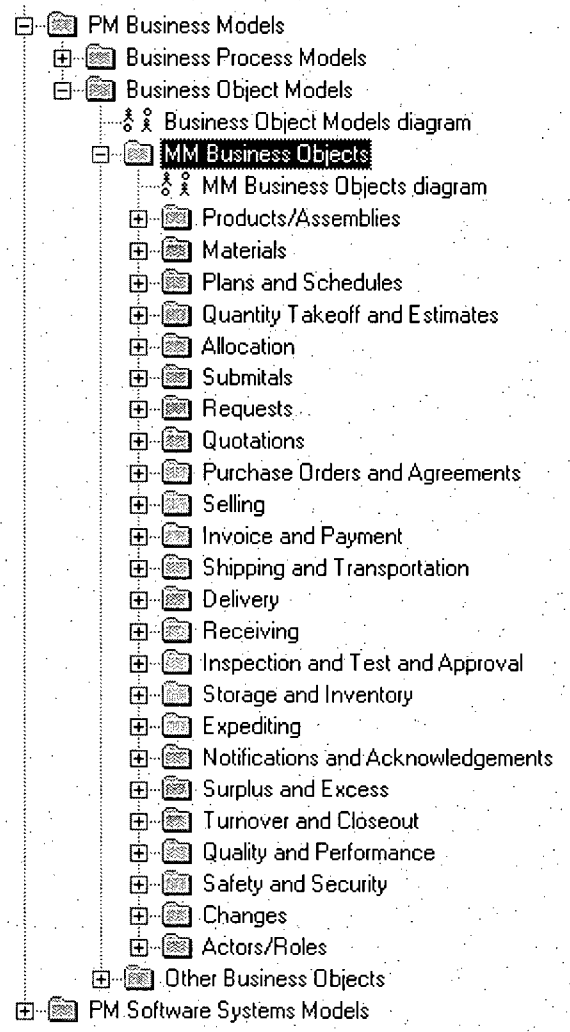


Figure 8–20: The Business Object View of the ITPMS Implementation Model

8.4.3 Types of Information Attached to Business Objects

In the ITMPS model implementation, the information attached to a business object varies from textual descriptions to links to various types of documents. The textual description includes the object's definition, characteristics and attributes, references to other model elements (e.g., objects and packages), and any other types of notes. Figure 8-21 shows two examples of images of typical sample construction documents attached to objects, as well as the textual information attached to the object (*Submittal Log*) and the document image (*Submittal Review Result*). Many such images were produced by scanning real construction documents collected during site visits and observation of field processes (Chapter 3).

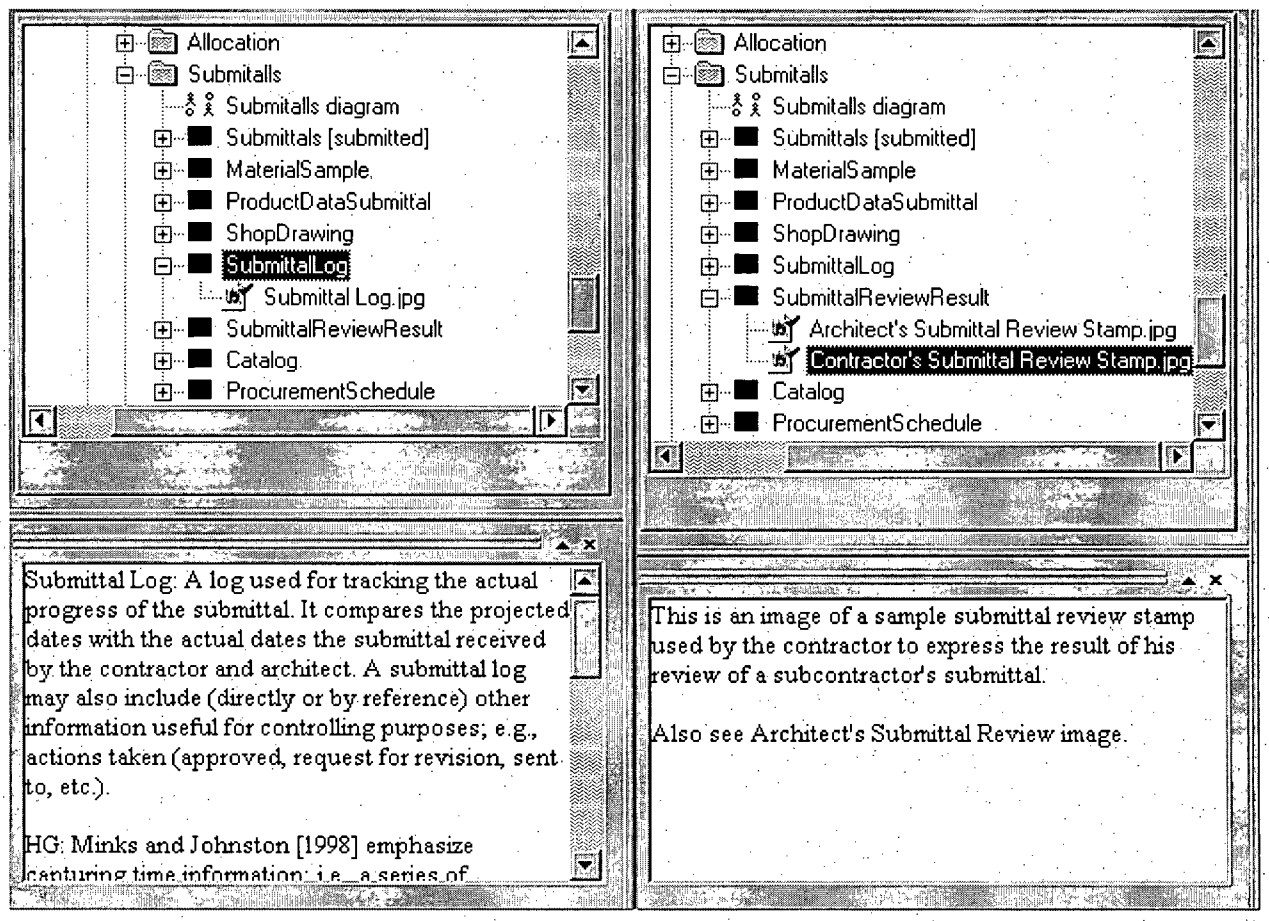


Figure 8–21: Association of Images of Real Objects (e.g., documents) with Business Objects

8.5 Chapter Conclusions

The main purpose of this chapter was twofold: 1) to elaborate the lower levels of the PM functions presented in previous chapter, specifically in the area of MM, and 2) to evaluate the feasibility of the models proposed in the previous chapter—i.e., the CPMF and ITPMS models. It explained the research approach to the classification of MM processes, a description of the implementation of the ITPMS model in a CASE tool and the results of this implementation.

As a result of the MM systems modeling, a large body of MM knowledge was formalized into MM business process and object models. The results of the top-down analyses of PM functions (Chapter 6) and MM systems (section 2.6) and the results of the bottom-up investigations (Chapter 3) were synthesized into the ITPMS model implementation. The immediate result of this synthesis was a set of MM business process models and object models, which was presented in this chapter. A complete listing of the MM business objects, with their definitions, is presented in Appendix D. The implementation model is generally considered as a proof-of-concept for the CPMF and ITPMS models. Moreover, the

suggested process and object models are the major building blocks for modeling software processes and classes required for development of integrated total PM systems.

While opportunities for further work and improvement are foreseen, the results are generally encouraging and useful (section 9.5). It is expected that the results of this chapter, in collaboration with other products of the research, considerably contribute to a better understanding of elements of MM systems (especially their related processes and information requirements) and to the existing standardization efforts in the CIC movement. The next chapter presents an evaluation of the research deliverables, including the MM models.

CHAPTER 9 Evaluation and Validation of the Research

This chapter describes the *evaluation/validation* (E/V) mechanisms and activities of the research and presents an evaluation of the research deliverables.

9.1 The Overall Research Evaluation and Validation

A scientific model validation generally involves an independent evaluation and testing of various aspects of the model (e.g., information *input*, model construction *process*, and model *output*). However, a full-scale validation of the research deliverables was well beyond the scope of the research. In a formal standards development project, the validation process is a major portion of the overall model-development process that is both complex and resource intensive [IFC R2.0, 1999c]. For example, with relatively small scope (energy analysis and HVAC systems in the early stages of building design), the COMBINE model consumed about 70 person-years of resources [Augenbroe 1995].

This research concentrated on the early activities of the software development process and resulted in relatively high-level models of information systems. Given the scope and schedule of the research, taking the models to the downstream activities of the software development process, including implementation and end-user testing, was not feasible. Nonetheless, several activities were incorporated into the research methodology to assess the validity of the research and the usefulness of the *primary research deliverables*.

First, the work followed a *concretization* development process, in which each model was intended to provide an abstract foundation to support the development of the subsequent, less-abstract (more concrete) models. In this way, each model (e.g., the CPMF model, Chapter 6) was evaluated through its use, as a framework, for development of a lower-level model (e.g., the ITPMS model and its implementation, chapters 7 and 8). The lessons learned from this testing were incorporated, as feedback, to reshaping of the earlier models.

Second, an extensive set of modeling criteria was developed from the variety of *bottom-up investigations* (Chapter 3), the review of previous models and other literature (chapters 2 and 4), etc. The research deliverables were then evaluated against these criteria to provide a further degree of validation.

Figure 9-1 illustrates the chain of E/V relationships among the primary research deliverables (boxes with underlined texts) and the bottom-up activities. Table 9-1 summarizes how the primary deliverables address the general modeling criteria set for the research, while the following sections summarize the detailed evaluation of the deliverables against the identified requirements.

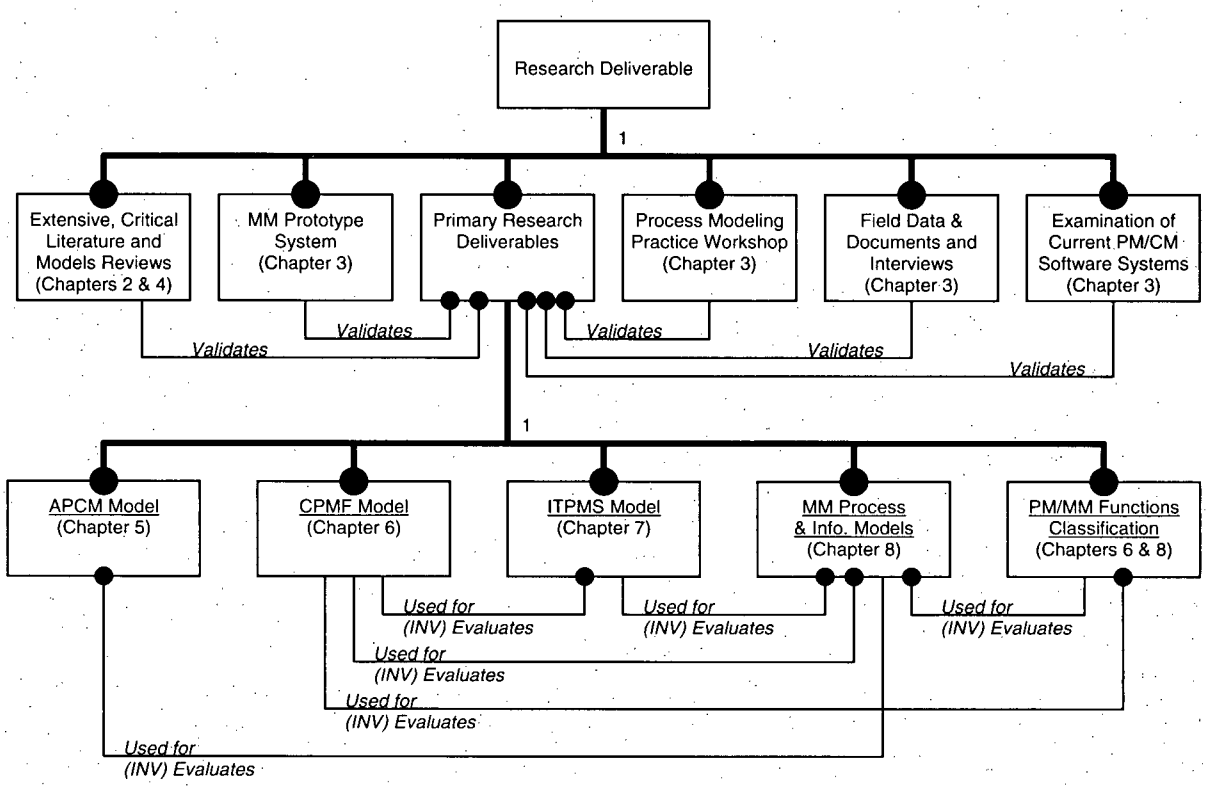


Figure 9–1: The Evaluation/Validation Relatinships Among Major Research Deliverables

Table 9–1: The Primary Research Deliverables Responding to the Research Criteria

(Continued on the next three pages)

Research Deliverable	Research Criteria					
	Comprehensiveness	**Integration**	**Neutrality**	**Flexibility & Openness**	**Modularity & Reusability**	**Realism & Usefulness**
APCM	-Concerns a whole range of project information. -Concerns all phases of an AEC/FM project; i.e., the whole life cycle of the facility (from cradle to grave).	-Integrates *product* and *process information* into a unified project core model.	-Does not imply any specific methodology, architecture, or implementation environment. -Does not concern any specific application area but generic, view-independent data constructs.	-Provides generic data concepts applicable to any application area. -Provides *dynamic relationships* among project objects. -Handles *objects views* through the *Role* and *Role Type* objects. -Is open to adding lower-level objects and their attributes.	-Incorporates object-oriented concepts and provides for *object reuse.*	-Builds and extends upon current technologies (e.g., object oriented methodologies and the IFC model). -Results from generalization of the findings of bottom-up investigations (e.g., prototyping and field investigations).
CPMF	-Encompasses a wide range of possible functions performed in managing a facility project. -Covers functions in various levels: managerial (e.g., sale forecasting), technical (e.g., cost estimating) and operational (e.g., request processing).	-Identifies PM functions through interfaces between various dimensions of PM (i.e., basic functions, project elements, and objectives).	-Does not reflect any specific contractual and organizational setup. -Does not dictate any methodology (i.e., how to be used). -Encompasses "generic functions"	-Due to the neutrality characteristic, it was proved that the model could be used in various ways (e.g., list functions in matrixes and hierarchies); i.e., no methodology constraints.	-Encompasses AEC/FM projects concepts categorized into PM dimensions.	-Was used usefully in the research for developing other deliverables (e.g., to identify PM/MM functions and to design the structure of the ITPMS model).

Research Deliverable	Research Criteria					
	Comprehensiveness	**Integration**	**Neutrality**	**Flexibility & Openness**	**Modularity & Reusability**	**Realism & Usefulness**
ITPMS	-Concerns the whole software development process and its related models. -Concerns all phases of an AEC/FM project; i.e., the whole life cycle of the facility (from cradle to grave).	-Integrates the various views and workflows involved in the software development process; i.e., integration of the process and information aspects of both business and computer systems.	-Does not imply any specific system architecture or implementation environment (uses UML, which is a *modeling* language). -Concerns no specific application area. -Concerns no project contexts (e.g., process sequencing and involved actors).	-Provides for efficient *model evolution* by organizing model elements into a hierarchy of packages. -Provides for flexibly adding, modifying, and specializing processes through the use of UML use cases, activity diagrams, and packages.	-Inherits the reusability advantages offered by the UML. E.g., specialization of process definitions through use-case modeling. E.g., object reuse though inheritance. -An object definition can be reused in multiple processes.	-Provides a mechanism for managing models of systems in an integrated manner. -Was usefully applied for developing other views of integrated MM systems. -Provide for configuration and total model management.
PM/MM Functions Classifications	-Include a wide range of possible functions performed in managing a facility project (operational-technical and managerial processes).	-Originate from intersection of various dimensions of PM.	-Do not convey any specific context, project type, phase, contracting strategy, method, or process sequencing.	-Can be extended to include more processes in various levels of details.	-Give an organized (e.g., hierarchical) view of PM/MM processes. -Allow repetition of processes; i.e., no *process- reusability* concern.	-Have theoretical foundations (literature and models reviews and the Process Modeling Practice Workshop). -Provide a high-level view of processes. -Demonstrated the usefulness of the CPMF model. -Were used for lower-level models (ITPMS implementation).

Research Deliverable	Research Criteria					
	Comprehensiveness	**Integration**	**Neutrality**	**Flexibility & Openness**	**Modularity & Reusability**	**Realism & Usefulness**
MM Models	-Incorporate a wide range of processes carried out by different MM functional bodies (operational-technical and managerial processes). -Incorporate a wide range of MM business objects.	-Integrate the process and information views of the business environment of AEC/FM projects. -Address the possible links (or plug-ins) between MM systems and other PM functions.	-Do not imply any design-and-implementation concern; i.e., no specific logical or physical computer application system architecture. -Generically portray the process models and object models of a MM business environment. -Contextual factors of projects (e.g., process sequencing) are not of concern.	-Can be extended to include other views (e.g., software process and class views) of software systems. -Model elements can be edited or added in various ways and orders.	-Model elements can be shared and reused throughout the model. -MM business objects are defined once and can be reused in many processes. -Specialization of processes is made possible through use-case modeling (i.e., relationships).	-Result from generalization of the findings of the bottom-up investigations (e.g., field data collections and interviews). -Can be used as basic building blocks for development of integrated MM systems.

9.2 An Evaluation of the APCM

This section elaborates on the advantages and limitations offered by the APCM (Chapter 5), as well as how the model's suggested data constructs can be mapped into the IFC model.

9.2.1 Advantages and Limitations of the Model

The APCM offers both advantages and limitations, though the former are believed to outweigh the latter. The key *advantages* of the model include:

1) ***Strong Development Basis:*** The credibility and usefulness of the APCM is generally supported by three facts. First, its development was justified with the results of the MMS prototype development (Appendix C). Second, its development was supported by low-level, filed data collections (section 3.3). Finally, it was verified by an extensive critical review of current literature and models. The model was developed through an analytical process, involving the critical review, and based on the requirements identified in the process (Chapter 4). It extends upon existing technologies; it may be generally viewed as an "IFC-inspired" core model that incorporates the advantageous features of current AEC/FM models (e.g., the IFC's) as well object-oriented methodologies.
2) ***Comprehensiveness:*** The model concerns a whole range of project information. It steps beyond product information an concerns the whole life cycle of an AEC/FM project (i.e., development of a facility from cradle to grave).
3) ***Project Information Integration:*** The model integrates *product* and *process information* into a unified AEC/FM project model, providing the basic data constructs required for development of integrated AEC/FM computer application systems.
4) ***Logical Compartmentalization of Information:*** The model logically distinguishes between diverse types of information mainly based on their *nature* and *uses* in business processes. This allows flexible modeling of project information from any particular view and level of abstraction; thus providing for interoperability between PM systems, without information redundancies and inconsistencies. The specific mechanisms providing this feature are explained later, under "view" and "flexibility".
5) ***Neutrality and Wide Range of Applicability:*** Being conceptual, the model does not imply any specific *methodology or implementation environment.* Any architecture, modeling method or technique may be adopted to structure the model to suit a specific purpose. Moreover, the model is independent of any specific *application area* as it was mainly originated through a high-level analysis of AEC/FM project information requirements coupled with generalization of the data collected in field investigations. This neutrality makes the model useful for development of any AEC/FM application area. The model's concepts can also be extended to be used for other industries.

Consequently, information sharing and data exchange within and between AEC/FM projects as well as other industries can be achieved.

6) ***Handling Types of Objects:*** The model allows definition of a *type* of any object (represented by *Type Object*), including *Occurrence Objects* (i.e., *Things*, *Roles*, and *Events*), *Type Objects*, and even *Aspects*; conceptualized in the *Of Type* relationship between *AECFM Object* and *Type Object* (Figure 5-1). This requirement is elaborated in sections 4.2.2 and 4.2.3.3. This feature of the model is comparable with other models. In the Coad's DNC model (Appendix A), for instance, typing is limited to only physical objects (i.e., "*Party/Place/Thing*"). The IFC model, on the other hand, allows typing of all subtypes of objects (i.e., *IfcObject*), but it does not handle defining a type of a *Type Object*.

7) ***Flexibility Through Relationship Objects:*** The *Relationship* object and its subtypes allow dynamically establish relationships among project objects as necessary. In addition to the advantages offered by their counterparts in the IFC model, the APCM's relationships provide many features, whose necessities were elaborated in section 4.2.1. These features are outlined in the following.

   a) ***Aggregation Relationship:*** is similar to *IfcRelDecomposes*, but allows definition of *Aggregation* relationships not only for occurrence objects but also for type objects.

   b) ***Assignment Relationship:*** may be viewed as an extension of *IfcRelAssigns* that additionally facilitates *type-to-occurrence* assignments.

   c) ***Association Relationship:*** may be mapped to *IfcRelAssociates*; though it offers a wider range of functionality, due to the coverage of various FFB/P properties by the *Aspect* object.

   d) ***Description Relationship:*** may be viewed as an extension of *IfcRelDefinesByType* that provides for *type-to-type* definition as well.

   e) ***Definition Relationship:*** may be mapped to *IfcRelDefinesByProperties*; though it offers a wider range of functionality, due to the coverage of various FFB/P properties by the *Aspect* object.

   f) ***Connection Relationship:*** may be viewed as an extension of *IfcRelConnects* that facilitates defining a *type-to-type* connection. Moreover, the *Function* object may be used to explicitly define the semantics of various connection relationships (e.g., the semantics of *Joining*, *Supporting*, *Embedding*, etc.). This relationship object is an important means for defining any other form of relationship that seem useful but is not captured by other types of relationship objects in the model. It is, in fact, an *extensibility mechanism* for relationships among objects.

8) ***Aspects of Objects and Aspects of Aspects:*** These features are facilitated through the *Definition* relationship between the *AECFM Object* and *Property Definition* (Figure 5-1). This acts as an *extensibility mechanism* that flexibly allows all types of objects defined as subtypes of AECFM

Object (except *Property* and *Property Measure* with which *Aspects* are measured), have *Aspects* (i.e., FFB/P: *Form*, *Function*, and *Performance*). This means that even an *Aspect* may be defined in terms of another *Aspect*. An example of this is the *cost* of a *risk*. Moreover, the *Aspect* object, with its FFB/P-oriented subtypes acts as a central resource, which may be used for defining other objects. However, limitations may be further defined as to which aspects are relevant to a specific type of object. This object is comparable with the defined in the GARM model's Aspect entity [Gielingh 1988], which is limited to characteristics of physical *products* (e.g., occurrences of building elements).

9) ***Handling Versioning and Changes Through Transactions:*** The *Transaction* object, which is intended to represent a record of a planned or performed business event, acts as a means of capturing *historical information* about various objects in a project (e.g., required, expected, and measured characteristics of a thing); therefore, it is a useful mechanism for managing *versioning and changes* (sections 4.1.5.5 and 4.2.4). Through this object and, possibly, the *Role* object (explained next), changes incurred on objects are recorded and can be accessed for various analysis and decision-making.

10) ***Handling Views on Project Information:*** This feature is made possible mainly through the *Role* and *Role Type* objects. The same *thing* can be referenced by different applications (representing actor's views) and regarded as different *roles*, while the integrity of the thing itself remains intact. For instance, the *Artifact* object can be used to model not only *products* (e.g., building, road, etc.), but also the *resources* (e.g., a trailer) and *external agents* (e.g., an adjacent building or the like) in a project. This feature, which offers *flexibility* (e.g., adopting things to fit specific needs) together with *consistency* and *integrity* of information, is not fully available in current models (section 4.2.3). The APCM applies the concept of *role* more explicitly and consistently throughout the model to both *physical* and *non-physical things* or *event* objects at two levels of specificity (section 4.2.2), conceptually represented with the *Role* and *Role Type* objects.

11) ***Usages of the External Agent Object:*** As a subtype of *Role* (section 5.3.3), the *External Agent* object offers some useful functionality for modeling the environment of an object (section 4.1.3). It allows to view an object (e.g., a concrete column) as an *external agent* in one domain (e.g., HVAC design, or piping activity) while being considered as internal to some other domains (e.g., structural design, interior design, cost control, concreting activity, etc.). Specific usage examples include the study of the constraints imposed by a building (adjacent to the site) on site operations, and corrosion of an object (e.g., a structure or a piece of equipment) under weather conditions. The information about impacts of the interacting objects can be captured using the *Impact* object (subtype of *Process*). The

scope and functionality of the *External Agent* object are broader than those of GARM's Agent entity, which is restricted to the effects of *natural agents* (e.g., wind, earthquake, insects, etc.) on *products*.

On the other hand, the APCM has some limitations. As mentioned earlier, since the model is conceptual, it does not reflect any implementation environment; therefore, it may not be directly useful for implementation. Moreover, in order to become useful, the model needs to be detailed into some lower-level application models. Such models would refine the model constructs into lower-level constructs (e.g., by referencing and specialization) to specify the information requirements of specific AEC/FM processes (e.g., MM, equipment management, and quality management). Many of the detailed attributes of the objects of the APCM, however, can be defined using the modeling guidelines suggested in different parts of in this dissertation. For example, the guidelines suggested in sections 4.2.3.4 and 4.2.4.4 can be used for detailed modeling of the *Role* and *Transaction* objects and their related objects respectively. The next section maps the APCM objects to the IFC model in order to demonstrate the added values offered by the APCM.

9.2.2 Linkage between the APCM and the IFC Model

The APCM, which may be viewed as an IFC-inspired deliverable of the research, can enhance the functionality provided by the IFC model in many ways. Table 9-2 presents a general mapping between the APCM objects and the IFC's. In this table, the (only) italized objects represents the objects (or schemas) that already exist in the IFC object model. Where underlined, these objects mean to somehow semantically differ from the ones originally defined in the IFC model (as explained in the last column). The double-underlined objects are those suggested to be added to the IFC's.

Table 9–2: A Mapping Between the APCM Objects and the IFC's

(Continued on the next two pages)

APCM Object	IFC Object/Schema	Notes
AECFM Object	*IfcRoot*	The former extends beyond building-projects objects.
Occurrence Object	*IfcObject*	The former implies more semantics than the current IfcObject; please see its subtypes.
Thing	*IfcThing*	This is a new object.
Physical Thing	*IfcPhysicalThing*	This is a new object, whose some of its functionality is supported by the IfcProduct object.
Tangible Thing	*IfcTangibleThing*	This is a new object.
Actor	*IfcActor*	
Artifact	*IfcElement*	Its semantic extends beyond the current IfcElement object, to include not only "building elements" but also the physical elements of other types of projects than just buildings.
Natural Thing	*IfcNaturalThing*	This is a new object.
Byproduct	*IfcByproduct*	This is a new object.
Space	*IfcSpatialStructureElement*	
Non-Physical Thing	*IfcNonPhysicalThing*	This is a new object.
Thing Group	*IfcGroup*	
Role	*IfcRole*	This is a new object.
Actor Role	*IfcActorRole*	
Product Role	*IfcProductRole*	This is a new object, whose some of its functionality is supported by the IfcProduct object.
Place Role	*IfcPlaceRole*	This is a new object.
Resource Role	*IfcResourceRole*	This is a new object.
External Agent	*IfcExternalAgent*	This is a new object.
Event	*IfcEvent*	This is a new object.
Process	*IfcProcess*	
Task	*IfcTask*	
Impact	*IfcImpact*	This is a new object.
Transaction	*IfcTransaction*	This is a new object that can replace the IfcControl object, though at a different location in the existing IFC's object hierarchy.
Request	*IfcRequest*	This is a new object.
Proposal Or Suggestion	*IfcProposalOrSuggestion*	This is a new object.
Order Or Agreement	*IfcOrderOrAgreement*	This is a new object.
Plan	*IfcPlan*	This is a new object. The IfcWorkPlan may be considered as one of its subtypes.
Evaluation Or Judgment	*IfcEvaluationOrJudgement*	This is a new object.
Certificate	*IfcCertificate*	This is a new object.

APCM Object	IFC Object/Schema	Notes
Allocation	*IfcAllocation*	This is a new object.
Movement Or Transfer	*IfcMovementOrTransfer*	This is a new object.
Gathering	*IfcGathering*	This is a new object.
Notification	*IfcNotification*	This is a new object.
Report	*IfcReport*	This is a new object.
List Or Label	*IfcListOrLabel*	This is a new object.
Log	*IfcLog*	This is a new object.
Phenomenon	*IfcPhenomenon*	This is a new object.
Event Item	*IfcEventItem*	This is a new object.
Type Object	*IfcTypeObject*	The former represents beyond types of physical things.
Thing Type	*IfcThingType*	Some of its functionality is covered by the IfcTypeProduct object.
Role Type	*IfcRoleType*	This is a new object.
Event Type	*IfcEventType*	This is a new object.
Generic Type Object	*IfcGenericTypeObject*	This is a new object.
Property Definition	*IfcPropertSetDefinition*	
Aspect	*IfcAspect*	This is a new object whose functionality is partially covered by the IfcProperty object.
Form	*IfcShape*	This can be mainly mapped into the IfcShape and geometric resource objects of the IFC's.
Function	*IfcFunction*	This is a new object.
Performance	*IfcPropertSetDefinition*	The former focuses specifically on objects' behavioral aspects, while the latter is more generic.
Aspect Group	*IfcPropertSetDefinition*	
Property	*IfcProperty*	
Property Measure	*IfcMeasureResource*	
Relationship	*IfcRelationship*	This represents a relationship between not only occurrence objects but also type objects.
Aggregation Relationship	*IfcRelDecomposes*	The former additionally suggests aggregation of type objects.
Assignment Relationship	*IfcRelAssigns*	The former additionally facilitates type-to-occurrence assignments.
Association Relationship	*IfcRelAssociates*	The former offers a wider range of functionality, due to the coverage of various FFB/P aspects.
Description Relationship	*IfcRelDefinesType*	The former additionally provides for type-to-type relations.
Definition Relationship	*IfcRelDefinesProperties*	The former offers a wider range of functionality, due to the coverage of various FFB/P aspects.
Connection Relationship	*IfcRelConnects*	The former provides for defining various connection relationships (e.g., inclusion, covering, supporting, filling, etc.) between type objects, in addition to physical objects.

9.3 An Evaluation of The CPMF Model

The CPMF model (Chapter 6) was evaluated through its use, as a framework, for development of lower-level (detailed) models. More specifically, it was used for:

- identification of high-level PM function (section 6.3);
- designing the ITPMS model's structure (section 7.2);
- exploring the MM processes (section 8.1);
- implementation of the ITPMS model; particularly MM business processes (section 8.2).

Of the major characteristics of the CPMF model is that it does not reflect any specific contractual and organizational setup. Therefore, it was applied as an effective analytical tool for exploring and identifying a wide range of *generic* PM functions. Due to scope constraints, however, other potential uses of the model was not explored in the research.

9.4 An Evaluation of the ITPMS Model

The proposed ITPMS model (Chapter 7) was evaluated mainly through its application for modeling the functional and information requirements of MM systems (Chapter 8). The model holds both advantages and implications. The ITPMS model embraces the benefits of integration of process and information models (section 2.2.6). Some of the major advantages, which respond to the modeling criteria set for the research (section 3.5), are as follows.

1) ***Managing Complexity in Modeling of Processes:*** Following the modularity criterion, and benefiting from the application of business use case modeling and UML packages, the model provides a modular structure within which the ever-complex processes involved in managing of construction projects can easily be defined, organized, located, and modified. The business process view (figures 7-2 and 7-3), which includes business process packages, is the primary mechanism for organization and formalization of the processes.

2) ***Integration of the Workflows of the Software Development Process:*** The model is capable of capturing the process and information requirements of both business and software systems without losing their identities. This integration was made possible through the incorporation of object-oriented concepts (i.e., abstraction, encapsulation, inheritance, and reusing) and the UML notations into the model. *Bilateral associations* may be established between *processes* and their relevant *objects* in both environments. Such association information, which may be maintained in the database of the CASE tool containing the model, can greatly help maintain the integrity and consistency of model elements.

Emphasizing software development as a *process*, the ITPMS model encompasses an *architecture* that facilitates the development of *integrated total PM systems* for the AEC/FM industry.

3) ***Responding to the Neutrality Criteria:*** The model does not suggest any specific system architecture or implementation environment. It is structured using the UML, which is a modeling language rather than a programming language or software tool. It does not represent any specific application area or project contexts (e.g., process sequencing and involved actors).

4) ***Facilitating the Ability to Add Details Progressively as Needed:*** Benefiting modularity, the instances of the model elements of the ITPMS may be added or their specifications completed progressively, as needed. For example, functionality can be dealt with at different levels of abstractions and details. Processes can initially be modeled as some black-box elements without their details; e.g., a use cases defined with their names, possibly, with relationships. Later, each use case could be dealt with for further detailed process modeling and object identification; e.g., using *activity diagrams*, *object-activity diagrams* (showing the usage of objects through activity flows), and interaction diagrams, for business and software process modeling. More details may be added to the specification of the model elements as more information becomes available about them, or likewise, as other existing constraints (e.g., technical and financial resource constraints) are eliminated.

5) ***Adderessing the Reusability Criteria:*** The model provides for *reuse of standard process and object definitions*. Processes and information are reused in many ways. For example, an approval (or authorization) activity that is used in a variety of processes, in general, might not dramatically be different for each process. Thus, it can be created once and reused using the generalization, extend, and include relationships. If required, an extended version of the generic approval process (i.e., use case) can also be created using extend relationship, for instance. Likewise, business objects are usually reused in business activities, with a possibility of change in their status. For instance, a Quotation object flows from one activity to the next; i.e., it is *created*, *sent*, *received*, *reviewed*, and *responded*. The "Business Object View" provides a pool of business objects that are groued into meaningful units (i.e., packages) and that can commonly be used by business processes (in the "Business Process View") as required. The same functionality is handled by the "Software Class View" and "Software Process View". This functionality eliminates the possibility of repetition and inconsistency of process and information elements in the model

6) ***Provision of the Means of Strategic Analyses:*** By maintaining information about model elements (e.g., business processes, business process groups, business objects, and business object groups) and their relationships, business and information strategies may be analyzed and planned more efficiently and effectively. For example, information on process-to-process and process-to-object relationships

provides an ability to usefully identify areas suitable for integration and automation at the strategic level. Such information can be used for process/object analyses (e.g., using such techniques as "affinity matrices": "CRUD activity/entity matrix" and "clustered activity/entity matrix" [ICON 1994, Final Report, pp. 22-26]). The hierarchical structuring of processes can further facilitate to scope various levels of functions.

7) ***Easing Configuration and Model Management:*** The aforementioned advantages, especially the *integration*, provide for better model use and management. The modeling elements are grouped into packages based on their nature and type. This compartmentalization of the elements helps easily locate an instance of a model element (e.g., an object or an activity describing a use case) among others in the model. Packages and/or their included model elements may be assigned to modelers and other actors and managed accordingly. A tree-view structured browser (like that of Streams), for instance, can transparently present the model structure to the user (e.g., modelers). The exploitation of these and previously described benefits, would require a suitable modeling tool (explained next).

The ITPMS model, on the other hand, imposes a number of implications, which are mainly oriented towards implementation issues. The following lists some of the issues associated with the model.

1) ***Model Management Issues:*** The model may not be exempted and safeguarded from the complexity associated with the processes and information in construction projects. As more process areas (i.e., topic-and sub-topic areas, Figure 7-3) are added to the model, more sophistication is introduced to the model. More sophistication requires a tighter management. A full-scale implementation of the model, which covers modeling of processes and information about all process areas, would require a formal application of PM concepts, methodologies and techniques to the software development process (i.e., the process of modeling) itself. This would encompass both managerial (e.g., staffing, procedures, and financing) and technical (e.g., ownership and versioning) issues related to the model and the organization; which are discussed elsewhere [Sommerville 1997; Jacobson et al. 1999].

2) ***Modeling Tool Issues:*** One of the very important issues in the process of software development is the modeling tools used for different purposes in the stages of the software development life cycle. In order to attain a high efficiency in the process and a reliable model of the system (e.g., in terms of consistency of information about all model elements), an UML-oriented CASE tool would be required for implementation of the ITPMS model. A full-scale implementation of the model requires either a very reliable, sophisticated CASE tool or a set of integrated CASE tools, which are interoperable and together provide for both forward and reverse engineering in a harmonized manner (explained in section 2.3.4). Supporting the basic UML notations used for representation of the model elements of the ITPMS model and handling the required associations between the model elements

(e.g., possibility of associating one or more activity diagrams to a use case) would be the essential requirements of such a CASE tool. A powerful *searching facility* capable of locating and listing the model elements (e.g., use cases, activities, objects, etc.) in the tool would definitely be an asset to effective modeling and model management. Other capabilities such as *zooming*, *filtering*, *comparing*, keeping and presenting *historical data* (e.g., access, changes, etc.), and business process/object analyses (e.g., clustered process/object matrix) would also add to the values of the tool.

The advantages of the ITPMS model (especially its emphasis on the "system architecture" and "integration") far outweigh its limitations. The limitations, which are mostly implementation issues (e.g., configuration and change management), may generally be viewed as some constraints that could be ultimately resolved, given the growing development technology of CASE tools. Although the model was intended to serve the purpose of the research, the advantages offered by the model can also be exploited in modeling other domains.

9.5 An Evaluation of the MM Systems Models

This section presents an evaluation of the MM systems modeling process and its resulted models—i.e., the conceptual classification of MM functions (section 8.1) and the MM business process and object models (sections 8.2 to 8.4). The models are mainly evaluated against the modeling criteria set previously (section 2.5.2).

9.5.1 Advantages and Limitations of the Results

9.5.1.1 The Conceptual Classification of MM Processes

The high-level classification of MM processes (section 8.1), similar to that of PM functions (section 6.3.2), generally revealed how the CPMF model and other analytical techniques (Chapter 6) could be used for identification of processes. Moreover, the process classification addresses the modeling criteria in many ways. For instance,

1) regarding *comprehensiveness*, it lays out a method for identification of a wide range of material-related processes that reflect different MM views (section 8.1.2) as well as operational and managerial processes (section 8.1.1); and
2) regarding the *business focus*, its main focus is *business processes*, and it deals with process elements (e.g., product and information) as far as they help identify processes.

Although the classification helped to freely explore and identify a wide spectrum of generic MM processes from various viewpoints, process overlaps and redundancies were observed. Moreover, the identified processes were generally very conceptual (e.g., no detailed definition and workflow).

Consequently, the results of this classification were used as a basis for modeling lower-level processes in the implementation model (evaluated next).

9.5.1.2 ITPMS-Based MM Models

The advantages of the ITPMS model implementation and its resulting MM process and information models were demonstrated through addressing the research's modeling criteria:

1) The defined model elements do not suggest any specific business setup or any logical or physical computer system architecture. The main focus is the *business system*, independent of contextual issues (e.g., process sequencing and involved actors). The hierarchical representation of processes with UML packages and avoiding actors representation greatly helped satisfy the *neutrality* criterion. Further neutrality was gained by *generalization* of field data into business objects.
2) Concerning the *flexibility* criterion, within the boundary of the adopted methodology, the model elements may be defined in various ways and in an iterative manner. A use case, for instance, may be initially defined with its name, and its specification (i.e., textual description, workflow, relationship with other model elements, etc.) may be completed as more information about the use case becomes available. While abiding to a systematic specification procedure, the information may be entered into the model in different orders. This accommodates real-world constraints such as availability of information and modeling priorities.
3) The *reusability* criterion is addressed by provision of the possibility for reusing existing model elements for defining new ones; e.g., using use-case relationships (section 8.3).
4) Regarding *usefulness*, the specified MM business processes and objects are the main building blocks for modeling software processes and classes required for development of integrated total PM systems. The credibility and validity of the MM models, especially object definitions, are generally supported by several facts: Their development was justified with the results of the literature and models reviews (e.g., section 2.6) and the MMS prototype development (Appendix C), and they are based on other bottom-up investigations, especially filed data collections (Chapter 3). The modeling of software processes and classes had been considered in the original ITPMS model structure (Chapter 7) but was not included in the implementation model's scope.

The MM models generally helped better realize the advantages and implications of the ITPMS model (explained previously). They also revealed the potentials and problems associated with existing CASE tools, as well as the functionality generally expected from such tools (explained next).

9.5.2 The CASE Tool Evaluation

The Streams CASE tool [Ensemble Streams 2000] proved useful as a tool for implementing the ITPMS tool and, thus, developing the MM models described in this dissertation. In carrying out this work, the following advantages of Streams were observed.

1) Due to its UML-modeling support, it provided the required tools for modeling of both functional and information requirements of PM functions. The possibility of assigning stereotypes to model elements was found to be particularly useful for distinguishing between model elements of the same type.
2) The organization of model elements into packages can be viewed in the Browser window, and an element can be moved from one place to another using drag-and-drop operations.
3) Files or links can be attached to the model elements (e.g., documents that provide examples of a business transaction).
4) Colors can be assigned to various model elements (which improves visualization and comprehension of the model).
5) A *shared modeling environment* may be established. A large model can be divided into several models, and each sub-model may include parts of other model as shared model elements (i.e., by reference to the source). The UML packaging mechanism and the model-import and Cut/Paste/Duplicate tools are among the elements that facilitate such an environment.

The following are some of Streams' limitations experienced in the research:

1) Inability to automatically sort the model elements in the Browser window.
2) Lack of class diagramming capabilities.
3) Lack of an efficient searching capability; e.g., for finding specific elements in the model.
4) Lack of location stability for elements placed in a diagram (i.e., locations change when the diagram or its containing package is moved to another place).
5) Lack of synchronization between objects and their corresponding software classes.
6) Inability to change the attributes of a group of model elements (e.g., changing the fill color of all model elements of the same stereotype or assigning a stereotype to several objects at the same time).
7) Lack of a constraint against using the same name for different classes or objects in different packages.
8) Inefficiency in using colors for model elements; e.g., fill-color operation is performed at the instance level rather than type level, and is quite time-consuming (i.e., assigning a fill color to instances of an object or a class in various diagrams may not be done all at once).

9) Lack of all required tools for a fully effective team-working environment established for software engineering. For example, it lacks the control of ownership of modeling elements, access control, and PM tools (i.e., time, cost, and resource management of a modeling process).

A full listing of the observed limitations of the Streams tool was communicated to the designers of the tool throughout the model development process. Improvements of the above limitations are critical to the establishment of an efficient and effective modeling process, especially in a team-working environment, which is usually the case in large projects such as AEC/FM modeling projects.

9.5.3 Other Observations and the Next Steps

The implementation of the ITPMS model is a proof-of-concept for the higher-level models proposed in this dissertation. At present, it focuses mainly on the consumption view of materials (section 8.1.2); however, placeholders for inclusion of other views of MM systems (i.e., standards and regulation as well as production and supply) and, more generally, PM systems do exist in the implemented model.

In the implementation model, the UML *package* was used to model major PM functions, while the concepts of *use case* and *activity* were used to model processes at different levels of abstractions. However, the demonstration of the potential of use case modeling was limited to a relatively small portion of the functions. Considering this and the fact that use case modeling (specifically the identification and definition of use cases) is "a very *iterative* process" [Jacobson et al. 1992, p. 162], the research results require further development in order to reach higher levels of maturity in modeling total PM systems. In this regard, some of the places for extensions include the following:

1) Extension of the use case modeling approach to PM functions other than MM functions.
2) Exploration of inter-functional relationships among various PM functions using use case modeling.
3) Extension of the ITPMS model implementation beyond business process modeling (i.e., into software modeling). The business objects defined in the model are one of the major inputs to this process. The elaboration of the *PM Software Systems Models* package (Figure 8-20), which is intended to include software model elements (e.g., classes, attributes, and relationships), was not considered within the scope of this research.

As expected, it was found that modeling the whole of PM functions and their information requirements is not an easy task, and it requires a team effort and organization. The study and design of the organization for such a modeling effort may be considered as a separate topic for research. Moreover, considering the complexity associated with the modeling of PM functions and their information requirements, the use of CASE tools may be considered as a key element to managing the complexity. To this end, the results of this part of the research (i.e., the MM business process and object models) are

considered to be an important outcome of the research—yet, considering the level of complexity of PM functions and their information requirements, this outcome is viewed as a starting point for further explorations and improvement.

9.6 Chapter Conclusions

This chapter described the mechanisms used in this research to assess the validity of the primary research deliverables (frameworks and models), which are collectively intended for development of integrated total PM systems. It also evaluated the deliverables against the research criteria set for this purpose earlier. The advantages and limitations of the deliverables were listed, and experience of the research in application of the Streams CASE tool was explained.

Overall, in addition to the validity gained from bottom-up investigations, the primary deliverables were developed in an iterative manner, and in an E/V relationships; i.e., one was used as a framework for development of another. In other words, the lower-level detailed model acts a proof-of-concept for the higher-level model. The next chapter concludes the dissertation.

CHAPTER 10 Conclusions and Recommendations

This chapter concludes the dissertation. First, it summarizes the major research dimensions presented in the dissertation—the context, objective, and achievements of the research. Next, it lists the contributions and benefits of the research. Finally, it presents the recommendations of the research as well as the future research.

10.1 Summary

This section summarizes the major dimensions of the research aimed at investigating *model-based integrated total PM systems*—a class of computer systems that include a suite of application systems that support a wide range of PM functions (section 1.2.3) and can flexibly and openly contribute to and draw from a shared pool of project information (referred to as a *unified project object model*, section 1.2.2.1) irrespective of the type of environment. It includes a summary of the issues tackled and the solutions proposed by the research.

10.1.1 Problems

CIC, which encompasses interoperability among AEC/FM computer systems, was addressed as a solution to the inherent fragmentation of the AEC/FM industry, and thus for improvement of efficiency and effectiveness of the industry. Over the past two decades, many research, development and standards projects have made an effort towards provision of the basic building blocks of CIC; i.e., formalization and standardization of AEC/FM project information. Included in this effort is the modeling of AEC/FM processes (within which the information flows), though with varying degrees of formality. While some may argue that the information modeling of AEC/FM projects (especially for the product) has reached its maturity level, some areas remain largely undeveloped and leave room for improvement. Acknowledging the valuable achievements of this effort, this research has attempted to address some of such problems, which are summarized in the following:

1) Due to the central role of product information to the performance of AEC/FM processes, product modeling has been a major focus to most AEC/FM information modeling efforts. However, product information and its closely linked materials information are modeled more from a designer's perspective and less from construction manager's view.
2) There is no AEC/FM project model that comprehensively integrates all types of product and process information as required by various PM functions.

3) PM functions have been traditionally limited to a few functions such as estimating, planning, and scheduling. This is also true for existing AEC/FM standards models and software applications. One of the most important application areas that is largely undeveloped is the MM function.

4) Current AEC/FM models have been developed in a restrictive top-down approach; therefore, their evaluation through bottom-up investigations would be useful.

5) The effectiveness and usefulness of AEC/FM systems in supporting PM functions is dependent on the underlying information and processes supported by such systems. The state-of-the-art software engineering methods generally suggest an integration of business process modeling and information modeling workflows—i.e., explicit elicitation of the information from models of business processes, which are intended to be supported by the software systems. However, this has generally not been the practice in the AEC/FM information standardization efforts.

10.1.2 The Overall Objective and Premise

Considering the research problems, the overall objective of this research was *to investigate how AEC/FM product and process information could be integrated into a unified AEC/FM project model that could support a wide range of construction project management (PM) functions, with an emphasis on materials management (MM), within the context of a CIC environment.* Considering the expected outcome of this work, the main research hypothesis was that *the distinct views of the functions involved in an AEC/FM project on the project information can be brought together into a unified project model, which can be used as a central repository for a wide range of PM application systems reflecting the CIC environment; this may be accomplished through the modeling of PM business processes and their information requirements, within a unified software development process.*

10.1.3 Actions, Results, and Values

The basic research challenge posed by the objective (*how to model project information in order to be useful for PM functions*) was decomposed into several research questions, which lead to the research deliverables, and the action plan was prepared accordingly. The primary results of the plan may be grouped into three levels:

1) *Methodological Level*: developing an integrated structure for modeling AEC/FM business and software systems' functional and information requirements, as well as managing the model elements in an integrated manner—i.e., the Integrated Total Project Management Systems (ITPMS) model (Chapter 7).

2) *Conceptual Modeling Level*: focusing on major AEC/FM projects' concepts and formalizing them in the form of some conceptual models useful for systems development. This includes:

a) devising a framework for classification of PM functions—i.e., the Conceptual Project Management Function Framework (CPMF) model (Chapter 6);

b) classifying PM functions (section 6.3.2) and MM Processes (section 8.1), mainly using the CPMF model; and

c) formalizing AEC/FM product and process information into a conceptual, unified project model—i.e., the AEC/FM Project Core Model (APCM) (Chapter 5)—through generalization of the requirements identified in the critical reviews of current models, low-level data collections, and the results of software prototyping.

3) *Implementation (or Detailed Modeling) Level*: implementing the ITPMS model in a CASE tool, as a proof-of-concept of the ITPMS and CPMF models. This resulted in two major results:

a) representing MM processes and their workflows (sections 8.2 and 8.3); and

b) identifying and defining MM business objects (section 8.4).

The research mainly focused on AEC/FM business process and information modeling (i.e., requirements analysis) with consideration of the totality of the software development process. Benefiting from existing technologies related to the knowledge areas of the research and exploiting the advantages of current models, the research has proposed a methodology and a set of frameworks (i.e., principles or ideas) and models (i.e., representations) to solve the identified problems (mentioned previously). While the general ideas of *total project management* and *unified project models* may not be new, the suggestion and application of some of the new technologies (e.g., PM/MM concepts, UML constructs, and CASE tools) and extension of current models' concepts (e.g., those of the IFC object model) for modeling integrated total AEC/FM PM systems are new.

Given the objectives and methodology of the research, the proposed solutions (listed above) were not intended to define or introduce new technologies, but rather to provide a better understanding of problems and opportunities as well as to specify solutions that could feasibly accommodate new, as well as established, technologies. The evaluation/validation mechanisms used in this research, especially the coupling of top-down and bottom-up approaches in the thinking process and development process of the research, is considered to be an important aspect of this research (section 9.1). The bottom-up accomplishments (review of current literature, models, and software applications; filed observations, interviews, documents collection and analysis; software prototyping; and Process Modeling Practice Workshop) reasonably added to the reliability and usefulness of the research's primary results.

It is envisioned that the results of this research will contribute to the CIC movement; however, the results are viewed as a starting point for further explorations and improvement, mainly because of the

research's *nature* (modeling as a front-end to a longer development process), *context* (the complex and ever-changing AEC/FM industry and software development, section 1.2.1), and *scope* (small portion of the overall context). The research confirmed that the detailed modeling of the whole of PM functions and their information requirements in an integrated manner is a major undertaking, and it requires a team effort, organization, and contemporary technologies.

10.2 Contributions and Benefits

The deliverables of this research (Figure 9-1) constitute the research contributions, which relate to the central theme of integrated total PM systems. The major contributions of the research and their key benefits are as summarized in the following:

1) *Extensive, Critical Review of the Knowledge Areas of the Research—specifically process and information modeling technologies and PM/MM concepts* (chapters 2 and 4): This review provided a better understanding of the context, issues and opportunities in the knowledge areas of the research. The critical review of existing AEC/FM process and information models specifically helped identify the major requirements of integrated AEC/FM systems.
2) *Formalization of AEC/FM Project Information into the APCM* (Chapter 5): This model, which was developed based on a partial result of the above item, extends the state-of-the-art models (e.g., the IFC model) and integrates product and process information into a unified project model. As one specific benefit of the model, the classification suggested for AEC/FM transactions (section 5.3.6) may be used as input to CIC projects such as the IAI's IFC model and the aecXML project, which require standard AEC business transactions definitions (section 4.2.4).
3) *Development of the CPMF model* (section 6.2): This framework model is useful for elicitation of a wide range of PM functions and processes (i.e., the functional requirements of AEC/FM systems). In this research, it was used for the classification of PM functions and MM processes and the development of the ITPMS model (explained in the following).
4) *Classification of PM Functions* (section 6.3.2) *and MM Processes* (section 8.1): These classifications, which were made possible through the use of the CPMF model, are useful for further detailing the processes and for elicitation and modeling their information requirements.
5) *Development of the Integrated Total Project Management Systems (ITPMS) Model* (Chapter 7): This model, which was developed through the application of the CPMF model and the state-of-the-art modeling language (i.e., the UML concepts), suggests an open, flexible architecture for integrating models of functional and information requirements of AEC/FM business and software systems within a unified model. It also embraces the benefits of integration of process and information models

(section 2.2.6) as well as facilitation for managing the model elements throughout the systems development process (thus benefiting the improved efficiency, effectiveness, consistency, etc.).

6) *Formalization of the Functional and Information Requirements of Integrated MM systems into a Set of Business Process and Object Models* (8): The models, which were developed based of the ITPMS structure and in a CASE tool, as well as the results of the field investigations, revealed a number of benefits. It primarily acts as a *proof-of-concept* for usefulness of the ITPMS model as well as the potential benefits of CASE tools in developing and managing the complex models of integrated AEC/FM systems. Moreover, the functional requirements (i.e., business process models) can directly be mapped into systems processes. Likewise, representing MM business objects, the object models can be used as the basic building blocks of integrated MM computer systems. Specifically, the defined objects may usefully be incorporated into the AEC/FM standards projects such as the IAI's IFC model and aecXML project.

Among the aforementioned contributions, items 2 to 6 have the highest importance. Other secondary products of the research (which are not listed here but never-the-less represent significant research effort, e.g., the field data and documents and the PMPW) gain their importance through their roles played in the evaluation/validation of the primary deliverables (section 9.1). Moreover, the modeling guidelines suggested in various parts of in this dissertation also contributed to the further detailing and enriching of the models. For instance, the guidelines suggested in sections 4.2.3.4 and 4.2.4.4 can be used for further detailing of the APCM.

Taken as a whole, it is envisioned that the deliverables of this research will contribute to the CIC movement. Building on the experiences gained in the research, the following section presents recommendations of the research for a full realization of the premise of the research.

10.3 Recommendations

The following recommendations are suggested by the research results:

1) *Model-based integrated Total PM Systems (MITPMS)*: MITPMSs are recommended for improvement of efficiency and effectiveness of AEC/FM projects processes.

2) *Information Integration*: AEC/FM information modeling should be directed beyond product information; i.e., integration of AEC/FM product and process information models into a *unified project model*, similar to the APCM proposed by this research.

3) *Process Integration*: The realization of MITPMSs requires a more rigorous application of the concept of *software development process integration*, which has been emerging in the software engineering domain. An emphasis should be placed on integration of information modeling and process modeling

activities into a unified software development process. The ITPMS model represents one possibility for such integration.

4) *Business and Software Modeling Integration*: In the *process integration*, a special emphasis should be placed on *business process modeling*. Models of AEC/FM software systems modeling should be based on the models of the business systems supported by such computer systems (i.e., software system modeling through business system modeling).

5) *Modeling Technique*: The UML is recommended as a modeling technique for modeling MITPMSs. As a *unified* generic modeling language, the UML can provide for both software development *process integration* and ease of *model management* through the use of its powerful modeling constructs (e.g., use cases, packages, and objects; Appendix A). Dependencies within and between functional and information components can be minimized, and modularity, reusability, and thus efficient development and maintenance of complex models may be realized. The research has found the UML packages and use cases as a good candidate for functional modeling of business and software systems. It recommends use-case modeling as an effective technique for classifying, organizing, and managing PM functions and coping with the complexity inherent in the functions as well as handling *variations of process setups and views of modelers*, which are commonplace in less standardized environments such as AEC/FM processes.

6) *Modeling Tools*: A full-scale implementation of the models such as the ITPMS model requires either a very reliable, sophisticated CASE tool or a set of integrated CASE tools that are interoperable and provide for both forward and reverse engineering in a harmonized manner. Such tools can potentially facilitate the development of better models and effective software development *process management* (i.e., from business to software modeling and deployment). Consideration of such a tool is a key element to managing the complexity associated with modeling MITPMSs.

7) *Modeling Criteria*: Modeling criteria, such as those defined for development of the models of this research (e.g., comprehensiveness, flexibility, and reusability), are recommended as a key for the successful development of MITPMSs.

Speculating upon the future of CIC, the next section presents some further work recommended by the research to better exploit the benefits of the research results.

10.4 Further Research

The realization of CIC depends largely on the willingness and real commitment of software vendors to adopt CIC-based standards into their application software tools for use in the industry as well as the willingness of the industry to adopt the software tools. The earlier requirement may be met by

initiatives such as the IAI consortium [IAI 2005], which encompasses an *industry alliance*. The latter requirement, however, may largely depend on how the tools satisfy the users' requirements, especially the functional requirements.

The development life cycle of CIC-based standards (i.e., the formalization and standardization of the data modeling and exchange technology required for CIC) encompasses three major overlapping processes: 1) development of the technology, 2) adoption of the technology by software industry, and 3) adoption of the software by AEC/FM industry. Figure 10-1 schematically shows the speculation of this research on these processes, under the influence of initiatives such as the IAI over a twenty-year horizon. As suggested in this figure, the level of technology development tends to rise. This rise will largely be due to the coverage of more business domains (application areas), e.g., in the IFC object model. Also, the gap between the three processes will continue to decrease, chiefly due to the trend of awareness on the importance of information technology, realization of the "industry alliance" for standardization of the technology, and thus, the software-and-AEC/FM industries' support of the standardization initiatives.

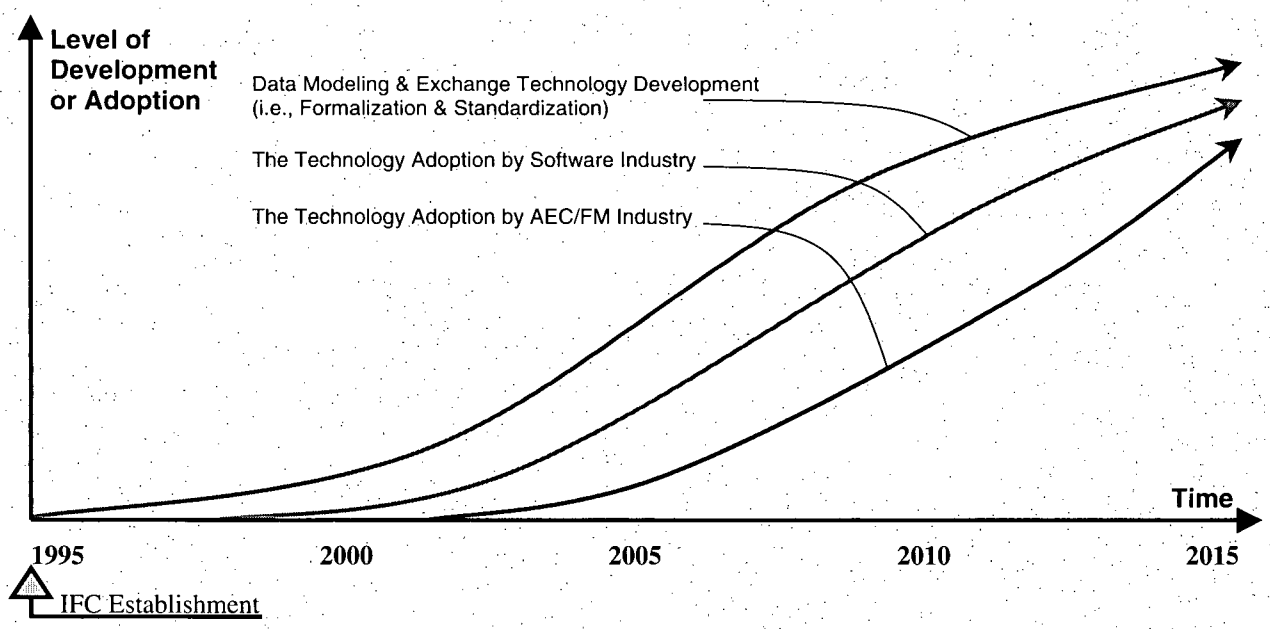


Figure 10–1: An Schematical Speculation of the Horizon of Data Modeling and Exchange Technology Development & Adoption

The results of this research, which suggests a model-based approach to integrated PM systems development, are envisioned to contribute to CIC developments. More specifically, they may well be incorporated to the IAI project components (i.e., the IFC model and the aecXML project). However, for many reasons—including the *nature* (i.e., modeling as an iterative and never-ending), *context* (AEC/FM projects with their complexity and the software development process with its diverse dimensions and

approaches), and *scope* (a very small portion of the overall context) of the research—further work is foreseen for them in order for the research premise (section 10.1.2) to be fully realized. Some of the areas for further work include:

1) a full-scale implementation of the ITPMS model covering all PM process areas (especially for MM) and their functional and information requirements, in both business and software models views (e.g., business process and object models and software process and class models);
2) a study of the organization required for the full-scale implementation of the ITPMS model;
3) an extension of the APCM constructs (e.g., through referencing and specialization), which were intended to serve as a conceptual model, to develop application models of various domains (e.g., MM, equipment management, and quality management);
4) the use of the suggested MM process and information models to develope MM application models within the ITPMS model in relation to the APCM (i.e., an explicit link between the two models);
5) an extension of the scope of the prototype MM application system (Appendix C) and inclusion of its model in the ITPMS model to examine the usefulness of both the MM business object models defined in the research and the ITPMS model; and
6) holding other process modeling workshops, similar to the Process Modeling Practice Workshop held in this research, to examine and compare the capabilities of the UML and other process and information modeling techniques.

To this end, this dissertation reported on various dimensions of the research aimed at investigating *model-based integrated total PM systems*, intended to improve efficiency and effectiveness of AEC/FM processes through interoperability among a wide range of computer application systems supporting the processes. The research demonstrated that the realization of the CIC environment may well be facilitated by model-based integrated total PM systems. Central to the development of such systems is a *unified project model*, which is developed through the acknowledgement and modeling of PM processes and their information requirements. Benefiting from existing modeling technologies and current models, the research took an initiative in development of such a model at three levels: methodological, conceptual, and implementation. The results of this initiative, whose reliability has been proved through evaluation/validation processes, are encouraging and their benefits are significant; yet they are viewed as a starting point for further explorations and improvement towards development of such integrated systems and a fully realization of the CIC environment. The research confirmed that, due to the complexity of AEC/FM projects, the detailed modeling of the whole of PM functions and their information requirements in an integrated manner is not an easy task, and it requires a team effort and organization supported with sophisticated modeling techniques (such as the UML) and modeling tools (including CASE tools).

References

Agapiou, A.; Flanagan, R.; and Norman, G. (1998).
"The Changing Role of Builders Merchants in the Construction Supply Chain". *Construction Management and Economics*, 16, 351-361

Ahuja, H. N. (1984).
Project Management. Techniques in Planning and Controlling Construction Projects. John Wiley & Sons, Inc., New York.

AIA A201 (1987).
AIA Document A201: General Conditions of the Contract for Construction. Fourteenth Edition, The American Institute of Architects (AIA), Washington D. C., USA.

Al-Hussein, M. (1999).
An Integrated System for Crane Selection and Utilization. Ph.D. Thesis, Department of Building, Civil, and Environmental Engineering, Concordia University, Montreal, Canada.

Ammer, D. S. (1974).
Materials management. 3rd Edition, Homewood, Irwin series in management and the behavioral sciences. ISBN: 0256015562.

Ammer, D. S. (1983).
Purchasing and Materials Management for Health-Care Institutions. 3rd Edition, Lexington Books, Lexington, Mass.

Amor, R. and Faraj, I. (2001).
"Misconceptions about Integrated Project Databases". *Electronic Journal of Information Technology in Construction, ITcon* September 2001, Vol.6, pp. 57-68. Available at URL: http://itcon.org/2001/5/.

Andersen, A.ur & Co., Flaatten, Per O., et al. (1992).
Foundations of Business Systems. Second Edition, Dryden Press.

Aouad, G.; Betts, M.; Brandon, P.; Brown, F.; Child, T.; Cooper, G.; Ford, S.; Khirkham, J.; Oxman, R.; Sarshar, M.; and Young, B. (1994).
Integration of Construction Information, Final Report. Published by University of Salford, Department of Surveying & Information Technology Institute, UK, July 1994.

Architects' First Source (2005).
Architects' First Source. Architects' First Source Inc., CMD Group Norcross, GA, USA. Web Site: http://www.afsonl.com

ArchStyler (2002).
ArchStyler. A CASE tool product of Interactive Objects Software, the home page at URL: http://www.iO-Software.com

ATLAS, EP-7280 (1997).
Architecture Methodologies and Tools for Computer integrated Large Scale Engineering (EP-7280). The ATLAS home page at URL: http://www.cstb.fr/ILC/ecprojects/atlas/atlas.html.

Augenbroe, G. (1994).
An Overview of COMBINE Project. First ECPPM Conference, Dresden. A pdf document at URL: http://erg.ucd.ie/combine.html

Augenbroe, G. and Rombouts, W. (1994).
"Extended Topology in Building Design Systems". *Proceedings of the First Congress on Computing in Civil Engineering*, Washington, D.C., USA, June 20-22, 1994 Published by the ASCE, New York, NY, USA, Vol. 2, pp. 1196-1203.

Augenbroe, G. (1995).
"The COMBINE Project: A Global Assessment". *Proceedings of the CIB Workshop on Computers and Information in Construction*, Fischer, M., Law, K. H., and Luiten, B. (eds). Publ 180, Stanford University, Stanford, California, USA, August 1995, pp. 163-171.

Battikha, M. (2000).
A Computer-Based System for Construction Quality Management with Application to Highway Pavements. Ph.D. Thesis, Construction Engineering and Management Group, Department of Civil Engineering, University of British Columbia, Vancouver, BC, Canada.

Bedard, C. P. E.; Gowri, K. and Fazio, P. E. (1991).
"Integration in Building Industry". Editorial, ASCE *Journal of Computing in Civil Engineering*, ASCE, 5(4):333-335.

Bell, L. C. (1989).
"Computer Systems for Construction Materials Management". *Proceeding of the Fourth Conference on Computing in Civil Engineering*, ASCE, 1989, pp. 454-460.

Bell, L. C. and McCullouch, B. G. (1988).
"Bar Code Applications in Construction". *Journal of Construction Engineering and Management*, ASCE, 114(2):263-278.

Bell, L. C. and Gibson, G. E. (1991).
"Computer Data Exchange Using National and International Standards". *Proceeding of the ASCE Construction Congress' 91*, Cambridge, MA, USA, April 14, 1991, pp. 441-445.

Bernold, L. E. (1990).
"Bar Code-Driven Equipment and Materials Tracking for Construction". *Journal of Computing in Civil Engineering*, ASCE, 4(4):381-395.

Beynon-Davies, P. (1993).
Information Systems Development: An introduction to information systems engineering. Second Edition. The MacMillan Press Ltd. London, UK. ISBN: 0-333-59506-8.

Bjork, B. C. (1992a).
"A Unified Approach for Modeling Construction Information". *Building and Environment*, 27(2):173-194.

Bjork, B. C., Huovila, P. and Hult, S. (1993).
"Integrated Construction Project Document Management (ICPDM)". In Beheshti, M., Zreik, K. editors, Advaned technologies - Architecture - Planning - Civil Engineering, *Proceedings of the EuropIAI'93 conference,* June21-24, Elsvier, Delft, Holland, pp:135-146.

BLIS Handout (2000).
"BLIS Project Update". A handout distributed in the BLIS meeting in London, October 26, 2000.

BLIS (2004).
Building Lifecycle Interoperable Software. http://www.blis-project.org/.

Booch, G. (1991).
Object-Oriented Design with Applications. Redwood City: Benjamin/Cummings.

Booch, G; Rumbaugh, J; and Jacobson, I. (1999).
The Unified Modeling Language User Guide. Addison-Wesley object technology series, Addison Wesley Longman, Inc. Massachusetts, USA, ISBN: 0201571684.

Brown, A.; Cooper, G; Rezgui, Y.; Brandon, P.; and Kirkham, J. (1996).
"The Architecture and Implementation of a Distributed Computer Integrated Construction Environment". *Proceedings of the CIB Workshop on Construction on the Information Highway*, Bled, Slovenia, June 1996.

Buildcore (2005).
BuildSource: Product Finder. Buildcore Inc., CMD Group. North York, Ontario, Canada. Web Site: http://www.buildcore.com

Capron, H. L. and Williams, B. K. (1984).
Computers and Data Processing. Second Edition, Addison-Wesley Publishers Limited, ISBN: 0201122308.

Cash, J.; McFarlan, W.; McKenney, j; and Applegate, L. (1994).
Corporate Information Systems Management: Text and Cases. Third Edition. IRWIN Inc., Illinois, USA.

Chambers, Andrew, and Rand, Graham (1997).
The Operational Auditing Handbook: Auditing Business Processes. By Management Audit Ltd., Published by John Wiley & Sons Ltd, West Sussex, England, ISBN: 0471970603.

Chandler, I. E. (1978).
Materials Management on Building Sites. Published by Longman Inc, New York, NY, USA, ISBN: 0904406237.

Cheng E. W. L.; Li, H.; Love, P. E. D.; and Irani, Z. (2001).
"An E-Business Model to Support Supply Chain Activities in Construction". *Logistics Information Management* 14: 68-77.

Cheng, M. Y. and O'Connor, J. T. (1996).
"ArcSite: Enhanced GIS for Construction Site Layout". *Journal of Construction Engineering and Management*, ASCE, 122(4): 330-334.

Cheng, M. Y., and Chen, J. C. (2002).
"Integrating Barcode and GIS for Monitoring Construction Process". *Automation in Construction*, Elsevier, 11(1): 23-33.

CII (1986).
Costs and Benefits of Materials Management Systems. Nov. 1986, 30 pages, (Publication RS7-1), Materials Management Task Force of the Construction Industry Institute.

CICE (1982).
Modern Management Systems. A Construction Industry Cost Effectiveness Project (CICE) Report, The Business Roundtable, Report A-6, reprinted on Jan. 1989, New York, NY.

CICE (1983a).
Report on Materials Management. Appendix A-6.5, A Construction Industry Cost Effectiveness Project (CICE) Report, The Business Roundtable, New York, NY.

CICE (1983b).
More Construction for the Money. Summary Report of the Construction Industry Cost Effectiveness Project (CICE), The Business Roundtable, reprinted on Oct.1988, New York, NY.

CIVI-523, UBC (2002).
Project Management for Engineers. A graduate course offered in the Department of Civil Engineering, University of British Columbia, Vancouver, Canada. http://www.civil.ubc.ca/courses/civl523/

Clough, R. H. and Sears, G. A. (1991).
Construction Project Management. Third Edition, John Wiley & Sons Inc., USA.

Coad, P. and Yourdon, E. (1991a).
Object-Oriented Analysis. Second Edition, Prentice-Hall, Inc., Englewood Cliffs, NJ, USA.

Coad, P. and Yourdon, E. (1991b).
"*Object-Oriented Design*". Prentice-Hall, Inc., Englewood Cliffs, NJ, USA.

Coad, P.; North, D.; and Mayfield, M. (1997).
Object Models: strategies, patterns and applications. Second Edition, by Object International, Inc. and published by Prentice Hall, Englewood Cliffs.

Coad, P.; Lefebvre, E.; and De Luca, J. (1999).
JAVA Modeling in Color with UML: Enterprise Components and Process. Prentice Hall PTR, Prentice Hall, Inc. Upper Saddle River, NJ, USA, ISBN: 013011510X.

CoadLetter #68 (2000).
Object Modeling: A New Beginning. CoadLetter, Issue 68, May 2000, URL: http://www.togethercommunity.com/coad-letter/Coad-Letter-0068.html.

CoadLetter #76 (2001).
Object Modeling: Modeling User Roles. CoadLetter, Issue 76, January 2001, URL: http://www.togethercommunity.com/coad-letter/Coad-Letter-0076.html.

CoadLetter #77 (2001).
Object Modeling: Comparing Object Models to the DNC. CoadLetter, Issue 77, February 2001, URL: http://www.togethercommunity.com/coad-letter/Coad-Letter-0077.html.

Coble, R. J, and Kilbert, C. J. (1994).
"Integrating Pen Computers with Other Keyless Data Systems for Construction Applications". *Proceedings of the First Congress on Computing in Civil Engineering*, Washington, D.C., USA, June 20-22, 1994, Published by the ASCE, New York, NY, USA, Vol. 2 pp. 1639-1646.

Cockburn, A. (2001).
Writing Effective Use Cases. Addison-Wesley, ISBN: 0201702258.

COMBINE (1995).
COMBINE Home Page. URL: http://erg.ucd.ie/combine.html

CSOM (1994).
Contractor's Safety and Orientation Manual. Published by the 'Campus Planning and Development' and 'Health, Safety, and Environment' of the University of British Columbia (UBC), July 1994, Vancouver, Canada.

Davenport, T. H. (1993). {Nadaaram}
Process Innovation, Reengineering Work through Information Technology. Harvard Business School Press, Boston, MA, USA.

Davis, B. J. (1976).
Information Sources in Transportation, Material Management, and Physical Distribution: An Annotated Bibliography and Guide. Greenwood Press, Westport, Conn. ISBN: 0837183790.

Dubois, A.M.; Flynn, J.; Verhoef M.H.G.; and Augenbroe, G.L.M. (1994).
Conceptual Modelling Approaches in the COMBINE project. First ECPPM Conference, Dresden. A pdf document at URL: http://erg.ucd.ie/combine.html

Eastman, C. M. (1999).
"Information Exchange Architectures For Building Models". *Proceeding of the Eighth International Conference on Durability of Building Materials and Components. Volume Four, Information Technology in Construction: CIB W78 Workshop*, Vancouver, Canada, May 30 - June 3, 1999. pp. 2139-2156.

Echeverry, D. (1996).
"Adaptation of Barcode Technology for Construction Project Control". *Proceedings of the Third Congress on Computing in Civil Engineering*, ASCE, Anaheim, CA, USA, June 17-19, 1996, Published by the ASCE, New York, NY, USA, pp. 1034-1040.

Echeverry, D. and Beltran, A. (1997).
"Bar-code Control of Construction Field Personnel and Construction Materials". *Proceedings of the Fourth Congress on Computing in Civil Engineering*, ASCE, Philadelphia, PA, USA, June 16-18, 1997, Published by the ASCE, New York, NY, USA, pp. 341-347.

Echeverry, D.; Guerra, C. A.; and Beltran, A. (1998).
"A Regional Attempt to Implement Bar-code Control in Construction Projects". *Proceedings of the Fifth Congress on Computing in Civil Engineering*, ASCE, Boston Massachusetts, USA, Oct. 18-21, 1998, Published by the ASCE, Reston, Virginia, USA, pp. 450-453.

Ensemble (2001).
Ensemble Systems Inc. URL: www.ensemble-systems.com.

Ensemble Streams (2000).
Ensemble Suite: User Guide, Release 3.2. Ensemble Systems Inc.

Ericksson, H. E. and Penker, M. (2000).
Business Modeling with UML: Business Patterns and Business Objects. Published by John Wiley & Sons, Inc., USA, ISBN: 0471295515.

Etter D. M. (1992).
FORTRAN 77 with Numerical Methods for Engineers and Scientists. Benjamin/Cummings Publishing Company, Redwood City, California, USA, ISBN: 0805317708.

Feldmann, C. G. (1998)
The Practical Guide to Business Process Reengineering Using IDEF$_0$. Dorset House Publishing Company, Incorporated, New York, N.Y., USA, ISBN: 0932633374.

Fischer, M. and Froese, T. (1996).
"Examples and Characteristics of Shared Project Models". *Journal of Computing in Civil Engineering*, ASCE, 10(3): 174-182.

Fischer, M.; Luiten, B.; and Aalami, F. (1995).
"Representing Project Information and Construction Method Knowledge for Computer Aided Construction Management". *Proceedings of the CIB Workshop on Computers and Information in Construction.* Fischer, M., Law, K. H., and Luiten, B. (eds). Publ 180, Stanford University, Stanford, California, USA, August 1995, pp. 404-414.

Fleming, A.; Lee, A.; Cooper, R.; and Aouad, G. (2000).
The Development of a Process Mapping Methodology for the Process Protocol Level 2. Third European Conference on Product and Process Modeling in Building and Related Industries, Portugal, 2000. The paper is also available at URL: http://pp2.dct.salford.ac.uk/publications.htm

Fowler, M. and Scott, K. (1997).
UML Distilled: Applying the Standard Object Modeling Language. Addison-Wesley, ISBN: 0201325632.

Froese, T. (1992).
Integrated Computer-Aided Project Management through Standard Object Models. Ph.D. Thesis, Department of Civil Engineering, Stanford University.

Froese, T. (1994a).
"Developments to the IRMA Model of AEC Projects". *Proceedings of the First Congress on Computing in Civil Engineering*, Washington, D.C., USA, June 20-22, 1994 Published by the ASCE, New York, NY, USA, Vol. 1, pp. 778-785.

Froese, T. (1996a).
"Models of Construction Process Information". *Journal of Computing in Civil Engineering*, ASCE, 10(3):183-193.

Froese, T. (1996b).
"STEP Data Standards and The Construction Industry". *Proceedings of the 24th Annual Conference of the Canadian Society for Civil Engineering (CSCE)*, Edmonton, Alberta, Canada, June 29- July 1st, 1996, pp. 404-415.

Froese, T. (1996c).
"STEP and the Building Construction Core Model". *Proceedings of the Third Congress on Computing in Civil Engineering*, ASCE, Anaheim, CA, USA, June 17-19, 1996, Published by the ASCE, New York, NY, USA, pp. 445-451.

Froese, T.; MASCE; and Yu, Kevin. (2000a).
"Architecture Issues for Distributed AEC/FM Systems". *Proceedings of the Eighth International Conference on Computing in Civil and Building Engineering (VIII-ICCCBE)*, ASCE, Stanford, CA, USA, August 14-16, 2000, Published by the ASCE, Reston, Virginia, USA. Vol. 2, pp.1285-1292.

Froese, T. and Rankin, J. and Yu, K. (1997a).
"An Approach to Total Project Systems". *Proceedings of the Fourth Congress on Computing in Civil Engineering*, ASCE, Philadelphia, PA, USA, June 16-18, 1997, Published by the ASCE, New York, NY, USA, pp. 1-8.

Froese, T., Rankin, J., and Yu, K. (1997b).
"Project Management Application Models and Computer -Assisted Construction Planning in Total Project Systems". *International Journal of Construction Information Technology*, 5(1):39-49.

Froese, T. and Rankin, J. (1998).
"Construction Methods in Total Project Systems". *Proceedings of the Fifth Congress on Computing in Civil Engineering*, ASCE, Boston Massachusetts, USA, Oct. 18-21, 1998, Published by the ASCE, Reston, Virginia, USA, pp. 383-394.

Froese, T.; Yu, K.; Liston, K.; and Fischer, M. (2000b).
"System Architectures for AEC Interoperability". Proceedings of the Construction Information Technology (CIT) 2000 Conference (CIB-W78, IABSE, and EG-SEA-AI), Reykjavik, Iceland, June 28-30, 2000.

GDCPP (2001).
Generic Design and Construction Process Protocol. A web site with a number of papers about the GDCPP project at URL: http://pp2.dct.salford.ac.uk/mainpage.htm.

Ghanbari, A. H. and Froese, T. (1999).
"Product Modeling for Construction Management". *Proceedings of the Eighth International Conference on Durability of Building Materials and Components. Volume Four, Information Technology in Construction: CIB W78 Workshop*, Vancouver, Canada, May 30 - June 3, 1999. pp. 2747-2757.

Ghanbari, A. H. and Froese, T. (2004).
"A Process Modeling Practice Workshop for Materials Management". Paper #606 in the *Proceedings of the CIB World Building Congress 2004 (Building for the Future)*, May 1-7, 2004, Toronto, ON, Canada.

Gibson, G. E.; Bell, L. C.; and Molad, C. (1994).
"EDI and It's Role in Information Management". *Proceedings of the First Congress on Computing in Civil Engineering*, Washington, D.C., USA, June 20-22, 1994, Published by the ASCE, New York, NY, USA, Vol. 2, pp. 1798-1805.

Gielingh, W. (1988).
General AEC Reference Model (GARM). ISO/TC184/SC4/WG1, Document 3.2.2.1, TNO report BI-88-150, Oct. 1988.

Halfawy, M. and T. Froese, T. (2003).
"Building Integrated Architecture/Engineering/Construction Systems Using Smart Objects: A Methodology and Implementation" Accepted for *Journal of Computing in Civil Engineering*, ASCE, Submitted July 2003.

Hammer, M. and Champy, J. (1993).
Reengineering the Corporation: A Manifesto for Business Revolution. Harper Collins, NY, USA.

Hannus, M. and Pietilainen, K. (1995).
"Implementation Concerns of Process Modeling Tools". *Proceedings of the CIB Workshop on Computers and Information in Construction.* Fischer, M., Law, K. H., and Luiten, B. (eds). Publication 180, Stanford University, Stanford, California, USA, August 1995, pp. 449-458.

Hannus, M.; Laitinen, K.; Seren, J. (1996).
"Integrating Design and Planning in Building Projects – System Architecture, Implementation and Experiences". *Information Representation and Delivery in Civil and Structural Engineering Design*, Civil-Computing Press, 1996, pp. 117-125.

Hassanain, M. A. (2002).
Integrated Systems for Maintenance Management. Ph.D. Thesis, Construction Engineering and Management Group, Department of Civil Engineering, University of British Columbia, Vancouver, BC, Canada.

Hegazi, T. and Elbeltagi, E. (1999).
"EvoSite: Evolution-Based Model for Site Layout". *Journal of Computing in Civil Engineering*, ASCE, 13(3): 198-206.

Hendrickson, C. and Au, T. (1989),
Project Management for Construction. Prentice-Hall, Inc., Englewood Cliffs, NJ, USA, ISBN: 013731260.

Howard, H. C.; Levitt, R. E.; Paulson, B. C.; Pohl, J. G.; and Tatum, C. B. (1989).
"Computer Integration: Reducing Fragmentation in AEC Industry". ASCE *Journal of Computing in Civil Engineering*, 3(1):18-32.

IAI (2005).
International IAI web site. IAI Home page at URL: http://www.iai-international.org/

IAI-NA (2005).
IAI North America Chapter. At URL: http://iai-na.com/

ICON (1994).
Information/Integration for Construction (ICON). Published by University of Salford, Department of Surveying & Information Technology Institute, UK, July 1994.

$IDEF_0$ (1981).
Integrated Computer-Aided Manufacturing (ICAM), Architecture. Part II, Vol. IV, Functional Modelling Manual ($IDEF_0$). SoftTech, Inc. Waltham, MA, USA, June 1981.

IFC R2x (2000a).
Industry Foundation Classes - IFC 2x. International Alliance for Interoperability, October 27 2000. A document at URL: http://cig.bre.co.uk/iai_uk/IFC2x.htm

IFC R2x (2000b).
Industry Foundation Classes - Release 2x: IFC Technical Guide. Editors: Thomas Liebich, Jeffrey Wix, International Alliance for Interoperability, October 27 2000. A document at URL: http://cig.bre.co.uk/iai_uk/IFC2x.htm

IFC R2x (2001).
Industry Foundation Classes - Release 2x: Model Implementation Guide. Version 1.0, International Alliance for Interoperability, Modeling Support Group, June 29, 2001. A document at URL: http://cig.bre.co.uk/iai_uk/IFC2x.htm

IFC R2x2-1 (2004).
Industry Foundation Classes – Release 2x, Edition 2, Addendum 1. International Alliance for Interoperability, 2004. A document at URL: http://www.iai-international.org/Model/

IFC R2.0 (1999a).
Industry Foundation Classes - Release 2.0. International Alliance for Interoperability, March 15, 1999.

IFC R2.0 (1999b).
Industry Foundation Classes - Release 2.0: An Introduction to the International Alliance for Interoperability and the Industry Foundation Classes. Release 2.0 documentation of IFC published by International Alliance for Interoperability, March 15, 1999.

IFC R2.0 (1999c).
Industry Foundation Classes - Release 2.0: IFC Specification Development Guide. Release 2.0 documentation of IFC published by International Alliance for Interoperability, March 15, 1999.

Illingworth, J. and Thain, K. (1988).
Materials management: is it worth it? CIRIA Special Publication 58, Published by Ascot, Berkshire: Chartered Institute of Building (CIOB), No. 93, 1988. Great Britain, ISSN: 0262-6632.

ISO (1991); Spiby, P.
EXPRESS Language Reference Manual. STEP Document, Part 11: ISO/TC184/SC4/WG5 N14, April 1991.

ISO (1992); Mason, H.
STEP Part 1: Overview and Fundamental Principles. STEP Document (CD ballot), Part 1: ISO/TC184/SC4/* , Sept. 15 1992.

ISO (1995a); Hass, I. W.
STEP Part 225: Building Elements Using Explicit Shape Representation. STEP Document, Part 225 (AP225): ISO/TC184/SC4/WG3 N434 (T12), July 1995.

ISO (1995b); Wix, J. and Tolman, F.
Building Construction Core Model. STEP draft Document, Part 106 (BCCM), version T100: ISO/TC184/SC4/WG3/T12 N411, October 1995.

ISO (1995c); West, M. and Fowler, J.
The Nature of Industrial Data - Some Architectural Aspects and Issues for SC4. STEP draft Document, ISO/TC184/SC4/WG10 N31, October 26, 1995.

ISO (1997); Wix, J. and Tolman, F.
Building Construction Core Model. STEP draft Document, Part 106 (BCCM), version T200, ISO/TC184/SC4/WG3/T12 N411, October 1997.

Jacobson, I.; Christerson, M.; and Overgaard, G. (1992).
Object-Oriented Software Engineering: A Use Case Driven Approach. The ACM Press, A Division of the Association for Computing Machinery, Inc. (ACM), Addison Wesley Longman Limited, England, Revised printing in 1998, ISBN: 0201544350.

Jacobson, Ivar; Ericsson, M.; and Jacobson, A. (1995).
The Object Advantage: Business Process Reengineering with Object Technology. The ACM Press Books, A Division of the Association for Computing Machinery, Inc. (ACM). Addison-Wesley Longman Limited, New York, N. Y., USA, ISBN: 0201422891.

Jacobson, I.; Booch, G; and Rumbaugh, J. (1999).
The Unified Software Development Process. Addison-Wesley object technology series, Addison Wesley Longman, Inc. Massachusetts, USA, ISBN: 020157162.

Jacobson, Ivar; and Bylund, Stefan (2000).
The Road to the Unified Software Development Process. Published in February 2000 by Cambridge University Press, Cambridge, UK. ISBN: 0521787742.

Jergeas, G. (1989).
Detailed Design and Constructability. A Ph.D. thesis. Civil Engineering Department, Loughborough University, England.

JDS (2002).
Jigsaw Distributed System. http://construction.civil.ubc.ca/jigsaw_technical/

Kagioglou M.; Cooper, G.; Aouad, G.; Hinks, J.; Sexton, M.; and Sheath , D. (1998a).
A Generic Guide to the Design and Construction Process Protocol. University of Salford, September 1998, ISBN: 0-902896-17-2.

Kagioglou M.; Cooper, G.; Aouad, G.; Hinks, J.; Sexton, M.; and Sheath , D. (1998b).
Final Report: Generic Design and Construction Process Protocol. University of Salford, September 1998, ISBN: 0-902896-19-9.

Karstila, K. & Björk, B-C. (1999).
"Models for the Construction Process - The MoPo-project". *Proceedings of the 2nd International Conference on Concurrent Engineering in Construction - CEC99*, Espoo, CIB TG33 and VTT, Finland, August 25-27, 1999.

Kavanagh, T. C., Muller, F. and O'Brien, J. J. (1978).
Construction Management -A Professional Approach. McGraw-Hill Book Co., New York, N. Y., USA, ISBN: 0070333866.

Kendall, K. E. and Kendall, J. E. (1995).
Systems Analysis and Design. Third Edition, published by Prentice-Hall, Inc., ISBN: 0134366921.

Khemlani, L. (2001).
XML for the Building Industry. CADENCE AEC Tech News #54 (July 6, 2001), http://www.cadenceweb.com/newsletter/aec/0701_6.html.

Kong, C. W.; Li, H.; and Love, P. E. D. (2001).
"An E-Commerce System for Construction Material Procurement". *Construction Innovation* 1(1): 43-54.

Kubeck, Lynn C. (1995).
Techniques for Business Process Redesign: Tying It All Together. Published by John Wiley & Sons Canada, Limited, New York, N. Y., USA, ISBN: 0471052957.

Kulwiec, R. A. (1985).
Materials Handling Handbook. Sponsored by the American Society of Mechanical Engineers and the International Material Management Society, Wiley, New York, NY. ISBN: 0471097829.

Laudon, K. C. and Laudon, J. P. (1988).
Management Information Systems. Published by Macmillan Publishing Company, a division of Macmillan, Inc., New York, N.Y., USA, ISBN: 0023681004.

Li, H. and Love, E. D. (1998).
"Site-Level Facilities Layout Using Genetic Algorithms". *Journal of Computing in Civil Engineering*, ASCE, 12(4): 227-231.

Love, P. D.; Irani, Z.; Li, H.; Cheng, E. W. L.; and Tse, R. Y. C. (2001).
"An Empirical Analysis of the Barriers to Implementing E-Commerce in Small-Medium Sized Constructors in the State of Victoria, Australia". *Construction Innovation*, 2001, 1(1):31-41.

Luiten, G.; Froese T.; Bjork, B-C.; Cooper, G.; Junge, R.; and Oxman, R. (1993).
"An Information Reference Model for Architecture, Engineering, and Construction". *Management of Information Technology for Construction*, editors: Mathur, K., Betts, M., and Tham, K., World Scientific & Global Publication Services, Singapore 1993, pp. 391-406.

Marca, D. A. and McGowan, C. L. (1993).
IDEFO-SADT Business Process and Enterprise Modeling. Published by Eclectic Solutions Corp., USA, ISBN: 0963875000.

Martin, J. (1982).
Strategic Data-Planning Methodologies. Prentice-Hall, Inc., Englewood Cliffs, NJ, USA.

Martin, J. and Odell, J. (1998).
Object-Oriented Methods, A Foundation. Prentice-Hall PTR, Upper Saddle River, NJ.

McCullouch, B. G. and Lueprasert, K. (1994).
"2D Bar-Code Applications in Construction". *Journal of Construction Engineering and Management*, ASCE, 120(4):739-752.

MEP (2001)
Net Knowledge Basics. A course designed by Manufacturing Extension Partnership (MEP), NIST, USA. Its electronic version is available at URL: http://www.mep.nist.gov/netknowledgebasics.

Mincks, William R., and Johnston, H. (1998).
"*Construction Jobsite Management*". Delmar Publishers, an International Thomson Publishing Company, Albany, New York, USA. ISBN: 0827371627.

Ng , S. T.; Chen, S. E.; McGeorge, D.; Lam, K. C.; and Evans, S. (2001).
"Current State of IT usage by Australian Subcontractors". *Construction Innovation*, 2001, 1(1):3-13.

Nijssen, G. M. and Halpin, T. A. (1989).
Conceptual Schema and Relational Database Design: A fact oriented approach. Prentice-Hall, Inc., Englewood Cliffs, NJ, USA.

Noran, O. S. (2003).
Business Modelling: UML vs. IDEF. School of Computing and Information Technology, Griffith University, Australia. A document available electronically at http://www.cit.gu.edu.au/~noran/Docs/UMLvsIDEF.pdf

Oberlender, G. D. (1993).
Project Management for Engineering and Construction. Mc Graw-Hill Inc., NY, USA. ISBN: 0070481504.

Oxford Dictionary (1989).
Oxford Advanced Learner's Dictionary of Current English. Fourth Edition, Oxford University Press, U.K.

OMG (2005).
Object Management Group. http://www.omg.org/.

Paulson, B. (1995).
Computer Applications in Construction. McGraw Hill. ISBN: 007048967X.

Peurifoy, R. L. and Ledbetter, W. B. (1985).
Construction Planning, Equipment, and Method. McGraw-Hill Series in Construction Engineering and Project Management, McGraw-Hill Book Co-Singapore, ISBN: 007Y664838.

Pheng, L. S. and Chuan, C. J. (2001).
"Just-in-Time Management of Precast Concrete Components". *Journal of Construction Engineering and Management*, ASCE, 127(6):494-501.

PMI (1996).
A Guide to the Project Management Body of Knowledge. Project Management Institute (PMI), PMI Standards Committee, Upper Darby, PA, USA.

Prabhu, V. and Baker, M. (1986).
Materials Management. An IPC (Institute of Production Control) publication, edited by Vas Prabhu and Malcom Baker, published by McGraw-Hill Book Company (UK) Limited, Maidenhead, Berkshire, England, ISBN: 0070849359.

Rankin, J. H. (2000).
Computer-Assisted Construction Planning. Ph.D. Thesis, Construction Engineering and Management Group, Department of Civil Engineering, University of British Columbia, Vancouver, BC, Canada.

Rational (2002).
Rational Software Systems. Rational Software Co., with Rational Rose™ as one of its software products, URL: http://www.rational.com.

Rezgui, Y. and Debras, P. (1995)
"An Integrated Approach for a Model Based Document Production and Management". *Electronic Journal of Information Technology in Construction, ITcon* 1995, vol.1, pp. 1-24. Available at URL: http://itcon.fagg.uni-lj.si/itcon/1995/1/.

Rezgui, Y. and Cooper, P. (1998)
"A Proposed Open Infrastructure for Construction Project Document Sharing". *Electronic Journal of Information Technology in Construction, ITcon* 1998, vol.2, pp. 11-24. Available at URL: http://itcon.org/1998/2/.

Riley, D. R. and Sanvido, V. E. (1995).
"Patterns of Construction-Space Use in Multistory Buildings". *Journal of Construction Engineering and Management*, ASCE, 121(4):464-473.

Riley, D. R., & Sanvido, V. E. (1997).
"Space planning method for multistory building construction". *Journal of Construction Engineering and Management*, ASCE, 123(2):171-180.

Riley, D. R. and Tommelein I. D. (1996).
"Space Planning Tools for Multi-Story Construction". *Proceedings of the Third Congress on Computing in Civil Engineering*, ASCE, Anaheim, CA, USA, June 17-19, 1996, Published by the ASCE, New York, NY, USA, pp. 718-724.

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., and Lorensen, W. (1991).
Object-Oriented Modeling and Design. Prentice-Hall, Englewood Cliffs, NJ, NY, USA.

Russell, A. D. (1997).
Project Views. A handout in a graduate course (CIVL 520: *Construction Planning and Control*) offered by the author at the Department of Civil Engineering, the University of British Columbia, Vancouver, Canada.

Russell, A. D. and Froese, T. (1997).
"Challenges and Visions for Computer-Integrated Management Systems for Medium-Sized Contractors". *Canadian Journal of Civil Engineering*, 24(2):180-190.

RS-Means (1998).
Means Construction Cost Data. 56th Annual Edition, R. S. Means Company, Inc., Kingstone, MA, USA.

Samuelson, O. (2002).
"IT-Barometer 2000 – The Use of IT in the Nordic Construction Industry". *Electronic Journal of Information Technology in Construction, ITcon*, March 2002, Vol. 7, pp. 1-25. Available at URL: http://itcon.org/2002/1/.

Sadonio, M.; Tommelein, I. D.; and Zabelle, T. R. (1998).
"The Last Designer's CAD-Database for Sourcing, Procurement, and Planning". *Proceedings of the Fifth Congress on Computing in Civil Engineering*, ASCE, Boston Massachusetts, USA, Oct. 18-21, 1998, Published by the ASCE, Reston, Virginia, USA, pp. 364-375.

Sanvido, V. E. (1987).
An Open Information Architecture to Integrate the Design, Construction, and Facility Management of Buildings. National Science Foundation Computer Integrated Construction Project Proposal.

Sanvido, V. E. (1990).
An Integrated Building Process Model. Technical Report No. 1, Computer Integrated Construction Program, Dept. of Architectural Engineering, Pennsylvania State University, University Park, Pa, USA.

Scheyer, W. L. (1985).
Handbook of Health Care Material Management. Aspen Systems Corp., Rockville, Md., ISBN: 0871891026.

Schneider, G. and Winters, J. P. (1997).
Applying Use Cases: A Practical Guide. Addison-Wesley, MA, ISBN: 0201309815.

Sommerville, Ian (1997)
Software Engineering. Fifth Edition, published by Addison Wesley Longman Limited, USA, ISBN: 0202427656.

Stukhart, G. (1995).
Construction materials management. With a contribution by David Kirby, Marcel Deckker Inc., New York, NY, USA., ISBN: 0824793609.

Stukhart, G. and Cook, L. (1990).
"Bar-Code Srandardization in Industrial Construction". *Journal of Construction Engineering and Management*, ASCE, 116(3):416-431.

Stumpf, A. L.; Chin, S; Liu, L. Y.; and Ganeshan, R. (1995).
"Use of a Relational Database System to Integrate Product and Process Information During Construction". *Proceedings of the CIB Workshop on Computers and Information in Construction*. Fischer, M., Law, K. H., and Luiten, B. (eds). Publ 180, Stanford University, Stanford, California, USA, August 1995, pp. 316-326.

Sullivan, R. F. (1989).
"*Integrated Field Systems*". A paper by Stone & Webster Engineering Co. *Proceedings of the 10th Annual Construction Productivity Improvement Conference*, University of Texas at Austin, Austin, TX, September 11-13, 1989, pp. 94-114.

Sweet's CD (2003).
Sweet's CD: Winter 2003. Sweet's CD Version 3.9, Sweet's Construction Product Marketplace™. The McGraw-Hill Companies, Inc. New York, NY. Sweet's Group Home Page at URL: http://www.sweets.com

Taylor, David. A. (1995).
Business Engineering with Object Technology. Published by John Wiley & Sons Canada, Limited. ISBN: 0471045217.

Teicholz, P. and Fischer, M. (1994).
"Strategy for Computer Integrated Construction Technology". *Journal of Construction Engineering and Management*, ASCE, 120(1):117-131.

Tersine, R. J. (1976).
Materials Management and Inventory Systems. Published by North Holland, New York, ISBN: 0720486025.

Thomas, H. R., Riley, D. R., and Sanvido, V. (1999).
"Loss of Labor Productivity due to Delivery Methods and Weather". *Journal of Construction Engineering and Management*, ASCE, 125(1):39-46.

Timmers, P. (2000).
Electronic Commerce; Strategies and Models for Business-to-Business Trading. John Wiley & Sons Ltd, England, UK, ISBN: 0-471-72029-1. Its eBook version is available by netLibrary at URL: http://www.netlibrary.com.

Tommelein, I. D. and Zouein, P. P. (1993).
"Interactive Dynamic Layout Planning". *Journal of Construction Engineering and Management*, ASCE, 119(2):266-287.

Tommelein, I. D., Dzeng, R., and Zouein, P. P. (1993).
"Exchanging Layout and Schedule Data in a Real-Time Distributed Environment". *Proceedings of the 5th International Conference on Computing in Civil and Building Engineering (V-ICCCBE)*, ASCE, Anaheim, CA, USA, June 7-9, 1993, Published by the ASCE, New York, NY, USA. Vol. 1 pp. 947-954.

Tommelein, I. D. (1994).
"MoveCapPlan*: An Integrated System for Planning and Controlling Construction Material Laydown and Handling*". *Proceedings of the First Congress on Computing in Civil Engineering*, Washington, D.C., USA, June 20-22, 1994 Published by the ASCE, New York, NY, USA, pp. 1172-1179.

Trus, S. and Collica, J. (1995).
Electronic Commerce and Electronic Data Interchange. Electronic Commerce group, Computer Systems Laboratory (CSL), NIST. URL: http://snad.ncsl.nist.gov/ecg/pubs/analyze_ec/tableofcontents2_1.html. Last time checked on January 2003.

Turban, E.; Lee, J.; King, D.; and Chung, H. M. (2000).
Electronic Commerce: A Managerial Perspective. Prentice-Hall, Inc., Upper Saddle River, N.J., USA.

UML 1.3 (2000).
OMG Unified Modeling Language Specification. Version 1.3, First Edition, March 2000, by Object Management Group (OMG).

UML 1.4 (2001).
OMG Unified Modeling Language Specification. Version 1.4, Final Edition, September 2001, by Object Management Group (OMG).

UML-BPM 1.1 (1997).
UML Extension for Business Process Modeling. Version 1.1, September 1997, by Object Management Group (OMG).

USEPA (1991).
Medical Waste Management and Disposal". Pollution technology review: 0090-516X, no. 200; U.S. Environmental Protection Agency, Office of Solid Waste ... [et al.], published by Noyes Data Corp., Park Ridge, NJ. U.S.A. ISBN: 0815512643.

van Nederveen, S. and Tolman, F. (2001).
"Neutral Object Tree Support for Inter-Discipline Communication in Large-Scale Construction". *Electronic Journal of Information Technology in Construction, ITcon* September 2001, Vol.6, pp. 37-46. Available at URL: http://itcon.org/2001/3/.

Walker, A. (1984).
Project Management in construction. Granada Publishing Ltd, London, UK.

Watson, D. A. (1986).
Construction Materials and Processes. Third Edition, McGraw-Hill, Inc., New York, NY, ISBN: 0070684766.

Wittenoom, R.; Howard Leslie, H.; and Mitchell, J. (1997).
"Industrial Data Requirements During Early Project Stages". A paper from the Australian STEP Community presented at the *Workshop on Architectures for Industrial Data*, ISO TC184/SC4/WG10 - Technical Architecture, National Institute of Standards and Technology, January 7-9, 1997.

Wix, J. and DP, B. (1995).
"Standardisation in the Building Industry: The STEP Building Construction Core *Model*". *Proceedings of the CIB Workshop on Computers and Information in Construction*. Fischer, M., Law, K. H., and Luiten, B. (eds). Publication 180, Stanford University, Stanford, California, USA, August 1995, pp. 184-195.

Wix, J., Liebich, T. (2001).
"*Intelligent Service Tools for Concurrent Engineering: Information Flow Scenario*". Draft 4, 5 JANUARY 2001

Wood, D. P. and Kang, K. C. (1992).
A Classification and Bibliography of Software Prototyping. A Technical Report (CMU/SEI-92-TR-13, ESD-92-TR-013 October 1992), prepared under Requirements Engineering Project, published by Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA.

Yu, K. (2002).
Computer Integrated Construction Based On Total Project Systems Framework. Ph.D. Thesis, Construction Engineering and Management Group, Department of Civil Engineering, University of British Columbia, Vancouver, BC, Canada.

Yu, K., Froese, T., and Grobler, F. (2000).
"A Development Framework for Data Models for Computer-Integrated Facilities Management". *Automation in Construction*, Elsevier Science B.V., 2000, 9:145-167.

Appendix A: Selected Modeling Techniques and Tools

This appendix describes several modeling techniques and tools that are used or referred to in this dissertation.

A-1 Model Grouping and the Domain-Neutral Component Model

Grouping of classes into a set of higher-level classes or categories has been used as a technique in information modeling for a variety of purposes, including managing model complexity. For example, Jacobson suggests that objects in analysis models be categorized into *Entity*, *Boundary*, and *Control* objects [Jacobson et al. 1992] (section 2.2.4).

In another approach, the STEP project uses the term *Unit of Functionality* (or UoF) [ISO 1992 and 1995a] to refer to a group of objects in a data model that share a specific knowledge area within an application domain. The UoF is defined as "a collection of application objects and their relationships that defines one or more concepts within the application context such that the removal of any component would render the concepts incomplete and ambiguous" [ISO 1992]. Examples of UoF are building elements, building composition, and design administration. However, the UoF is only a mechanism for *grouping* of objects, and no explicit relationships are defined between the UoF's themselves.

In a similar approach, though more formally, the IFC model [IFC R2x, 2000a] groups objects into a set of schemas (or modules), which are assigned to the layers of the model architecture. Each schema represents a business concept (e.g., actor, approval, etc.), a property concept (e.g., cost, geometry, topology, representation, etc.), or a discipline (e.g., architecture, HVAC, construction management, etc.). The schemas act like UoF's; however, some access and use constraints have been defined among them based on the IFC model architecture (section 2.4.4.7.2).

The UML modeling constructs [UML 1.4. 2001] can effectively be applied to represent these typing and grouping of objects. For example, the *Stereotype* and *Package* notations can be used to represent types of objects and groups of objects, respectively.

With a slightly different view to object grouping, Coad [Coad et al. 1999] introduces the concept of *Archetype* as a mechanism for grouping of objects of the same type, based on their responsibilities (i.e., attributes, links, methods, and interactions). An archetype is, in fact, a high-level class that "describes a model that classes (i.e., domain objects) with that archetype *more or less* follow." The "*more or less*" constraint implies that classes of a specific archetype may not be considered as subtypes of and inherit from that archetype. Coad uses the UML stereotype notation to represent an archetype. He suggests four types of high-level classes (i.e., *Archetypes*), to which domain objects may belong: 1) *Moment-Interval*

(something that one needs to work with and track for business or legal reasons, and that occurs at a moment in time, or over an interval of time), 2- *Role* (a way of participation by a party, place, or thing), 3) *Party/Place/Thing* (the role-player, which is uniquely identified and may play different roles in different situations, and 4- *Description* (a catalog-entry-like description; e.g., aircraft description, vehicle description, parts-catalog entry, and clothing-catalog entry).

In order to highlight this categorization mechanism in a model for better understandability, he suggests using colors for the archetypes: pink (for *Moment-Interval*), yellow (for *Role*), green (for *Party/Place/Thing*), and blue (for *Description*); white is also occasionally used for notes, for plug-in points (i.e., interfaces), and for system-interaction proxies. Applying this categorization of concepts, Coad formulates and suggests a generic domain object model, called the *Domain-Neutral Component (DNC)* model. The DNC model defines a pattern of archetypes that occurs again and again within problem domain object models of business systems. It suggests typical attributes and methods for classes of different archetypes as well as interactions between them. Coad et al. states: "*We've built hundreds and hundreds of models. All of them follow this domain-neutral component model*" [Coad et al. 1999, p. 15]. Using UML class diagram notations, Figure A-1 shows the DNC model. The models uses colors and UML stereotype notations to distinguish between different kinds of archetypes. Due to the black-and-white nature of the figures presented in this section, however, the archetypes' colors are denoted with a single letter in the lower-right corner of each object: P = pink, Y = yellow, G = green, and B = blue.

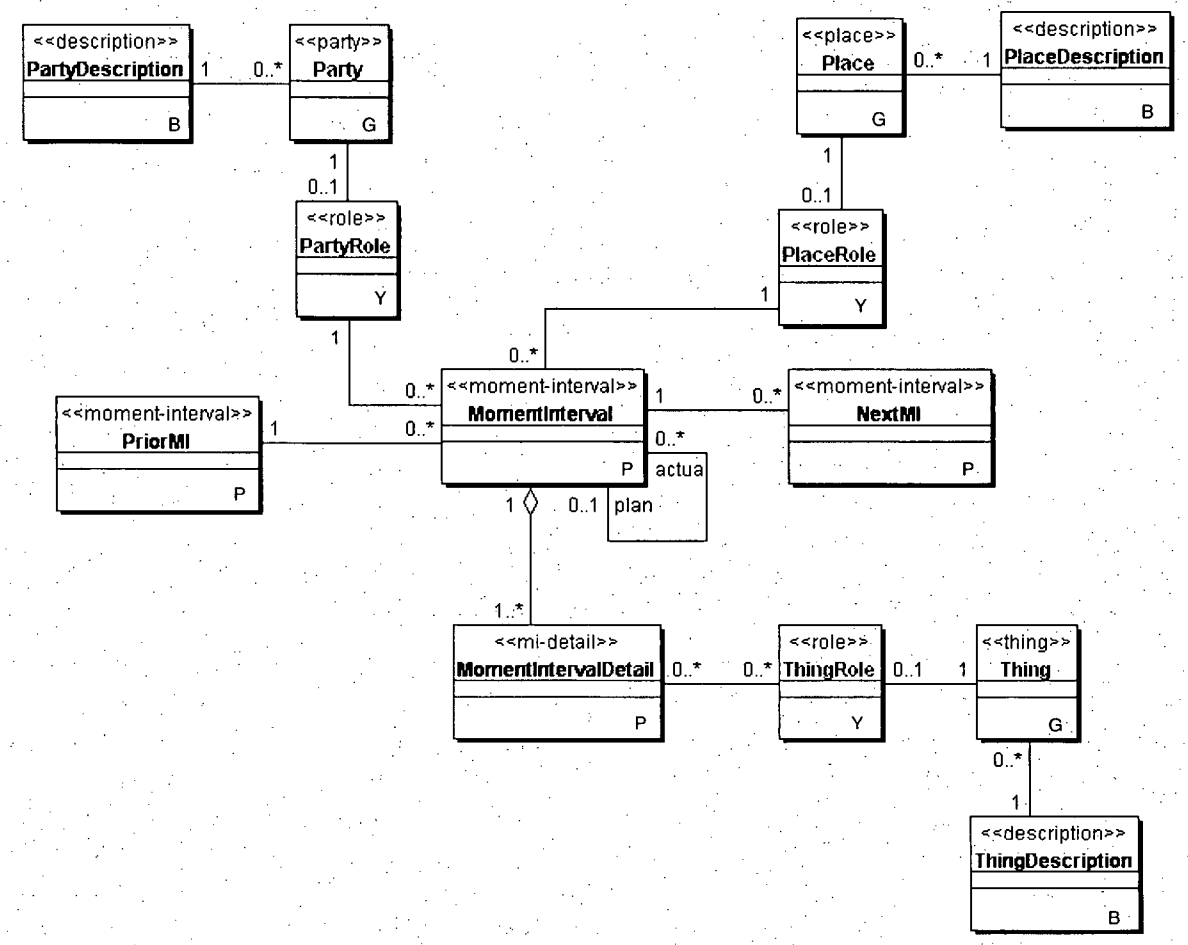


Figure A–1: The Domain-Neutral Component

[Adapted from CoadLetter #68, 2000]

The DNC model offers several advantages. First, it provides a *framework* that facilitates domain object modeling: business objects of a domain can be easily found and grouped into the four basic types of objects (i.e., archetypes). Second, it provides a *template* for specification of objects: the basic behaviors and properties defined by the archetypes could be exploited and examined for objects of a specific archetype. Finally, the four archetype colors considerably enhances the *understandability* of the model for the model users (e.g., analyzer, reviewer, client, etc.).

Nevertheless, despite recommendations, the DNC model may not be considered as a universally applicable pattern for domain object modeling. The CoadLetter [#68, 2000] gives an example of different approaches to domain object modeling where, depending on preferences and the interest of the application system, the object model may change from a complex model with multiple objects to a model with a single object.

More importantly, the DNC model may dictate rigidity in terms of *objects' relationships*. The necessity of relationships between type objects (i.e., the *Description* archetype) is a perfect example in this regard. Such a necessity may be examined in a material testing database application scenario, in which the testing requirement for different material types is managed. In order to retrieve or access the types of tests that are required for a given type of material, a direct link (i.e., relationship) may normally be required between a *Material Type* object (i.e., the *Description*) and a *Material Test Type* object; i.e., without any need to involve an occurrence of *Material* object (i.e., the *Thing*) and/or *Material Test* object (i.e., the *Moment-Interval*). Construction PM processes, which heavily rely on knowledge management, demonstrate a high need for such a *type-to-type* relationship [Rankin 2000].

Among other scenarios is the relationship between the *Description* and *Moment-Interval* archetypes. A *Purchase Order* is a typical example in which a number of type objects (e.g., some *Material Type* objects like one-inch gravel or type II concrete) are entered in the line items of the purchase order. This shows that not all possible relationships between the four archetypes are suggested by the Coad's generic DNC model.

Considering the advantages and limitations offered by the DNC model, especially the *rigidity of relationships*, it may be generally concluded that the model is best suited to systems with a high degree of standardization and control (i.e., controlled environment).

A-2 EXPRESS Modeling Language and EXPRESS-G

As a part of the STEP standard [ISO 1995a], the EXPRESS data definition language and its graphical technique, EXPRESS-G [ISO 1991], have been widely used for information modeling in AEC/FM information systems development research. Almost all STEP information models [Froese 1996c] and the IFC model [IFC R2x, 2000a] are specified using this technique. Information models specified using this technique are independent of any implementation context.

EXPRESS-G is a graphical modeling notation that is used to describe classes (or entities) and their attributes and relationships; much like an entity-relationship diagram, but using object-oriented concepts. All model elements represented in an EXPRESS-G model can be coded in EXPRESS, but the reverse may not be true. The graphic symbols of EXPRESS-G may be categorized into three major groups: definition, relationship, and diagram-composition symbols. Most of these symbols, together with their names and definitions, are summarized as follows (also see Figure A-2):

1) *Definition symbols* (i.e., different forms of rectangular boxes): A labeled solid box represents a *class/entity*, and a rounded box placed in another simple box with continuous or dashed lines shows a class that is modeled in another schema but is being used or referenced respectively. Types (i.e.,

select, *enumeration*, and *defined data type*) are shown with labeled dashed boxes. The select data type and enumeration, however, have a thin dashed line on the left and right side of their boxes respectively. A labeled solid box with a thin vertical line on its right side represents a *simple data types* such as real, integer, number, logical, binary, and string.

2) *Relationship symbols* (i.e., lines connecting definition symbols): Relationships are bi-directional, but one direction is emphasized, using an open circle in the emphasized direction (in this dissertation, for ease of drawing, a filled circle is conventionally used instead of the open circle). A labeled thin line shows a relation between two classes or between a class and a data type. A select type relation is also shown with an unlabeled thin solid line branching from a select type. On the other hand, thick solid lines represent generalization/specialization relationships. Everything that is related to a class is considered as an attribute of the class.

3) *Diagram Composition symbols* (i.e., referencing from one page/diagram to another): When there is a relationship (i.e., reference) to a class which is defined in another page/diagram, the class will be shown with a rounded box that contains a page number (i.e., the referenced page) and a reference number. Then, in the source page, another rounded box labeled with the same reference number and a page number (i.e., referrer page) would resemble the referrer. This box would have a link to the referenced class.

The aggregation relations supported in EXPRESS include Array, Bag, List, and Set. However, the last two are the most widely used aggregations within the IFC specification. LIST is a collection of things with order (sequence) and no duplication (e.g., L[1:?], which means a list of one or many). On the other hand, SET is a collection of things with no order and no duplication (e.g., S[1:?], which means a set of one or many). ARRAY is a fixed size collection of things with order represented (e.g., A[1:?]), and BAG is a collection of things with no order and allowed duplication (e.g., B[1:?]).

The EXPRESS language allows the definition of separate schemas, and the linkage between them through import declarations such as USE and REFERENCE clauses. This allows declarations made in other schema to be used or referenced but does not make them a part of the current schema; i.e., the declarations remain remote. COMBINE project uses this feature as a way to decompose its IDM (Integrated Data Model) in a number of small schemas and to make explicit their relationship and dependency links (section 2.4.4.5) [Dubois et al. 1994]. Also, the IAI uses this feature for linkage between classes of different schemas (i.e., modules) of the IFC object model (section 2.4.4.7) [IFC R2x, 2000a].

Graphical Notation	Name	Description
ClassName	Class/Entity	A thing that is of interest in a domain.
(ABS) ClassName	Abstract Class	A class that has no instance in the application. Its presence in a model is usually to act as a parent whose properties could be inherited by its children (i.e., subtypes).
STRING INTEGER NUMBER REAL BOOLEAN LOGICAL BINARY	Simple Data Type	An atomic data types that cannot be subdivided into any smaller thing. The simple data types defined in EXPRESS and EXPRESS-G include: STRING (alphanumerical characters), INTEGER (a whole number with no decimals), NUMBER (either real or integer), REAL (a rational number with decimals), BOOLEAN (true or false), LOGICAL (true, false, or unknown), and BINARY (sequence of 1 and 0).
Enum	Enumeration Data Type	A data type providing for a range of possible values defined in an enumeration list by an attribute of a class. Only one of the values may be assigned to the attribute.
Select	Select Data Type	A data type enabling the choice among a number of classes. It functions similar to supertype/subtype.
Label	Defined Data Type	A text that is used as a value for an attribute, like a description (e.g., "A 15% waste should be considered" as the value of the "description" attribute of a "product" object).
Relation	Mandatory Relation	*one and only one (1)* relation exists between the elements related by the relation (here, from left to right). *
Relation	Optional Relation	*zero or one (0-1)* relation exists between the elements related by the relation (here, from left to right). *
Relation S[1:?]	Mandatory Set Relation	*one or many (1-n)* relation exists between the elements related by the relation (here, from left to right). *
Relation S[1:?]	Optional Set Relation	*zero, one, or many (0-1-n)* relation exists between the elements related by the relation (here, from left to right). *
(INV) Relation	Inverse Relation	An inverse relation exists between the elements related by the relation (here, from right to left). *

Figure A–2: The Graphical Notations of Basic Modeling Elements in EXPRESS-G (p. 1/2)

Graphical Notation	Name	Description
	Select Type Relation	*one and only one* of the child classes may be selected. *
1	Exclusive Supertype/Subtype	The instance of the supertype can be of type of *only one* of the subtypes. *
	Inclusive Supertype/Subtype	The instance of the supertype can be of type any number of the subtypes (i.e., ANDOR relation). *
Schema.Class	USE from interface	The shown class or data type is being used from another model (or schema). The name of the class or data type and the name of the model in which the used class or data type is defined are shown in the middle rounded box. Note that in this dissertation, the rounded box in the middle is shown with a simple box.
Schema.Class	REFERENCE from interface	The shown class or data type is being referenced from another model (or schema). The name of the class or data type and the name of the model in which the referenced class or data type is defined are shown in the middle rounded box. Note that in this dissertation, the rounded box in the middle is shown with a simple box.
Page, Ref , Class	Connector to Class in another Page	This related Class is defined in another Page. To find the other end, find the connector numbered as Ref in the Page.
Page, Ref, (from Page)	Connector onto this Page	A relationship has been defined by another class "from another Page" to the related class in this "Page".
	Relations (*Our Conventions*)	* **Note**: In this dissertation, for ease of drawing, a filled circle is conventionally used instead of the open circle at the end of all relations (shown in the most left column).

Figure A-2: The Graphical Notations of Basic Modeling Elements in EXPRESS-G (p. 2/2)

A-3 IDEF$_0$ Modeling Technique

This section presents the background and a brief description of the IDEF$_0$ modeling technique [IDEF$_0$ 1981; Feldman 1998].

A-3.1 Background and Modeling Elements

IDEF$_0$ is one of the modeling techniques in the IDEF (ICAM (Integrated Computer Aided Manufacturing) DEFinition) suite of techniques [Noran 2003], which was originally intended for use in systems engineering. Loosely based on the Structured Analysis and Design Technique (SADT) [Marca and McGowan 1993], IDEF$_0$ is one of the most widely used graphical techniques for activity and process modeling with an emphasis on transformation of some input into an output (i.e., flow of things:

information, physical artifacts, etc.). Although it was originally intended for the manufacturing industry [$IDEF_0$ 1981], others such as the AEC/FM's information modeling community have also used this technique for modeling processes. Figure A-3 shows the graphical representation of the basic elements of an $IDEF_0$ diagram using an example. An *activity* is shown as a box, and its logical relationship with other activities are conveyed through the use of arrows, connecting activities. An arrow may represent an *input* (towards the box from the left), an *output* (outward from the left side of the box), a *control* (towards the box from the top), and/or a *mechanism* (towards the box from bottom). Inputs represent things such as information and material that are used and/or processed (typically transformed), while mechanisms represent the resources (e.g., equipment, tools/software, organizations, and people) that perform or are applied to perform the process (thus, typically not consumed). Controls represent things that influence or constrain the process but are not affected or changed in the process. Finally, outputs are the result of the process. Like inputs, outputs can be either information or physical objects. Outputs of one process can be fed into one or more of the following processes (as inputs) and/or constraints on the processes (i.e., as controls).

Each instance of the elements of the $IDEF_0$ diagramming technique should be uniquely identifiable. This is usually done by assigning unique labels (or names) to the elements, except for activities, which must also have a unique "*activity id*" number, as an indicator of the activity's position in the activity decomposition hierarchy. The following shows the general $IDEF_0$ representation format of a process and its related elements (i.e., input, control, output, and mechanism— ICOM) with examples. The example process in called "*Construct Stud-Wall Frame*".

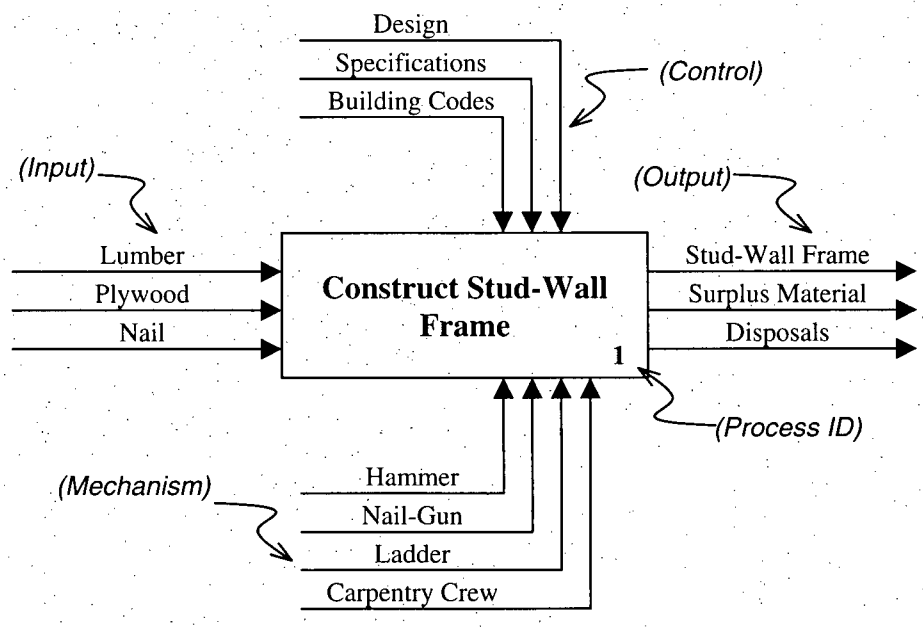


Figure A–3: Example $IDEF_0$ Activity Diagram (Construct Stud-Wall Frame)

A typical $IDEF_0$ diagram consists of related parent and child diagrams. The topmost diagram is called the context (or A-0, read as A minus zero) diagram. The context diagram shows only the topmost parent activity and its related ICOM elements. It also indicates the "*purpose*" (or "*viewpoint*") of the model. The next immediate, lower-level diagram shows the child activities of the parent activity and their related logical ICOM relationships. This may be continued on several levels to present a more detailed breakdown (Figure A-4).

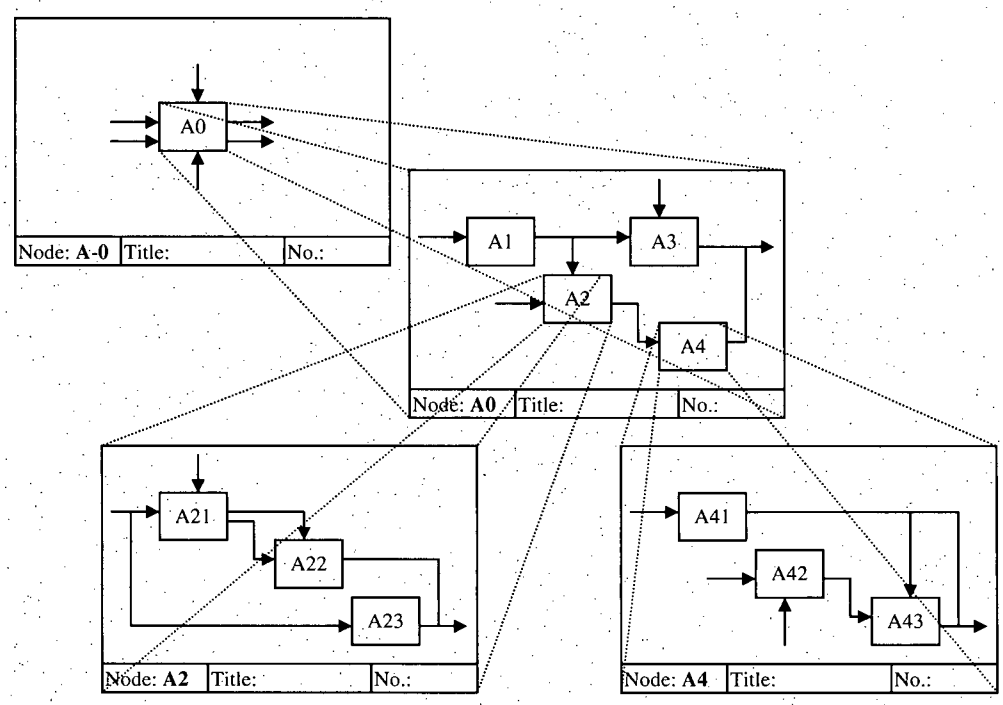


Figure A–4: The Breakdown Structure of $IDEF_0$ Diagrams

In addition to the $IDEF_0$ diagrams, a "*node tree*" diagram is usually used as a starting point for brainstorming and hierarchical structuring of the activities of the process [Feldman 1998]. Each node of the tree (a box or circle) represents an activity, and thus, is labeled based on the activity's name and ID (Figure A-5). In large models, separate node trees may be used to show the activity breakdown structure at different chosen nodes. The IBPM model [Sanvido 1990] is an example of an AEC/FM process model that has extensively used this technique together with the $IDEF_0$ technique. The node tree representation, however, may be preceded or replaced by just a textual representation of the hierarchy using an indentation technique. This technique, which is faster in production and gives more information in less space, is used in this dissertation for presenting AEC/FM process models (e.g., sections 2.5.1.2 and 2.5.1.3).

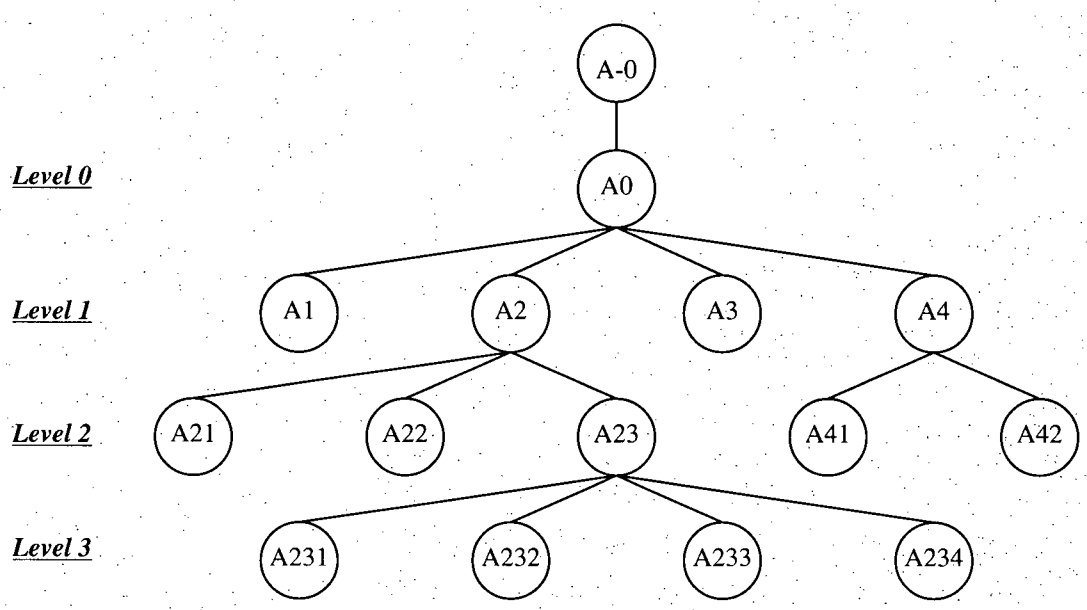


Figure A–5: A Node Tree of an $IDEF_0$ Model

A-3.2 Advantages and Limitations

The $IDEF_0$ technique holds both advantages and limitations. $IDEF_0$ may be used for different purposes, including business process modeling and software process modeling. $IDEF_0$ may be used to model "as-is" and "to-be" processes as well as to identify their *information requirements* using the input and output notations. A simplified version of $IDEF_0$ (with activity, input, and output elements) can also serve as a means of modeling procedural design and user interfaces of computer application systems.

$IDEF_0$ has been used in data modeling projects—e.g., in the STEP [ISO 1995a] and IAI [IAI 2005] projects—as an educational tool and as an introduction to data modeling. It serves to familiarize and educate the data modeler and the reader to the overall processes involved in the domain of concern. There is often no explicit mapping between the ICOM elements of the $IDEF_0$ models and the data objects defined in the resulting data models. Nevertheless, others have not found the $IDEF_0$ technique useful for achieving their goals in information modeling [Froese et al. 1997a; Froese and Rankin 1998; Rankin 2000]. It is reasoned that because of "little strict sequential ordering of the processes" involved in construction PM, "it makes it difficult to fit into the $IDEF_0$ notion of domain processes" [Froese et al. 1997a, p. 5].

Among the key advantages of $IDEF_0$ is its ability to model moderately complex processes in a structured fashion. This ability, which is one of the primary purposes of $IDEF_0$, is accomplished through a hierarchical breakdown of activities of the process and their elements (i.e., input, output, mechanism, and control) into lower level activities and their flows.

In order to reduce complexity and increase understandability of a model, it has been recommended that the model be presented from a specific *viewpoint* [Marca and McGowan 1993; Feldman 1998]. A viewpoint, which is a term originally used in SADT (Structured Analysis and Design Technique) process modeling technique, is a perspective through which one may look at the domain of the model. It may be thought of as "a place, person, or thing one can stand in to view the system in operation" [Marca and McGowan 1993, p. 9]. For example, a formwork production process may be viewed from the viewpoint of a carpenter, a foreman, an inspector, a shop manger, or a project manager. Each one of the viewpoints may result in a different model encompassing the interests of that viewpoint.

However, $IDEF_0$ has some limitations. The following lists a few of these limitations in comparison with other techniques such as UML use cases and activity diagrams (Appendix A):

1) $IDEF_0$ does not offer any mechanism to support for integration of multiple-*viewpoints* models. In order to more comprehensively model processes of a domain (e.g., materials management), shifting from one viewpoint to another may become a necessity. For example, in order to model a wider range of MM processes (e.g., *Determine Requirements*, *Request*, *Order*, *Quote*, *Contract*, *Manufacture*, *Ship*, *Deliver*, *Receive*, *Store*, etc.), consideration of viewpoints such as project manager, contractor, supplier, manufacturer, etc. would be needed. However, the inclusion of multiple viewpoints in a single $IDEF_0$ model has not been recommended [Marca and McGowan 1993; Feldman 1998]. The UML activity diagram is one of the techniques that supports for this functionality through its modeling constructs of Swimlane and Actor.
2) The strict *hierarchical* nature of an $IDEF_0$ model imposes limitations on *flexibility* of the model. The impossibility of *reusing processes* in a model, or over several models is an example of lack of flexibility in an $IDEF_0$ model [Hannus and Pietilainen 1995]. Due to the strict hierarchical breakdown of a process into activities, every activity is assigned a unique ID number that represents its location within the breakdown tree. Therefore, an activity instance can be used and referred to only within one diagram. The UML, on the other hand, provides more flexibility through provision of various constructs (e.g., use case and activity) for modeling processes. The UML's use case modeling is based on the *reusability* concept (Appendix A).
3) *Conditional* activities (i.e., decision points) are not supported by $IDEF_0$. It is not uncommon, especially in AEC/FM processes, that start of an activity be subjected to satisfaction of a condition or criteria (e.g., order materials when the inventory falls below a certain level; approve a document if it satisfies the requirements; etc.). Such functionality is, on the other hand, supported by some other techniques, such as flowchart and UML activity diagrams.

4) *Concurrency* of processes belonging to different parents (i.e., diagrams) cannot be modeled in an $IDEF_0$ model. UML activity diagrams [UML 1.4, 2001] provide such a mechanism through the *Synchronization* modeling construct.

In summary, $IDEF_0$ has widely been used by software development projects for modeling business processes and their information flows. In comparison with the flat, box-and-arrow flowchart technique, $IDEF_0$ offers additional possibilities; e.g., modeling more dimensions of processes (e.g., inputs, constraints, outputs, and resources) and dealing with complexity (through hierarchical breakdown of processes and the viewpoint mechanism). Some of modeling issues associated with the $IDEF_0$ (e.g., lack of support for conditionality, concurrency, synchronization, integration of models of multiple viewpoints, and reusability of processes), most of which were observed in the process modeling practice workshop (Chapter 3), may seriously limit the effectiveness of the process models produced for the purpose of integrated total PM systems development. Despite its relative infancy, the UML overcomes such limitations through the use of its modeling constructs.

A-4 The Unified Modeling Language

This section provides the background and a brief description of the Unified Modeling Language (UML) and its basic modeling constructs.

A-4.1 Background of the UML

The UML is a general-purpose graphic language primarily designed for object-oriented modeling; though modeling of other paradigms is also possible with its modeling constructs. The generality of UML allows it to be used for analysis and design of a variety of development processes. However, UML has been used for specifying, constructing, visualizing, and documenting the artifacts of software-intensive systems. It came into existence with collaboration between three object-oriented methodologists, Grady Booch, Jim Rumbaugh, and Ivar Jacobson to unify their methods (i.e., OOD/Booch, OMT, and OOSE/Objectory) with the OMG (Object Management Group, Inc.), which is a vendor-neutral, international organization [OMG 2005; UML 1.4. 2001; Fowler and Scott 1997]. Founded in 1989, the OMG's primary goals are the reusability, portability, and interoperability of object-based software in distributed, heterogeneous environments by establishment of industry guidelines and specifications for application development.

Since its adoption, the UML specification has gone though a number of revisions. Version 1.1 of the UML was first submitted to the Object Management Group in September 1997 in response to an OMG RFP requesting a standard approach to object-oriented modeling. The proposal was approved by the OMG in November 1997. Version 1.3 of the UML was finalized in June 1999, and its first edition was

published in March 2000 [UML 1.3, 2000]. The UML 1.4, which was published in September 2001 is the version referred to throughout this dissertation. The UML is intended to bring all software development methodologies into a shared language of communication (of both models of the system and models of the process of software development), and is becoming an international standard (i.e., ISO) [OMG 2005].

The UML is *process free*. It has no notion of process and is not dependent on any methodology or implementation environment. It is labeled as a *modeling language*, not a *method*, since it does not contain anything about the way a system is developed. Being process independent, it can be used as a technique with any appropriate process to record the results of analysis, design, and implementation of a system. Although the UML is not necessarily tied to any particular application area or modeling process, its greatest applicability is deemed to be in the area of object-oriented software design because of its provision for object-oriented concepts [Fowler and Scott 1997; Booch et al. 1999; UML 1.4 2001].

The following sections provide a brief description of the UML's basic modeling constructs referred to in this dissertation. Figure A-6 shows some of the basic modeling elements defined in the UML [UML 1.4. 2001]. Some of the useful literature on the UML that were cited by this research includes Fowler and Scott 1997 & 1999; Booch et al. 1999; Cockburn 2001; Jacobson 1992; Jacobson et al. 1999, Jacobson and Bylund 2000; OMG 2005; Schneider and Winters 1997; UML 1.4 2001.

Graphical Notation	Name	Description
«stereotype name»	Stereotype	A subclass of an existing model element with the same form (attributes and relationships) but with a different intent (i.e., semantic). It represents a usage distinction.
	Association Relationship	A structural relationship that describes a set of links, each of which denoting a connection between objects (e.g., between two classes, an actor and a use case, and so on).
	Aggregation Relationship	A special kind of association, representing a whole and a part relationship. The whole object does not control the life of the pointed-to objects. A part can be included in one or more whole objects.
	Composition Relationship	A special form of aggregation, which requires that a part instance be included in at most one composite at a time, and that the composite object is responsible for the creation and destruction of the parts.
	Generalization Relationship	A relationship between a more general element (i.e., supertype) and a more specific element (i.e., subtype). The subtype is fully consistent with the supertype and contains additional information. An instance of the subtype may be used where the supertype is allowed.
	Dependency Relationship	A semantic relationship between two things in which a change in one thing (at the arrow's head) may affect the semantics of the other thing (at the arrow's tail).
	Object Flow	In an activity diagram, it represents the passing of an object from the output of actions in one state to the input of actions in another state.
Actor 1	Actor	A coherent set of roles that users of use cases play when interacting with these use cases. An actor has one role for each use case with which it communicates.
Package A	Package	A mechanism for "logical" (not physical) grouping of some model elements (including packages) into a collection.
Use Case A	Use Case	The specification of a sequence of actions, including variants, that a system (or other entity) can perform when interacting with actors of the system.
«include» «extend» 	Use Case Relationships	*Include*: The source/base use case (at arrow's tail) explicitly incorporates the target use case's behavior (at arrow's head). *Extend*: The base use case implicitly (under certain conditions) incorporates the target use case's behavior. *Generalization*: Like generalization among classes, the child use case (at the arrow's tail) inherits the behavior and meaning of its parent use case (at the arrow's head).

Figure A–6: The Graphical Notations of Basic Modeling Elements of the UML (p. 1/2)

Graphical Notation	Name	Description
--------	Note Anchor	Attaches a note to a model element in a diagram.
This is a NOTE	Note	Provides an arbitrary explanatory text usually attached to a model element for adding details and clarifications.
Object A	Object	An entity with a well-defined boundary and identity that encapsulates state (represented by attributes and relationships) and behavior (represented by operations, methods, and state machines). An object is an instance of a class.
Class A	Class	A description of a set of objects that share the same semantics, attributes, relationships, operations, and methods. A class may use a set of interfaces to specify collections of operations it provides to its environment.
●	Start State	Identifies where the flow starts.
◉	End State	Identifies where the flow ends.
Activity A	Activity	Indicates a step in the flow that involves a set of operations and has some duration.
▬	Synchronization	Indicates a forking (i.e. branching) of the flow (i.e., sequence) into parallel/concurrent flows in which activities take place concurrently or in any order, or a joining (i.e., merging) of parallel flows that must complete before the flow continues.
→	Transition	Represents a relationship between two (action/activity) states indicating the completion of one and the start of another. It may be used to connect from a *start state* to an *activity* or a *synchronization*, or from an *activity* or a *transition* to an *activity* or an *end state*.
Decision?	Decision	Indicates a branch or merge in the flow.
Component A	Component	A modular, deployable, and replaceable part of a system that encapsulates implementation and exposes a set of interfaces; It is typically specified by one or more classifiers (e.g., implementation classes) residing on it and implemented by one or more artifacts (e.g., binary, executable, or script files).

Figure A-6: The Graphical Notations of Basic Modeling Elements of the UML (p. 2/2)

A-4.2 Use Cases and Use Case Models/Diagrams

One of the early activities in the system development process is the transformation of the requirement specification, offered by the client of the system, to the *requirements model* (section 2.1.4), which includes a *use case model* (or diagram) describing the system's functional requirements. A *use case diagram* presents an external view of the system by showing the actors, use cases, and (possibly) some

interfaces, and the relationships between these elements within a system [UML 1.4, 2001]. The concept of actor represents a stimulus (e.g., a human or a system) that exists outside the system and interacts with the system, and the concept of use case represents the tasks performed by the system to produce something of value to one or more actors [Jacobson 1992]. A use case can be either *abstract* or *concrete* (i.e., without or with an instance in the final system, respectively). Figure A-7 illustrates an example of such a diagram.

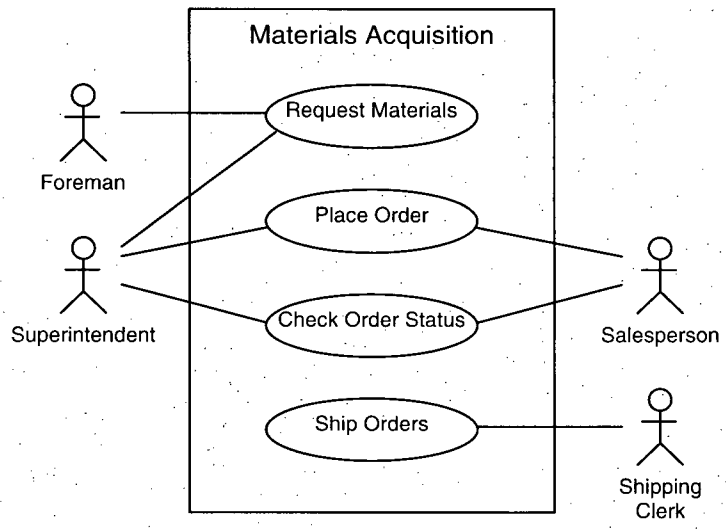


Figure A–7: An Example of a Use Case Diagram

According to the UML 1.4 [2001], three types of relationships can be defined between use cases: *generalization*, *extend*, and *include*. (e.g., Figure A-8). *Generalization* relationships among use cases are very similar to generalization among classes, and they follow the same principles. This means that the child use case inherits (i.e., contains) the meaning and behavior (i.e., all attributes, sequences of behavior, and extension points) of the parent use case and participates in all relationships of the parent use case [UML 1.4, 2001, p. 2-143]; i.e., possibility of use of the child in place of the parent, and the child extending or overriding the behavior of the parent.

The *extend* relationship is generally used to model extensions to other complete, meaningful use cases (e.g., *Cancel Order* in Figure A-8). The relationship is suggested [Cockburn 2001; Jacobson and Bylund 2000; Jacobson et al. 1992; Booch et al. 1999; Fowler and Scott 1997] to be used in situations such as: to write *additions* to a locked requirements document; to model *optional* or *alternative* behavioral parts of use cases (e.g., various order payment options); and to model *conditional* behaviors of use cases (e.g., to place an order if the inventory gets to its specified minimum level). The *include* relationship, on the other hand, is used when a mandatory behavioral part of a use case is factored out and

modeled as an included use case (e.g., *Arrange Payment* in Figure A-8). The *extend* and *include* relationships are a type of dependency relationship stereotyped as «extend» and «include».

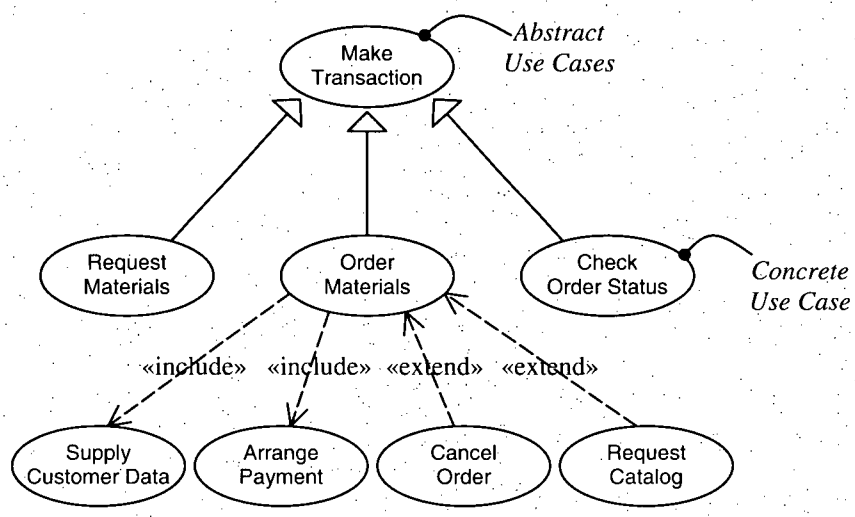


Figure A–8: Use Case, Actors, and their Relationships in a Use Case Diagram

A-4.3 Packages

The UML package notation is a "*generic mechanism*" that can be used for *logical* (not physical) grouping of some logically related modeling elements in a hierarchical format (i.e., each element may be owned by one and only one package, though with the possibility of inter-package referencing). In contrast with such other modeling elements as classes and components (which exist at runtime and, thus, in the model), a package is purely *conceptual* and has no instances in the final system, and thus no existence at runtime). The elements of a package could be any combination of all kinds of logical elements (e.g., actors, use cases, objects, packages, classes, and various diagrams) and physical elements (e.g., component, subsystem) [Booch et al. 1999; UML 1.4, 2001].

Packages may be used as a mechanism for system architecture and configuration control, storage, and access control, and they have proved to be more useful in modeling and managing large and complex systems [Booch et al. 1999, p. 177]. Stereotypes may be used to distinguish one kind of package from another. Packages and their relationships can be presented only in class diagrams.

The UML package notation can be used for managing model views. Referring to the complexity involved in the process of development of software-intensive systems, Booch et al. [1999, p. 30] highlight systems' architecture as a means of control of the inherent complexity in the development process:

> *"Visualizing, specifying, constructing, and documenting a software-intensive system demands that the system be viewed from a number of perspectives. Different stakeholders—end users,*

analysts, developers, system integrator, testers, technical writers, and project mangers- each bring different agendas to a project, and each looks at that system in different ways at different times over the project's life. A system's architecture is perhaps the most important artifact that can be used to manage these different viewpoints and so control the iterative and incremental development of a system throughout its life cycle".

Referring to system's architecture as a set of some significant decisions about different aspects of the system at each incremental step of the development process, Booch et al. [1999, pp. 31-32] suggest five architectural views: 1) *use case view* (system's functional requirements: use cases and use case diagrams); 2) *design view* (specification of the static and dynamic aspects of the system: classes, interfaces, class diagrams, object diagrams and interaction diagrams, activity diagrams, etc); 3) *process view* (threads and processes forming system's concurrency and synchronization mechanisms: the same types of diagrams as in design view but focusing on the active classes, representing the threads and processes); 4) *implementation view* (components and files or code used to assemble and release the physical software system); 5) *deployment view* (physical nodes in the system's hardware topology, e.g., printer, scanner, etc.).

They further suggest that UML packages be used to represent these architectural views. The use case view is considered as a central integrating package among others. These five major packages are recommended as a top-level decomposition of the system's models that is appropriate for even the most complex system one might encounter in practice [Booch et al. 1999, p. 180]. Taking all views into one picture, the use case view may encompass the problem and requirements, while others portray the suggested solution. Nevertheless, in practice, each view defines some requirements for, and thus an input to, the next view for defining and developing a solution.

Presentation of packages and their relationships in a class diagram can help capture a high-level design and architecture of a system. Chapters 7 and 8 explain how the UML package construct is used in the models developed in this research.

A-4.4 Activity Diagrams

The UML Specification defines an activity diagram (or graph) in terms of a state machine: "*A special case of a state machine that is used to model processes involving one or more classifiers*" (e.g., classes, use cases, and subsystems) [UML 1.4, 2001, p. B-3]. Booch et al. [1999, p. 25] suggest, "An activity diagram is a special kind of statechart diagram that shows the flow from activity to activity within a system. Activity diagrams address the dynamic view of a system. They are especially important in modeling the function of a system and emphasize the flow of control among objects."

An activity diagram can include elements such as *activity*, *decision*, *synchronization*, *swimlane*, *transition*, *object*, *object flow*, *start state*, and *end state*. A complete listing of the elements used in this dissertation, together with their definitions and graphical symbols, are presented in Figure A-6.

Activity diagrams may be applied for different purposes. They can be used to visualize and document both business and software processes. In the context of the processes involved in the systems analysis and design, however, activity diagrams can be of great help in identifying and documenting the details of specific desired functionality of the system. Activity diagrams can be applied to:

1) Model *business processes* (i.e., flow of *business activities* and *objects* from one section or department of the business to next). Here, activities represent the physical activities in the business (e.g., process order, pull materials, ship materials, etc.). The result of such an application would be the identification of a set of use cases (i.e., software processes), which can assist the business in handling the business processes, and a set of objects, which represent instances of the classes in the targeted application system. The identified use cases would represent the business processes to be automated.
2) Model *software processes* (i.e., flow of control from one *software activity* to another and creation or changes in the states of *objects* in a software process): Here, a software process may be viewed as a *use case*, which is identified in the first approach. Software activity, on the other hand, would be a specific activity or action (within the use case).
3) Model *operational details* of a software process. Looking at the overall flow within a use case, each flow path can be modeled to visualize and document the set of steps taken in a specific scenario of the use case. Here, activity diagram would act more like a flowchart.

In addition to the provision for *decomposition* and *sequencing* of activities (which are also provided by $IDEF_0$), UML activity diagrams can explicitly represent *conditional* behavior (using *decision* notation) as well as *concurrencies*. The UML activity diagrams together with use cases can provide for modular, reusable process models (using synchronization notation). Activity diagrams can be used to model the *realization of use cases*. The textual description of use cases would communicate the software requirements to the client (who is not usually technical) more effectively. One or more activity diagrams may be associated with a use case; each activity diagram would visually describe a scenario of that use case (i.e., flow of objects information between activities); what a sequence diagram may do to show the flow of messages among objects in a scenario. The pre-and post-conditions may be shown as notes in the activity diagram using *note* notation.

This research uses activity diagrams to model the internal behavior of use cases. Activity diagrams with object flows are also used for the purpose of elicitation of the objects involved in the performance of a use case (chapters 7 and 8).

A-4.5 Stereotype

Stereotypes are one of the built-in extensibility mechanisms of the UML. A *stereotype* is defined as "a new class of metamodel element that ... represents a subclass of an existing metamodel element with the same form (attributes and relationships) but with a different intent. Generally a stereotype represents a usage distinction. A stereotyped element may have additional constraints on it from the base metamodel class" [UML 1.4, 2001, p. 3-31]. A stereotype is presented with a keyword string (as the name of the stereotype) within a couple of matched guillemets (i.e., « ») or double angle brackets (i.e., << >>). The new model element, which has stereotype, is graphically represented using the symbol for the metamodel base element (e.g., use case), but the stereotype symbol (i.e., the guillemets with the keyword string) is placed above the name of the element, if any (Figure A-9).

Certain stereotypes are predefined in the UML (e.g., «extend» and «include», which are used for use case relationships). However, user-defined stereotypes may also be used in a model. Stereotypes are introduced at modeling time and may be interpreted later by code generators and other tools to treat stereotyped elements especially for such purposes as code generation and so on. In this dissertation, for example, stereotypes of «business use case» and «software use case» are used to distinguish between the use cases representing business processes and those representing software processes (i.e., functionality) respectively (Figure A-9).

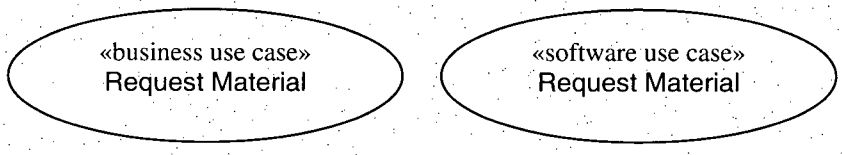


Figure A–9: An Example of Presentation of Stereotypes

A stereotype represents the subclassification of a model element (i.e., any UML modeling element such as a class, a package, a use case, an activity, an activity diagram, and so on). Stereotypes may extend the semantics, but not the structure of pre-existing types and classes.

A-5 Ensemble Streams™ and Business Process Modeling

Ensemble Streams™ (Streams) [2000] is a CASE tool for Business Processes Modeling (BPM) using Activity Diagrams of the UML, which has been widely accepted by the software industry. It supports such UML notations as those needed to draw UML activity diagrams (e.g., start state, end state, activity, transition, dependency, synchronization, decision, swimlane, note, note anchor) and others notations such as package, actor, use case, object, and class. It also allows the user to add GUI window designs and associate them with activities in the diagrams. Streams can also export information to

Rational Rose™ (Rose) [Rational Rose 2002], which is a CASE tool for software engineering, to support the design of support systems for the analyzed business processes. The actors, use cases, classes, and associations generated in Streams can be exported to a Rose model through the export add-in. The Rose model can then be used for further software engineering activities in the Rose environment.

Process models, developed in Streams, specify and document business process flows in the form of the UML activity diagram, together with the business objects involved in the performance of activities of the process. The model may be used for three major purposes in a software development project. First, they can be considered as a part of the specification and documentation of a project. Second, they can be used for communication purpose (e.g., between business analyst, system analyst, and end user of the system). Finally, and more importantly, the business process models can be synchronized with the analysis and design models of the software system. The latter is handled through the export/ import capability of Streams with Rational Rose™. A brief description of Streams is presented in the following section. This includes the modeling notations and methodology as well as the user interface of the tool.

A-5.1 The Streams Modeling Elements: Notations and Relationships

Figure A-10 presents the names, the graphical notations, and the meanings of the basic modeling elements supported by the Streams. Most of theses element have been used in the models of this dissertation. Moreover, using EXPRESS-G notations (Appendix A), Figure A-11 shows a simplified metamodel of the concepts represented in the Streams. With the exception of activity diagrams, all modeling elements can be placed on an activity diagram. However, the activity diagram may not own any of the elements. A modeling element may be exclusively owned by one element other than an activity diagram, but it can appear on one or more activity diagrams. This is facilitated by importing the included elements from their owning elements into an activity diagram.

Some modeling elements can possibly be a composition of (i.e., own) other model elements. For example, a number of activity diagrams, activities, objects, etc. can be associated with an activity. Such an association is, in fact, treated as a composition relationship defined between the activity (i.e., the parent) and its related elements (i.e., the children). This means that the elements are exclusively owned by the activity. The owned elements are graphically represented as children of the parent in a tree format in the browser window (Figure A-12). There is synchronization between a model element and its copies, appearing on different activity diagrams, and any change incorporated into the specification of one would be reflected in the others. Table A-1 shows the possibility for a model element (in the first column of the table) to own other elements (in the first row of the table) in a Streams model. It also shows which elements can be included in an activity diagram. The arrow in the lower left corner of the table depicts the direction of the relationship.

Graphical Notation	Name	Description
Actor 1	Actor	Represents a role that the user of the system (i.e., a human, a hardware device, or another system) plays when interacting with the system that is under study.
Package A	Package	A mechanism for "logical" (not physical) grouping of some model elements (including packages) into a collection.
Use Case A	Use Case	Describes a functionality of a system that is of value to the user of the system.
<<software use case>> <<business use case>>	Stereotype	Streams™ methodology suggests that use cases represent functionality of software systems. In this research, however, *stereotype* notation is used to represent both business use cases and software use cases. A stereotype notation is placed on the top of a use case to explicitly indicate whether the use case represents a function in the business or in a software system.
Object A	Object	A business object (i.e., a piece of information or a physical object) created, used, manipulated by an *activity* in a business system. It can be either an input to or output from an *activity*.
[text]	State	Placed in an object symbol, after the name of the object, it describes a distinguishing characteristic of the object at a particular point in the process.
------>	Object Flow	An object flow symbol is used to show an object entering an activity as input or leaving an activity as output.
------>	Dependency	A semantic relationship between two things in which a change in one thing (i.e., the independent element, at the arrow's head) may affect the semantics of the other thing (i.e., the dependent element, at the arrow's tail).
————	Association (In UML, not Streams)	A structural relationship that describes a set of links (i.e., a connection between objects); e.g., between two classes, an actor and a use case; not supported in Streams.
------>	Association (In Streams)	An association between an actor and a use case or two classes (A convention in Streams; a simple continuous line in UML).
<<include>> ------>	Use Case Relationships	*Include*: The source/base use case (at arrow's tail) explicitly incorporates the target use case's behavior (at arrow's head).
<<extend>> ------>		*Extend*: The base use case implicitly (under certain conditions) incorporates the target use case's behavior.
<<generalization>> ------>		*Generalization*: Like generalization among classes, the child use case inherits the behavior and meaning of its parent use case (at the arrow's head). In UML, the notation is a simple empty-triangle head arrow with no stereotype (————▷).

Figure A–10: The Graphical Notations and Meanings of the Basic Modeling Elements of a Streams Model (p. 1/2)

Graphical Notation	Name	Description
●	Start	Identifies where the flow starts.
◉	End State	Identifies where the flow ends.
Activity A	Activity	Indicates a step in the flow that involves a set of operations and has some duration. Though, duration is not of concern in a Streams model. An activity with a plus sign in its upper-left corner would mean the activity is not atomic and includes other sub-activities.
▬	Synchronization	Indicates a forking (i.e.branching) of the flow (i.e., sequence) into parallel/concurrent flows in which activities take place concurrently or in any order, or a joining (i.e., merging) of parallel flows that must complete before the flow continues.
→	Transition	Indicates the completion of one activity and the start of another. It may be used to connect from a *start state* to an *activity* or a *synchronization*, or from an *activity* or a *transition* to an *activity* or an *end state*.
text	Trigger	It is placed on a transition, and it indicates an event that causes the transition. It can be placed only on the transition from a start state.
[text]	Guard	A condition placed as a label on a transition. It can be true or false; if true, the transition can take place.
Decision?	Decision	Indicates a branch or merge in the flow. A criterion may be evaluated at the branching point (i.e., decision) to determine which path of the branch would be executed.
This is a NOTE	Note	Provides an arbitrary explanatory text usually attached to a model element for adding details and clarifications.
- - - - - - -	Note Anchor	Attaches a note to a model element in a diagram.
Class A	Class	The collection of all information about the business objects of a particular type.

Figure A-10: The Graphical Notations and Meanings of the Basic Modeling Elements of a Streams Model (p. 2/2)

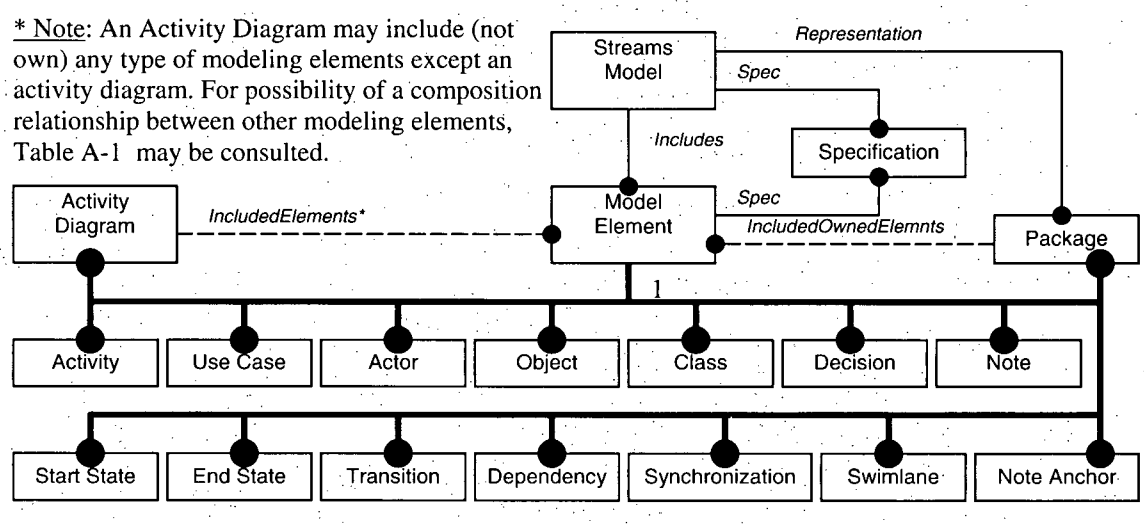


Figure A–11: A Simplified Metamodel of the Streams

Table A–1: Possiblity of Inclusion and Composition Relationship Between Model Elements in the Streams

	Mdl	ADm	Pkg	UC	Act	Actr	Obj	Cls	Dec	Nte	SS	ES	Trn	Dep	Syn	NAn	Sln
Mdl		✓✓	✓✓	✓✓	✓✓	✓✓	✓✓	✓✓	✓✓		✓✓	✓✓			✓✓		
ADm			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Pkg		✓✓	✓✓	✓✓	✓✓	✓✓	✓✓	✓✓	✓✓		✓✓	✓✓			✓✓		
UC		✓✓		✓✓	✓✓	✓✓	✓✓	✓✓	✓✓		✓✓	✓✓			✓✓		
Act		✓✓		✓✓	✓✓	✓✓	✓✓	✓✓	✓✓		✓✓	✓✓			✓✓		
Actr																	
Obj																	
Cls		✓✓		✓✓	✓✓	✓✓	✓✓	✓✓	✓✓		✓✓	✓✓			✓✓		
Dec																	
Nte																	
SS																	
ES																	
Trn																	
Dep																	
Syn																	
Nan																	
Sln																	

Notes:

1- The following abbreviations are used in the above table: Mdl=Model, ADm=Activity Diagram, Pkg=Package, UC=Use Case, Act=Activity, Actr=Actor, Obj=Object, Cls=Class, Dec=Decision, Nte=Note, SS=Start State, ES=End State, Trn=Transition, Dep=Dependency, Syn=Synchronization, NAn=Note Anchor, Sln=Swimlane.

2- A ✓ sign means an inclusion relationship, which may be defined between an activity diagram and other modeling elements. The included elements are, in fact, copies of their corresponding elements, which are imported to be appeared on one or more activity diagrams.

3- A ✓✓sign means that a composition relationship is supported, and thus, the parent and its children are graphically presented in the model tree of the browser window of the Streams user interface (Figure A-12).

A-5.2 The Method Suggested by Streams

Streams suggests its own method for structuring and developing a business process model, although some flexibility has been provided by the application for modification of the methodology. The underlying core of the Streams model suggested by the method is a "*hierarchical process decomposition*" [Ensemble Streams 2000]. The method relies heavily on a hierarchical process breakdown structure of activities very similar to the $IDEF_0$ modeling method, though some exceptions apply. For example, it benefits from the advantages suggested by UML activity diagrams. Sequencing of activities can optionally include *concurrencies* using *synchronization* symbols, and *optional paths* may be included using the control elements (i.e., *decision* symbols).

The method suggested by Streams can be summarized into five steps:

1) business process modeling using UML activity diagrams,
2) identifying of business objects in activity diagrams,
3) identifying of (software) use cases,
4) software process modeling using UML activity diagrams, and
5) identifying and defining of software classes and user interface dialogs.

The method relies heavily on activity diagrams to identify and define business objects, software use cases (i.e., functional view of the supporting system), software classes, and user interface elements. It does not include any business use cases. However, it allows the UML stereotype notation be assigned to most model elements. This brings some flexibility for inclusion of some other concepts into a model and, thus, modification of the suggested method. This research uses two stereotype notations of «*business use case*» and «*software use case*» for use cases to model the functionality of a business system and a software system respectively.

The Streams user interface provides three windows, which can be optionally viewed within the main window of Streams: *Browser* (which displays a hierarchical view of the basic elements of the model, such as packages, activity diagrams, activities, objects, actors, use cases, and classes), *Drawing* (for retrieving and editing of an activity diagram) and *Documentation* (showing the textual description attached to the model element that is selected in the browser or drawing window) (Figure A-12). The information about each model element can be retrieved and edited in the *specification editor* of the element. Figure A-13 shows two specification windows, one for an activity and one for an object.

In summary, this research uses Streams as a CASE tool for the purpose of process modeling and object specification in both business and software contexts in an integrated manner. However, unlike the modeling method suggested by Streams, which is *activity centric*, the methodology of this research is

shaped around *use cases*. In this research an activity diagram is used as a means of realization of a use case. Using Streams, the description of a use case—which is normally described in the *documentation* associated with the use case—is explicitly and graphically represented by an activity diagram. Business requirements are seen as two components: *processes* (i.e., functions) and their related *workflows* (i.e., flow of works and events). The former are captured into some use cases and use case models, and the latter are modeled into activities and activity diagrams, which are associated with use cases. Depending on the desired level of granularity, an activity may include a number of sub-activities.

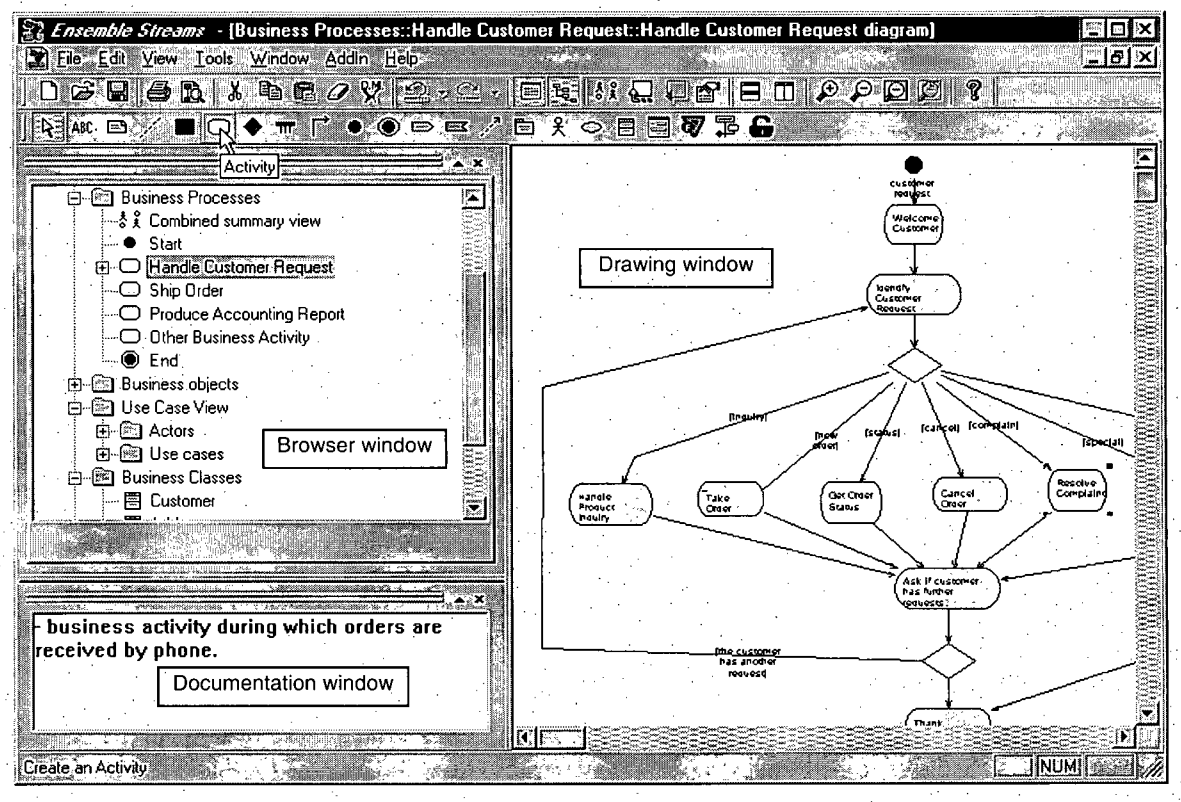


Figure A–12: The Streams User Interface Elements

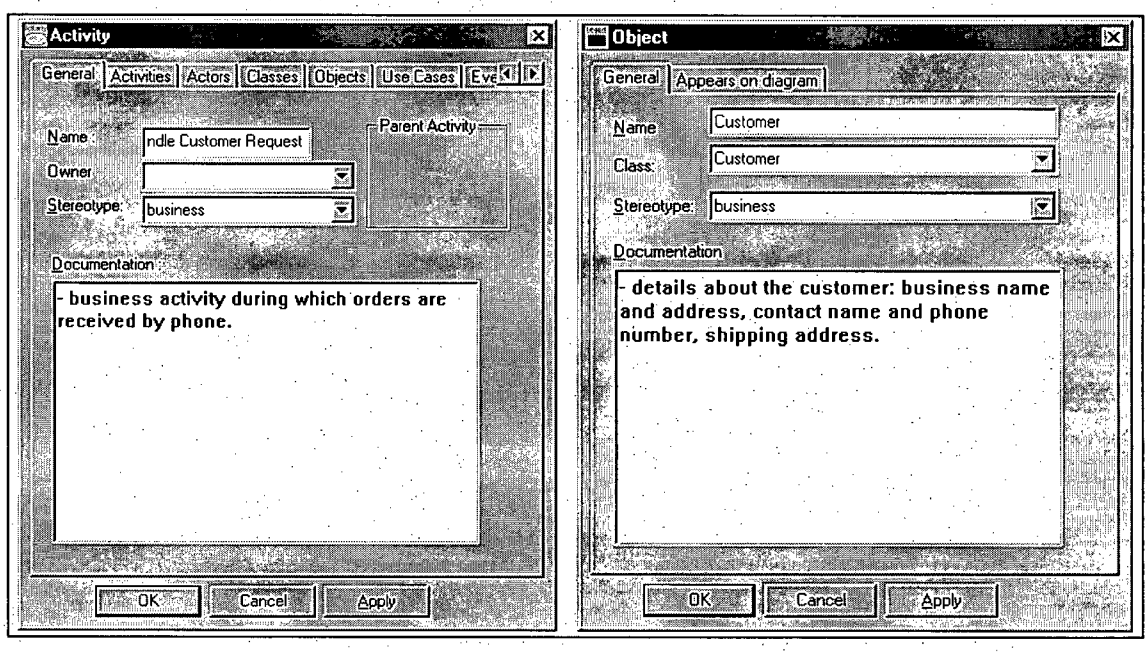


Figure A–13: Two Specification Windows for an Activity and an Object

Appendix B: Selected Commercial PM/CM Software Studied

This appendix presents the attributes of some of the current commercial PM/CM software applications, which were studied as a part of the bottom-up investigations of the research (explained in Chapter 3). Table B-1 communicates this information.

Table B–1: Attributes of Selected Commercial PM/CM Software Applications Studied

(Continued on the next six pages)

Developer	Software Name	Main Functionality	Detailed Functionality & Important Features	Address
ACS Software, Inc.	AutoEDMS	Document Mngt.	Engineering document management and workflow system with several packages (AutoEDMS, AutoEDMS Anywhere, and AutoEDMS Redline) for workgroups and multi-site enterprises; computer-based files, data and programs over Internet or other connections.	acssoftware.com
AEC Software, Inc.	Details	Document Mngt.	Submittal logs (for tracking approval processes), project activities, timeline graphs, contact fields, work calendars, import and export with MS Office tools (Word, Access, and Excel).	aecsoft.com
AEC Software, Inc.	FastTrack Schedule	Project Scheduling	CPM scheduling (time and resources)	aecsoft.com
Architects' First Source, Inc.	BuildSource, BuildSelect, and BuildSpec	Web-based Construction Product Information Center and Product Specification	Used by architects, engineers, interior designers and other construction professionals to find construction products (BuildSource) and get technical information and datasheets (BuildSelect) and prepare specification (BuildSpec); based on the three-part specification of the MasterFormat (general, products, and execution), SelectionFormat, or PageFormat standards; available on the Internet; paper-based information also available.	firstsource.com
Atlantic EC	Atlantic Projections	Resource Scheduling	Scheduling of resources across multiple projects; based on a comprehensive skills database; skills shortages identification.	atlantic-ec.com
Axium	Protrax	Financial Mngt.	With three modules: Protrax, Protrax Plus and Protrax Enterprise; project control, invoicing and accounting with inquiry and report designers.	axiumae.com

Developer	Software Name	Main Functionality	Detailed Functionality & Important Features	Address
Builcore, Inc.	BuildSource, BuildSelect, and BuildSpec	Web-based Construction Product Information Center and Product Specification	The Canadian version of the Architect's First Source, focusing on Canadian construction products and standards, used for product selection and specification; online and on paper.	buildcore.com
Building Cost Information Service Limited	BCIS Online	Online Building Cost Forecasting	Subscription-based website allowing online building construction /refurbishing cost forecasting in the UK, using examples priced in the marketplace, indices, £/m2, Briefing and Dayworks; cost figures for specific time and location factor.	bcis.co.uk
Causeway Technologies	Specnet	Web-based Product Information Center	Online construction products information, including specifications, CAD drawings, and technical data; search, select, and brows information about; order samples, make cost enquiries through email templates. Also serves other industries: mechanical and electrical manufacturing, tourism, real-estates, stock market, etc.	buildingwork.com
Chief Architect & More	National Repair & Remodeling Estimator	Construction Estimating	Pricing for dwelling repairs and remodeling and for high-and low-volume builders; material costs and labor figures based on historical project data; recommends crew sizes, average production rates, exact material, equipment, and labor costs, a total unit cost and a total price including overhead and profit.	qualityplans.com
Colonial Systems, Inc.	Colonial Construction Management	Construction Accounting	Job cost accounting, contract/ subcontract control; financial reporting and information management features; interface with imaging, spreadsheet, word processor, and ODBC.	colsys.com
Comprotex Software, Inc.	Construction By Design Estimating	Construction Cost Control	Construction costs control; integrated with a simple cost-to-complete accounting system for small custom builder; calculates lumber, roofs, sheetrock, foundations, stucco, brick, countertops and other items using spreadsheets.	comprotex.com
D.R.E.C. Inc.	Construction Concepts	Construction Cost Mngt.	Designed specifically for residential to light commercial builders; project cost estimating, planning, and control; purchase orders, completion forms, lien waivers; color charts and terms sheets.	constructionconcepts.net

Developer	Software Name	Main Functionality	Detailed Functionality & Important Features	Address
Framework Technologies Corporation	Active Project	Project Web Sites	Project Web Sites with CAD files, project schedules, spreadsheets, database queries and other documents.	frametech.com
Geac AEC Business Solutions	Construction Manager	Estimating and Accounting	Information and financial control through job costing & accounting; payroll, estimating, purchase orders & inventory.	aec.geac.com
Geac AEC Business Systems	StarBid	Bid Estimating	Evaluate competitive vendor and subcontractor quotes; customizable pre-built RS Means databases; take-off and bill of materials through a CAD interface; interfaces to accounting and job costing; import/export with spreadsheet, word processor or database tables.	construction.geac.com
Geac AEC Business Systems	StarBuilder	Construction Job Cost Accounting (for contractors)	Job costing, general ledger, accounts payable, accounts receivable; employee-vendor-& customer related cost management (payroll, purchase orders, subs, progress/contract billing); project management (job contacts, contracts, & performance), and estimating.	construction.geac.com
Geac AEC Business Systems	StarProject	Project Document and Profile Mngt.	Job contacts, contracts, & performance; project documents like transmittals, submittals, request for information, request for proposal, and change orders; integrates with StarBuilder	construction.geac.com
Geac AEC Business Systems	The Construction Manager (TCM)	Job cost Accounting & Estimating	Accounting & estimating modules: job cost, general ledger, accounts payable, accounts receivable, inventory, payroll, purchase orders and estimating.	construction.geac.com
Innovative Technology, Inc.	NMS/DDN	Products Specification	Canadian National Master Construction Specification (or Devis Directeur National), a large, boiler-plate construction specification covering more than 700 topics in all 16 MasterFormat divisions. It includes eight customized packages of sections for specific disciplines.	innovative.ca
Innovative Technology, Inc.	NMS-Edit	Specification Writing	A specification writing system with specialized word processing, project control, and project verification tools; ideal for use with the NMS (National Master Specification), but may be used with any master specification.	innovative.ca
InterPlan Systems, Inc.	Conceptual Cost Estimator	Estimating	Conceptual estimating based on the study of historical project cost factors for feasibility studies and budgeting; based on the John Page's (The Conceptual Cost Estimating Manual) database used in the USA.	interplansystems.com

Developer	Software Name	Main Functionality	Detailed Functionality & Important Features	Address
InterPlan Systems, Inc.	eTaskMaker	Planning and Scheduling	Creates detailed project plans and exports project data to scheduling tools (Artemis, Primavera Systems, Microsoft, Sciforma and Welcom), and to Microsoft Excel for cost estimating / bidding.	interplansystems.com
Management Information Control Systems, Inc.	Builder Information System (BIS)	Construction Accounting	With 3 editions, 17 modules aimed for contractors; AIA billing, certified payroll reporting, and job costing; Windows based and ODBC compliant.	micsonline.com
McGraw-Hill Companies, Inc.	Sweets Product Marketplace	Web-based Construction Product Information Center and Product Specification	Online searching, selecting, construction products; Online construction products information, including specifications, CAD drawings, and technical data; search, select, and brows product information; order samples, make cost enquiries through email templates; links for online, free preliminary cost estimates; both Web-and PC-based versions available; paper-based also available.	sweets.construction.com
Meridian Project Systems, Inc.	Prolog Application Suit (Prolog Manager, Prolog WebSite, Prolog LT)	Construction Project Document and Portfolio Mngt., Field Administration, and Job Cost Accounting	Intended for large to mid-sized AEC organizations (general contractors, architects, engineering firms and owners); includes three module: Prolog Manager (the core of the system), Prolog WebSite (web-based collaboration of project information), Prolog LT (light version of Prolog Manager for general contractors and subcontractors to manage field activities, costs, and documents); project documentation and information retrieval (e.g., budgets, costs, requests for information, transmittals, drawings, quotes, change orders, purchase orders, bills, and invoices, cash flows); built on Microsoft SQL Server database platform; cost-control module; multiple projects in one database; integrates with Microsoft Word and Project, SureTrak Project Manager, Primavera Project Planner (P3), Welcom Open Plan, and Bentley ProjectWise.	mps-inc.com
Pertmaster Limited	Pertmaster Professional Project Management Software	Planning and Scheduling	Plan, track, and control time, cost, and resources in projects; CPM scheduling, bar charts, precedence networks, resource graphs, and cash flow.	pertmaster.com

Developer	Software Name	Main Functionality	Detailed Functionality & Important Features	Address
Pertmaster Limited	Pertmaster Risk Expert (PRE) & Pertmaster Project Risk (PPR)	Schedule Risk Analysis	Project schedule risk analysis using Monte Carlo techniques; projects created directly in Pertmaster or imported from applications such as Primavera and Microsoft Project.	pertmaster.com
Primavera Systems, Inc.	Primavera Project Planner (P3)	Construction Planning and Scheduling	CPM scheduling of large-scale projects (up to 100,000 activities); time, cost, and resource analyses; 24 activity codes, 16 custom data items, 19 levels of sort, 28 levels of selection criteria, and 31 activity calendars; filtering and sorting of activities, projects, and resources; project files access for multiple users, ODBC-compliant.	primavera.com cpmsolutions.ca
Primavera Systems, Inc.	Primavera Expedition	Contract Mngt. and Administration	Contract control, project management and administration for AEC firms; organize contract changes, drawings, submittals, minutes of meetings, material deliveries, daily reports, correspondence, et.; linkage of drawings, submittals, and daily field activities with P3, P3e, or SureTrak for administrative purposes.	primavera.com cpmsolutions.ca
Primavera Systems, Inc.	Primavera Project Planner for the Enterprise (P3e)	Enterprise Planning and Scheduling	More features than P3; plan, budget, monitor and control multiple projects; multiple baselines and what-if alternatives; cost and schedule performance across multiple programs; web-based team communication, timekeeping, and portfolio management; timesheet interface and progress report; project details captured on handheld devices and uploaded that to the project server using a docking cradle; built on Oracle and Microsoft SQL Server relational databases.	primavera.com cpmsolutions.ca
Primavera Systems, Inc.	Primavera P3e/c (engineering /construction)	Web-based Collaborative Planning and Scheduling	P3e-tailored-to construction; field status and project plan and control of multiple projects through a web browser or mobile reporting tool such as a Palm Pilot or Pocket PC; online collaboration of project team; schedule analysis, cost forecasting; construction images, project templates (based on past projects), and standard reports; multiple rates for equipment, labor, and material costs; integration with P3, SureTrak, Primavera Expedition P3e, and Microsoft Project.	primavera.com cpmsolutions.ca
Primavera Systems, Inc.	SureTrack Project Manager	Planning and Scheduling (time, resource, cost)	CPM scheduling of small to medium-sized projects; web reporting and graphics.	ctnontime.com

Developer	Software Name	Main Functionality	Detailed Functionality & Important Features	Address
TDOC Projects Ltd.	The TDOC System	CM Documents Mngt.	Recording and issuing technical documents (acknowledgement, instructions to contractors, requests for information, proposals for change, etc.); Event Scheduling capability.	tdoc.org.uk
Third Millennium Software, Inc.	iBUILD (Interactive Building)	House Arch. Design, estimating, and bidding	Physical scope definition (architecture); quantity takeoff, and request for proposal preparation (bid scope to vendors).	ibuilt.com Kelwna, BC, Canada
Timberline Software Corp.	Precision Collection	Construction Estimating	An integrated suite of applications for construction estimating; built around the core Precision Estimating, which comes in Standard Edition and Extended Edition versions for conceptual and detailed estimating; includes several optional analysis tools (e.g., Bid Analysis, Buyout, Cut & Fill Products, Digitizer, Explorer, & Unit Price), third-party software interfaces (e.g., to P. Job Cost, AutoCAD, MS Project, & Primavera) and pre-built databases.	timberline.com
Timberline Software Corp.	Gold Collection	Construction Project Accounting	An integrated suite of applications for construction accounting (e.g., job cost and equipment cost modules) and property management (e.g., property management and advanced retail modules); shared modules: accounts payable, accounts receivable & contracts, general ledger, billing, payroll, remote time entry, information assistant, inquiry designer, report designer, and ODBC.	timberline.com
WinEstimator, Inc.	Win*Cost*	Job Cost Accounting	Accounts payable & receivable, general ledger, purchase orders, inventory, payroll, subcontracts, change orders, job cost tracking; integrates with WinEst.	winest.com interwld.com
WinEstimator, Inc.	Win*Est*	Construction Detailed Cost Estimating	Construction detailed cost estimating; a combination spreadsheet and database technologies; scheduling interface, cost accounting interface (Budget Export); digitizer takeoff; integration capability with CAD and accounting systems.	winest.com interwld.com

Appendix C: A Prototype MM Application System

This appendix describes the Material Management System (MMS) Tool, which was developed in a distributed environment and as a part of the bottom-up investigations of this research, mainly for the purpose of exploration and validation of the concept of model-based integrated total PM systems. The results of this activity were used as an input to shaping the research's ideas, frameworks, and models. The following sections provide the background and description of the MMS Tool as well as a discussion of the results of this effort.

C-1 Introduction

The emergence of iterative and spiral life cycle models and prototyping paradigms in the field of software engineering [Wood and Kang 1992] may be considered as a response to two major characteristics of software development projects. First, in contrast with physical product development (e.g., buildings), which is usually managed in a sequential fashion, the process of software development is generally *evolutionary* and *incremental* in nature (section 2.1.3). Second, like many other types of projects, requirements specification has been recognized as the most difficult and crucial part of the software development process that has great potentials for savings (e.g., by early validation of functional and information requirements) [Sommerville 1997].

The Unified Process [Jacobson et al. 1999], for instance, is suggested over traditional methods (e.g., different variations of the waterfall method [Beynon-Davies 1993]), which support discrete phases of software development process, to integrate the whole process in an iterative manner. Varying in purposes and approaches, however, software prototyping paradigms generally aim at reducing the failure risks associated with the process through early requirements *validation* [Wood and Kang 1992]. The approach of this research to prototyping may fall under "*concept prototyping*", which focuses on a *specific development stage or part of a system* with the purpose of "validation prior to commitment" [Wood and Kang 1992, p. 8]. The research attempted prototyping as a means of validation of the concepts, ideas, and models suggested by the research, rather than validating the requirements of a software system aimed for commercial production. More specifically, the research used prototyping as an approach to exploration of *model-based integrated total PM systems*, which are based on a *unified project information model*, and requirements of such systems and their underlying models.

This appendix describes a prototype application system, called Material Management System Tool (referred to as *the prototype system* or *MMS Tool* herein), which was developed in a distributed environment. The prototype system was developed as a component of a larger system (i.e., Jigsaw Distributed System, JDS, version 0.6), which facilitates a wide range of data exchanges and software

interoperability through provision of a set of software components [JDS 2002; Hassanain 2002]. The following sections provide a description of the goal, objectives, and methodology adopted in the development of the prototype, a description of the JDS, the functionality, underlying process and data models, and user interfaces of the MMS Tool, a discussion of the results, and a conclusion of the materials presented in this appendix.

C-2 The Goal and Objectives

The main goal of the prototyping was to provide some valid grounds for the research through software implementation; i.e., supporting the research with a bottom-up investigation of model-based integrated total PM systems. Within this goal, two basic objectives were assumed for the prototype:

- To examine the IFC's data exchange capabilities for supporting MM functions in a distributed integrated PM systems environment, and
- To use the lessons learned from the prototype development as input materials for developing the research deliverables; i.e., a bottom-up validation approach.

C-3 Assumptions and the Methodology

This section describes the hypothetical user and functionality requirements of the prototype system as well as the approach to its development.

C-3.1 The User and Required Functionality of the System

The MM staff of a hypothetical (ABC) construction company was assumed to be the user of the MMS Tool. In the long-term plans of the company, the adoption of information technology has received a great attention. The company has envisioned a fully integrated computerized information system, which could support a variety of PM functions sharing and exchanging their information. Nevertheless, in order to safeguard risks, it has decided to take an *incremental* approach to the development of the system.

As a first step, due to the importance of MM processes in its profile, the company has required a *proof-of-concept* of such a system providing some basic functionality; i.e., a system that could later be extended to include a variety of MM functions and interact with other PM systems such as financial management, quality management, and cost management. The basic functionality required for the system is as follows:

1) a database holding information about various types of materials used in a building construction project and possibility of querying and updating the information;
2) interfaces for defining a new project together with its related building elements;

3) opening an existing project (created by this or other application systems);
4) interfaces for selecting material types from the database and associating them with project elements such as building elements and construction tasks;
5) interfaces for data querying on the whole project model or specific pieces of information; and
6) interfaces for saving (or re-saving) the project data for possible use by this or other application systems (e.g., scheduling).

C-3.2 Out-of-Scope and Future Plans (Extensions of the System)

Any other functionality than those listed as required functionality (previous section) is considered as out-of-scope. The system is not required to support versioning and change management (e.g., change tracking and notification and access control) in a multi-system environment. However, the system was required to be examined for possibility of its extension to support some other MM processes, such as request handling, purchasing, receiving, storing, and inventorying of materials. Considering the long-term strategic IT plans of the company, the examination of possibility of integration of the MMS Tool with other PM systems (such as asset management, financial management, quality management, and cost management) around a unified project model is also encouraged.

C-3.3 The Approach to the Development of the System (The Platform and Tools)

The company's requirements generally suggested an *evolutionary prototyping* approach to the development of the integrated PM systems; nevertheless, considering the goal and objectives of this part of the research and the scope of the MMS Tool, the development of the system was found to fall more under *concept prototyping* [Wood and Kang 1992, p. 8]. Moreover, in order to demonstrate the potentials of integrated, distributed PM systems, the JDS version 0.6 [JDS 2002] and the IFC R2x object model [IFC R2x, 2000a] (section 2.4.4.7.4) were chosen as the platform and the underlying data model of the system respectively. The Microsoft Visual Basic™ 6.0 (MS-VB) and Microsoft Access™ 2000 (MS-Access) were also used as implementation tools for development of interfaces and materials database correspondingly. The following section presents a description of the JDS and the underlying architecture of the prototype, while the MMS Tool itself is described next.

C-4 The Background (The Architecture and The Platform)

This section describes the platform (i.e., the JDS) and the architecture on which the MMS Tool was built.

C-4.1 The Reference Architecture for AEC/FM Distributed Systems

Elaborating on the need for distributed AEC/FM systems that support multi-modal exchange of data (i.e., a variety of data exchange paradigms; e.g., file-based exchange versus server-based database) and transaction-based services (i.e., message exchange paradigms; e.g., e-commerce services), Froese et al. [2000a and 2000b] suggest a reference (or typical) architecture for distributed, model-based, integrated AEC/FM systems. The architecture includes three major layers, which incorporate a typical set of components that can be configured to fit various modes of data exchange needed to support AEC/FM processes in a distributed environment (Figure C-1):

1) *Application (or Presentation) Layer*: consists of commercial or custom *applications* (synchronized to a shared project model) and application-specific *adaptors*. An adaptor may be encoded or be used as add-on or macro within an application, or be implemented as a separate program interacting with the application through the Application Programming Interfaces (*API*). It maps project data (generated by an application) into *shared project data* within the integrated system through the middleware layer.
2) *Middleware (or Business Objects) Layer*: consists of *local model proxies* (i.e., local copies of the shared project data model), which make the *data* and *services* (e.g., from a shared database and distributed systems) available to local applications, and *business objects* (which implement *business logic* processes common to all the applications using the data). Business objects components can distinctively reside on servers. They can also be implemented within other components (e.g., local model proxies).
3) *Data Layer*: consists of the *data server* component and the *project data model* (i.e., the common, physical project data, e.g., in the form of XML files or relational databases residing in a local network or on the Internet). The data server handles the persistence of common project data model.

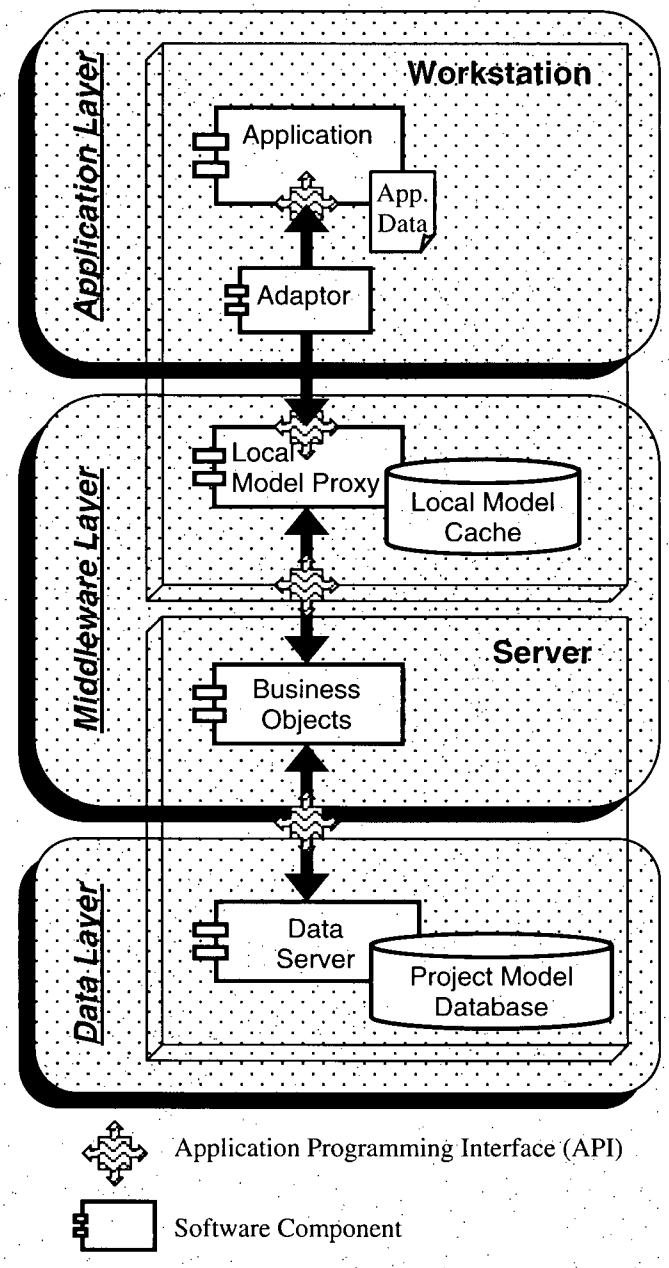


Figure C–1: A Reference Architecture for Distributed, Model-Based, Integrated Systems

[Based on Froese et al. 2000b]

C-4.2 The Jigsaw Distributed System (JDS)

The Jigsaw Distributed System (JDS, or simply called Jigsaw herein) has been developed in Construction Engineering and Management Group at the University of British Columbia, based on the reference architecture explained in previous section. The JDS aims at integration of PM tools (e.g., estimating, scheduling, MM, asset management, etc.) through a common project model (section 1.2.2.1), which is based on the standard information objects defined in the IFC's. Using the IFC R2x schema [IFC R2x, 2000a] (section 2.4.4.7.4) as a "*vocabulary*" for communicating AEC/FM project information among the PM tools, the JDS version 0.6 interface supports project data queries and encodes IFC-based data in various formats such as XML file, MS-Project file, MicroROOFER database, or BLIS file.

Figure C-2 shows the major components developed in the JDS version 0.6. These components are grouped into three categories (represented with a UML stereotype notation, Appendix A): *data client*, *data server*, and *business objects*. The *data client* components (i.e., application tools) include:

- ***JsInfoBrowser***: general browsing of Jigsaw data and linking to external information, and
- ***JsCAD***: CAD-based access to product and process information [Halfawy and Froese 2003],
- ***PECAD***: cost estimating through Timberline™ Estimating Software [Yu 2002],
- ***JsCost***: simple costing,
- ***AMS Tool***: an asset management tool [Hassanain 2002], and
- ***MMS Tool***: a materials management tool (described later in more details).

The Jigsaw's *data-server* components, which provide interfaces for importing and exporting data records from and to their corresponding applications, are as follows:

- ***JsXml*** (JsXmlFile06): reads and writes XML Files.
- ***JsBlis*** (JsBlisDS06): maps project data to IFC 2.0 /BLIS files [BLIS 2002].
- ***JsRoofer*** (JsRooferDS06): maps project data to MicroRoofer™ database.
- ***JsMSProject*** (JsMSProjectDS06): maps project data to Microsoft Project™ (MS Project) file (i.e., tasks information, such as task name, duration, start date, finish date, predecessor and successor tasks) using MS Project's interfaces.
- ***JsWeb*** (JsWebDS06): provides secure access to remote data source over World Wide Web.

Another component, which plays a central role in the JDS, is the ***Jigsaw Application Objects*** (JsAppObjs06). This component, which is implemented as a DLL (Dynamic Link Library) file, includes the definition of business objects (i.e., the "*unified project information model*", section 1.2.2.1) shared among the communicating applications. The objects are a subset of the IFC R2x model. Figure C-2 shows

how, in the JDS version 0.6, various applications (i.e., data clients) communicate through the interfaces provided by this component and data server components to access the *project data*, which may be encoded in various formats such as XML file, MS-Project file, MicroROOFER database, or BLIS file.

As a *data client* component of the JDS, an application initiates data exchange by calling Jigsaw's standard interface through its specific (built-in or external) adaptor, and Jigsaw provides access to the data source (through *data server* components). The Jigsaw's interfaces provide a mechanism that individual data clients do not need to know the details of the individual data servers and vise-versa.

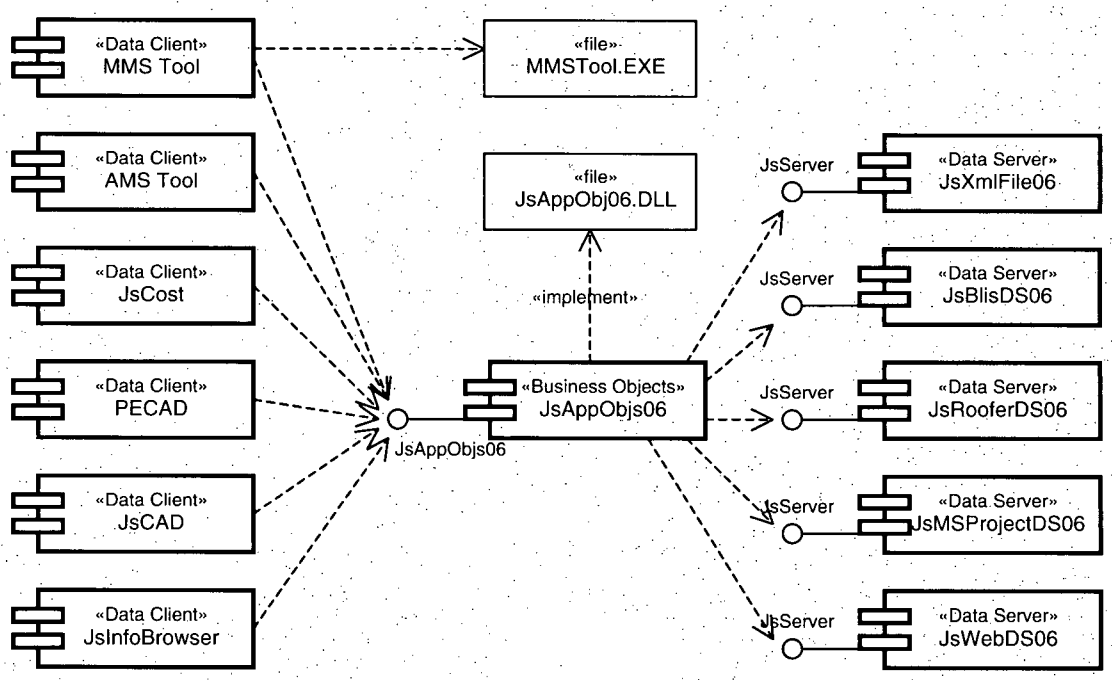


Figure C–2: The Major Components of the JDS version 0.6

C-4.3 The Jigsaw Modeling Tool (JMT)

The Jigsaw Modeling Tool (JMT) is a utility that was developed as a part of the JDS's development process. Using Jigsaw's data server components, the JMT provides interfaces for importing and editing object models or schemas (representing the *business objects* component of the JDS described above) and saving them into their original format or exporting them from one format to another. A specific application of the JMT is described in the next section where the MMS Tool is explained.

C-5 The Prototype Materials Management System (MMS) Tool

The prototype MMS Tool is a custom application implemented as one of the collaborating applications (i.e., a data client, in the application layer) of the JDS version 0.6. Using the IFC R2x schema

[IFC R2x, 2000a] (for business objects definitions) and JDS's components (for accessing the project data and communicating with other JDS applications, explained earlier), the MMS Tool provides interfaces for performing such functionality as creating a new project, opening an existing project (defined by the MMS or any other Jigsaw's collaborating tools), defining project elements—such as products (without geometry), construction tasks, and materials-related information and transactions (e.g., required materials, materials request, and purchase orders)—and their relationships (e.g., assigning materials to tasks), performing data queries (the whole model or specific pieces of information), and saving the project as an XML file. The following describes the various aspects of the MMS Tool.

C-5.1 Creating the "Jigsaw Application Objects" Component

Using UML activity-diagram notations (Appendix A), Figure C-3 illustrates the workflow involved in creating the *Jigsaw Application Objects* component, which contains the business objects definitions commonly shared among various Jigsaw data clients (i.e., applications, including the MMS Tool). The same procedure can also be used for creation of the business objects that are specific (i.e., internal) to only one application system; i.e., only the system would include these objects. Using the JMT's main interface (Figure C-4), the common object model (i.e., a subset of the IFC R2x schema) is edited and exported to a VB project (Figure C-5), which is then is compiled into an ActiveX component to be commonly used by the applications for communication of project data.

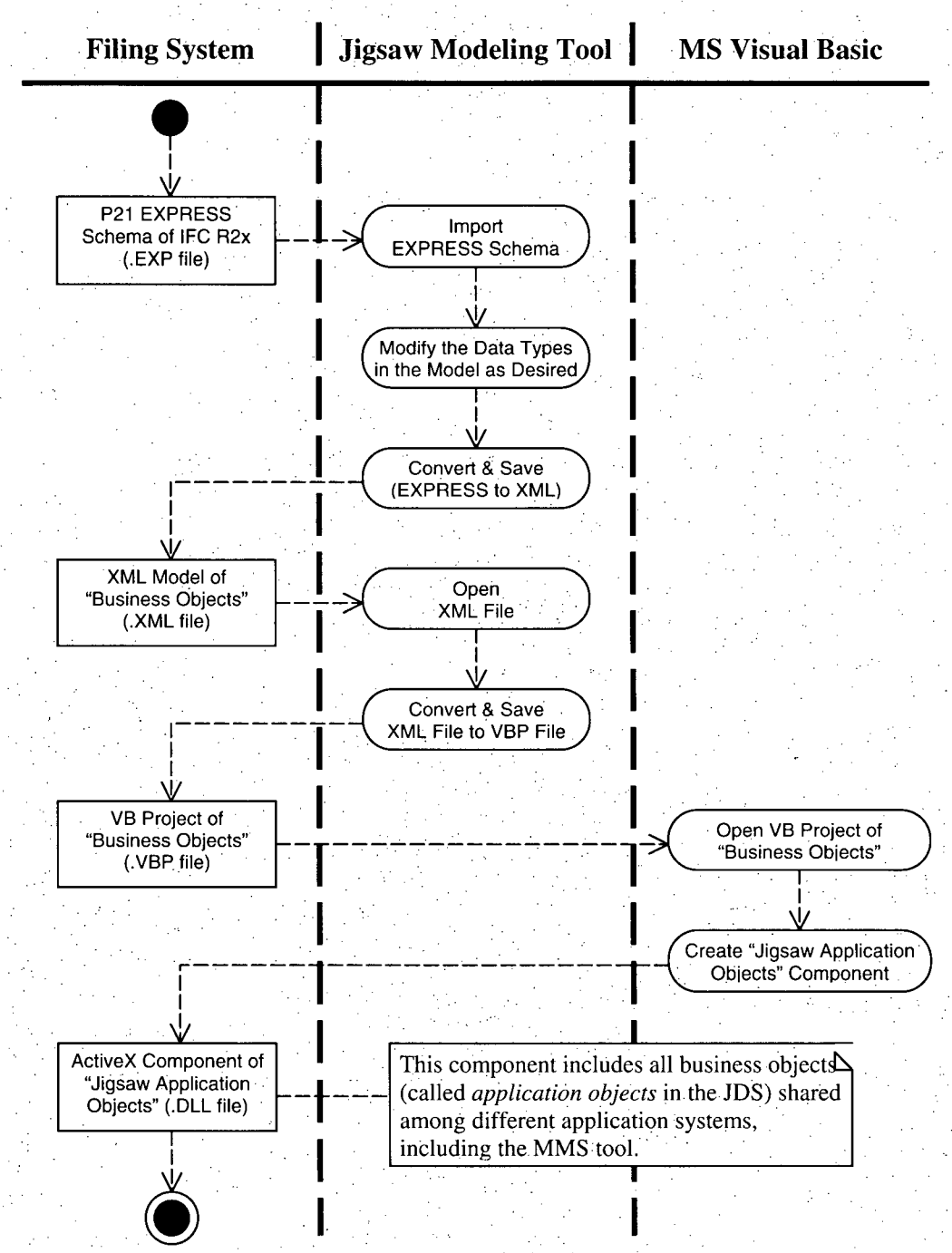


Figure C–3: The Workflow Involved in Creating the "Jigsaw Application Objects" Component (Using the JMT and MS-VB)

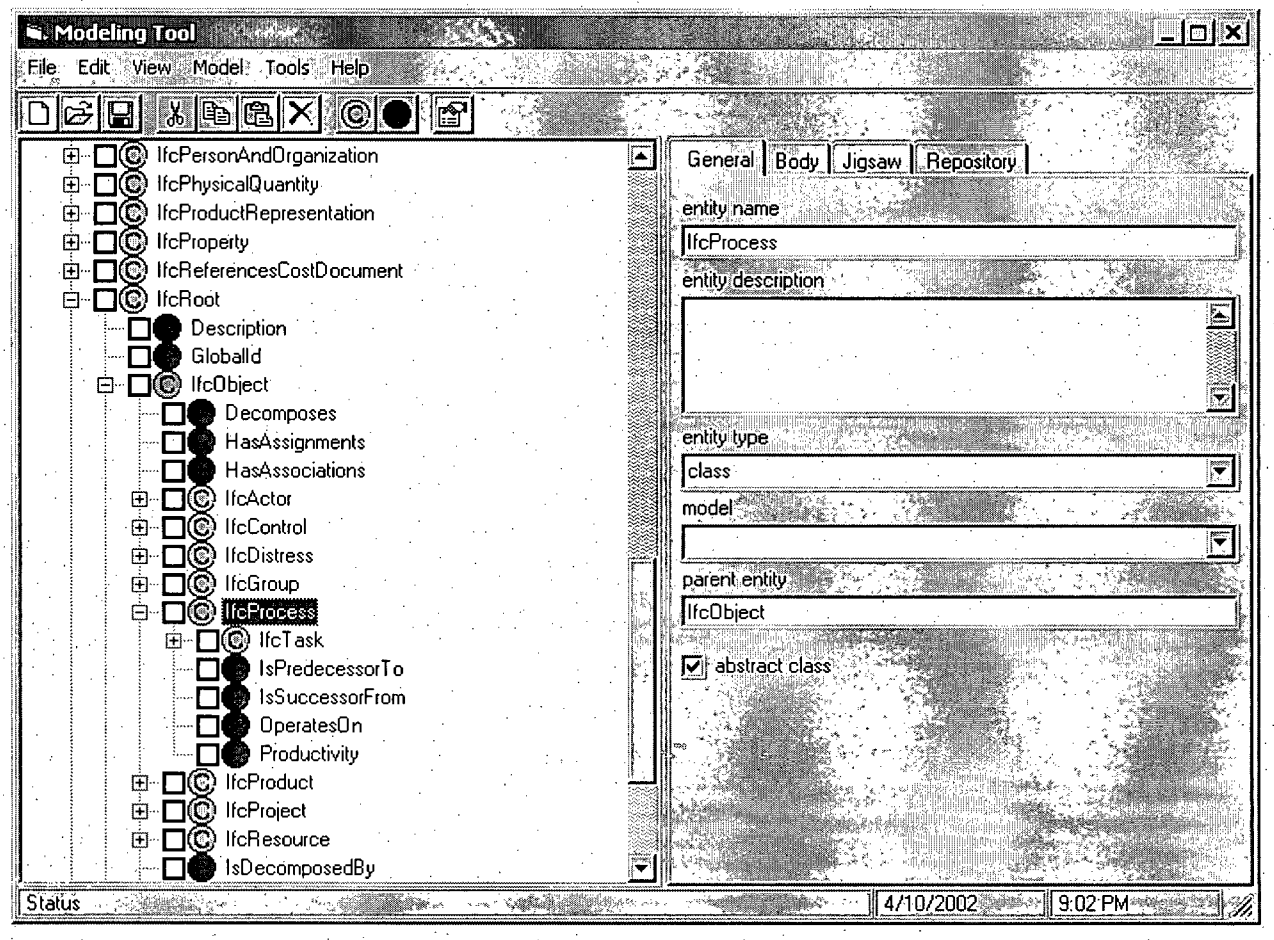


Figure C–4: The Main JMT Interface

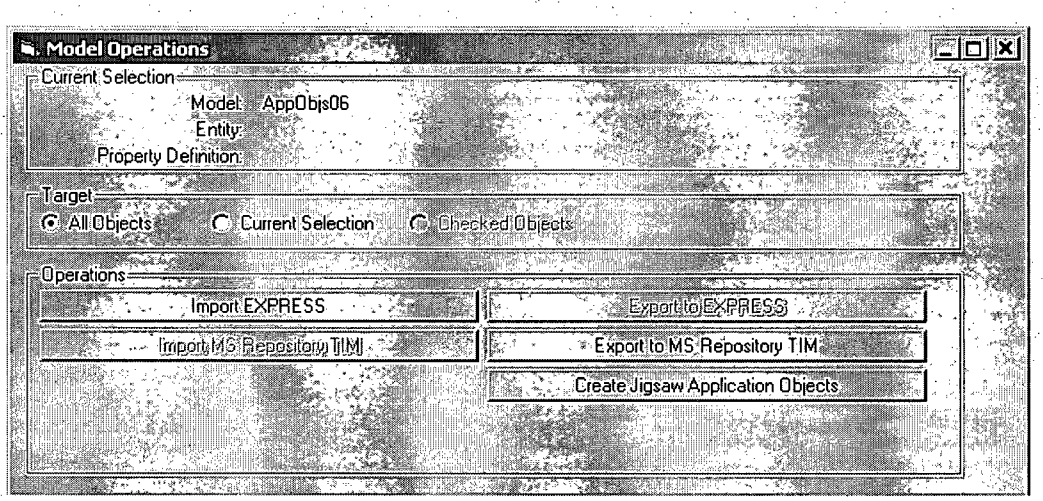


Figure C–5: The JMT's Import/Export Interface

C-5.2 The Materials Database

As a part of the MMS Tool development process, the MS-Access 2000 was used for developing a relational database [Kendall and Kendall 1995], which includes materials information (including cost data). Figure C-6 shows the database structure (i.e., relationships between tables) for the part considered in the current version of the MMS Tool. This structure is based on the format suggested by the Means Construction Cost Data [Means 1994], which has adopted the CSI (Construction Specifications Institute) MASTERFORMAT system of classification and numbering. The database was populated mainly with a portion of the year 1994 Means data. Figure C-7 shows a partial view of the populated database tables. This data was later used for definition of material types in the MMS Tool.

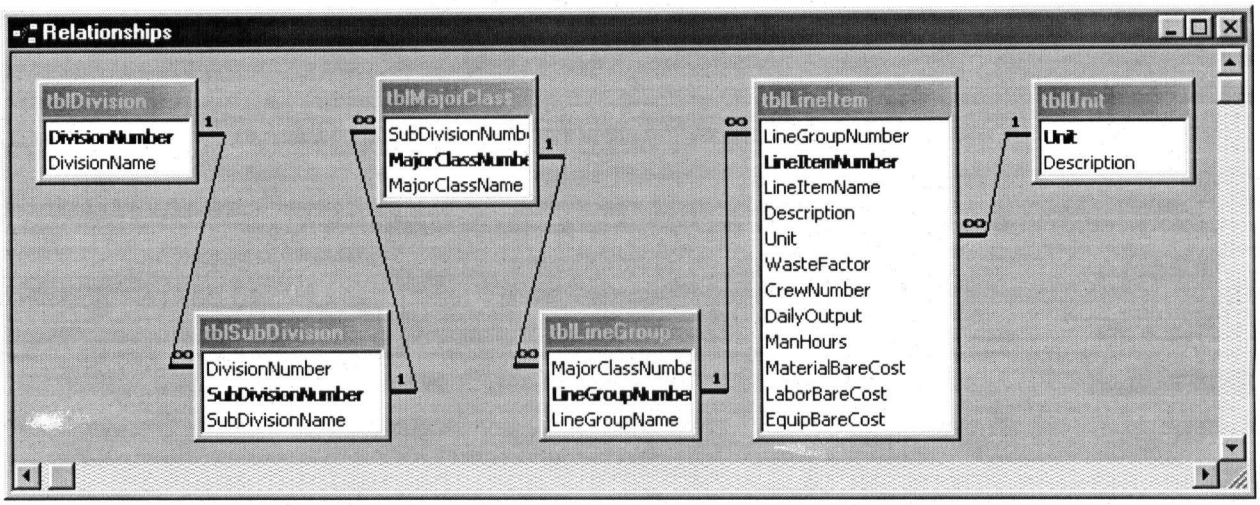
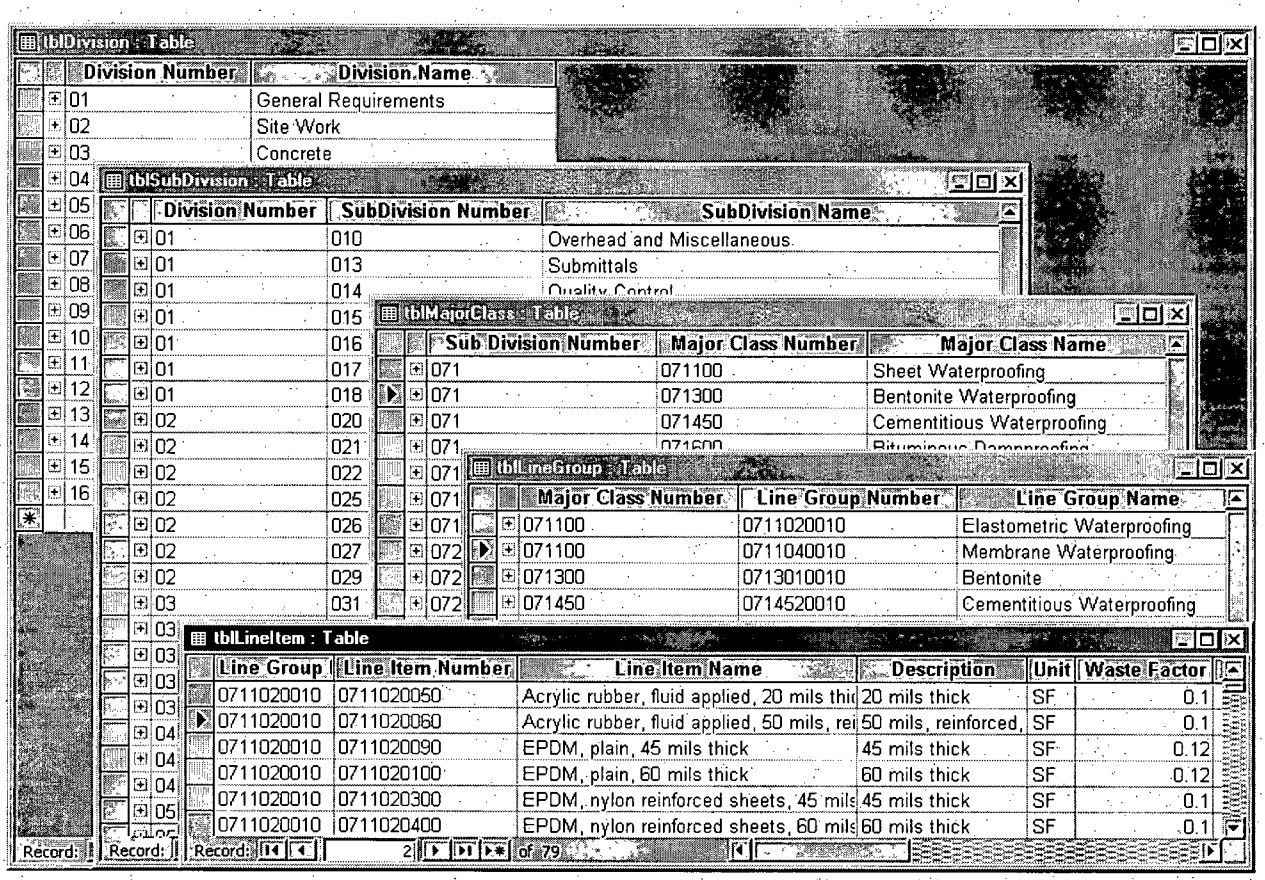


Figure C–6: Relationships Between Tables



tblDivision : Table

Division Number	Division Name
01	General Requirements
02	Site Work
03	Concrete
04	
05	
06	
07	
08	
09	
10	
11	
12	
13	
14	
15	
16	

tblSubDivision : Table

Division Number	SubDivision Number	SubDivision Name
01	010	Overhead and Miscellaneous
01	013	Submittals
01	014	Quality Control
01	015	
01	016	
01	017	
01	018	
02	020	
02	021	
02	022	
02	025	
02	026	
02	027	
02	029	
03	031	
03		
03		
03		
03		
04		
04		
04		
05		

tblMajorClass : Table

Sub Division Number	Major Class Number	Major Class Name
071	071100	Sheet Waterproofing
071	071300	Bentonite Waterproofing
071	071450	Cementitious Waterproofing
071	071600	Bituminous Dampproofing
071		
071		
071		
071		
072		
072		
072		

tblLineGroup : Table

Major Class Number	Line Group Number	Line Group Name
071100	0711020010	Elastometric Waterproofing
071100	0711040010	Membrane Waterproofing
071300	0713010010	Bentonite
071450	0714520010	Cementitious Waterproofing

tblLineItem : Table

Line Group	Line Item Number	Line Item Name	Description	Unit	Waste Factor
0711020010	0711020050	Acrylic rubber, fluid applied, 20 mils thi	20 mils thick	SF	0.1
0711020010	0711020060	Acrylic rubber, fluid applied, 50 mils, rei	50 mils, reinforced,	SF	0.1
0711020010	0711020090	EPDM, plain, 45 mils thick	45 mils thick	SF	0.12
0711020010	0711020100	EPDM, plain, 60 mils thick	60 mils thick	SF	0.12
0711020010	0711020300	EPDM, nylon reinforced sheets, 45 mils	45 mils thick	SF	0.1
0711020010	0711020400	EPDM, nylon reinforced sheets, 60 mils	60 mils thick	SF	0.1

Record: 2 of 79

Figure C–7: The Database Tables Populated with Cost Data

C-5.3 A Demonstration of Selected Features of the MMS Tool

Figures C-9 through C-21 present several screen captures of the MMS Tool intended to describe several selected tool's features aimed to respond to the requirements of the prototype. However, in order to better highlight the features, the example project used for demonstration was considered to be simple and include a relatively small amount of data. The presented figures collectively illustrate the sequence of actions involved in loading and reviewing the *shared project data* (XML file), adding materials and procurement task data to the project, assigning materials to tasks, and saving the information for further use by this or other Jigsaw applications.

C-5.3.1 The User Interface

A Multiple-Document Interface (MDI) forms style was used in development of user interface of the MMS Tool. The main MDI form allows the user to display multiple forms as its children (inside the main form) at the same time. Each child form is displayed in its own window. Browsing and editing facilities are provided through a typical tree-view (explorer-style) interface, menus, and toolbar icons.

The MMS Tool provides a flexible tree-view interface for viewing and manipulating project information. Double-clicking on a node for reviewing or editing its associated information and drag-and-drop operations for adding and associating project model elements are examples of the features supported by this interface. Browsing project data in multiple tree-view windows is supported for simultaneous browsing of project information from different views. Multiple tree-view windows are synchronized to the same single project data; i.e., changes made to a model element (e.g., a task or material) results in refreshing all windows (Figure C-8). In addition to the facilities provided through the tree-view window, menus and toolbar icons also facilitate creating new objects and retrieving and editing objects data.

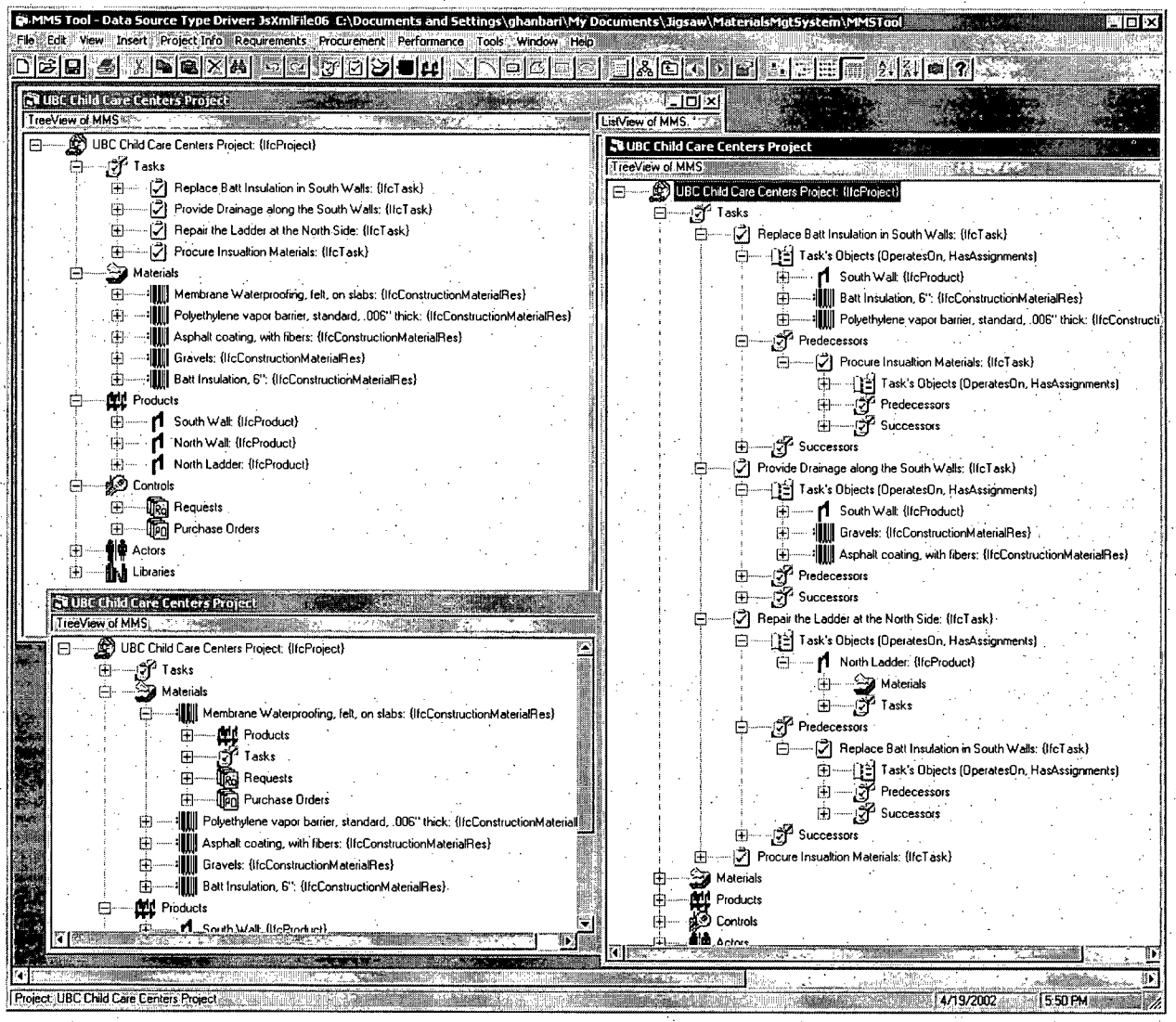


Figure C–8: Multiple Tree Views for Simultaneous Reviewing and Editing of Project Data

C-5.3.2 Creating and Opening/Importing a Project

A project can be defined from scratch, as an existing project can be opened. Figure C-9 shows (left side) the initial setup of the "*File*" menu, which provides for basic functions such as setting *connection type* (i.e., data source type, to use one of the data-server components, explained earlier, for accessing project data), defining a new project, and opening an existing project. The "*Save*", "*Save As*", and "*Close*" menus will be enabled after a project is opened. Selecting the "*Set Connection Type*" command displays the "*Connection Information Dialog*" window (right side of Figure C-9), which allows the user select one of the four data source types supported by the JDS version 0.6 (i.e., JsXmlFile06, JsBlisDS06, JsRooferDS06, and JsMSProjDS06; explained above). In the importing process, only that part of information is imported that is supported by the data source (i.e., by the schema of the application into which the data is imported), while unsupported information is simply ignored.

After choosing JsXmlFile06, which is the primary component that is used by the MMS Tool for reading and writing IFC-based, XML-format project files, the "*Open*" command of the "*File*" menu can be executed for opening an existing *shared project data* file. This command displays the "*Open*" interface (Figure C-10) through which a project file is selected and loaded. In importing a project, only the information that is supported through the application's schema is imported, while unsupported information is simply ignored. Consequently, a subset of the project data that is of interest to the MMS Tool is retrieved in a tree-view window (Figure C-11), which is one of the key interfaces used in the tool for browsing and manipulating the project data.

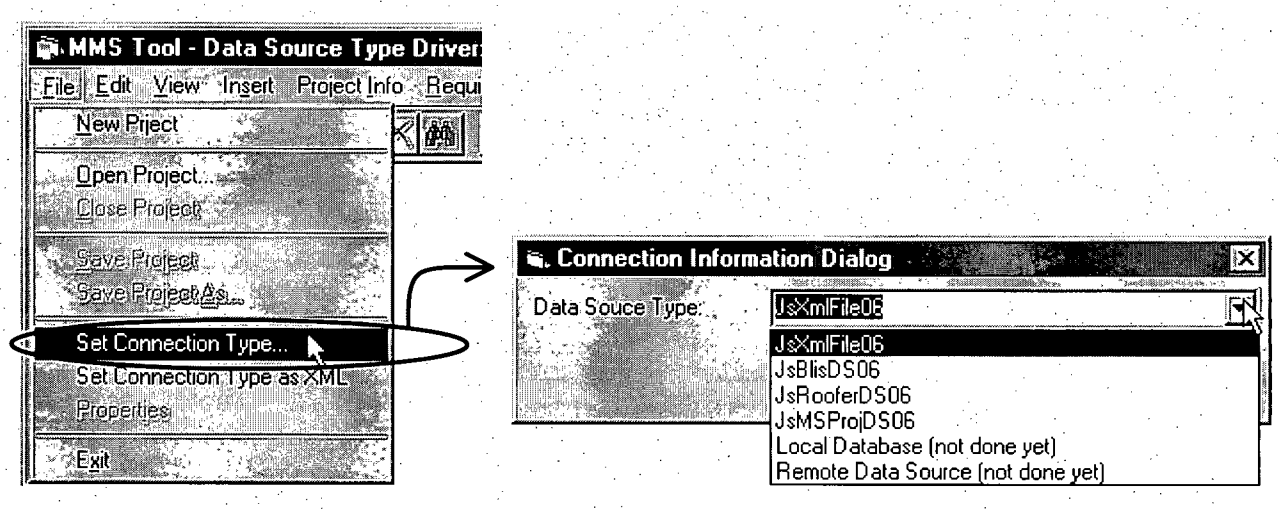


Figure C–9: The Intitial Setting of Connection Type to XML Data Source

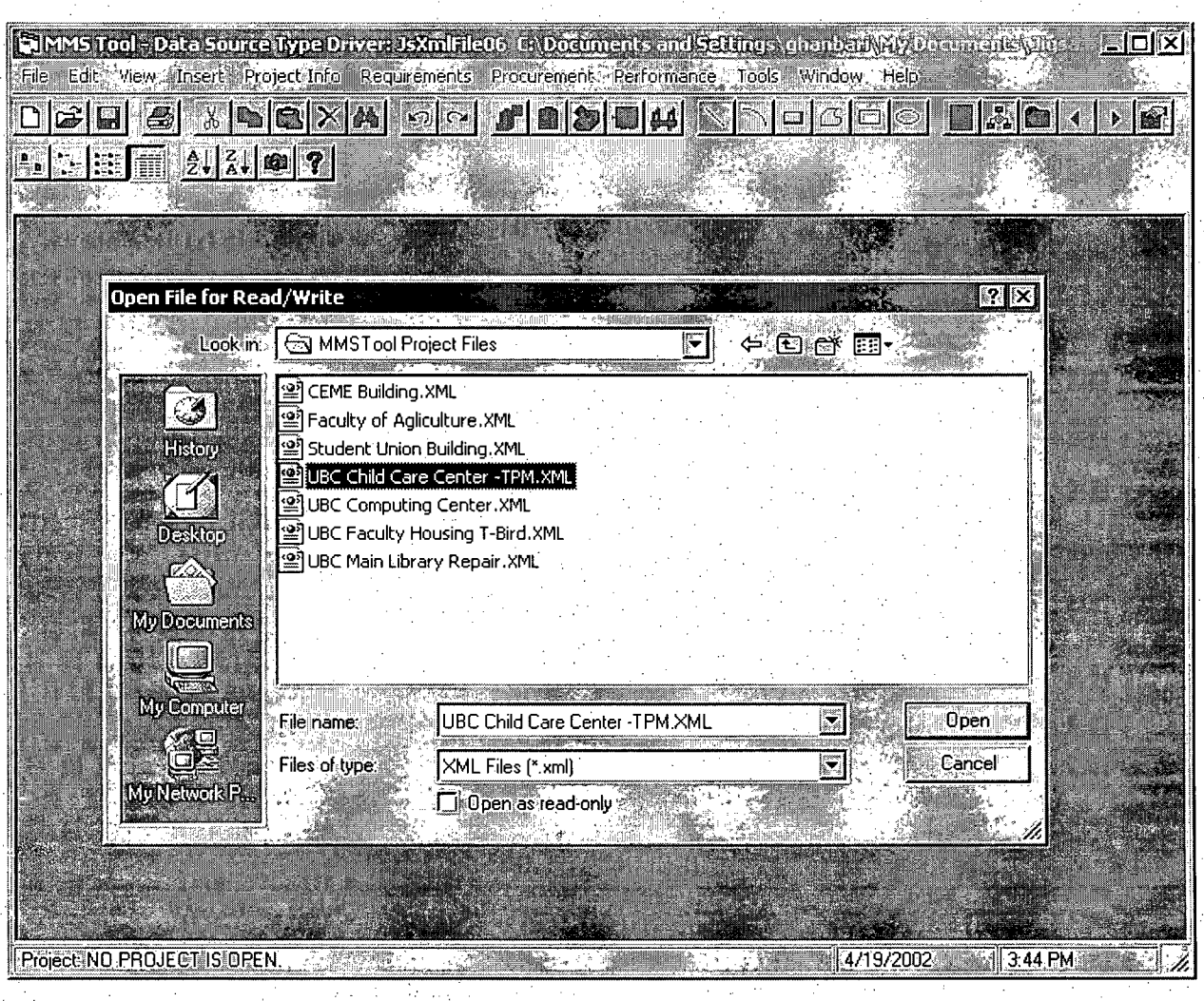


Figure C–10: Opening an Existing Project File (XML Format)

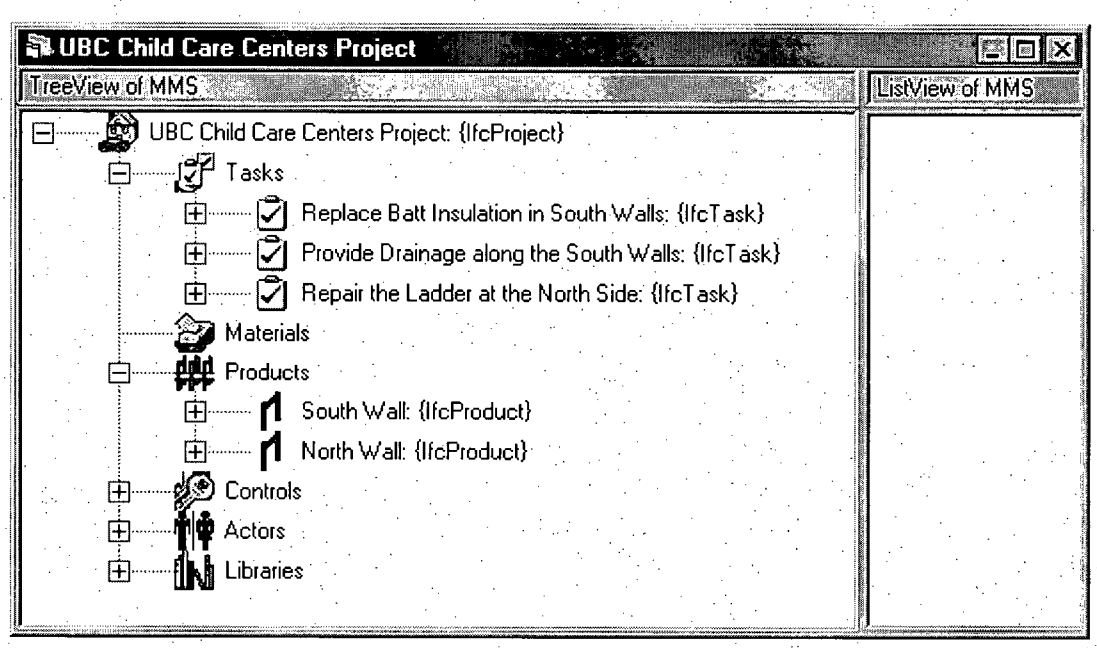


Figure C–11: The Project Information Displayed in the Tree-View Window

C-5.3.3 Editing General Project Information

Figure C-12 illustrates the interface used for editing the general project information. This interface can be viewed using the "*Project Info*" menu or by double-clicking the project icon at the highest level of the tree-view hierarchy. In the current version, the four text-boxes in the lower left corner of the form are enabled for data entry; however, other parts of the form is intended for further development to be used for retrieving information about other projects as well (though editing would still be allowed only for the current loaded project).

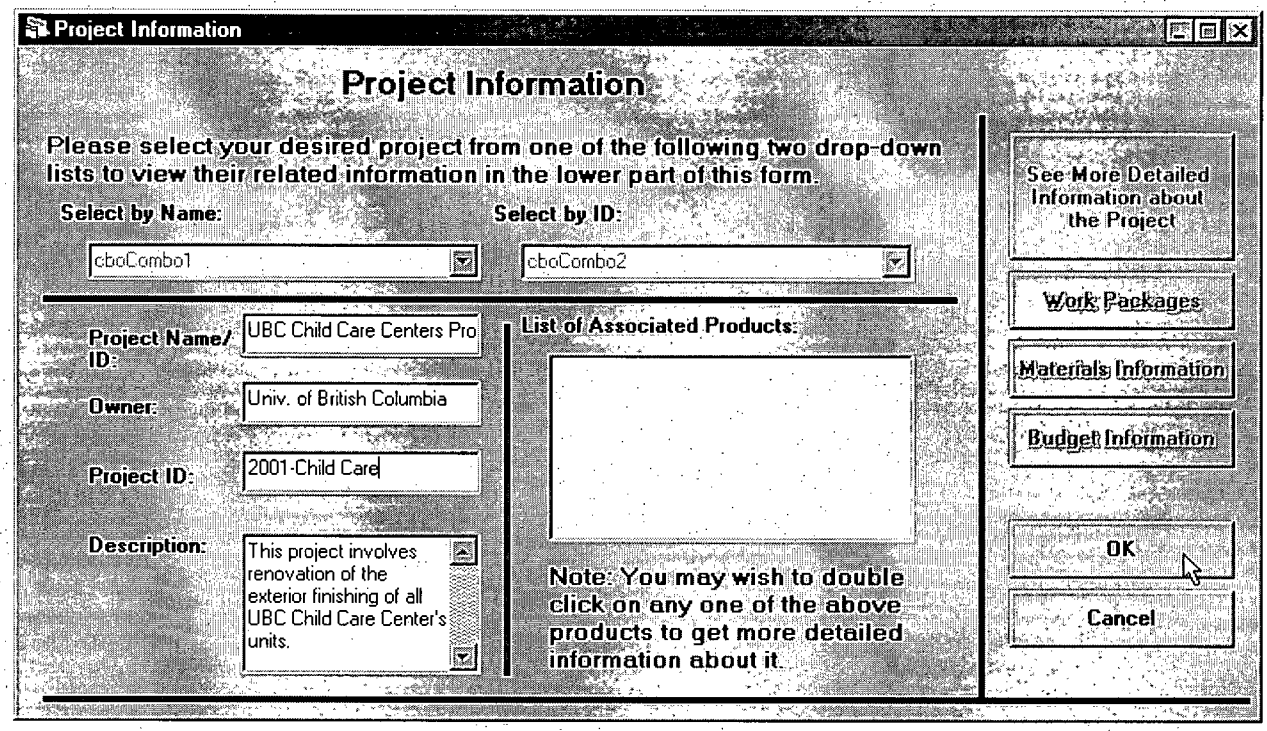


Figure C–12: Viewing and/or Editing Grneral Project Information

C-5.3.4 Adding and Editing Project Elements

New project elements (e.g., products, tasks, and materials) can be added to the project using menu items or the toolbar icons. Figure C-13 illustrates the procedure involved in adding a material type (represented with the *IfcConstructionMaterialResource* object) to the project. By double-clicking the node representing the new added material in the tree view, the "*Project Material Information*" form is displayed. Clicking on the command button next to the Material Category displays the "*Material Selection from Database*" form. This form allows selecting a material type from he materials database using the form's combo-boxes. Confirming the selection (clicking on the OK button) results in updating the earlier form with detailed information about the selected material (e.g., unit of measure, unit cost, and waste factor; Figure C-14). The text box and the list boxes in the lower part of the form are placeholders for further development of the prototype to retrieve other information related to the material. After entering the required date for the material, the final confirmation can be expressed by clicking the OK button. This results in updating the project (in the memory) and the tree view with the added information (Figure C-15).

A similar procedure can be followed to add a new task (i.e., a project activity, an instance of the *IfcTask* object). Figure C-15 shows the interface used for defining a new (or modifying an existing task) task; i.e., "*Procure Insulation Materials*" in this example. At present, the "*Task Information*" form allows

entering data such (task ID, type, name, description, status, and start and finish dates). The other parts (i.e., list-boxes) are intended for future development to list (read-only) the task's predecessor and successor tasks and required materials.

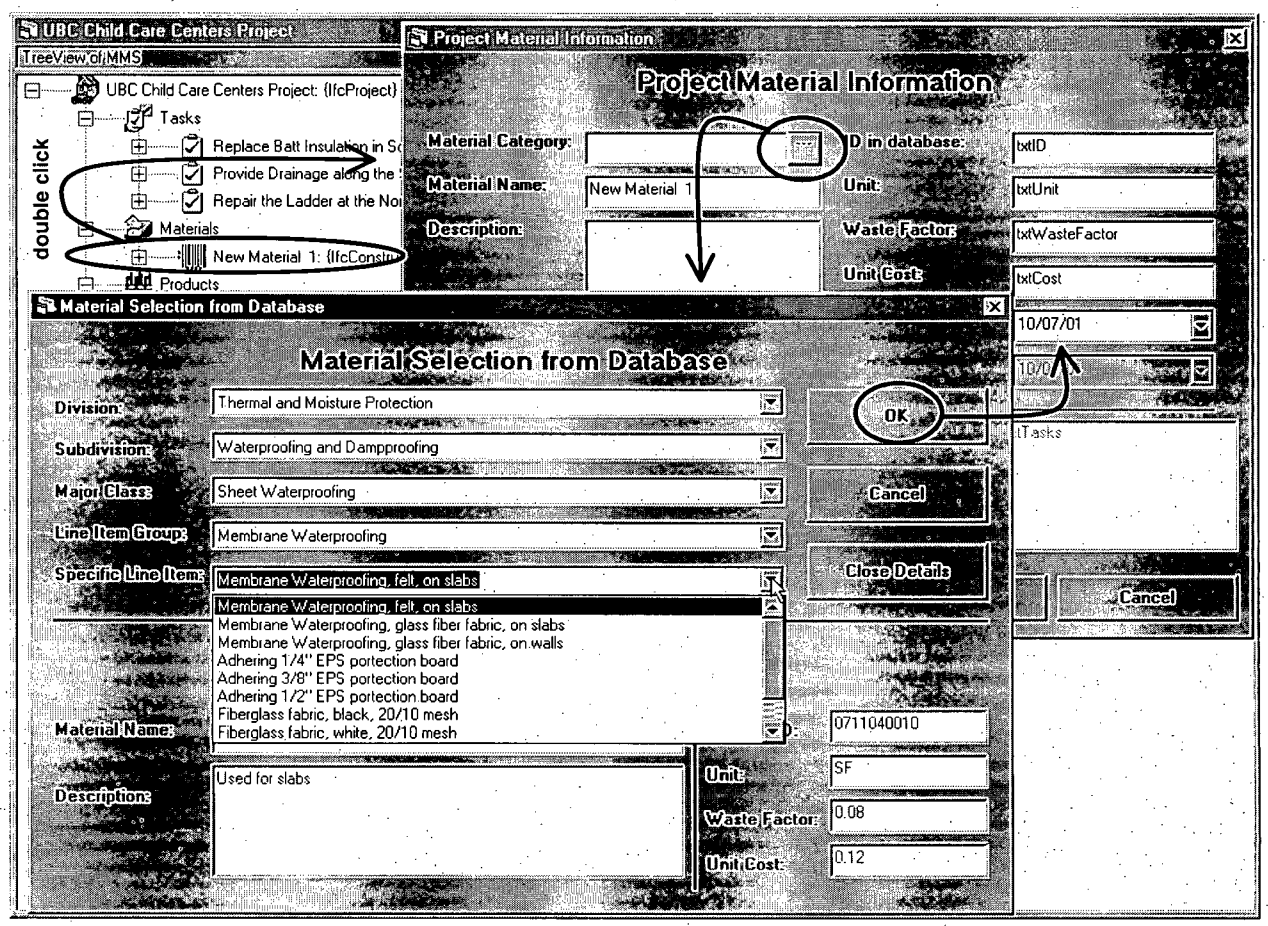


Figure C–13: Selecting and Adding a Type of Material from the Materials Database

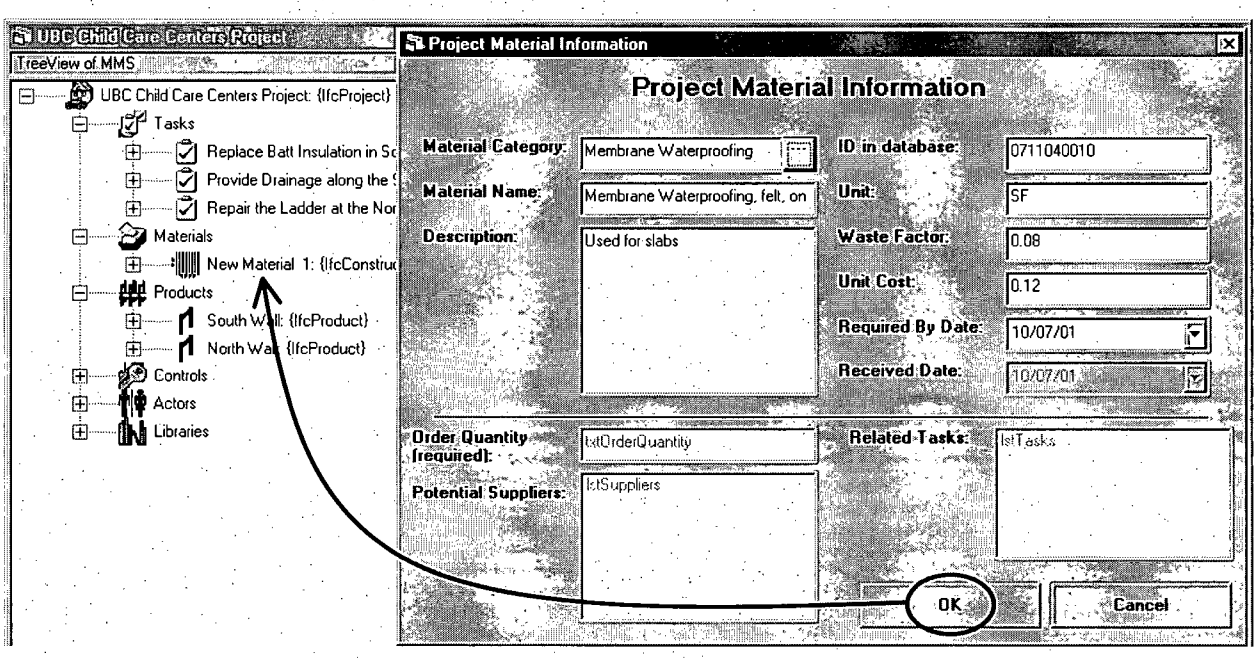


Figure C–14: Project Material Information Form Updated with Detailed Information from the Materials Database

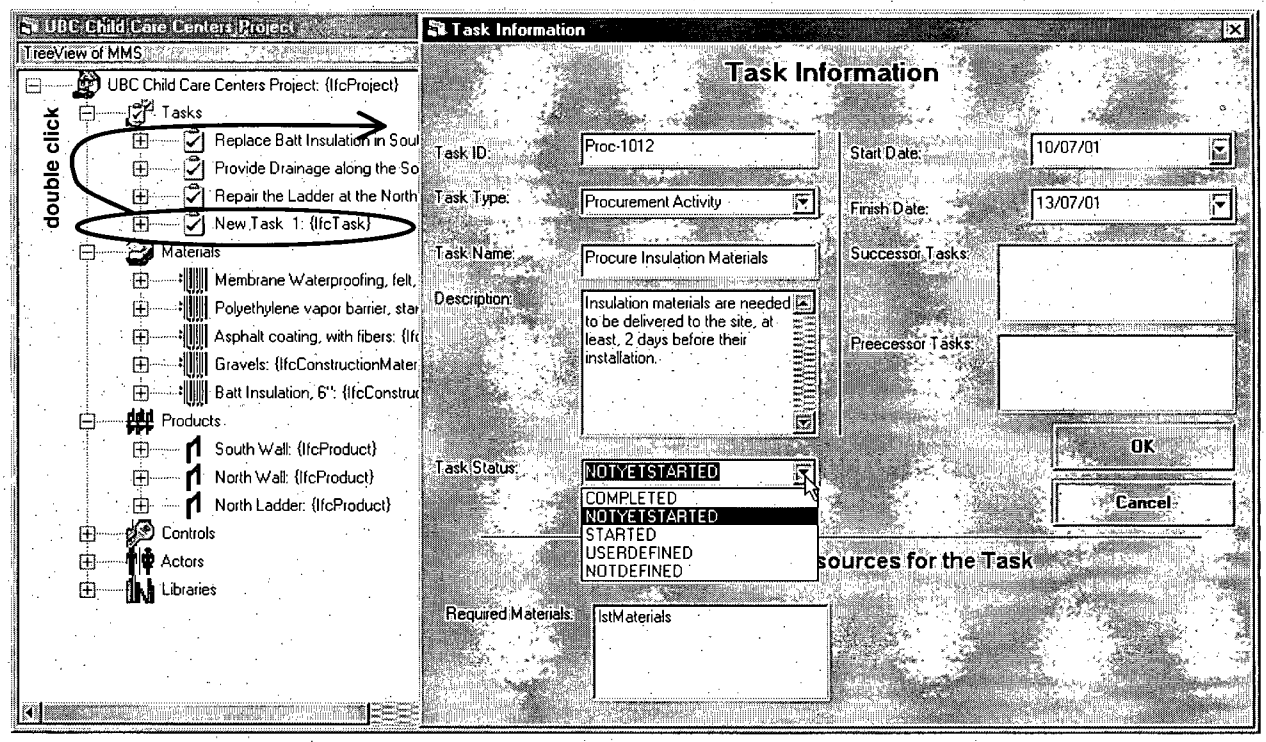


Figure C–15: Adding A New Procurement Task to the Project

C-5.3.5 Associating Project Elements

The assignment of various project elements to each other is performed using drag-and-drop operations in the tree-view window. For example, Figure C-16 shows how the "*Batt Insulation*" material is assigned to the "*Procure Insulation Materials*" task. Other assignments (e.g., materials to products and products to tasks) are also supported in the current version of the prototype. Such assignments are implemented using the various objectified relationships defined in the IFC model (section 4.2.1.2).

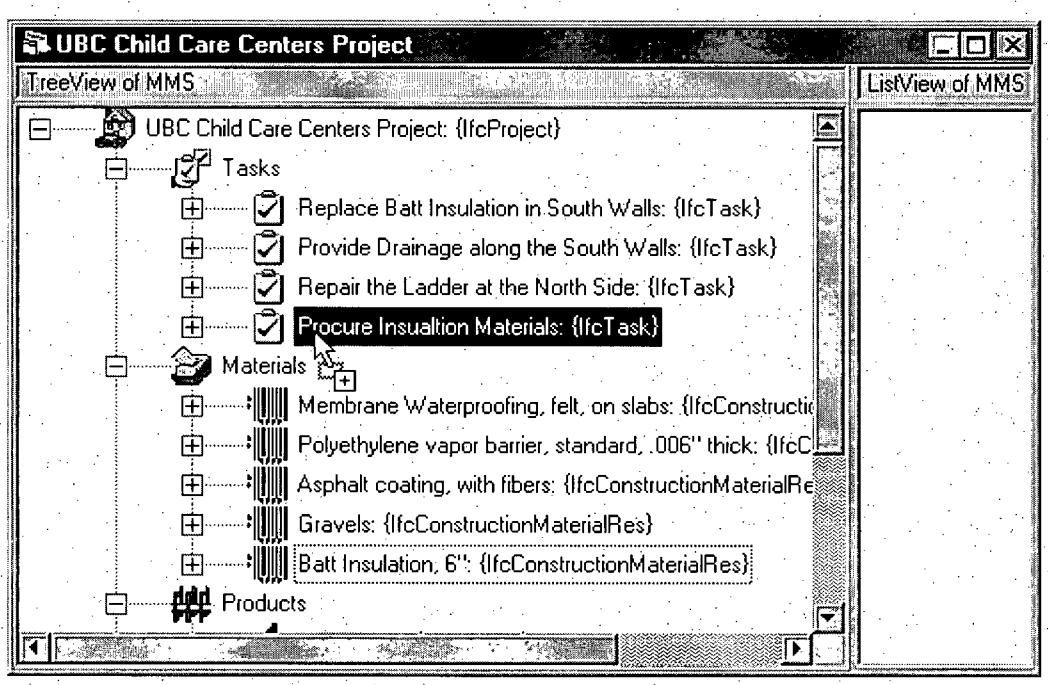


Figure C–16: Assigning a Material Type to a Task Using Tree View Window

C-5.3.6 Saving/Exporting Project Data Model

A project model completed in the MMS Tool can be saved or exported into one of the four data source types supported by the JDS (i.e., JsXmlFile06, JsBlisDS06, JsRooferDS06, and JsMSProjDS06; explained above). This requires the user to first choose the data source of interest (as explained earlier for the case of opening the project file, Figure C-9), which depends on the further actions planned for using the project data. Then the exporting process may continue by using the toolbar's save button or the "*Save*" or "*Save As*" command in the "*File*" menu (left side of Figure C-17).

Figure C-17 depicts the interfaces involved saving the project when the JsMSProjDS06 has been chosen (i.e., exporting the project information in the form of a MS-Project file). In the exporting process only that part of information is exported that is supported by the data source (i.e., by the schema of the application importing the data), while unsupported information is simply ignored. In this example, only the task information (name, start and finish dates, predecessor, and successor) is exported. Figures C-18

through C-20 show the subsequent interfaces of this exporting process. The newly created file can then be opened in MS Project for scheduling purposes.

Alternatively, the user may choose to save the project as an XML file so that this or other Jigsaw applications could open it for other purposes. For instance, the model can be opened in the AMS Tool for asset management purposes (Figure C-21).

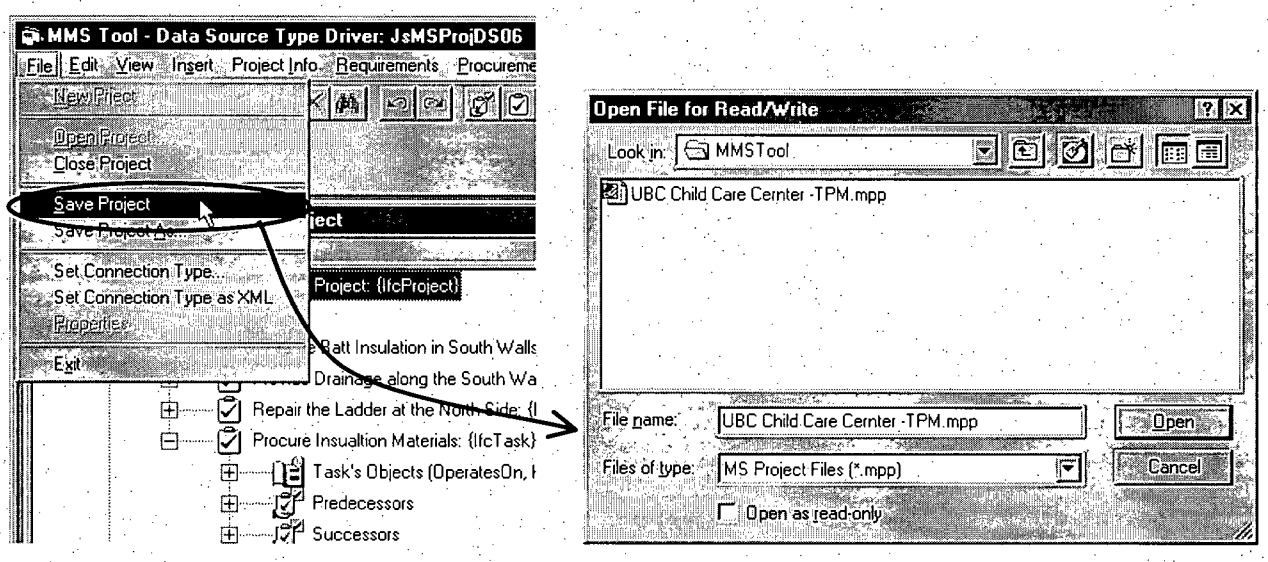


Figure C–17: Exporting the Project Model as a MS Project File (Saving Interfaces)

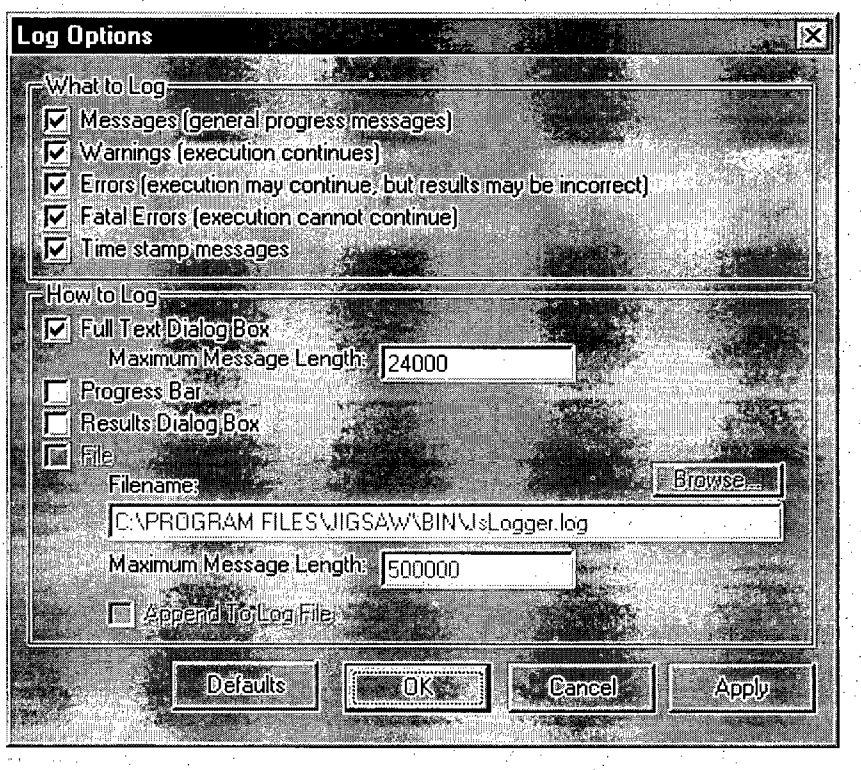


Figure C–18: Exporting the Project Model as a MS Project File (Log Options Window)

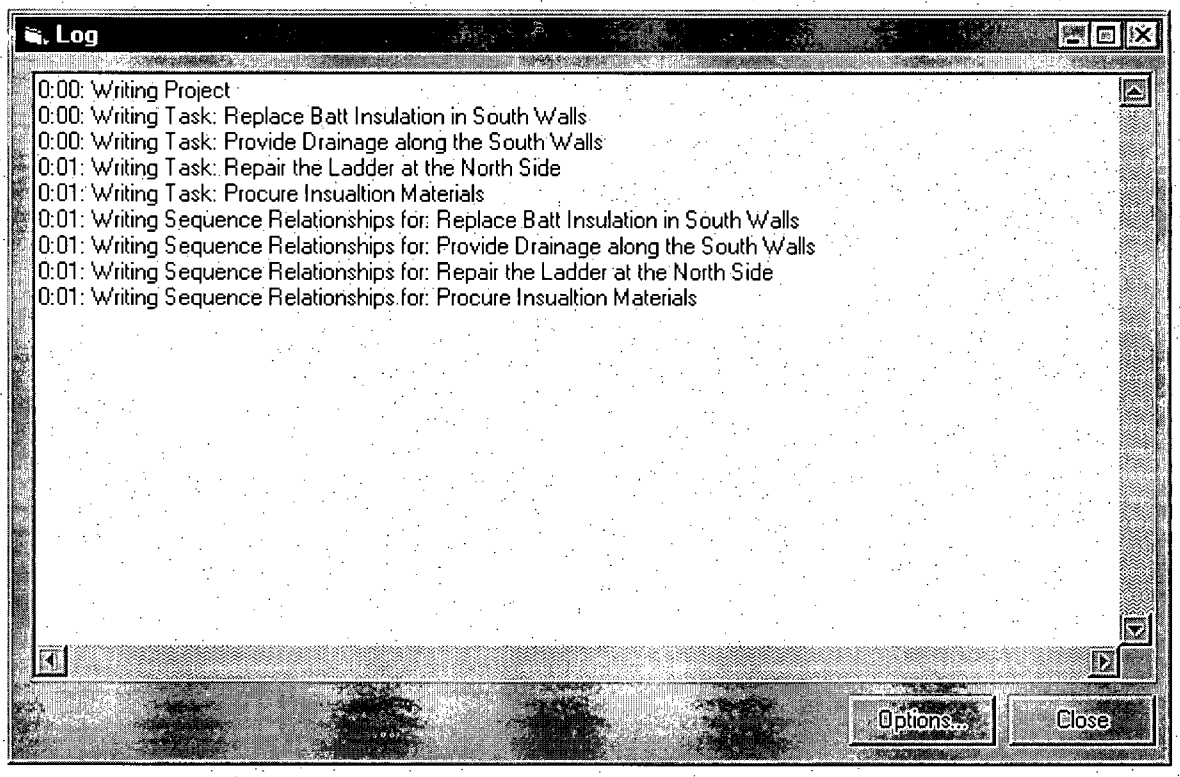


Figure C–19: Exporting the Project Model as an MS Project File (Log Window)

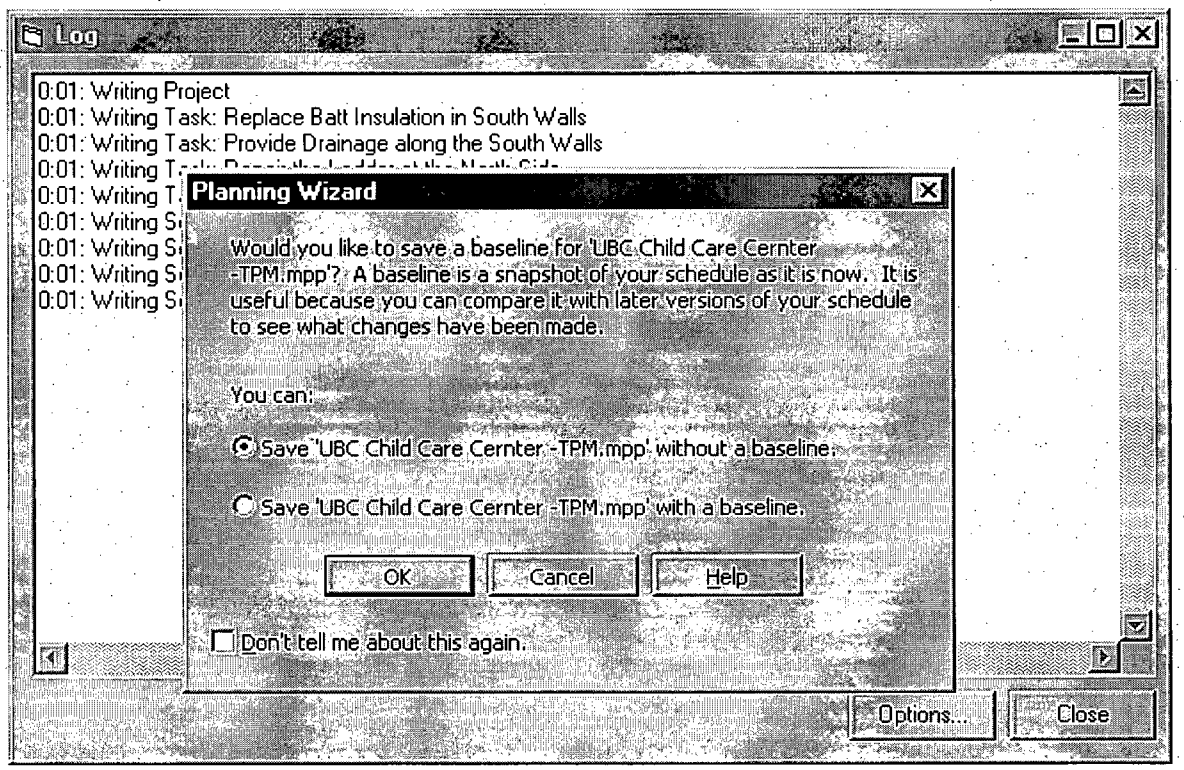


Figure C–20: Exporting the Project Model as an MS Project File (Planning Windows)

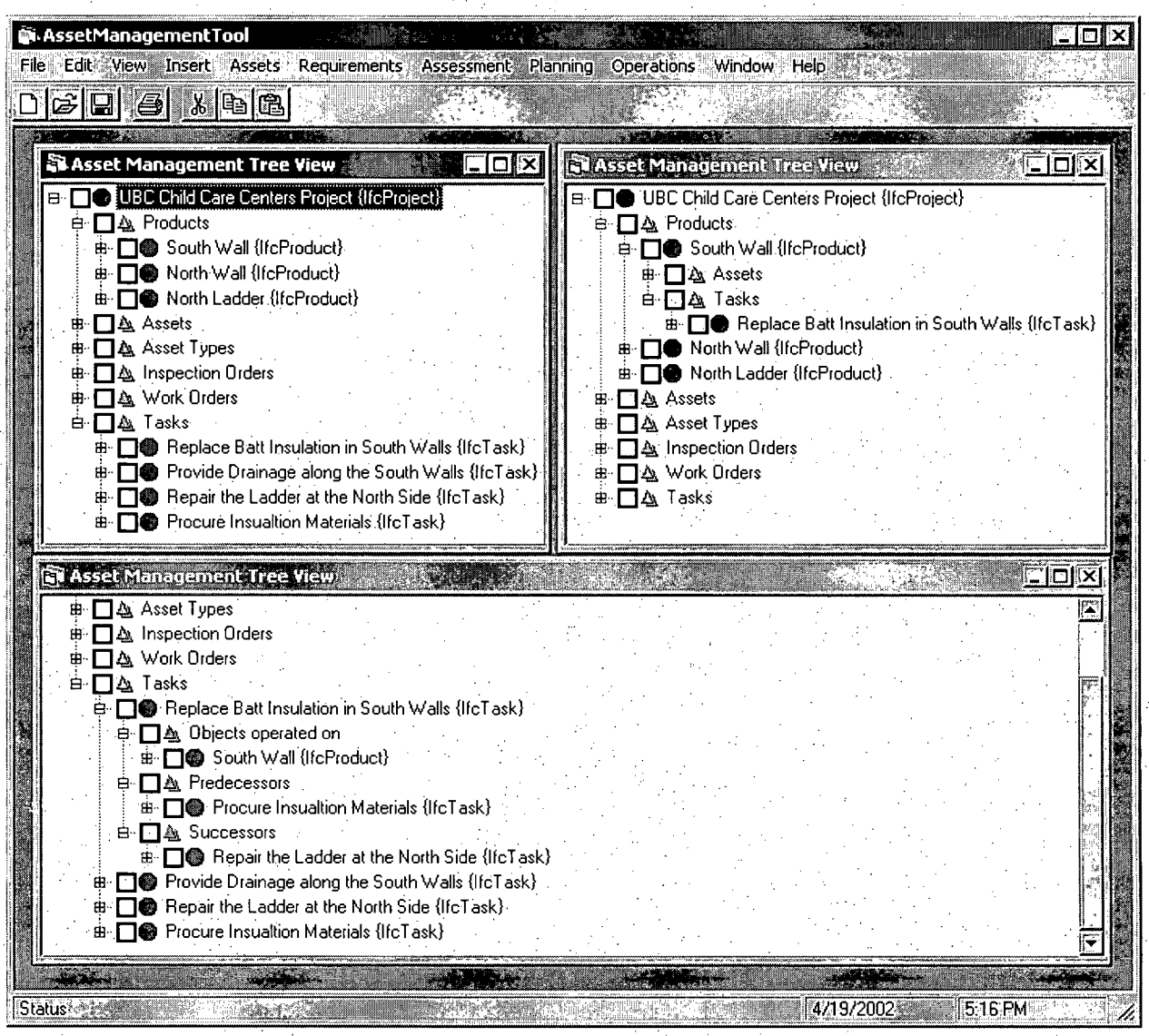


Figure C–21: Importing the Project Model in the AMS Tool

C-6 Discussions

This appendix presented the background and results of a prototype system development, which aimed at exploring model-based integrated total PM systems at an implementation level and within the overall objective of the research (section 1.4.1). The prototype system (i.e., the MMS Tool) was designed and implemented as a data-client component of a larger system (i.e., Jigsaw Distributed System, JDS, version 0.6), which facilitates a wide range of data exchanges and software interoperability through provision of a set of software components. Central to the prototype development was the use of the internationally recognized data standards (IFC R2x), as a *unifying vocabulary*, for structuring and communicating project information (i.e., the business logic commonly shared by PM functions).

The MMS Tool successfully demonstrated software interoperability in the context of construction PM. It is capable of flexibly exchanging data with other Jigsaw applications as well as commercial systems (e.g., MS Project) to support the various functions involved in managing AEC/FM projects. Despite its support of limited areas of functionality (as it was aimed to be a "proof of concept"), it has laid out the basic ground for further improvement. It also highlighted research and development issues, which are outlined in four major groups:

- *MMS Tool-specific Improvement*: Due to the scope limitation of the prototype, a large number of MM functionality areas were left undeveloped, though in many instances they have been flagged by considering "placeholders" in the system. Describing the system's features and interfaces, a number of these placeholders were highlighted in earlier sections. Among other areas identified for further development of the prototype is the definition and implementation of the *business objects* included under collections such as materials *Requests*, *Purchase Orders*, *Actors*, *Documents*, and materials *Information Libraries*. Except for a few, many of the objects considered under these collections have not yet been specified in the current version of the IFC model (i.e., a research need). Examples are the many types of project transactions performed throughout the project life cycle (e.g., *request for materials*, *request for quote*, *submittals*, and *request for information*). Another issue identified was the lack of explicit representation of *occurrences of materials* (i.e., uniquely identified material items) as opposed to *material types*. In the current version of the MMS Tool, to a large extent, the *IfcConstructionMaterialResource* appeared useful enough for representing the materials specified transactions such as a purchase order or request (i.e., as a material type), but it may not represent a specific material identified in a *receiving* or *inspection* transaction. Last (but not the least), resolving data ownership, access control, and security, change management, and versioning (out of scope of the prototype) remain to be critical issues in implementing integrated total PM systems.
- *PM/MM System Development*: The prototype development showed that the gap between the current level of support of MM functions and the state of computer integrated MM systems is vast, and MM is at large an undeveloped area. Moreover, the study of interactions between MM functions and other PM functions (e.g., facility and asset management) remain to be a subject for research. A part of the effort of this research (Chapter 8) was directed to address this gap. Moreover, the MMS Tool uses the file-based data exchange mechanism (which has its own implications) to communicate with other applications. The development of other data exchange mechanisms (e.g., server-based and transaction-based data exchange) using unified project models may also appear useful.
- *Process Modeling*: The complexity of MM processes was truly and practically realized. In fact, the prototype development took place early in the research, i.e., prior to most of the process and

information modeling work and with little insight into the complexity of processes. The complexity was visibly experienced in designing the UI elements (especially, of the main MDI form) and in organizing the menu commands of the prototype. This confirmed the necessity and importance of the process modeling part of the research (chapters 6 and 8).

- *Information Modeling*: As mentioned above, the prototype development revealed both the potentials and gaps that exist in the area of PM/MM information modeling. The lack of explicit representation of occurrences of materials in current data standards (section 4.1.6) and the necessity of identifying and formalizing various types of AEC/FM project transactions (section 4.2.4) are among specific examples of the areas identified as research needs; i.e., justification of the information modeling part of this research (chapters 5 and 8).

In summary, despite its relative limited functionality, the MMS Tool, in concert with other components of the JDS, proved the importance of data standards such as the IFC model in playing an integrating role among AEC/FM application systems. The development of the prototype also revealed places for improvement. The prototype was generally intended to serve as a means of investigation and validation of the concepts, ideas, and models suggested by the research. It contributed to the development of the primary deliverables of the research, though it may not be considered as a contribution by itself.

C-7 Conclusions

This appendix presented the results of development of a prototype application system (MMS Tool), which was a part of bottom-up investigations of this research on model-based integrated total PM systems. The goal, objectives, and methodology adopted for this effort were explained. The underlying platform and architecture of the MMS Tool and the functionality and interfaces of the system were described. A discussion of the results was also presented. While the prototype system is not considered as a part of contributions of the research, it represents a significant research activity. It gains its importance through its role played in the evaluation/validation of the primary research deliverables presented in the main body of this dissertation. It helped gain insight into model-based integrated total PM systems from an implementation perspective. This prototype was developed early in the research, prior to most of the information modeling work. The result of development of this system, which was purely based on the usage of the IFC object model, generally validated the research hypothesis (i.e., effectiveness of model-based PM integrated systems in supporting PM functions). It also confirmed some of the research needs identified early in the research—e.g., the necessity for explicit representation of materials, as viewed by construction crews, for instance, as well as the need for working towards covering the many areas of functionality not supported in existing models and application systems; section 1.3.

Appendix D: MM Business Objects

Table D-1 presents a list of MM business objects and their corresponding object-group packages defined in the ITPMS model implementation (section 8.4). The definitions suggested for the objects are based on various references [e.g., Ammer 1974; Wass 1980; Webster's 1982; AIA A201-1987; Oxford 1989; Clough and Sears 1991; CSOM 1994; Stukhart 1995; Minks and Johnston 1998].

Table D–1: MM Business Objects in the ITPMS Model Implementation

(Continued on the next fifteen pages)

Object-Package Name	Object Name	Definition
Products/Assemblies		
	Occurrence Product	The physical output of a process aimed to serve a permanent use (e.g., a house) or temporary use (e.g., a formwork) in an AEC/FM project.
	Product Type	A catalog-entry-like description of a collection of products; e.g., a door foundation type.
Materials		
	Occurrence Material	The physical material intended to be incorporated into a product; e.g., what is delivered to the receiving function.
	Material Type	A catalog-entry-like description of a collection of materials; e.g., a door type. It can be a generic material or a brand name material.
	Generic Material	A type of Material Type that may not be specific to one manufacturer. For example, under the "cast-in-place concrete" section of the Technical Specification, "curing and sealing coating" may be specified to "clear, liquid acrylic based polymer compound for curing and sealing concrete slabs." As a second example, under the "waterproofing & Dampproofing" section, the "Self-Adhesive Membrane Flashing" material may be specified to "Blueskin SA", which is a generic material and is manufactured by a number of manufacturers, including Sopraseal and Soprema. The real manufactured product of each manufacturer may, thus, be named differently.
	Brand Name Material	A specific Material Type branded to (i.e., produced by) a specific manufacturer. For example, under the "cast-in-place concrete" section of the Technical Specification, "curing and sealing coating" may be specified to "KURE-N-SEAL, manufactured by Sonneborn Building Products." As a second example, under the "waterproofing & Dampproofing" section, "Self-Adhesive Membrane Flashing" may be specified to "Sopraseal Stick 1100 as manufactured by Soprema."
	Material Category	The class or category to which a material belongs; e.g., the position of the material within a classification hierarchy (e.g., wall-mounted cabinets vs. floor-supported cabinets, self adhesive vs. non-adhesive). Thus, it may not provide any explicit information about properties (e.g., size) of the material. Such properties are directly associated with the material itself.

Object-Package Name	Object Name	Definition
	Material Type Database	A repository of information about different material types.
	Material Property	A collection of properties attached to a material. They could be "expected properties" (e.g., promised by the manufacturer, in the material's datasheet), "required properties" (suggested by design requirements, imposed by the owner, designer, and/or standards; usually recorded in the design and specifications), or "actual properties" (i.e., as measured/observed form occurrences of materials at a given time).
	Engineered Material	Items that are either engineered and fabricated specifically for a project or manufactured to an industry specification and are often stocked by the manufacturer or distributor. They are uniquely identified and referred to (usually with a unique number) on drawings and through the life cycle of the project. This includes tagged items and materials that require detailed engineering data sheets (e.g., mechanical and electrical equipment, operating doors, finish hardware, and folding partitions). An engineered material may have one or more Installation and O&M Documents (usually in the form of manuals).
	Bulk Material	Items purchased by lot quantities, standard length, or other quantity measurement. In contrast with engineered materials, they are normally allocated at the time of fabrication and or construction per schedule priorities, and they usually loose their identity in the finished product; e.g., sand, gavels, and cement.
	Fabricated Material	An assembly of basic stock materials or component parts that are joined together to produce a finished part or a complicated component; e.g., a steel beam with wholes, beam seats, and or connecting angles added. Also see the definition of the Total Parts List object.
	Direct Material	The materials that enter into and become part of the finished product and that can be identified with and assessed against a particular part, product, service, or group of parts, products, or services accurately and without undue effort and expense; i.e., the raw materials directly incorporated in the product.
	Consumable	Materials that are purchased in bulk basis but that lose their identity in the assembly, e.g., welding rods, gases, paint, and shims.
	Defective Material	A material or component that has one or more properties that do not comply with specified requirements.
	Spare Part	An extra part for a machine (e.g., washing machine), equipment (e.g., crane), car, etc. used to replace an identical part if it gets lost, damaged, etc.
	Spare Parts Checklist/Log	A list of spare parts that provides information on the parts and their related subcontractors and status (e.g., quantity and whereabouts).
	Surplus	Usable things (e.g., materials, equipment, or parts), which are in excess of construction or repair requirements. Surplus may sometimes be kept and reused (e.g., in future projects), returned to the supplier (e.g., through buyback arrangements), or sold to third parties.

Object-Package Name	Object Name	Definition
	Scrap	Materials or parts that have no value except for basic material content. Such items are usually dumped or recycled.
Plans and Schedules		
	Construction Document	A generalization of all various types of documents that communicate the design of the facility (envisioned by the architect and/or engineer) and transmitted to the contractor for realization of the facility; It includes project manual and construction drawings (see their definitions).
	Project Manual	A set of documents included as construction documents. Such documents, whose formats are dependent on the construction type and local customs, may include Advertisement/ Invitation For Bid, Instructions To Bidders, Bid Form, General and Supplementary Conditions to the Contract, Additional Information To Bidders (e.g., soil reports), General Requirements, and Technical Specifications.
	Technical Specification	A statement of particulars of a given job (construction work) as relates to the quality and performance of the labor, material, and equipment and procedures required to accomplish the job. It is accompanied with design drawings and is usually a part of contract documents.
	Construction Drawings	A collection of drawings describing the various subsystems (i.e., architectural, structural, mechanical, and electrical) of a facility in detail, to be used as a guide for construction of the facility.
	Work Schedule	A definition of timing relationships among the tasks involved in the accomplishment of a job.
	Materials Schedule	A schedule that defines the timing of materials requirements, i.e., materials over time.
	Procurement Schedule	A type of Work Schedule focusing on materials-related activities, from submittals through delivery to the jobsite.
Quantity Takeoff and Estimates		
	Bill Of Materials (BOM)	A listing of all direct materials (i.e., directly incorporated into) a product or a set of products. Consumables, temporary pieces, and tools are usually not included. A BOM includes such information as descriptions and quantities of the materials as well as their related incorporating products. Frequently this list consists of all items on a drawing.
	Materials Quantity Estimate (MQE)	An approximation of the amount of materials required for a specific assembly, task, project, etc. It can include direct materials, wastes, etc.
	Quantity	A number or amount that makes it possible to measure things.
	Materials Requirement	A collection of Material Types (often defined generically) required for a specific task, a project, an assembly, and so on.

Object-Package Name	Object Name	Definition
	Estimation Method	A method type used for estimating a specific type of item. Such methods may differ from one location to another. Examples include 1- Volumetric Method (e.g. cubic foot) vs. Surface-Area Method (e.g. square foot) for brickwork estimating; 2- Volumetric Method (used for large, deep concreting) vs. Surface-Area Method (for thin-layer concreting such as in a floor slab, driveway, or sidewalk.
	Unit Of Measurement	A physical quantity with which an aspect of a thing is measured. For example, "number of batches" and "cubic meter" are used for measurement of volumetric objects (e.g., the amount of concrete required for a foundation), while "square meter" is used for measurement of the surface area of layered objects (i.e., thickness relatively smaller than length and/or width; e.g., floor slabs, driveways, or sidewalks).
	Correction Factor	A number used for quantity adjustments. Examples are waste factor (used in estimating the required bulk materials) and swell/shrinkage factor (used for estimating the quantity of the soil involved in an earthmoving operation).
Allocation		
	Source	The place from which something (e.g., material) comes or is obtained or allocated.
	Sink	The place to which something (e.g., material) goes or is moved or allocated.
	Materials Allocation	The generic process of assigning specific materials from one or more sources to be used for a specific assembly, place, or task .
	Committed Allocation	The process of assigning materials from the inventory to a specific assembly, place, or task; e.g., allocating materials in the inventory documents.
	Trial Allocation	The process of ensuring availability of materials for scheduled construction tasks by checking their required materials against materials on hand.
	Physical Allocation	The process of (physically) setting materials aside and bundling and tagging them for their issue and ultimate use.
Submittals		
	Submittals	Shop drawings, material data, and samples that are required to be submitted by the contractor to the architect and engineer for approval (as specified in the construction documents, specifically the technical specifications).
	Sample	One of a number of things, or part of a whole, that can be looked at to see what the rest is like. A sample is usually intended for examination (visually or with instruments) and decision-making (e.g., selection, approval/rejection). Examples are material and test samples.

Object-Package Name	Object Name	Definition
	Product Data Submittal	A collection of the manufacturer's product information. It includes information such as manufacturer, trade name, model or type number, use, expected performance characteristics, size and physical and finish characteristics (e.g. color, texture, etc.), application, limitations, installation guides, estimation guides (e.g., coverage area), standards compliances, warrantees, and packaging.
	Shop Drawing	A drawing or set of drawings produced by the contractor, supplier, manufacturer, subcontractor, or fabricator, for the purpose of illustrating how specific portions of the work shall be fabricated/installed. It usually showing more detail than the construction documents.
	Submittal Log	A log used for tracking the actual progress of the submittal. It compares the projected dates with the actual dates the submittal received by the contractor and architect. A submittal log may also include (directly or by reference) other information useful for controlling purposes; e.g., actions taken (approved, request for revision, sent to, etc.).
	Submittal Review	The event of reviewing a submittal by the architect or contractor. It provides such information as the timing, subject, involved actor and the results of the review.
	Submittal Review Result	The information about the results of reviewing a specific submittal.
	Catalog	A document (usually booklet) listing a set of types of products of a manufacturer together with their related quantitative property information. A catalog is usually intended for marketing and helping the buyer in selecting a product. A catalog received from a supplier may be recorded in the catalog system of the (potential) buyer, i.e., numbered, dated, etc.
Requests		
	Materials Request	A formal inquiry made to receive specific amounts of some types of materials. A request can potentially be a follow-up, and thus reference, to another request that has partially or totally been rejected in an approval process. It can also be directed to a storeroom for withdrawal (i.e., act as a stores requisition) or to the purchasing agent (i.e., act as a purchase requisition).
	Materials Request Approval	An approval event for a materials request.
	Request Receipt Acknowledgement	A confirmation of the receipt of a materials request expressed by the receiver of the request.
	Request Rejection Notification	A notification sent to the requesting agent regarding the rejection status of the request.
Quotations		
	Request For Quotation (RFQ)	A solicitation document used in negotiated procurement to obtain an offer from a specific material/equipment supplier.

Object-Package Name	Object Name	Definition
	Quotation	An offer submitted in response to an RFQ (Request For Quotation); i.e., a statement of price, terms of sales, and description of goods or services offered by a supplier to a prospective buyer. A quotation may be followed by negotiations between the buyer and the supplier.
	Quotation Terms And Conditions	The terms and conditions attached to a Quote to be included in the General and/or Special Conditions of the Purchase Agreement/Contract after its acceptance by the buyer.
	Invitation For Quotation (IFQ)	A document used in competitive sealed bidding procurement (usually published in the form of an advertisement) to obtain offers from any interested company or person supplying a specific type of material or equipment. It is similar to an Invitation For Bid, but used mainly for equipment and materials.
	Bid Analysis Summary (BAS)	A compilation of commercial and technical evaluations that summarizes the bids and sets forth the reasons for recommending award to a particular bidder.
	Invitation For Bid (IFB)	A document used in competitive sealed bidding (usually published in the form of an advertisement) to obtain offers, from any interested company or person, for performing a work.
	Approved Suppliers List	A list of suppliers determined by the owner and contractor to meet the minimum set of standards of business competence, reputation, financial ability, and product quality for placement in the bidders list from which bids, proposals, and quotations can be solicited.
	Project Suppliers List	A list of suppliers of materials and services for a project. Such a list, which is usually used in superintendents' offices as a reference for various purposes (e.g., materials ordering/reordering), contains information about suppliers (name, contact, address, etc.) and their related materials and services.
Purchase Orders and Agreements		
	Purchase Order	A formal document used by the purchaser to formalize a purchase transaction with a supplier. It includes description, quantity, and price of the goods or services ordered; agreed terms as to payment, discounts, date of performance, and transportation; and all other agreements pertinent to the purchase and its execution by the supplier. It can have a status: approved, rejected, filled (i.e., items shipped), or closed (invoice/bill paid).
	Order For Material Release	An order placed (e.g., by the superintendent) to release a part or the whole of the materials that are the subject of a blanket order.
	Purchase Order Acknowledgement	A confirmation of the receipt of a purchase order by the supplier.

Object-Package Name	Object Name	Definition
	Purchase Agreement	An agreement between a buyer and a supplier setting forth, in general, the price and the terms and conditions of the sale; e.g., the involved parties, right, duties, obligations, schedule, QA/QC requirements, test info, drawing approvals, data submittals, expediting, terms of payments, vendor's technical service agreements (e.g., supervision of installation and/or erection by supplier's technical representative).
	Blanket Order	A type of purchase agreement aimed at reducing the number of small orders; i.e., the delivery dates and quantities of each delivery is subjected to the receipt of a release (i.e., order for material release) from the buyer. It requires the supplier to furnish certain goods for a certain period of time and at predetermined prices or, on the basis of a formula, for revising prices due to market or other conditions.
	Field Purchase Order	A limited and specific purchase order used in situations where authority to make the type of purchase involved has been delegated to designated agencies (e.g., the superintendent).
	Open Purchase Orders List	A list of all open purchase orders (i.e., those orders whose materials have not been received yet).
	Confirming Order	A purchase order issued to a supplier, listing the goods or services and terms of an order placed verbally or otherwise in advance of the issuance of the usual purchase document. A confirming order has a reference to an informal (e.g., verbal) order.
	Back Order	The portion of an order which the seller cannot deliver at the scheduled time and which has been re-entered for shipment at a later date.
	Back Charge	A sum deducted from the amount owed to the (sub)contractor (e.g., from progress payment) because of faulty or deficient work; i.e., costs that have been incurred by the contractor that should have been the subcontractor's responsibility (e.g., cleaning the site from material wastes).
	Discount	An allowance or deduction granted by the seller to the buyer, usually when the buyer meets certain stipulated conditions, resulting in reduction of the cost of the goods purchased.
Selling		
	Sale	The event of selling specific goods to a customer.
	Sale Item	The information pertaining to the item sold in a sale transaction. A sale can have one or more sale items, and every sale item belongs to one and only one sale.
Invoice and Payment		
	Invoice	A document showing the description, quantity, price, terms, nature of delivery, and other particulars of goods sold or of services rendered; a bill.
	Invoice Rejection Info	The information pertaining to the reasons for rejection of an invoice; e.g., the disagreements (in terms of quantity, quality, price, etc.) found between the invoice and the receiving reports, purchase agreement or quotation(s) of the materials invoiced.

Object-Package Name	Object Name	Definition
	Invoice Rejection Notification	A notification sent to the invoice generator (i.e., supplier) regarding the rejection of the invoice. The notification addresses the disagreements (in terms of quantity, quality, price, etc.) found in the invoice.
	Payment	A sum of money (to be) paid for acquiring goods or services. It can be invoice-based; i.e., originated after receipt of (thus have a reference to) an invoice.
	Cost Center Code	The lowest organizational unit or activity for which costs are accumulated. A cost code equates to an organization performing a mission where the supervisor has a degree of control over resource consumption.
Shipping and Transportation		
	Shipment	The action of shipping (or sending) goods by the supplier to the buyer.
	Shipping Method	The contractual arrangement selected for transferring the ownership and responsibility of goods from the seller to the buyer; e.g., FOB (Free On Board), C&F (Cost and Freight), and CIF (Cost, Insurance, and Freight).
	Packing Slip/List	A document that itemizes in details the contents (assemblies, subassemblies, and loose pieces) of a particular package or shipment.
	Shipping List	A list that enumerates each piece of material to be shipped in a shipment. The list is used to supplement the purchase order for receiving purposes.
	Total Parts List	A list of the parts of the fabricated equipment furnished by suppliers, especially when sub-suppliers are involved in multiple shipments. This list, which identifies the parts and their related shipment, is usually sent to the job-site (i.e., receiving point) by the supplier a few weeks prior to the final shipment of materials.
	Shipping Notice	An advance notice sent by the supplier to inform the buyer of a load that is being shipped. The notice provides information on the content, destination and arrival time of the shipment.
Delivery		
	Delivery	The transfer of possession; as applied to shipping, it occurs when lading is surrendered and title to goods passes to the receiver or consignee.
	Delivery Schedule	The required or agreed time or rate of delivery of goods or services purchased for a future period.
	Delivery Ticket/Slip	A document used by the driver of a shipment as a receipt from the receiver. It contains such information as ticket number, date, delivery address, time (loading, arriving at the site, unloading, leaving the site), customer (name, ID, etc.), purchase order reference, truck or hauling unit information, haul number, driver (name, ID, etc.), and receiver (name, signature, etc.).
	Vehicle Information	A collection of information about a vehicle; e.g., model, plate number, size, and capacity.

Object-Package Name	Object Name	Definition
	Bill Of Lading/Waybill	A document issued to a shipper by a carrier as both a contract (with the shipper) and a receipt for the goods that are promised to be transported and delivered to a designated person by the carrier. The document provides information on the goods (description, quantity, weight, etc.), invoice number (of carrier), shipper, carrier, type of delivery (regular, rush, special, etc.), charges, type of payment (prepaid, collect, etc.), destination, and receiver.
	Certified Bill Of Lading	An ocean bill of lading certified by a consular officer to meet certain requirements of a country as to goods imported.
	Release	A delivery that takes place subsequent to a request from the buyer, against a purchase agreement, blanket order, or price agreement.
	Demurrage	A charge made on cars, vehicles or vessels held by or for consignors or consignee for loading or unloading, forwarding directions, or any other purpose.
	Demurrage Log	A record of arrival and departure of rail cars, trucks, barges, and ships involved in a delivery process.
Receiving		
	Receiving	The action of taking delivery of an item at the designated location, including item accountability, quality, and quantity (i.e., tracking of shortages).
	Receiving Report	A form used by receiving function of a company to formally inform others of the receipt of goods purchased. Copies of the report are usually distributed to the purchasing and accounting departments and the store room.
	Receiving Location Information	A collection of information about the place at which the buyer receives a shipment; e.g., street address, plant gate number, and working hours.
	Gate	The entrance or exit point of a site, usually controlled for security and safety purposes.
	Gate Log	A record of the objects (including delivery vehicles) passed through a gate. The record includes such information as gate number, date, vehicle information, and enter/exit time.
	Gate Issue	Any unexpected event that takes place at a gate; e.g., a vehicle not registering at its exit time.
Inspection, Test and Approval		
	Inspection And Test Plan	A statement of the method (i.e., organization and regular, orderly, logical procedure) of proceeding with inspection and testing to satisfy quality objectives in a project. It describes the organizational structure (e.g., roles), responsibilities, procedures (e.g., timing and frequencies), processes, and resources (e.g., tools and equipment) required for inspection and testing.
	Request For Inspection Or Test	An inquiry for an inspection or test to be performed. The request includes such information as ID, date, requesting party, the subject and type of inspection/test, and date and time suggested for inspection/testing.

Object-Package Name	Object Name	Definition
	Inspection	The process of examining something to assure its conformance with specific requirements. This may be performed visually or with special equipment; e.g., looking for any hidden defects or damage, comparing actual material characteristics with their counterparts in specifications and blueprints, using gauges, and field and laboratory testing.
	Receipt Inspection	An examination of the goods receipt (e.g., at the site) before acceptance to note the completeness (i.e., the right item and quantity) of the delivery and any obvious external damage. This inspection is usually performed by the receiving clerk and is documented in a receiving report.
	Field Inspection	A thorough examination of incoming materials shortly after receipt to verify their conformance to specification requirements. As a second step in the receiving process, this inspection is usually performed by the quality control inspector (who often works in the same area as the receiving clerk) by looking for any hidden defects or damage. The results of this verification process are then recorded in an inspection report.
	Shop Inspection	A thorough materials inspection performed at the supplier or fabricator's shop for the purpose of assuring conformance of the materials to their requirements. This may be done visually or using performance tests.
	Inspection Report	A report that includes actual results of an inspection. In the receiving process, this report, which may be a separate form or part of the receiving report, it is usually prepared by the quality control inspector, and its copies are sent to the supplier (often if the material does not meet specifications), as well as to several departments of the buyer's company (e.g., purchasing and accounting).
	Test	The determination or verification of the capability of an item to meet specified requirements by subjecting the item to a set of physical, chemical, environmental, or operating conditions.
	Material Test Report (MTR)	A report that includes actual results of testing a material. It may be certified by a qualified person or organization.
	Certification Of Compliance	A written statement, signed by a qualified party, attesting that the items or services are in accordance with specified requirements.
	Certificate Of Occupancy	A document that is generated and issued by a local building code authority specialized in a specific area (e.g., plumbing, electrical, elevator, fire alarm, environmental/storm water drainage, and so on) after its final inspection/ certification on the entire project.
	Defective/Correction Notice	A notice issued by a local authority requesting for correction of a defective work. It is used when no life-threatening issue is found in an inspection, and corrections can be made without an immediate response.
	Stop-Work Notice	A notice issued by a local authority to stop the work. It is used when work is outside the permit or when public safety is threatened.

Object-Package Name	Object Name	Definition
	Reject	The action of refusing the acceptance of an item or discarding it as worthless, useless, substandard, or unsuitable for its intended purpose; e.g. a nonconforming material that is economically or technically incapable of rework or repair to be kept or used.
Storage and Inventory		
	Jobsite	A place in which a construction work takes place; the location of a construction project.
	Container	A generalization of any object that physically contains a set of objects.
	Transportation Unit	A means of transporting things from one location to another location (truck, aircraft, etc.); a type of container.
	Package Unit	A wrapped or boxed thing that contains an item or a group of items (e.g., bundle, box/carton, role, crate, parcel, etc.); a type of container.
	Storage Unit	A general concept that represents a place used to store materials. It can have different subtypes, such as loading/unloading duck, laydown area, warehouse/storage yard, section, shelf, aisle, bin, etc.
	Materials Storage	The action of physically storing materials in a place (i.e., storage unit) for future uses.
	Materials Disbursement	The action of expending materials; e.g., withdrawing materials from storage for use.
	Materials Inventory	A historical record of available materials in a given location or project. The record provides an update of on-hand materials by referencing to open purchase orders, receiving reports, and disbursements. It mainly encompasses a synthesis of the information posted by the purchasing, receiving, and storing functions.
	Materials Inventory Control	A one-time counting/measurement of the on-hand materials in a given location. It provides an itemized list of individually identifiable items, i.e., with the notion of "location" (e.g., a volume of bulk material, a set of blocks of bricks, a piece of mechanical equipment, etc.), and it is usually used as a control against the materials inventory record's balance.
Expediting		
	Expediting	The tracking, confirming, and/or accelerating of delivery of an item that has been purchased.
	Shortage	A deficiency in supply of an item; i.e., not having enough quantity or amount in hand and/or expected to receive.
Notifications and Acknowledgements		
	Notification/Notice	A generalization of any formal announcement or warning imparting required or pertinent information. A notification may be informative (e.g., a shipping notice) and/or instructive (i.e., requiring an action to be taken; e.g., a stop work notice).
	Acknowledgement	A formal statement of receipt of an item (e.g., a letter, a request, a package, or an order).

Object-Package Name	Object Name	Definition
	Memo	A notice to help one remember something or remind one to do something. A memo can be an "informative notice" (e.g., to inform what happened in an event, as in a cover letter of a meeting minute), or a "reminder" for an action that should be (or have been) taken.
	Meeting	A purposeful gathering intended to discuss or decide on specific business matters; e.g., a safety meeting intended to review and discuss safety issue at the construction site. The results of a meeting are usually recorded in its minutes.
	Meeting Minutes	An official record of what was said, done, and decided at a meeting.
Surplus and Excess		
	Surplus And Excess Plan	A detailed method, formulated beforehand, for managing surplus in a project or enterprise. The method might suggest, for instance, such techniques as materials substitutions where possible, prompt action on engineering changes, and surplus transfer to other projects, use as plant spares, returning to suppliers (through buyback arrangements), and sales to third parties.
	Buyback Agreement	An agreement whereby surplus materials are sold back to the firm from which they were originally bought.
	Disposal	The action of getting rid of something; e.g., through transferring to other processes or projects, recycling, returning to the supplier, and selling to other parties. The subject of a disposal event is usually scraps and surplus items.
Turnover and Closeout		
	Punch List	A list of the (sub)contractor's uncompleted (or to be corrected) work items. Depending on the contracting arrangement, a punch list is usually prepared by the architect, engineer, project manager, or the owner's maintenance personnel in coordination with the architect. Accordingly, the contractor prepares and issues specific punch lists to related subcontractors and suppliers.
	Keying Schedule	A list of keys required for a project. This list, which acts like a door schedule, usually applies to doors, may be used by the finish hardware supplier, the contractor, and the owner's facility management agents during the submittal, keying, key turnover, and maintenance and facility management periods.
	Warranty	A formal guarantee or an assurance given by a supplier or contractor (i.e., warrantor) to the buyer or owner (i.e., warrantee), stating that the goods or service is or shall be as represented or stated in the contract, or repair, replacement, and other forms of compensations will be provided by the warrantor as necessary.
	Warranty Checklist	A list of the warrantees associated with a piece of work or project. The list includes such information as the related section of the technical specification, warrantor (e.g., subcontractor, warranty type (e.g., 1 year standard, 5 year warranty bond, or 2-season special warranty), and warranty dates (e.g., receiving and delivery to the owner).

Object-Package Name	Object Name	Definition
	Certificate Of Substantial Completion	A formal, standard document signed by the architect, the contractor, and the owner, announcing that the facility or a portion of the facility has reached to the stage of 'beneficial occupancy' and is acceptable for owner use and occupancy. It may also list some remaining minor defected items to be completed by the contractor. A list of warrantee dates, which are established by the issuance of the certificate, should also be attached to the certificate.
	O&M Manuals	A set of documents that contain technical information or instructions on installation and O&M (operation and maintenance) of specific types of technical equipment, such as mechanical and electrical equipment, as well as other operating systems; e.g., operating doors, finish hardware, folding partitions. Such manuals are usually furnished by the contractor or suppliers.
Quality and Performance		
	Quality	Conformance to established requirements (i.e., to acceptance criteria).
	Quality Policy	The overall intensions and direction of an organization with regard to quality, as formally expressed by top management.
	Quality Plan	A place in which a construction work takes place; the location of a construction project.
	Quality Requirement/Acceptance Criteria	Specified limits placed on characteristics of an item, process, or service. Such limits are usually defined by codes, standards, or requirement documents.
	Quality Control Checklist	A collection of quality control (QC) points or reminders, which relate to a work item (e.g., concrete work) or a physical item (e.g., cement), that should be checked off by a quality controller upon satisfaction of their related quality requirements. QC jobs are usually allocated to superintendents (or their assistants), engineers, or field engineers, using QC assignments.
	Quality Control Assignment	A formal allocation of a specific QC job to a QC actor, who is usually the superintendent (or his assistant), engineer, or field engineer. A QC assignment has an association with an actor and a QC checklist.
	Vendor Performance	The level of satisfaction of a buyer from a vendor. This, which is usually expressed as a "rating" (e.g., excellent, good, satisfactory, or unsatisfactory), may be considered as a function of both pre-award factors (e.g., experience, personnel, financial, and location) and post-award factors (e.g., the quality of work, compliance with delivery schedule, and cooperation).
	Defect	A nonconformance of an item with the specified requirements. It is usually used for physical things, and it refers to the specific part or piece of the defected thing (e.g., the part that does not function as required).
	Failure	The event of inability of an item to function as expected or required; e.g., not able to perform within previously specified limits.

Object-Package Name	Object Name	Definition
Safety and Security		
	Material Safety Datasheet (MSDS)	A form that provides information about a chemical product. It is supplied by chemical manufacturers and must be kept on file on the site. The MSDS information includes "chemical makeups of the substance, fire and explosion hazard data, reactivility data, health hazard data, precautions for safe handling and use, and control measures.
	Safety Manual	A collection of rules and instructions pertaining to safety considerations prepared by the owner for the contractor's attention.
	Supplier Label	A label placed on a product/material by the supplier. It provides some essential safety considerations and usually has a reference to the MSDS (Material Safety Data Sheet) of the product/material for more detailed information.
	Workplace Label	A label that identifies the place a hazardous material is used. Workplace labels are usually required to be placed on containers used to store hazardous materials made or filled at the workplace.
	Safety Issue	An important safety matter that needs to be questioned, discussed, decided, and/or disputed.
Changes		
	Request For Information (RFI)	An inquiry (usually from contractor to architect) for further information or clarification on an issue, which has not been addressed fully or at all in a project's contract documents. An RFI is normally intended to eliminate unnecessary disputes.
	Request For Change	A request sent to the contractor (usually by the owner and/or architect after an issue raised in an RFI) to implement a change. The contractor may, in response, evaluate the cost and time consequences of the change and prepare a Change Proposal, with a reference to the request.
	Change Proposal	A document prepared by the contractor or supplier, usually in response to a request from the owner or purchaser, to suggest his conditions for a change in the contract.
	Construction Change Directive	A written order used in the absence of a Change Order by the architect and signed by the owner and architect, directing a change in the work stating a proposed basis for possible cost and/or time adjustment.
	Change Order	A document signed and issued by the purchaser (or owner and/or architect), authorizing the supplier or contractor to modify, add to, or subtract from a purchase order (or contract). A change order can have a reference to a Change Proposal, and it is an all-parties-agreed document.
	Change	An increase, decrease, or revision in the work (or items) ordered by the owner (or purchaser) within the general scope of the contract (or purchase order).
Actors/Roles		
	Manufacturer	A person or company that produces goods in large quantities.
	Distributor	A wholesale middle-person or company handling a wide range of products for a number of manufacturers.

Object-Package Name	Object Name	Definition
	Supplier/Vendor	A person or firm selling goods. The supplier may or may not be the manufacturer.
	Buyer	A person or agent who purchases goods and may perform purchasing-related functions (e.g., vendor selection, negotiation, order placement, follow-up, and vendor performance control).
	Product Information Center	An agency that provides pools of information about products of different manufacturers in the form of product catalogues to the public. Examples are Architect's First Source Inc. (with product of Architect's First Source) [Architect's First Source 2005], Buildcore Inc. (with products of BuildSourceTM, BuildSelectTM, and BuildSpecTM) [Buildcore 2005], and Sweet's Group (with product of Sweet's Construction's Product MarketplaceTM) [Sweet's CD 2003].
	Materials Requester	A person or agent requesting for materials from the provider.
	Salesperson	The supplier's liaison between customers and the home office and factory.
	Shipper	A person or agent who arranges for goods to be shipped.
	Freight Forwarder	An agent that acts on behalf of the shipper, as a carrier or middle-person, in arranging for transportation of goods (through ocean, air, or land) to the ultimate consignee.
	Common Carrier	A person or agent, licensed by an authorized governmental agency, engaged in the business of transporting personal property from one place to another for compensation.
	Fabricator	A person or agent who builds, constructs, or manufactures a completed item (e.g., a roof truss) by fitting together standardized parts.
	Consignee	A person or agent (usually the buyer) named on a bill of lading to receive the goods from the carrier.
	Witness	A person who is present at an event (i.e., sees an event take place), especially the signing of a document, and thus can testify to the fact that it took place.