The Role of Object-oriented Databases in Project Management

by

Fabio Massimo Mondini

Laurea in Ingegneria Civile Edile, University of Naples "Federico II" (1992)

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

Master of Science
in
Civil and Environmental Engineering

at the
Massachusetts Institute of Technology
June 1994

Signature of author..
Department of Civil and Environmental Engineering
June 24th, 1994

Certified by..
Prof. Robert D. Logcher
Thesis Supervisor

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

The Role of Object-oriented Databases in Project Management

by
Fabio Massimo Mondini

Submitted to the Department of Civil and Environmental Engineering in partial fulfillment of the requirement for the degree of Master of Science in Civil and Environmental Engineering at the Massachusetts Institute of Technology on June 1994

Abstract

The objective of this thesis is to analyze the state of the art, new tools and prospective of computer aided project management. The thesis work will emphasize the greater flexibility, higher intelligence and significantly better integration of Object-Oriented Databases as compared with Relational Databases.

An introduction to object-oriented database management systems will highlight its potential for integrated computer aided project management. The advantages of the object-oriented technology toward the relational database are also discussed. After a brief history of computer-aided project management, the thesis will analyze the use of software applications in the main areas of construction project control. The thesis will use a survey of G.C. computer use to show the different utilization ratios between computer applications. An interpretation of this survey implies that system integration might improve utilization.

The goals, benefits and barriers to the implementation of an integrated project management system are also discussed with an analysis of the major issues and problems of construction project control and management in the CIC (Computer Integrated Construction) environment. The research has identified three key points:

1. A piece of data, gathered or created by one user, once put in the system can have value to other users of the system.
2. Data combined from multiple users is more useful than separate groups of isolate data. They can enhance existing applications and create new system capabilities.
3. Data sharing is more complicated than just data storage. There is a strong need for interpretation.

Finally, the thesis has shown that that the interaction of active databases and project management tools can achieve two important goals:

1. interpret data automatically, creating data useful to an application different from the one in which the initial data has been collected.
2. enhance control procedures for problem recognition, combining data from different application to evaluate performances and anomalies.

Thesis Advisor: Robert D. Logcher
Title: Professor of Civil and Environmental Engineering

Dedication

To my father: a wise advisor, a constant point of reference and a wonderful friend.

Acknowledgments

I would like to thank:

My mother, for having taught me the real value of education.

Prof. Logcher, my thesis supervisor, whose support is present in every part of the thesis. He had always the right answers to all my doubts and a useful critic to my ideas.

Geom. Tamburini and ing. Pietroni, for their support.

Frank Altobelli, for his availability and his advises.

Lucio Soibelmann, for his friendship and the several discussions we have made, that have brought new interesting ideas.

Mehmet Emre Camlibel, and all my new friends, that have made this MIT experience more interesting and pleasant.

All my Italian friends and relatives, that have supported me and the phone companies in this year of permanence at MIT.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

A single construction project must handle a massive amount of information. The correct use and management of information is one of the key points for the implementation of a good Project Management System (PMS). Several features of the construction industry hamper good information flow. This thesis will discuss the issues in construction project control regarding information flow and will show some possible uses of object-oriented technology to enhance the current PMSs.

1.1 The role of Information Systems in the A/E/C industry

The main purpose of an information system is to increase productivity. The system should be able to improve the utilization of the information, adding value to the data present in the database (DB). The implementation of a project management system has a significant impact on the organization structure, and it is likely to require changes in the company organization. Some of the project management features that can be enhanced with a PMS are the following:

- Efficient management of a large volume of data in an electronic format with easy retrieval
- Data communication links with project participants
- Planning and simulation of field operations (with analysis of possible scenarios)
- Project monitoring, management of changes in design and field conditions
- Summary and reports on a timely basis
- Evaluation of project personnel

- Problem recognition on a timely basis
- Accurate projections of remaining time and cost

From the list, it is clear that a DB plays a fundamental role in the company PMS. In some current PMS, there is a database management system (DBMS). This thesis will analyze the current use of computer aided project management in construction and identify areas in which the performance of the applications can be improved. The analysis of the current state of the art will suggest that integration is a key point of the development of a PMS. Moreover, since object-oriented systems (OOS) offer many features that can add value to the information in the system, this thesis will focus attention on object-oriented database management system (OODBMS).

1.2 Object-oriented technology

Preliminary research has suggested that object-oriented technology can effectively aid the development of PMS in construction. Thanks to several of its advantages, discussed in chapter 2, object oriented systems (OOSs) are increasingly used for many engineering applications. This thesis will focus on how some of the object oriented features, such as behavior and message passing, can provide an OODBMS with more dynamic life of information in the system. OOSs can also reduce software complexity, and are able to model the knowledge domain at a much higher level, handling large quantities of complex engineering data more effectively than traditional approaches such as relational databases. Overall, an object-oriented approach makes project modeling and system integration schemes more flexible and powerful, as well as much easier to create.

1.3 An active database

The role of current DBs is essentially that of a static repository of data, with some tools which aid the user to retrieve the data he needs in different ways. This thesis suggest that an OODB can go further and interact much more with the PMS. Within an integrated PMS, an OODBMS can process and interpret information in the system. An "active" (or "intelligent") database can reason about the values it has and react according to them. Sriram and Logcher write:

> *In short, every object has the ability to: store information, process information, create new information, and communicate with other objects. Thus OOP facilitates encoding design and construction knowledge in a disaggregated and modular form.* [1]

This thesis will show how object-oriented features, such as behavior and message passing, can add value to the information that has been input into the system interpreting data in two ways:

1. create some data, that otherwise must be collected separately
2. analyze the data present in the system to identify problems and anomalies, according to knowledge that has been input into the system

Some examples are discussed in chapter 5.

1.4 Thesis outline

The primary objective of this thesis is to investigate the role of OODBMS in construction project management. To perform this task, the thesis will show some examples of how an OODB can interpret data present in the system. This work is organized as follows:

---

[1] Sriram 1989, pg. 7.

- Chapter 2 introduces the reader to the characteristics of the object-oriented database (OODB) technology, its state of the art, and the main differences from the relational databases (RDBs).
- Chapter 3 analyze the CAPM history development and current state. Great attention is given to the trends that are driving the development of current PMSs, such as integration and project modeling. The chapter interpret a survey of Information Technology (IT) use in construction, identifying the reasons for different levels of integration in different construction companies. Finally, a current advanced PMS is described.
- Chapter 4 discusses the issues in Construction Project Control regarding the implementation of an integrated PMS. After an analysis of the benefits of the system and the barriers to its implementation, possible solutions are outlined.
- Chapter 5 shows some examples of data interaction in an OODB within an integrated PMS. In the examples shown, scheduling and cost data present in the system will automatically interpret the information in the DB to create new data that otherwise would have to be collected separately.
- Finally, Chapter 6 concludes this study with a discussion of possible future directions of research.

Chapter 2
Object-oriented Databases

The ever-increasing complexity of database applications and the increasing richness of the data has prompted the development of more efficient and cheaper DBMS (Database Management Systems). The goal of DBMS improvement was often the offloading of some tedious and repetitive bookkeeping functions from the applications into the DBMS. In general, a conventional DBMS has to include the following facilities:

- *A Data Model*. This provides the schema to model and manage the data.
- *Persistence and Recovery*. The data are in a non-volatile memory and it is possible to recover them from a crash.
- *Concurrency and Transaction Management*. The database may interact with different users; the transactions are atomic and consistent
- *Query language*. The program utilizes a high-level, easy-to-use language for accessing information systematically.
- *Security*. The program offers also a selective access to the data itself.

However, as described later, the data model and structure both reveal themselves to be unable to handle computer-intensive applications like CAD, and non sufficiently flexible for cooperative work.

2.1 The evolution of Database Technology

The database technology for information systems has constantly been improved in the last 30 years by adding new features and processing larger and more complex data and database applications. The technology has passed through four generations and the fifth generation is now the core of much research. This work will analyze in this chapter the last two, while first three generations are here described by Kim:

The first generation was file system, such as ISAM and VSAM. The second generation was hierarchical database systems, such as IMS and System 2000. The third generation was CODASYL database systems, such as IDS, TOTAL, ADABAS, IDMS, etc. The second and the third generation systems realized the sharing of an integrated database among many users within an application environment. The lack of data independence and the tedious navigational access to the database gave rise to the fourth-generation database technology, namely, relational database technology[1] .

2.1.1 Relational Databases

In 1993, relational databases were the most common in business application. They are characterized by declarative queries and they can be easily described as a collection of tables. Therefore, the software for an RDBMS (Relational Database Management Systems) will be essentially a program for managing these tables. An RDB, as defined by Codd, has three major parts:

1. **Data Structure**. An RDB it is just a collection of tables. The columns and the rows are called attributes and tuples (Fig.2.1). Every attribute has a domain which is a pool of values from which one or more attributes draw their actual values. Every value in the database must belong to the domain of its attribute or be null (missing or unknown).

Attribute: a value, indicating the ID#

Dog name	Owner	Plate number
Tony	John	2341
Tobias	Frank	2342

Tuple: the row with information about Tobias

Figure 2.1 - Data Structure

[1] ref. Kim, 1990, pg. 1

2. **Data Manipulation**. The most popular query language for RDB is SQL (Structured Query Language). It can retrieve, order or manipulate columns and rows of the tables, selecting and sorting them with several types of clauses or comparison conditions.

3. **Data integrity**. In 1972, E. F. Codd proposed a relational model with the semantic of the relation attributes. Essentially, he highlighted two general integrity rules: entity integrity and referential integrity. Entity integrity dictates that each table have exactly one primary key. A primary key is a combination of one or more attributes and a unique identifier for the table. All pairs of the primary key in the same table must differ from each other. The primary key, therefore, provides the basic tuple-level addressing mechanism in a relational system. A primary key embedded in another table is a foreign key. Therefore, the referential integrity requires that the RDB keep each foreign key with its corresponding primary key. However in 1994, 22 years after, most applications still do not provide the referential integrity.

   Moreover, Codd also defined some restriction rules to avoid redundancy and update anomalies (in terms of insertion, deletion and modification of the data). He introduced the Functional Dependencies and the first four Normal Forms. A table is in the first Normal Form (NF) when each attribute value does not contain a repeating group. It is in the Second Normal Form (2NF) if each row has a primary key, each non primary key attribute is fully functional-dependent on the primary key and it is in the NF. A table is in the Third Normal Form (3NF) if it is in 2NF and no non-prime attribute is transitively dependent on the primary key. The fourth and the later Boyce-Codd Normal Form give additional criteria.

The introduction of these criteria and the establishment of SQL as the standard query language for RDBMS, increased the popularity of the RDBs.. In 1994, the RDBs represent the vast majority of the database in commerce.

2.1.2 Limitations of Relational Databases

The RDBMS has proved to be very efficient for managing tables of alphanumeric fields of fixed length. However, there are several limitations, including:

- *A simple model.* The relational model is too simple for modelling complex nested entities, such as design and engineering objects, and complex documents
- *Simple data types.* The conventional RDBs support only a limited set of atomic data types, such as integer, string, etc.; they don't allow the storage and retrieval of long unstructured data (the so-called long data) such as images, audio and textual documents.
- *Low performance.* The performance of a RDBMS is good only for simple types of applications, while it is unacceptable for various types of compute-intensive applications, such as CAD and project management.
- *Impedance mis-match.* The application programs are implemented in some algorithmic programming language (such as COBOL, FORTRAN, or PL/1) and some database language embedded in it (such as SQL, DL/1 or codasyl DML), that are very different from the programming languages, in both data model and structure.
- *Poor transaction management.* In a co-operative, interactive environment involving complex data, the serial model of transactions supported in relational database systems is not able to allow multiple simultaneous access to the same data used by cooperating designers.

New and more complex needs and shortcomings of the relational database have pushed toward new and more complete technologies. The fifth generation of databases is Object-oriented. The next part of the chapter describes what an object is and how the Object-oriented Systems can overcome the limitations of the RDB leading to an OOBDMS.

2.1.3 Network Databases

The data structure in a network database is quite different. While in the RDB, the access to a record is achieved through the value of the primary key or of an another attribute, in a database system like CODASYL, a piece of data can be retrieved directly, with the so-called *navigational access*. There are essentially two different approaches: hierarchical and network[2].

In the hierarchical approach, each record would usually include several data-items. In the example in Fig. 2.2, there are three occurrences of DEPT and nine of EMP. These records occurrences are grouped into three occurrences of a set called DEPTEMP. Each set type is defined in the schema to have a certain type of record as its member; in the example, set type DEPTTEMP would be declared in the schema with DEPT as its owner and EMP as its member. Each occurence of a set represents a hierarchical relationship between the owner occurence and the corresponding member occurences. These connections can be made via a chain of pointers that originates the owner occurence, runs through all the member occurences, and finally returns to the owner occurence.

Figure 2.3 shows an example of a database with a network situation, that represents the current plans for a series of orchestral concerts. Each concert will include works by several composer and each composer will have works in several concerts. This situation is represented by introducing a "connection" record type (WORK), whose function is to connect the two basic types of entity (CONCERT and COMPOSER). Each WORK occurence connects one concert and one composer and contains data describing the connection it represents, i.e., the name of the appropriate work.

---

[2] ref. Date, pg. 395-415.

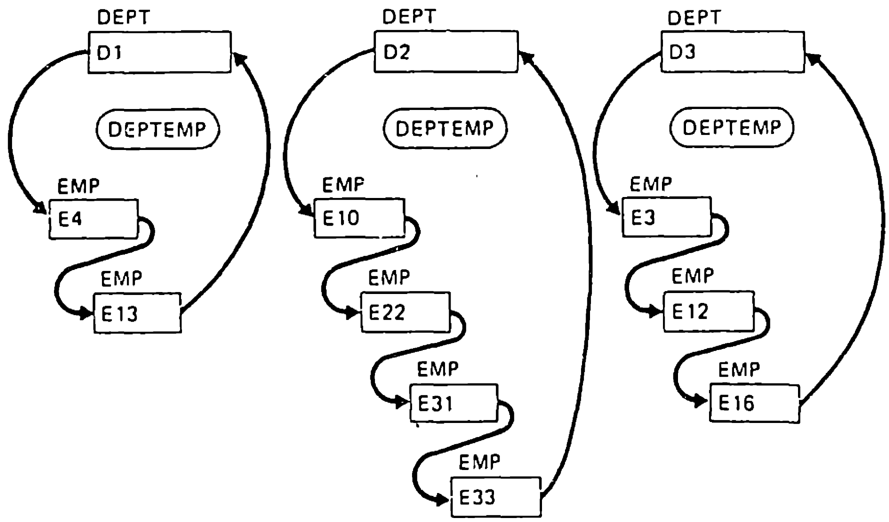


Figure 2.2 Hierarchical Approach[1]

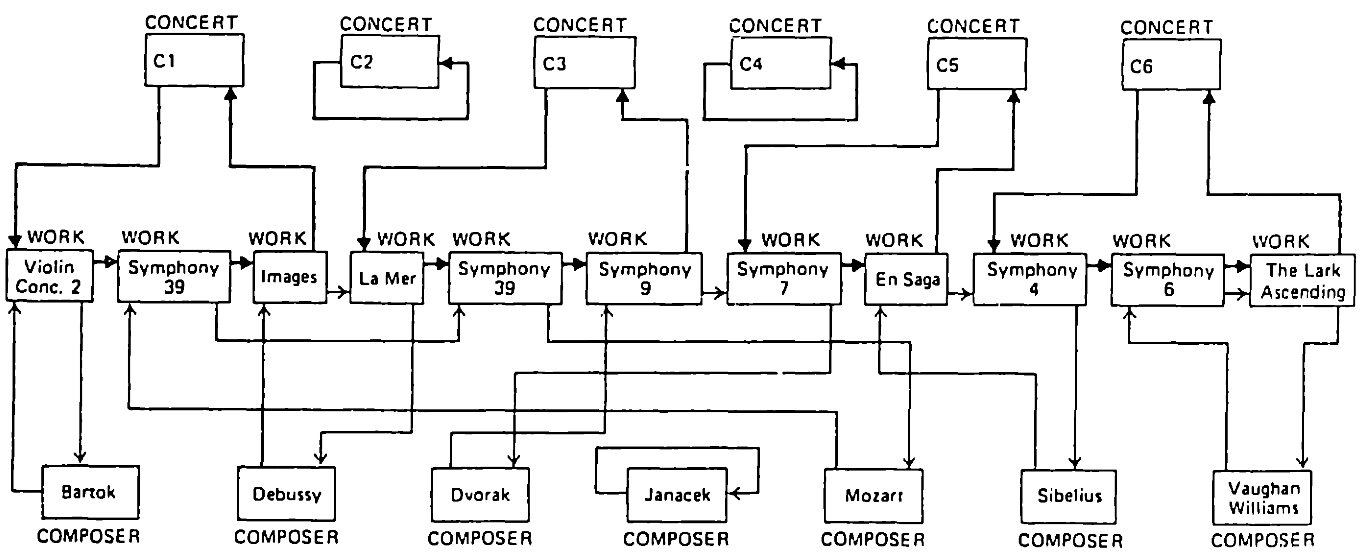


Figure 2.3 - Network Approach[2]

[1] rf. Date, pgg. 348
[2] rf. Date, pgg. 360

As we will see later this approaches are closer than the relational to the object-oriented approach.

2.2 What is an Object?

It is not easy to define what an object is. Booch has introduced the concept in this way [Booch]:

> *we informally defined an object as a tangible entity that exhibits some well-defined behavior. From the perspective of human cognition, an object is any of the following:*
>
> - *A tangible and/or visible thing*
> - *Something that may be apprehended intellectually*
> - *Something toward which thought or action is directed.*
>
> *... the term object was first formally applied in the Simula language, and object typically existed in Simula programs to simulate some aspect of reality.*[3]

An object can therefore be something physical or conceptual. It can have attributes and it can exhibit behavior. An object could be: an activity, a resource, a line in a drawing, or a worker. All of these objects have attributes which define them and distinguish them from others. This group of attributes comprises its state. Some of these attributes may be relationships to other objects. The lock on the left car door may have identical properties to the one on the right door, but it is part of a different door. As we will see later, similar objects can be grouped into a *class*; individual objects of a class, with attribute values, are called *instances*. Moreover, at the moment of its creation, every object will have its own identity, that will allow it to be absolutely unique. The definition of an object given by Booch is:

> *An object has state, behavior, and identity; the structure and behavior of similar objects are defined in their common class;*[4]

---

[3] ref. G. Booch, pg. 76.

[4] ref. Booch, pg. 77.

2.2.1 State and attributes

In object-oriented programming (OOP) and in OODB, an object is a unique block of data and methods. The data provides the values of the attributes of the object, which might represent the external appearance of the object. For example, the attributes of a car can be colour=red, brand=Ford, model=Escort. Some attributes can be hidden, from external access. Booch defines the set of the attributes (state) of the object[5] :

The state of an object encompasses all the (usually static) properties of the object plus the current (usually dynamic) values of these properties.

If we think of the same car, a static property can be the brand or the model, while a dynamic one can be the colour or the owner. All the data of the object will represent its state.

2.2.2 Behavior

Booch defines the behavior of an object in this way:

Behavior is how an object acts and reacts, in terms of its state changes and message passing.[6]

An object may send *messages* to another object in order to influence the other object's behavior or to stimulate sending other messages. In OOP, the operations that an user can perform upon an object are called also methods. The object can be designed to reason about the domain and react in different ways depending upon the value introduced. In this way, the presence of the behavior in the object itself assures a dynamic reply to external and internal *messages*. In OODB it is vital and it is the basis of many features of the object-oriented system (OOS).

---

[5] ref. Booch, pg. 78.

[6] ref. Booch, pg. 80.

2.2.3 Relationship

Objects are not only independent pieces of data; they are linked to other objects. A linkage between two objects forms a relationship. The object *student* will interact with the object *professor*: in this specific case, the relationship can be defined as *advises* or *is being advised*.

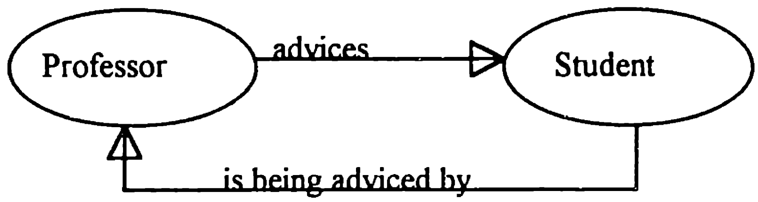


Figure 2.3 - Object Relationship

Relationship are usually attributes in classes, used as pointers to link different objects. A primordial example of a relationship can be found in the network databases (like CODASYL) to link the pieces of data belonging to the same set.

2.2.4 Identity

Identity is described by Khoshafian and Copeland as:

> *Identity is that property of an object which distinguishes it from all other objects.*[7]

Object-oriented systems are identity-based. The identity of an object is preserved even if two objects have identical values, attributes, and relationships. This concept is very useful for the notion of ownership of the data and the notion of sharing, which allows having references from various points to the same object. A similar concept does not exist in RBDMS without an explicit, unique identifier.

' :f. Booch, pg. 84.

2.3 An object-oriented approach

Different programming languages have utilized "object features" in the past, but the new approach developed in object-oriented programming and applied to database management comprehends all these features in a synergic way. Object-oriented concepts like *encapsulation* and behavior used together can give the database dynamic responses and other capabilities never implemented before. As will be described later, the object-oriented approach will provide the possibility of handling complex data in a much simpler way.

In the next pages, several features of the object-oriented approach are analyzed: encapsulation, abstraction, hierarchy and inheritance.

2.3.1 Encapsulation

Oscar Nierstrasz writes:

> *We have put forward the proposition that the term object-oriented is best interpreted as referring to any approach that exploits encapsulation or "packaging" in the process of designing and building software.*[8]

Encapsulation is extremely important in computer science to reduce software complexity by organizing data and code into software objects. This process can be performed by decomposing large systems into smaller encapsulated subsystems that can be more easily developed, connected and maintained. Another strong characteristic of encapsulation is the functional distinction between two set of codes of the objects: one that can be activated directly by the user and one that can be activated only within the object itself. Therefore, in OOP languages that support encapsulation, there is a limited set of *operations* that represent the visible interface of the object's procedures - called

---

[8] ref. Kim, 1991, pg. 17.

public interface - while the details of the data structure and how the procedures are implemented is kept private. This will restrict the access of the user to only the public attributes, while the system programmer would be the only person that can access the private attributes and methods. This approach will allow the

2.3.2 Extendibility

An object-oriented approach also relies heavily upon extendibility, in terms of behavioural extension and inheritance. Extendibility refers to the ability to expand an existing system without introducing changes to it. Once again, it is a particularly powerful feature for developing, evolving and maintaining large and complex software systems. The behavior of an object may be extended by simply including additional procedures, which does not affect the validity of all the existing structure and codes. To clearly define the inheritance, it is important to define two other fundamental concepts of OOS: abstraction and hierarchy.

2.3.2.1 Abstraction, classes and instances

The purpose of abstraction is to focus on the peculiar characteristics of a group of objects, that can distinguish them from all the others groups of objects. In this way, abstraction involves concentrating only on the development of essential structure and behavior and deferring other detailed description to a later stage[9]. This approach will reduce the complexity of the analysis, by introducing, at every stage, a limited number of variables and specific characteristic. Therefore, if for example, we want to analyze the performance of all the Ford cars in New York, it can be very helpful to divide the analysis into several stages, increasing step by step, the level of detail. We can first create a group of object by identifying the general characteristic of a vehicle, then we

---

[9] ref. D. Sriram 93, pg. 13.

can define the specific properties of a car. At this stage we can divide the cars by the manufacturer: one group of objects will represent all the Ford cars, then it will be possible to go further by defining a limited number of objects to identify every Ford Model. Then, for every model, define the objects for a specific year, and so on (see Fig. 2.5). Similar objects can also be grouped in *classes*. A class is a set of objects that share a common structure and a common behavior. Examples could be: the class of the Ford_car (containing all the car made by Ford) or the class of Ford_Escort_90 (containing all the Ford Escort made in 1990), and so on. The objects belonging to a class are called *instances* of that particular class. This abstraction procedure allows the development and use of abstract and logically complex object. Booch defines abstraction in this way:

> *An abstraction denotes the essential characteristics of an object that distinguish it from all other kinds of objects and thus provides crisply defined conceptual boundaries, relative to the perspective of the viewer.*[10]

2.3.2.2 Hierarchy and Inheritance

An object-oriented approach further promotes the reuse of information. The attributes and the behavior defined in an object can be reused to define more specialized objects. As we have seen in the last paragraph, objects can be organized on different levels of abstractions. An *hierarchy* is a set of abstractions. In the previous case, the hierarchy shows the different levels of details to represent the objects (see fig. 2.5). All the characteristics of a Ford car, defined in the class Ford_car, are inherited into the new objects of the class Ford_Escort, representing one of the model of a Ford car. The same object representing the Ford car probably has inherited part of its attributes and behavior from the class *car* (like number of wheels=4, composed by=engine, chassis, etc...). This process is called inheritance and defined in a much simpler way complex object, using

---

[10] ref. Booch, pg. 39.

attributes and behavior of an object of a superior level of abstraction and adding only the addition specification needed. At instance inherit by default all the attributes and methods of the class object from which it has been generated. In addition, it can have other attributes and methods on its own, and assume dynamically various forms during runtime.

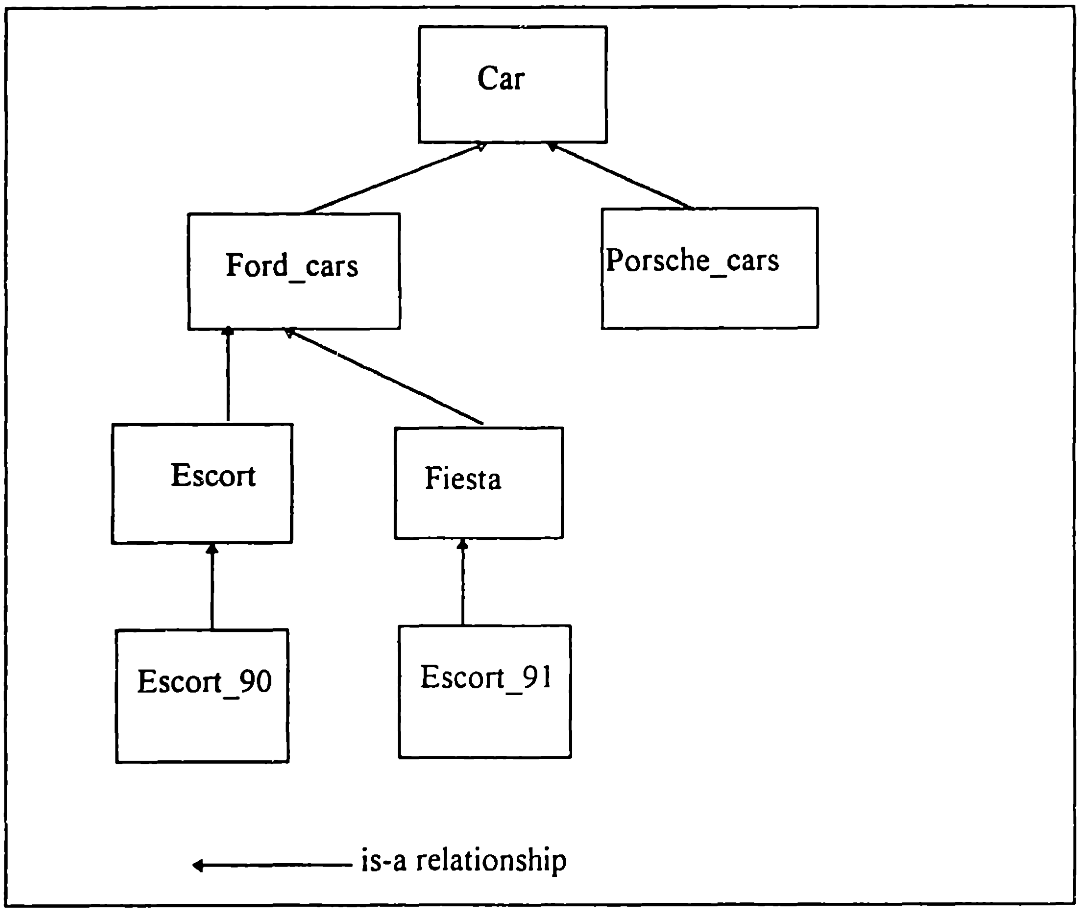


Figure 2.5 Hierarchy

2.4 Object-oriented Databases

As computer science develops and programming languages expands, database technology must be able to handle more and more complex data and issues. Databases should be more than just a static repository of data: they should interact with to the main applications in a more dynamic way. If the initial role of a database was only to store information in an electronic format, now the database must contribute to the performance of the system with query capabilities, efficient structures, etc. The next generation databases, whether they are object-oriented or not, must include the requirements described in the next section. This need and other developments in Artificial intelligence has brought to the birth of Object-oriented Databases.

2.4.1 Next generation Database.

The following capabilities represent some basic useful features that have to be provided to the users. Any conventional database system may be extended with these features, without utilizing the concepts of behavior, identity, encapsulation, abstraction and hierarchy[11].

1. **Definition of arbitrary data types**. At present the data are not only numbers or text, as was the case originally, but also audio, images, drawing objects, and many other types of data. The definition and manipulation language for these new types of data must be easy and efficient.
2. **Long Data Modeling capability**. The database must be able to store, retrieve (and ideally also manipulate) long data[12]. Effective techniques must be supported to minimize storage space for long data and the time for transferring long data between main memory and secondary storage. The possibility to combine the process with a conventional query capability is also useful.

---

[11] For a more complete discussion, refer to Kim 1991.

[12] Long data is a common expression for every type of data with high storage requirement (essentially every data but text and numbers).

3. **Complex nested object**. As said before, the database technology has to handle different and complex types of data. The ability to represent and manipulate complex nested objects is essential. The representation of data produced by CAD/CAM/CAE/CASE applications is normally complex for the various functional or structural links. Message passing should be efficient. Applications, like scheduling or CAD/CAM, also require simulation capabilities, at different levels of detail.
4. **Semantic schema design**. The database must have the ability to represent and manipulate various semantic modeling concepts for use in applications. One such concept is the assembly-part hierarchy in the context of computer-aided manufacturing, open documents, etc.. Since different users prefer different semantics, it is very difficult to capture the full range of semantics of these modeling concepts correctly.
5. **Rigorous constraint management**. The ability to specify rules and extended constraints is essential in order to provide support inferencing and constraint management, two basic mechanisms for supporting knowledge-based applications. The database system must have an efficient unification mechanism to support forward and backward rule processing, and it must also provide for efficient implementation of highly flexible constraints on the contents of data and the relationship among data.
6. **Data versioning**. The database must be capable to represent and manage changes over time, including a variety of concepts: time, time interval, versioning of single and complex nested objects, and versions of the schema entities.
7. **Cooperative work environment**. With the development of distributed computing, it becomes very important to be able to manage long-duration cooperative transactions. In interactive, cooperative work environments, the notion of seriability is inappropriate for maintaining database consistency. A more flexible definition of

database consistency is necessary, one which would allow flexible and concurrent sharing of common information among multiple users and deal with versioning, notification and merging when data values change while others work with the same data.

2.4.2 Feature of Object-oriented Database

Kim has defined an OODBMS step by step:

> *A data model is a logical organization of the real-world objects (entities), constraints on them, and relationship among objects. A data model that captures object-oriented concepts is an object-oriented data model. An object-oriented database is a collection of objects whose behavior and state, and the relationships are defined in accordance with an object-oriented data model. An object-oriented database system is a database system which allows the definition and manipulation of an object-oriented database*[13].

Object-oriented database technology has developed from different fields, such as object-oriented programming, data management, Graphical User Interfaces (GUIs) and artificial intelligence. In the last paragraphs, some fundamental principles (abstraction, encapsulation, hierarchy, inheritance, and identity) have been described. Precedent DBs have used some of these concepts. CODASYL, for example, has introduced the concept of *set* relationship, which include *hierarchy*. Some authors underline a difference between object-based and object-oriented database. CODASYL id fundamentally object-based. While many database systems use some of the object-oriented concepts, the utilization of behavior integrated into the data definition represents the uniqueness of the object-oriented database.

While the data model will be discussed in Chapter 5 - after an analysis of the specific needs of the Computer Aided Project Management (CAPM) in Construction - at this point the features of an object-oriented database are discussed.

---

[13] ref. Kim 90, pg. 9-10.

Database technology has followed in a certain way the technology of programming languages. As the languages have become more and more expressive, databases have developed more declarative and complex structures.

It is widely acknowledged in research environment that OODBMS will be used for next-generation database applications, such as CAD/CAE/CASE/CAM systems, collaborative engineering applications, knowledge-based systems, multimedia information systems and advanced human-interface systems. The reason for this general acceptance can be seen in the list of requirements in section 2.4.1; it is clear that some of the features required by next-generation databases are already part of the core concepts of an object-oriented approach. These object-oriented concepts now include such data modeling concepts, found also in conventional database languages, and grouping objects in a class, aggregation relationship between a class of objects and classes of objects specialized from that class. The definition of arbitrary data types, long data modeling capability, complex nested object and rigorous constraint management - listed above - have already been integrated into the object-oriented approach.

As for the semantic schema design in the database area, research into semantic data models has led to data modeling concepts similar to those embedded in object-oriented programming and in knowledge representation languages. The class concept captures the instance-of relationship between an object and the class to which it belongs; the concept of a subclass specializing its superclass captures the generalization (IS_A) relationship; and the composition of an object in terms of its attributes captures the aggregation relationship. Also, the concept of identity is the basis for both data versioning and the use of the DB in a cooperative work environment.

In addition to the capabilities discussed above, there are a number of additional characteristics, unique of the OODBMS:

1. **Data Abstraction**. This feature allows the development and use of abstract and logically complex and flexible data types

2. **Powerful information modeling capabilities**. Information is modeled in the form of classes and objects both of which represent and capture the structure and behavior of real world entities in the computer environment, making them ideal for simulation and design purposes.
3. **Object identity**. OODBMS defines and maintains unique identifiers for objects. This characteristic allows "equal" objects (which have the same name, attributes and equal attribute values) to coexist; it frees the user from the need to define unique keys for entity instances;
4. **Message passing**. The interaction of objects by the invocation of each other methods - a feature that enhance simulation.
5. **Encapsulation and data-hiding**. Objects are manipulated by operations that are defined by their types, and the inner codes are hidden from external access (or sometimes inaccessible as in certain languages implementations). The implementation of these operations may change without invalidating their use;
6. **Intelligent data**. The encapsulation of procedures along with the data provides the database with an ability to reason about its domain, consistency, validity, etc. This feature enhances the capability of the program to perform intelligent operations by sending message to other objects on its own initiative.
7. **Inheritance**. This characteristic allows the reuse and/or incremental definition of new class structures in terms of existing ones, and enable the transfer of information to derived objects.
8. **Polymorphic data and functions**. This feature enables the object to respond differently according to the forma of the data. The variety of behavior depends on the methods of the object, of its class and of its superclasses.

However, the lack of a universal standard (such as the normalization rules for relational databases established by Codd's) has, in a certain way, slowed down the implementation of object-oriented databases, especially for business applications.

Companies are afraid to utilize an OODBMS that in two years will become incompatible with the "standard" OODBMS in commerce.

Several research projects are under way to establish an OODBMS standard. One of the most important is being developed by the Object Data Management Group-93 (ODMG), a working subgroup of the Object Management Group (OMG), consisting of vendors of OODBMS, who represent more than 90% of the market. T. Atwood, chairman of Object Design Inc., one of the voting members, writes:

> *The ODMG-93 standard to be published this fall will play the same role for OODBMS that the SQL standard did for the relational era*[14].
>
> *....The standard consists of an object model and a set of programming languages bindings. The object model gives a precise, formal meaning to the basic concepts that determine how information is structured in the database - concepts such as object, attributes of an object, relationship between objects, and operations on objects*[15].

Atwood's view seems to be highly optimistic, but on the other hand object technology is no longer a new technology; it has already to his credit more than ten years of research and, because of its high potentialities, is likely to penetrate rapidly in the business applications.

---

[14] ref. T. Atwood, pg. 37

[15] ref. T. Atwood, pg. 40.

Chapter 3

Computer Aided Project Management

Computer Aided Project Management (CAPM) is a powerful and necessary set of tools required for today's competitive A/E/C industry. This industry is characterized by the need to process a huge amount of information. In construction, every product (i.e. a building, a road, pipelines, etc..) is unique with its own data to process. Kotanchik and Logcher write:

> *The reason that the construction industry requires a set of management skills and tools different from other industries is primarily due to the non-repetitive nature of its process.* [1]

Another reason for the complexity of a construction project can be found in the number of participants in a single project. For a mid-size project of 3 million $, there are about 30-50 subcontractors to add to the classical triangle of General Contractor (GC), Architect and Owner. Good management of information flow between the parties is critical.

In this chapter, the past history of CAPM, the trends that are shaping the development of the current project management system, its present use and a specific example of an advanced system, are described.

3.1 CAPM History

For a long time, construction planning and scheduling was based on the intuition and expertise of experienced site personnel, who took few notes on paper.

Project management systems (PMSs) have been available since the late 1950s with the advent of CPM, network techniques and "modern project management". These

[1] cf.. Kotanchik and Locgher, pg. 1.

mainframe systems were practically inaccessible to project control staff. The systems were absolutely not user-friendly, requiring input forms, which had to be prepared very precisely, processed by EDP staff, and returned much later to the user. The main application areas were accounting and financial management, estimate, CPM scheduling and inventory control. These applications were controlled by separate systems

In the 1970s, the general dissatisfaction of these inflexible computer systems and the development of the first DBMSs led to the birth of another generation of PMSs based on mini-computers and characterised by modest interactive methodologies implemented in large scale systems. Some interactions with time sharing systems increase the user responsiveness of the system and began to spread the idea of an integrated system. More sophisticated project managers began to take advantage of this improved responsiveness and flexibility and to understand the benefit of integration between different application areas. At that time the most successful general purpose PMS reached a few hundred units sold in the business environment, with typical installation cost of hardware and software ranging between 200 and 500 thousand dollars.

In the 1980s, these systems and research in the universities were the foundation of the PMSs based on the PCs. Most of the new capabilities stemmed from the use of sophisticated DBMS, with a lot of features not present in previous systems. The use of a powerful DBMS was one of the most important innovations and increased significantly their performance.

The reduction of software and hardware costs led to a wide expansion of use, bringing a PC to almost every desk. By 1984, hundreds of thousands of microcomputers had been installed and more and more project managers began to use these systems. This trend, supported by the use of single-user interactive PC systems, conflicted with the growing trend of integration, which was supported by broader, more integrated systems, based on a corporate DBMS. If the PCs have helped in spreading the use of CAPM software, lowering the hardware and software prices and making the cost/benefit trade-off more feasible, they have also brought more decentralised data and applications in several small stand-alone islands of work. As it will be described later, this trend of integration is now stronger than ever, and is now facing the fragmentation brought by the wide use of PCs.

In the last 15 years, software and hardware have developed at an incredible pace. The increasing speed of personal computers and new graphical interface software has completely changed the appearance and performance of these systems. Distributed computing has reduced the physical distances. The LANs and WANs have broadened the boundaries of the single departments and networking has made possible cooperative work. However, the basic approach to planning, underlying algorithms, and levels of system integration have not changed much since the introduction of PERT/Cost and financial systems.

3.2 Trends in CAPM development

It is very hard to forecast in a field like CAPM in the construction industry, where a fast changing technology like IT meet an old style type of production like construction. However, current research indicates that significant changes can be expected in the near future. This section will first analyze the general environment for CAPM and then discuss some specific trends in technology that are heavily affecting these systems.

3.2.1 General trends

The amount of information is increasing, as is the complexity and the performance of project control systems. Information needs to be analyzed, processed and filtered. In 1987, Logcher forecasted these changes in the following way:

Our future systems will be far more integrated and deal in a common manner with all types of project, environment, organizational and other data, using knowledge to integrate and utilize the data.[2]

With the exclusion of the implementation of the KBSs, which are developed more in the research environment than in the commercial application, this vision is happening right now and represents the general trend of commercial PMS software development. The possibility of using shared databases in design and construction is encouraging firms to integrate these functions in order to gain competitive advantage. A shared database and a Work Breakdown Structure (WBS) are the basis for an integrated control system, where different applications work on the same data. Architects will tend increasily to integrate design and engineering analysis. The building model will be developed entirely on the computer, with every block of related data (e.g. a wall) containing all the necessary information (e.g. dimensions, material involved, estimation code number, relationship in scheduling). These two trends are described in figure 3.1.

As it will be described later, some large firms have already implemented their systems in this direction. The three most important driving forces that are shaping the development of CAPM are integration, collaborative work and project modelling. These will be now analyzed.

3.2.1.1 Integration in DBMSs

Information sharing and the integration of software applications are the key points of the system development. A CII Design Task force committee has defined integration as follows:

Integration provides a common database that is accessed, used and updated by multiple application or users. The information in an integrated system is

[2] cf.. Logcher 1987, pg. 86.

organized in a logical way and demonstrates a centralized behavior with consistent and non-redundant data.[3]

The degree of integration of PC applications used on construction projects varies widely from company to company. From the most part, small and medium construction firms adopt different stand-alone applications only, not systems connected each other through use of common or related data. A few larger firms, as described with an example in section 3.3, do have a server, a LAN (see section 3.2.3) and a corporate database stored on a mainframe. CAPM requires integration, since project management tasks are inherently integrated. The integrated DB system must be able to handle an issue involving more than one aspect of the project simultaneously. Moreover, system integration is essential to do collaborative work and to implement efficiently the technological improvements listed in the next sections: communication links, graphical applications, simulation and knowledge based systems.

A DBMS can present different aspects of integration; the most important are:

1. **Hardware communication:** This involves the possibility of transfer data between two computers in different locations. Networks and communications links are examined in section 3.2.3.
2. **System software and application integration:** Different applications must be able to "talk" each other. The trend is toward having one detailed database and a number of specific applications. The DBMS should make it possible to process, filter and manage any data by any of these applications. Data should be able to go beyond Operating Systems and other type of software barriers (see section 3.2.3.2 and 3.2.5). Different programming technologies (e.g. KBSs and DB query languages) should be able to interact.

---

[3] cf.. C.W. Ibbs, K.C. Choi, Cost Effectiveness of Computerization in Design and Construction, University of California at Berkeley, Construction Engineering and Management, Technical Report No. 13, February 1989. in Teicholz, 1989, pg. 1

3. **Single data entry**: Every single piece of information should be entered into the system as soon as possible (*i.e.* as close as possible to the source of it) and only once. Redundancy implies vain effort and more expensive storage and maintenance cost. Building maintenance and facility management will be much easier if drawing specification, HVAC system, material and other related information are available in an electronic format. A connection of a similar database with a KBS will greatly increase the performance of the system.
4. **Monitoring**: The system must be able to monitor the flow of information coming from different applications and different functions, providing performance evaluation and analysis of the problem. This issue will be discussed in more depth in the next chapter.
5. **Information sharing**: More information flow will be necessary inside the company itself and between the different participants involved in the construction process. The difficulties of this information exchange in the construction industry and possible solutions will also be described in the next chapter.

The main purpose of a common database is to add value to the data already in the system, making this data useful to people other than the person that has collected or created the data. The database system itself also presents a typical economy of scale, for both the implementation and the maintenance, and requires significant organizational changes as well. Therefore, an integrated approach might discourage small contractors, who are not willing to sustain a considerable initial expense and a reorganisation to gain a limited benefit from the system. As the cost of the different components of the system decreases, it is likely that more and more medium and small contractors will consider it worthwhile to implement the system.

3.2.1.2 Collaborative work

To face the development of collaborative work, the multimedia interaction and the increase of users, the computer has also to perform new tasks. Froese writes:

> *... the computer will support future project management in the following three roles: First, the computer will be a supplier of, and medium of information. Second, it will act as a device for multimedia communication and coordination. Third, the computer will be used for processing information.*[4]

The project manager should be able to communicate with the owner, the structural engineer and other participants of the project through the computer itself. Changing orders, reviewing the drawings and other actions could be performed in an electronic format directly on the database. Moreover, inside the same organization, different people in different departments must be able to access the data simultaneously, without losing consistency.

To speed up the life cycle of the project, concurrent design will also be more and more common,. The system should support collaborative work, using two-phases locking or other similar features to assure consistency of the database. Security issues and ownership support should also be developed. These features are critical in the construction industry, where information is of great value and there is still the idea of a zero-game between the GC, the architect and the owner: access to the information will probably be under strict control for a long time to come. In chapter 4, it will be described how an OODBMS can provide a powerful set of tools to manage the ownership of data, giving the possibility to the person that has created the object to regulate and manipulate the access to that specific object.

---

[4] cf.. Froese 1991, pg. 8.

3.2.1.3 Project Modelling

The trend for integration will provide common access to all the information about a single project. This block of data will represent the model of the project itself and will develop over the different phases of the development of the facility. One such example is found in the Stone & Webster system called COMANDS, described in paragraph 3.4, where all geometric data are linked to a common project database containing several construction details. Project modelling must become more and more powerful to perform different tasks. Some of the general features follow:

1. *Preliminary model*: The ability to build a model prior to construction provides a prototype that designers can observe, analyze and review. The 3-D model will also provide additional information and general input on issues like interaction with the surroundings, impact on the environment and other similar issues. In most cases, it can replace the plastic model, which has been necessary for complex projects, and can add new characteristics. The new system will also provide walk-through capability, modelling in virtual reality and other highly interactive capabilities.
2. *Drawing Generator*: The preliminary model will evolve into a more detailed one that can be used for drawing production and review. From the 3-D model, it will be very easy to "slice" a 2-D drawing and add some specifications. At the same time, consistency could be guaranteed, errors will reduced and the visual evaluation facilitated.
3. *Planning*: The model can be used to simulate the building process before and during construction, and can help in the development of "what-if" scenarios to choose the most appropriate technologies and construction methods. The results, in terms of cost, time, and resources utilized, will provide a sound base to reevaluate the preliminary studies.
4. *Project Control*: Discussed more in depth in the next chapter, project control will be a major issue. A complete model will evaluate the work in progress, comparing it to

the planned model. This feedback allows for analysis of the problems of the building phase, and the identification of the similarities.

5. *Information Supply*. The model itself will provide a whole range of information regarding all the phases of the building process and different functions performed by the project manager. This information is available to different users at any time.
6. *Future references*: The model will also provide future access to all the information of a specific project that is based on the as-built data. Storage and future references will be easier to access and more reliable.

The model will increase the level of detail step by step, beginning with the definition of the geometry, if any, and adding during different stages the information required, through message passing (automatically by the computer) or by the operator. For a better understanding, it would be helpful to read section 3.4, where the project modeling approach of the Stone & Webster COMANDS is further described.

The whole database will be affected by the modelling capabilities. Through project modelling, programs can use the actual project as a framework for organizing information in the computer. The modelling capabilities are expected to improve significantly, especially with the introduction of OODBMS, where the data model is much more powerful and flexible.

A powerful model and the general representation of the project, with all the relevant data, will also be the base for the information processing made by the KBSs.

3.2.2 Hardware

The performance of the hardware, in terms of processing speed, memory capacity and graphical capabilities, will continue to increase, probably at a slightly slower pace than current rates. However, the price will remain stable, therefore increasing the ratio of performance/$ spent. The computer industry will continue to expand its market, reaching a percentage of utilization in the construction industry closer and closer to 100%.

PCs will be able to handle a huge amount of information, merging capabilities with what are now called workstations. Prices of laptops are expected to decrease and portable computing will become more and more common for project managers. New interfaces will assure the possibility of faster intercommunication between the main office and the laptop, by cellular phone or phone line. Interfaces and peripherals like Fax, modems, Scanners, Video phone will be much more common. The computer will be a multimedia station with the ability of interact more dynamically through text, audio and images.

3.2.3 Communication physical links: Local Area Network (LAN) and Wide Area Network (WAN)

Collaborative work is significantly increasing. In the next future, in the main office all the computers will be linked together in a LAN. The network will connect members of a project team and staff departments. The LANs will be linked to each other and probably to WANs to other branch or other elements of the system value chain. The speed of modems and of the line will make intercommunication easier and faster, with online interaction.

3.2.4 Graphical User Interfaces

Application of graphics is increasing in nearly every software application and CAPM will not be an exception. In the future, there will be adequate resolution and speed to display and manipulate 3D graphic models and *walk-through* simulations. Graphical Users Interfaces (GUIs) will welcome project managers in a more user-friendly environment, where every object and tool can be represented with an icon. Other multimedia features will integrate graphical ability.

3.2.5 Knowledge Based Systems (KBSs)

As the variety of data increases, PMSs may offer many new features and capabilities to the participants in the project. On the other hand, to make this vastly increased amount of data meaningful, it should be processed and interpreted. KBSs can perform many of these routine operations, like simple but voluminous checking. Increasing the system's intelligence will provide more useful information, adding value to the data present in the system.

A large part of actual middle management has as its main function the transfer with interpretation of data from and between the low levels of the organization and senior management. This approach has essentially created the hierarchy structure of several industries. Artificial intelligence will allow information to be transferred and filtered automatically, with customized KBSs, reducing the need for middle management. The senior management will have the necessary information ready on his screen. Coordination among project participants can be provided more effectively and more productively.

This change has already happened in construction industries like OTIS; the introduction of the OTISLINE system has almost completely eliminated the middle management and greatly increased the flow, collection and evaluation of information. The OTISLINE application is part of the OTIS Service Management System (SMS), an integrated DBMS. Prior to OTISLINE, the SMS database contained the customer master file (customer name, building location, contract information) and other information that was used to monitor and control the service business. The owner of the elevator had to call the local OTIS number, where his complaint was recorded on paper. Then the local office dispatched personnel to performthe repair. When the service was done, a paper report would be filed in the local office. With OTISLINE, the customer calls to an "800" number, where the operator in triage would transfer the call to the right specialist. He will use the computer log and send a dispatch to the nearest employee available. When the service is done, the report will go back to the central office, where an electronic file will be updated. With all the information from the service performed, the OTISLINE system is able to identify frequent problems, exceptions and create useful report for problem recognition and performance evaluation. Therefore OTISLINE is not only able to reduce response time, but it also provides an interpretation of the data collected, a task that before was performed (if done) by the local manager.

Current research in construction is developing the so-called Intelligent Agents, software packages which are able to capture the information at its source, transfer it, process it, and prepare the summary (with anomalies and overview results) for the upper management. Even if the management of a construction project is far more complex and unpredictable than other examples in manufacturing, the current effort toward the implementation of the KBSs and the support given by the OODB technology make it believe that these system will be realized very soon.

3.3 Current use of CAPM

Having analyzed the driving forces that influence project management software development, the thesis provides an overview of the current use of commercial project management software in the A/E/C industry. While many large firms have developed some of their own software and integrated software with DBMSs, most firms rely on vendor supplied software.

3.3.1 Typical uses of CAPM in the construction industry

In construction industry, CAPM utilizes information technology to perform and integrate several project management functions during the different construction phases. The areas in which project management systems are currently used are as follows:

- Project and financial planning
- Scheduling
- Resource management
- Estimating
- Project accounting
  - payroll
  - general ledger
  - accounts payable
  - accounts receivable
  - job cost accounting
- Purchasing
- Inventory control
- Resource allocation
- Equipment management
- Engineering and surveying support
- CAD/CAM/CAE

In 1988, a survey of more than 1700 G.C. provided statistics on computer use in the construction industry. Table 3.1 shows applications by decreasing popularity[5]: of the survey respondents 10% were small contractors (0-1 million $ annual volume of work), 58% medium-small (1-10 million $), 39% medium (10-100 million $) and 3% big (more than 100 million $). Half of these are in the building market, the others are divided into Highway, Heavy Construction and Utilities.

[5] cf.. Constructor, December 1988, pg. 35 and 36.

Table 3.1: The most popular applications in the Construction Industry:

Accounts Payable/Receivable	92%
Payroll	92%
General Ledger	90%
Job Cost Accounting	90%
Word Processing	68%
Spreadsheets	65%
Estimating	43%
Database	30%
Equipment Management	30%
Scheduling	30%
Inventory Control	15%
Document Control	14%
Personnel / Human Resources	12%
Purchasing	9%
CAD-CAM	6%
Engineering Design	3%
Other	2%

There are clear differences in application rates. Some of the reasons for these differences can illustrate the impacts of the lack of integration. The applications have been grouped in four groups:

1. *Accounting* data is involved in the first four applications: Accounts Payable/Receivable, Payroll, General Ledger, Job Cost Accounting, with a percentage of 90 or 92%. These applications are clearly widespread, involve a large volume of data, are in part required for legal and tax purposes, are used in many other industries and have therefore a well-developed, although slightly tailored software. Since these applications deal with mostly the same, well structured data, the use of some common file, a database typically, tie these applications together, thus improving productivity of data and people. Especially because of the need to present a precise tax report, the use of accounting applications is also widespread in the medium and medium-small contractors, and it is often considered more a need than a productivity improvement.

2. *Word processing and Spreadsheet* range from 65% to 68%. These often represent stand-alone applications, with a very low cost of implementation. They require almost no organizational changes and very little training. Because of their ease of implementation, they are common in all the industries. In site of their lack of integration, these applications are popular because they provide distinct productivity improvement to the single user.
3. *Estimating, Database, Equipment Management and Scheduling* represent a very useful class of application, but also one that's harder to implement. The use of these applications has increased dramatically with more user-friendly interactive software. Different scope and coding does hamper the integration of these functions with the accounting. In all the larger firms, the cost/benefit ratio is already low and they tend to integrate these function with other applications. In small and medium-small companies, it is not uncommon to find these operations performed by hand or not considered at all.
4. *Inventory Control, Document Control, Personnel, Purchasing, CAD-CAM, Engineering Design* software applications are normally used only in medium-large and large companies.

The reasons are explained more in depth in the next paragraph. However, in a general analysis, it is possible to highlight some key points:

1. applications in group 1 provide a high return if integrated each other. Similar scope and type of data make this integration easy to implement and profitable.
2. applications in group 3 or 4, in this order, have a lower return for data without integration. So, they are more likely to be implemented in large companies that will integrate them with the other applications, while few companies use them as a stand-alone application.

3.3.2 Different levels of integration

It is possible to see how for certain tasks, like accounting, computer support has become indispensable, while in others only a small part of the industry has adopted computer-aided techniques. This situation is clearly related to the size and the strategy adopted by the specific firm. A small contractor, whose strategy is to be a low cost competitor, without supporting a decent project control process, is likely to have minimal computer experience and unsophisticated management methods. On the other hand, in an construction company like G.H. Macomber, whose strategy is to differentiate itself with high quality and management of complex systems, the use of CAD applications is essential.

There are two types of economies of scale in the implementation of a CAPM software in construction. One, the classical one, is linked to size. The larger the company, the easier it is to divide expenses like training, LANs, software upgrades, and an information system division over a larger number of users, and thus reduce the cost/benefit ratio. The second type of economy of scale relates to the level of integration reached by the system. The broader the information gathered, the greater are the capabilities, features and the benefits that the system can offer. It is a synergic environment, where, for example, the major advantage of implementing an estimating software application could be the potential connection to the accounting software, and not merely the benefits brought about by the estimating software itself. In complex projects, this synergy has been show to be extremely powerful for simulation, *what-if* analyses, and early detection of construction problems.

With the decreasing cost of hardware and software, more and more companies tend to utilize information technology in different functions of project management, but very few, for the lack - or the very high price - of the appropriate software, tend to integrate all the software application in a single integrated PMS. Moreover, the use of information technology, and CAPM in particular, doesn't always stem from a rational decision or from the firm's business plan and goals. Instead it can result from the lack of involvement and distrust of senior management with PMSs.

3.3.3 Commercial Software

Available commercial software mirrors the different levels of integration used in different sizes of the construction companies. The software market for PMS respond to different needs. Some products are specifically for the A/E/C industry, while other are offered for a broader market but applicable to this industry. While there are many stand-alone applications, there are also few commercial products that already provide integrated PMSs, thus making noteworthy advances toward the PMS of the future. One example could be the ARGOS package, that allows the definition of a product model in 2D with ties to all information required for Accounting, Purchasing, and Estimating and some routines to implement scheduling. From the 2D model it is possible to visualize the product in 3D. Once the preliminary model it is defined, special routine will calculate automatically the estimate and produce the purchase order for the materials required. The program relies upon a central RDB, from which the single *element*, with all its data (geometry, material, estimate cost, cost code, manpower resources and machinery required for installation, and so on), can be retrieved, inserted in the design and modified, if necessary. The price of the software is high, but justified for a narrow market by its specific features. Its target market is large homebuilding contractors, where the purchase of the software could be easily paid off with a small number of retail sales and the product used repetitively to provide frequent reuse of information.

However, it is very hard for a small contractors to afford more than $18,000 for software alone, and then buy hardware and train staff. Therefore, they tend to buy cheaper software packages for single applications. This situation clearly hampers the integration of different functions. However there are inexpensive systems available even for the small contractors.

Timberline, Autodesk and Primavera System, Inc., for example, offer different packages of software for accounting and estimating, CAD, and scheduling, respectively. These firms have worked together to link their software together:

Precision Collection (Timberline): accounting and estimating software packages, able to handle the General Ledger, accounts payable and receivable, payroll, job cost accounting and estimating. It also allows data input in the *AutoCAD* (Autodesk) drawing format to estimate line items and quantity takeoffs directly from the drawing of the Design department.

Project Planner (Primavera): a scheduling software for generating resource-loaded schedules.

Other two possibilities are: a) the purchase of a commercial database to develop integration of applications or b) using lower level programming languages such as C or COBOL with a database to trigger the different applications. The trade off between the two options will be the development speed and coordination versus flexibility. While the first will provide a platform which is easy to implement but with limitation inherent to the specific software used, the latter can provide a greater flexibility, but would be much harder to implement.

3.4 Stone & Webster Integrated Management System [6]

A few large construction companies, like Stone & Webster and Bechtel, have developed integrated computer systems to support all the stages of large construction projects, from preliminary design, drawings and specifications, scheduling, etc.. These systems include 3D-CADD models of the project (with "walk-through" graphic simulation capabilities) that link to large mainframe project databases which, in turn,

[6] for a broader reference, see the special section of the bibliography.

connect to estimating, procurement, scheduling, and other construction support systems. These applications also use Knowledge Based Systems. To complete the analysis of the state of the art of computer aided project management, this section will briefly describe the purposes and capabilities of the integrated system developed at Stone & Webster Engineering Corporation.

The Boston based company engineers, constructs, and manages projects from small renovations to very large utility and industrial construction projects. The company has always been interested in CAPM and in 1989, its *Advanced Systems Development Services* division developed its new COnstruction MANagement Display System or COMANDS, software able to simulate the construction process to identify obstacles before a project goes to the fields.

Nevins and Zabilski write:

> *COMANDS is used to conduct 3-D construction process simulation by developing construction activity sequences of modeled plants, conduct automatic quantity take-off of modeled materials and equipment, link project schedule generation to 3-D construction sequence models, manage bulk material installation with trial warehouse withdrawal situations, and report construction progress by identification of completed components in the model.*[7]

The system is based on a IBM relational database (DB2) and on the graphic systems CATIA, a three-dimensional solids modeling application, and CADAM. In addition the database utilizes expert systems. Following the general trends described before, this software allows an integrated approach for cost, estimating, scheduling, purchasing, construction management and materials management. Integration is achieved through Stone & Webster-developed tools and routines that link DB2 with CATIA and other applications. Great attention is also given to project modeling. 3-D graphics have already proved to be extremely useful in giving more accurate information in the first phases of the design than possible with the 2-D graphic. The 3-D models are

[7] cf.. Nevins and Zabilski (1989), pg. 1.

also extensively used for interference avoidance, through CATIA's walkthrough capabilities.

The data model developed by Stone & Webster is based on graphical "objects". These graphical "objects" can be any physical or logical element, such as machinery, equipment beams, columns, etc. A graphical representation of each "object" can be stored in the geometric database. Table 3.1 is an example of how the information is stored in the database. Each "object" has an identifier and several pieces of data linked to it. All project information is stored in the central RDB, while the only information stored in the 3-D model is the geometry identifier. A graphical user interface assures a user-friendly environment, where the user can access any of the data associated with the object by identifying the desired object on the screen and clicking it with the cursor.

Component	Graphical Data (CADAM or CATIA)	Non-graphical Data (DB2)
	Object identifier	Object identifier
	Part Number	Specification number
Schematic,	Location	Purchase order number
Diagram,	Connectivity	Drawing references
Drawing,	Flow direction	Vendor Name
or	Dimensions	Spare parts inventory
Floor Plan	Orientation	Mainteinance History
		Preventive maintenance

Table 3.1 - Data model

Figure 3.2 illustrates some of the basic elements of the integrated system developed by Stone & Webster. The three basic components of the system: the graphic system, the RDB DB2 and the Expert System shell. The reason for this configuration is to store in the RDB only alphanumeric data. and to not overload the graphic model with all the information about the "object". At the same time, the program link between DB2 and

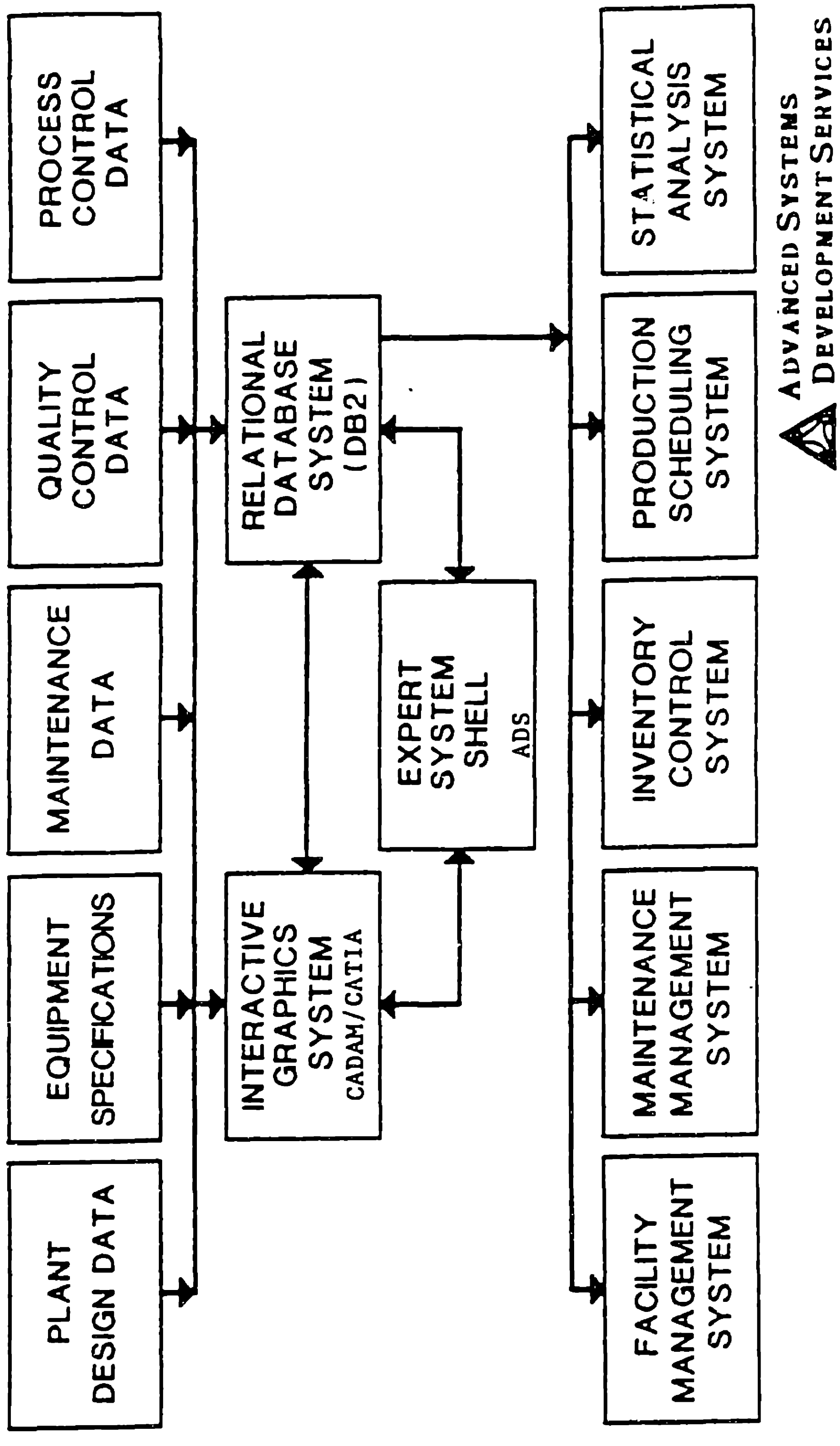


Figure 3.2 - Stone & Webster integrated system[8]

[8] rf. Reinschmidt, 90, pg. 8

CADAM or CATIA permits geometric and non-geometric data to be tied to the same entity. This model perform the tasks outlined in Section 3.2.1.3 about project modeling and provides software bridges for other applications such as structural analysis and pipe stress analysis.

As shown in Figure 3.2, the integrated database has the main function of communicating data to additional computer applications. The process will be automatic and error-free. Simple interfaces allow the transfer of data and the passage from one application to another.

The model is developed in the planning phase, the first step of COMANDS. Starting from its base components it is assembled by a project team, simulating the sequence of construction activities in the same order as the expected construction. Figure 4 is an example of construction sequencing on the lower portion of a turbine building. The preliminary model can provide the early identification of construction problems and interferences and can develop what-if scenarios to compare different possible solutions.

Another possible use of COMANDS is the study of the site layout (see Fig. 3.3). The model can show the existing facilities, new structures, as well as coal and limestone piles and the material handling equipment. The model is utilised to identify congested areas and sequencing problems in the early stages of the project; also here, the studies of different scenarios could be very helpful.

After the construction process is finalized, the system generates the scheduling. Each modeled component is assigned to an activity descriptor from the WBS, such as "Install Turbine Building Grade Beam". The activity descriptors are the linkages to the associated elements in the database. First, a list of activities that reflects the logic of the construction sequence model is generated. Then the accounts that are associated with the activities are assigned from the accounts list. Once the activity and account structure is developed the modeled components are assigned to an activity and associated accounts. "Install Turbine Building Grade Beam" and "Reinforcing Steel," Form work,"

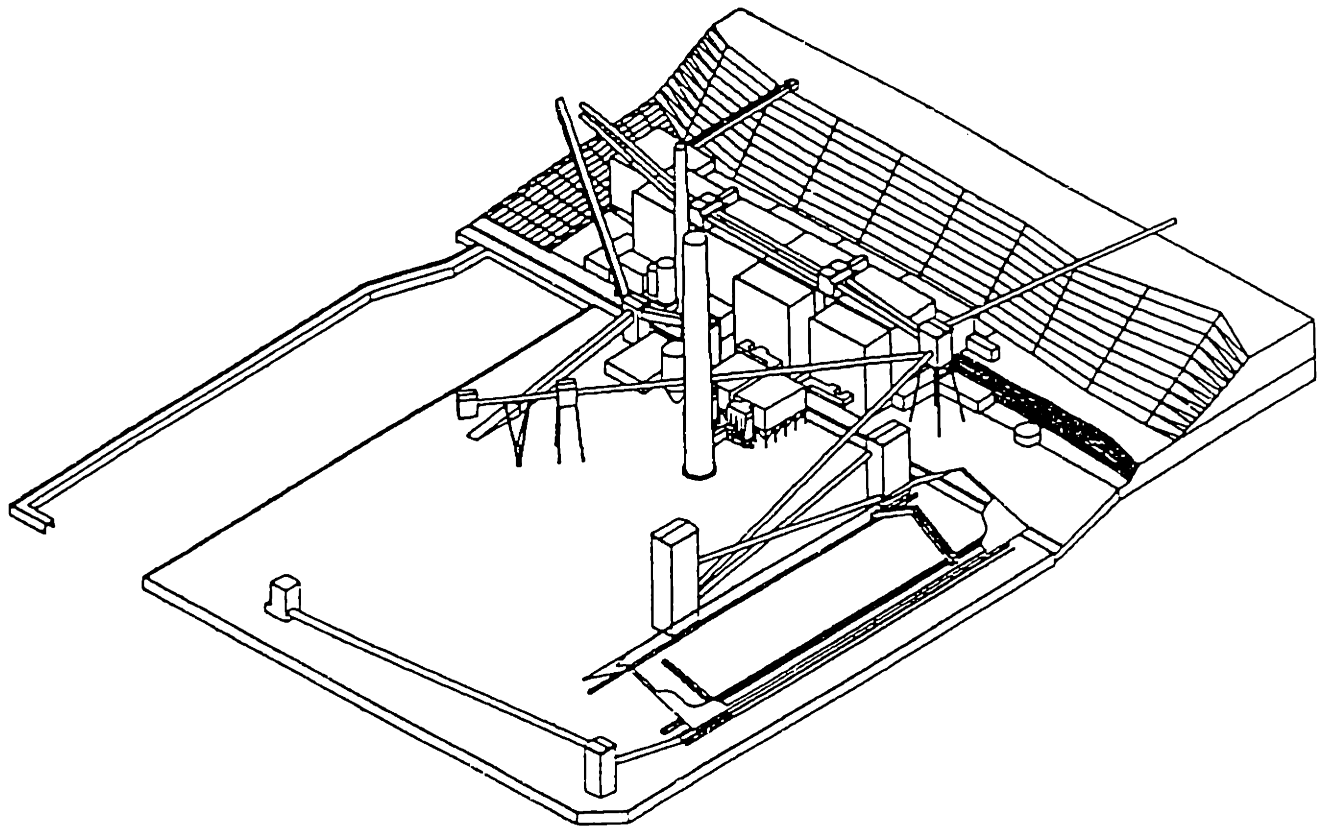


Figure 3.3 - Site Layout[9]

and Concrete are examples of an activity and its accounts. When the WBS has been linked to the 3-D components, quantities can be calculated by activity from the model's geometry and then summed by the account's unit of measure. The quantities are then multiplied by the account's labor installation rate, giving the total hours of labour required for that activity. By assigning a crew to that activity, the duration is calculated. The activity bar is then placed on a Gantt chart, visualized in CATIA and stored in the relational database. This information is used as a data deck to feed a network processor such as Primavera where the critical path and floats are calculated. Figure 3.4 provides an example.

[9] rf. Nevins and Zabilski, April 1989

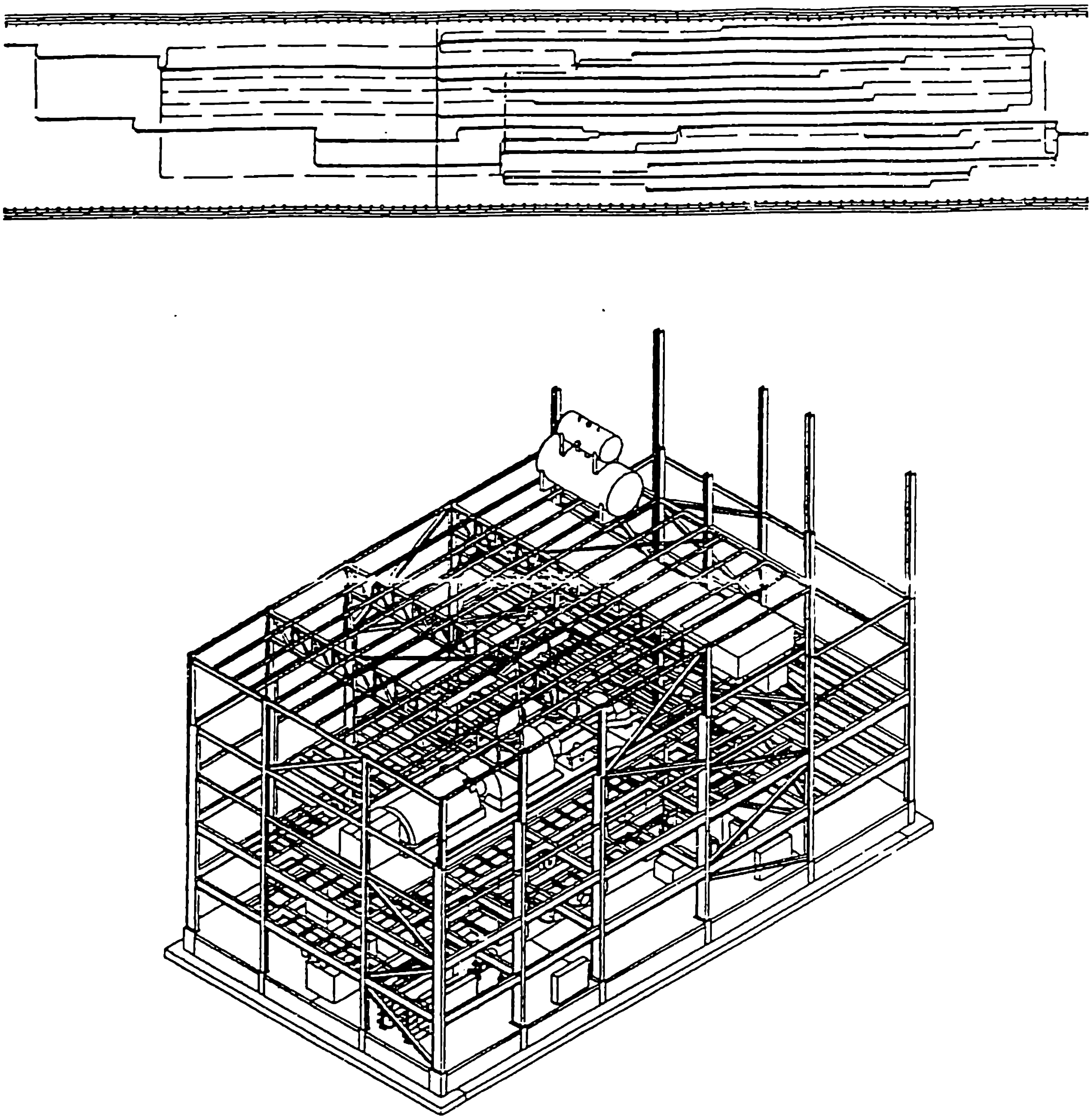


Figure 3.4 - Schedule Generation from 3-D Model[10]

[10] rf. Nevins and Zabilski, April 1989

The estimating and the materials management are provided through an automatic take-off and bill of material generated from 3-D models, as shown in Figure 3.5. The material requirements are collected in the database at the line number level. These data are then summarized into material summary reports that display the total material requirements of a model and cross reference these requirements to their source line numbers. The generation of these material lists requires loading of specification information into the database to establish the relationship between the model's components and their correspondent material grades. The inventory module contains the related purchase order data so that all purchase orders and promised delivery dates are incorporated into the database. The system will integrate the scheduling process with material available by using withdrawal simulation techniques to identify material shortages.

During the construction of the project, COMANDS is used to keep track of the material installed. The engineer selects the completed modeled components on the display screen, and the information in the database is automatically flagged as the components are installed. Actual installation labor-hours are then transferred from the payroll system and accrued to the activities. When the progress update has been completed for the current report period, COMANDS calculates the earned value of progress for each activity and automatically updates the percentage completed of each activity in the bar chart schedule.

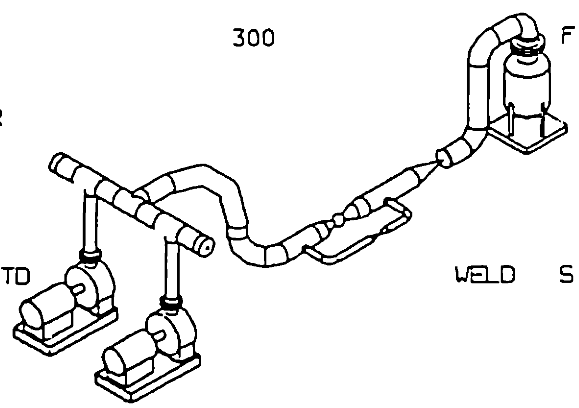
The system uses the Work Breakdown Structure (WBS), as a basis for: defining project scope, establishing project schedules, assigning resources, collecting progress and summarizing reports for various levels of management. The communication links provided by the system go beyond the connection of different applications inside S&W, providing an interconnection with the various project participants using the same hardware system, such as architects, engineers, suppliers, contractors and the client.

STONE & WEBSTER ENGINEERING CORPORATION
MATERIALS MANGEMENT SYSTEM
BILL OF MATERIALS

ELEMENT: *PIP27

PIPE LINE NUMBER: 1-ABC-010-7-301

COMMODITY CODE	QUANTITY	SIZE1	SIZE2	SCHED1	SCHED2	RATING	FACING	END TYPE	FAB FLAG
CB10135-1000	9.000	10		STD				WELD	S
PIPE CS. - ASTM A106 GR B									
CB20185-1000	1.000	10		STD		300	RF	WELD	S
W.N.FLANGE STD FIN BORE FS.ASTM A105 GR2 ANSI B16.5									
CB7007--01000614	16.000	1	6-1/4						F
STUD BOLTS A193 B7 CR MOLY STEEL W/TWO NUTS ASTM A194 GR 2H CARBON STEEL HEX.SEMIFIN.HEAVY SERIES ANSI B18.2.2									
RSWGAS-31000	1.000	10				300			F
GASKET FLAT RING SPIRAL WOUND TYPE 304 SS WITH FLEXICARB FILLER FLEXITALLIC STYLE CG OR EQUAL WITH 1/8 IN THICK RING ANSI B16.5									
CB30225-1000	3.000	10		STD				WELD	S
ELBOW 90 DEG SHORT RADIUS CS.SMLS.ASTM A234-WPB ANSI B16.9									
CB30265510000800	1.000	10	8	STD	STD			WELD	S
CONCENTRIC REDUCER CS.SMLS.ASTM A234-WPB ANSI B16.9									



PAGE 1 OF 2
FWD / BWD

Figure 3.5 - Bill of materials [11]

[11] rf. Nevins and Zabilski, April 1989

Nevins and Zabilski write:

Stone & Webster experience with IBM systems DB2 and CATIA has shown that the integration of the project database with the 3-D modeling and construction management process is a cost-effective tool. This is true for projects that are designed on the system and constructed using COMANDS, as well as for projects where the system is used only for construction management. The project where the system is used only for construction management. The project database is a tool that can improve communication and coordination among the owner, engineer, and constructor during project design and construction. Moreover, it is a permanent asset for improved operational management of the facility over its entire life cycle.

The description of this system is just an example of how some of the features described earlier in this chapter could be implemented. Integration has played a major role also in the S&W system. This system present some limitations derived from the relational database technology. The model itself require the division of the database in two parts: graphical and non-graphical parts. There are severe limitations for further enhancements with multimedia application. The model itself and the various software bridges, required to allow the integration of the different application slow down the system itself, lowering its performance.

Even it the S&W has put a lot of effort in securing the data through the implementation of the GUIs, the system still lack in the management of the ownership of the data.

The system can be improved even with the relational database technology, through the use of a more powerful KBS, giving particular attention to the processing and filtering of information. In the next chapter, the thesis will describe how further significant capabilities can be easily implemented through the use of OODBMS.

3.5 Conclusion

In this chapter, CAPM history, state of the art and current trends have been analyzed. Even if all contribute to the development of future PMSs, integration and information sharing seem to play the major role. The analysis of the GCs' use of information technology has highlighted the different utilization rate for different applications. With little or no integration, many project management software applications can't offer enough benefits in most of the construction companies, if implemented as a stand-alone island of automation. As a consequence, the companies are less likely to implement these functions with a computer application.

On the other hand, one of the key benefits of an integrated systems has to be found in the possibility of gathering data for one application and utilizing it in another one, adding value to the data and to the capabilities of the system itself. In fact, not only can the system improve efficiency with single-data entry or the reuse of data, but, as it will be described in the next chapter, data combined from different applications allows project monitoring and other features more efficiently and dynamically.

Chapter 4

Issues in project control: the role of an integrated PMS

The main reason that can push a company to implement a new technology is the need to increase productivity. The implementation of an integrated PMS is not an exception. In most cases, a company will not pursue the development of a new information system if it is not clear that it will increase the productivity and/or the quality of its work. Gathering data in a common database for an integrated CAPM system can add value to the data by allowing data gathered for one purpose to be used for others. In this chapter, this thesis will describe the goals of such a system, the problems it has to face and some suggestions to overcome them. An example will also be provided.

4.1 The goals of an integrated Project Management System

There are several ways in which an integrated project management system can increase the productivity. For a common DB, it is possible to identity three key points:

1. A piece of data, gathered or created by one user, once put in the system, can have value to other users of the system itself. The system must retain it and present it in the way that is more useful for every other user who may be interested in it.
2. Data combined from multiple users is more useful then separate groups of isolated data; it can enhance existing applications and create new system capabilities. It also avoids duplication in data collection and fosters an higher likelihood of consistency among data.

3. Data sharing is more complicated than just data storage. There is an opportunity to create some data, useful to the system or to the user, by interpretation and transformation of other data, already present in the system.

Paragraph 3.2.1.1 already has pointed out monitoring, single data entry and information sharing as some of the aspects of integration. They represent possible benefits that the system can achieve.

4.1.1 Information Monitoring

It is easy to reason about what should have been done or what should not have been done after the activity is finished or at the end of the project. It is instead a big challenge to be able to analyze early results to recognize problems and issues of construction in progress and to evaluate and forecast successive performances. The monitoring of information provided by the system, strengthen with forecast tools, could be very helpful to recognize as soon as possible mistakes made during the estimating and pre-planning phases and problems created by contingent factors. The system should be also able to provide the necessary information for making the right decisions for recovering from the situation. In fact, the monitoring of information is essential to provide:

1. statistic results to evaluate the previous work and forecast problems of the work in progress and similar future activities.
2. reports and graphics on a periodic basis, with an accurate projection of remaining cost, time and resource needed (trying to avoid the *Hockey stick*[1] effect).
3. additional data to assure data integrity.

Due to the complexity of construction planning and forecast, problem recognition needs a wide base of data. To discover the reason why the time or the cost expended for the

[1] This relates to the shape of a hockey stick - almost horizontal until the end when it suddenly increases the gradient to the actual cost in function of the percentage of work completed, stating that during the last 10% of work there is usually an unforeseen growing of the cost (after have been stable for the first 90%).

completion of an specific activity has been more than expected, it is necessary to have a range of data about the estimator of the activity, the foreman in charge of that activity, the availability of material on site and so on, to try to identify the probable cause of the delay or cost overrun. As it has been stated before, in general the broader the information, the better will be the performance of the system. With the right information and tools, the evaluation of the problems and delays after a certain percentage of work (variable between 10% and 40%) could be used to identify critical issues and prevent problems and delays in the last part of the activity or in activities that are likely to present similar issues.

As already seen in the project modelling section, the systems take progress and status data, compare them against reference standards such as budgets or schedules, providing useful information to the project managers and other parties involved in the project management. This information can provide the basis for evaluation of project personnel, early recognition of problems and delays, and an integrated view of cost, schedule, material and quality issues. The need for project monitoring and information processing are two of the main driving forces toward an integrated PMS that will be able to capture and process from the central database all necessary data for performing the various applications.

4.1.2 Single data entry and information sharing

Information monitoring requires an huge amount of data, but one of the major benefits of an integrated project management system is the possibility of entering each data item only once in the system, and being able to use that data for other application, without re-entering it again. In this way, the system can add value to the specific data gathered for a particular function, processing and using it in another functions at no extra cost. Useful information can be provided also from other participants of the projects. Co-operation between GC, architect and owner can be facilitated and enhanced by an integrated system.

4.2 Barriers to the implementation of an integrated PMS

An integrated project management system clearly presents very high benefits. But, at the present time, there are some barriers to its implementation. The barriers can be: organizational, of scope, technical or legal. This section will highlight the main ones.

4.2.1 Barriers in information gathering

Two main issues appear in the gathering of information for an integrated PMS:

1. Construction company employees gather and interpret information primarily for their own use or, sometimes, for their department. Companies have schedulers, cost engineers, etc.., that understand the meaning and use of a particular type of information and are not willing to collect and interpret information that they don't use or that they think it is not useful.
2. Owning information provides power. An employee with information needed by others is able to extract forms and exert influence over others who need or could benefit from such information. General availability removes this power. The transfer (or sharing) of such information will be resisted if there aren't other incentives to foster it.

One of main problem can be found in the lack of job descriptions, where data collection tasks that go beyond the scope of the department itself are often omitted. Therefore, as people establish their own individual priorities, it will be necessary to provide some incentives to assure that the user would collect all the data required by the system. Sections 4.3.1 and 4.3.3 will address this issue.

4.2.2 Information Sharing

An integrated information system needs a wide variety of data, from different sources, both within and outside the lead management company. Most of the barriers to information sharing come from the complex structure of the building process and the required interaction between the parties involved in it. The main issues are:

1. The presence in the same project of different companies with their own objectives (often in clear contrast with the ones of an another participant).
2. The discontinuity of relationship between the different companies involved in the same project makes it very difficult to implement a common strategy for increasing the use of a compatible information system.
3. The use of different, incompatible systems and software packages (for CAD, CAE, CASE, scheduling and accounting) also presents another barrier (the use of the standard format is increasing, but slowly).
4. Legal issues, not unusual during and after the construction process, that often require signatures and documents on paper, hampering the flow of information in electronic format (*i.e.* through the system).

If the G.C. has a detailed schedule of the project, the G.C. is very likely not willing to give it or its updates to the owner: it doesn't want to be controlled, and it would prefer to hide some slack for itself, avoiding public disclosure. When required to report schedule status, many G.C.'s maintain separate internal and external versions. Similar situations can be found in many other parts of the building process and, often, they can happen inside the same company: the estimating department will not allow sharing of its database or other information or expertise with the accounting department and vice versa. Moreover, without a central database, these difficulties can also be due to physical location.

4.2.3 The need for interpretation

Inside the project organization, different people have different information needs. Every user will tend to interpret the data he has gathered or generated keeping in mind his own needs. The benefit of integration for the implementation of applications that require significant manual interpretation is greatly reduced. This limits the number of companies that integrate applications such as cost control and scheduling, where the needs and the scope are quite different and there is a strong need for interpretation to produce input data. While databases can provide interfaces between different applications that allow data transfer between each other, they are still not able to interpret data created in or for other application to make them useful for project management analysis or applications that go beyond the scope for which that data have been gathered. These systems are often not able to quickly react to changes in design and field conditions or provide the basis for evaluation of project personnel. Section 4.3.4 will illustrate some possible solutions to enhance the capabilities of an integrated systems.

4.3 Possible solutions

The previous section has outlined some of the main barriers to the implementation of an integrated PMS. As the nature of this problems varies, from organizational to technical, so do the possible solutions. While this thesis has highlighted in the last chapter problems common to the whole construction industry, the analysis has focused on G.C.s. In this section, while the focus remains more toward G.C. organizations, many issues are the same for design firms and owner companies.

4.3.1 Redefinition of the job tasks

An integrated PMS requires significant changes in the organizational structure of the company and its information flow. To implement effectively an integrated system, the concept of the value of information must be introduced and people must understand that it is important to provide data to the system even if they will not benefit directly from this data. Every employee establishes his own priority in his own job. It is very unlikely that a task that goes beyond the needs of the specific department, like gathering data for the system or interpreting data for someone else, would attract the attention of a general user. This activity has no direct benefit to him in coming out his own responsibilities so that if it is not present in his job description, it will not have an high priority between his own tasks. Therefore, it is very likely that he will simply not do it or he will do it with no zeal and against his will, so that the accuracy and the timeliness of data will suffer. A solution to this problem is the review of the job definition: it has to clearly define the necessity of gathering all data needed by the system and input it, unrelated to his contribution to the department role, even if they can't benefit directly the user. Even better is to provide the employee with a useful combination of the data he is asked to collect with other data collected for the system. This provides the user with a concrete reward for supplying data to the system, making his job easier.

Careful development of job descriptions are required. Data gathering and interpretation of data must lead to an useful combination of data available to data suppliers and management. As we have discussed in the previous section, the need for manual interpretation of data is one of the causes of reduced benefit even in an integrated PMS, but the thesis will describe in the next chapter how some tools, like OODBMS, can be used to overcome this problem. The system itself can provide data interpretation, filter and process data according to customized interfaces.

One of the most significant changes brought by an information system is the flattening of the organization structure. The example given with OTISLINE in section

3.2.6, has already shown how the information flow can dramatically change with the introduction of an information system. If the system is able to interpret data generated at the lower levels of the organization and to analyze these data for the upper management, it has done a significant part of the work of the middle management. Moreove, the system can also provide an efficient way to transfer and interpret data horizontally, to transfer data interpreted by one manager to another, between different department and applications. This would also clearly reduce the people required for interpret data and the number of levels that the organization structure requires.

4.3.2 Early capture of information

One advantage of an integrated system is the possibilities to increase the performance of the project control through early problem recognition. To enable this feature and to help to overcome difficulties in information gathering, it is crucial to promote input of data into the system the earliest possible. In many construction projects, some data are collected on site on paper and, when they reach the main office, directed to one or more departments where these data are approved. These data is often not entered into a system until after approval. This mechanism slows down the ability to use this information, and hamper the possibility that the system can provide early recognition of problems in the construction, one of the unique opportunities of an integrated PMS. Moreover, it presents the risk of loosing information in the passage from the site to the office and, with the passage of time, the knowledge required to correct possible data collection errors.

In an integrated system, a partial solution to this problem can be provided if a PC is used on the worksite to keep track every day of the progress of the project in terms of activities performed, time spent, materials and man power utilized, and resources still available on site. The PC could be connected to the PMS in the main office. It would transmit data to the home office and allow the use of this information the earliest possible. Adequate precautions and careful job descriptions should assure the accuracy of the data gathering. Once the required precautions have been taken, the chances that a piece of information will be misunderstood or misplaced will be greatly reduced. In this way, the early capture of information will allow "real time" information monitoring, allowing immediate analysis on progress status, productivity and problem recognition. This will clearly represents an incentive, providing to the main office an easier handling of the data coming from the job site.

4.3.3 Incentives for information gathering and sharing

Adequate incentives must be provided both for information gathering inside the company and for information sharing with other companies. For example, the owner must provide the G.C. adequate incentive to give up its own schedule information, because the G.C. has disadvantages from doing that, mainly loss of flexibility in time control. These incentives can be a money premium (lump sum or in percentage), useful information that can increase the productivity of the G.C. or other forms of benefits.

The system itself can help the users to perform their jobs in different ways:

- provide an easier access to an user-friendly database, with all the data collected from other department and on site.
- provide information, gathered from other employees, and processed by the system for being useful to the user itself
- overall, increase their productivity and performance.

Moreover, at an organizational level, the idea of the department could be integrated in a more general view of the whole company, stressing the importance of the success of the company as a whole and not of a single department. From this point of view, the accounting will benefit from transferring its information to the estimating department also with a higher performance of the whole firm.

However, if we look at the cycle of information gathering and sharing, incentives for this, information processing and then from the beginning, there is a chain reaction effect. More and broader information the system gathers and processes, more it will be able to give the right incentives and show how it can benefit to the whole company (or the whole value system). Therefore, while at the beginning of the development of the system it will be necessary to give high incentives and to have the support from the top management, once the system has started, it should be able to give enough feedback to provide itself the right incentives.

4.3.4 Automatic data interpretation

Once pieces of data are entered into the system, the system should be able to use them for application other than the specific one for which the data has been entered. An integrated PMS can be strongly enhanced with two very powerful tools:

1. automatic data interpretation, able to create data useful to an application different from the one in which the initial data has been created.
2. automatic control procedures for problem recognition, able to analyze quickly data from different applications. A problem identified in one application automatically alert the other applications affected by it.

The primary objective of the first tool is to add value to the information already present in the system. At this moment, this type of data interpretation requires human interpretation, reducing the possible benefits of the system in terms of productivity. The main objective of the second tool is the earliest possible recognition of problems and the possibility of a quick response to the problems identified.

The issue of interpreting data is also present with the transfer of data to different level of the organization. For the system to be effective, it must be able to present different types of information at the different level of management. A project manager can need and use most of the data involved in the specific project he is following, but the

Vice President of Operation, that is interested in many projects and other tasks, would prefer to have fewer more significant data, in a summary format.

This thesis shows that, even if many improvements can be done with an integrated PMS, based on a RDBMS, there are some features that can be implemented, or are easy to implement, only with a OODBMS. One of these is the automatic interpretation of data, through a KBS approach, based on the behaviour and message passing of the objects of the database. The next chapter will illustrate with an example how object-oriented database can improve the automatic interpretation of data inside the system.

Chapter 5

An example of OODBMS approach: cost data interpretation

In the last chapter, the benefits, goals and issues of an integrated PMS have been analyzed. Some of these benefits are already present in systems like the Stone & Webster COMANDS, some can be enhanced by organizational change, a redefinition of the jobs or an alignment of goals. Others require a less common approach with the use of KBS and/or OODBMS. In this chapter, this thesis wants to provide some examples of the benefits within an integrated DBMS that can be brought by the use of object oriented technology.

Section 4.3.4 has highlighted how a powerful capability of an OODBMS can improve a PMS: the automatic interpretation of data. Essentially two major benefits have been identified: a) the possibility of analyzing data for early problem recognition and for the identification of their causes and b) the creation of additional data useful for project management by the system itself. The thesis focuses on the latter, illustrating, with some examples, how the system, provided with some cost and schedule data, can automatically create data that are normally collected separately. This chapter will first analyze the cost and schedule data needed, then explain the interpretation process and, finally, provide some specific examples.

5.1 Cost and schedule data collection

The need for interpretation is an issue common to many project management applications. However, the need to focus on specific examples that can well illustrate the potential capabilities of automatic data interpretation of the system requires the choice of a restricted area. The interaction of cost and scheduling data provides a fertile field to develop some good examples.

Cost control systems are widely used within the construction industry. These systems normally produce weekly and monthly report that summarize cost, budget, man-hour and forecast information against a chart of accounts. These systems are often well integrated with accounting systems, and sometimes also with scheduling and material management (as seen with the two systems Argos and COMANDS). The purpose of this section is only to clarify which data the system will process to create new data shown in the example in section 5.4. Table 5.1 list the construction documents produced in construction project management, with the person that is typically responsible for creating the data. Figure 5.1 examines the information flow inside the different functions of the company. Finally, the table 5.2 shows, for every document, the data used in the examples. The two tables and Figure 5.1 represent only a part of the general information flow.

5.2 Possible interpretation of cost data

One of the advantages of the OODBMS is to provide the data with its own behavior. Methods can model the behavior of the objects specifying the actions to be taken in different context. Methods, integrated with message passing, allow the creation of intelligent data: events such as accessing data can trigger objects to reason about the situation and react accordingly.

Event notification, in conjunction with intelligent data, provides the basis for the event-driven reasoning in which processing is performed based on events that happen within the system. For example, changes made to an object representing a construction

Table 5.1 Cost and scheduling document production

Responsible Party	Produces	using	Document
Estimator		Firm Database	Cost Analysis
		Material Supplier	Supplier Bid
		Subcontractor	Subcontractor Bid
		Designer	Blueprints
	Estimate		
Purchaser		Project Manager	Order Sheet
		Estimator	Estimate
		Material Supplier	Mat. Delivery Ticket
		On site inventory	Mater. Arrival Form
	Purchase Order		
Accountant		Material Supplier	Invoice
	Job Cost Report		
	Transaction Journal		
Scheduler		Estimator	Estimate
		Firm Database	Precedence Chart
		Subcontractor	Subs schedule
	Schedule		
Project Manager		Estimator	Estimate
		Scheduler	Schedule
		Accountant	Job cost report
	Order Sheet		
	Daily Log		

activity could result in messages that trigger a recalculation of the network dates; the same changes may also update a material procurement schedule or a cash-flow analysis. Essentially, the combination of behavior and message passing gives to the system the possibility of creating, starting from the interpretation of some data already collected, new data that otherwise would have to be collected. This will clearly add

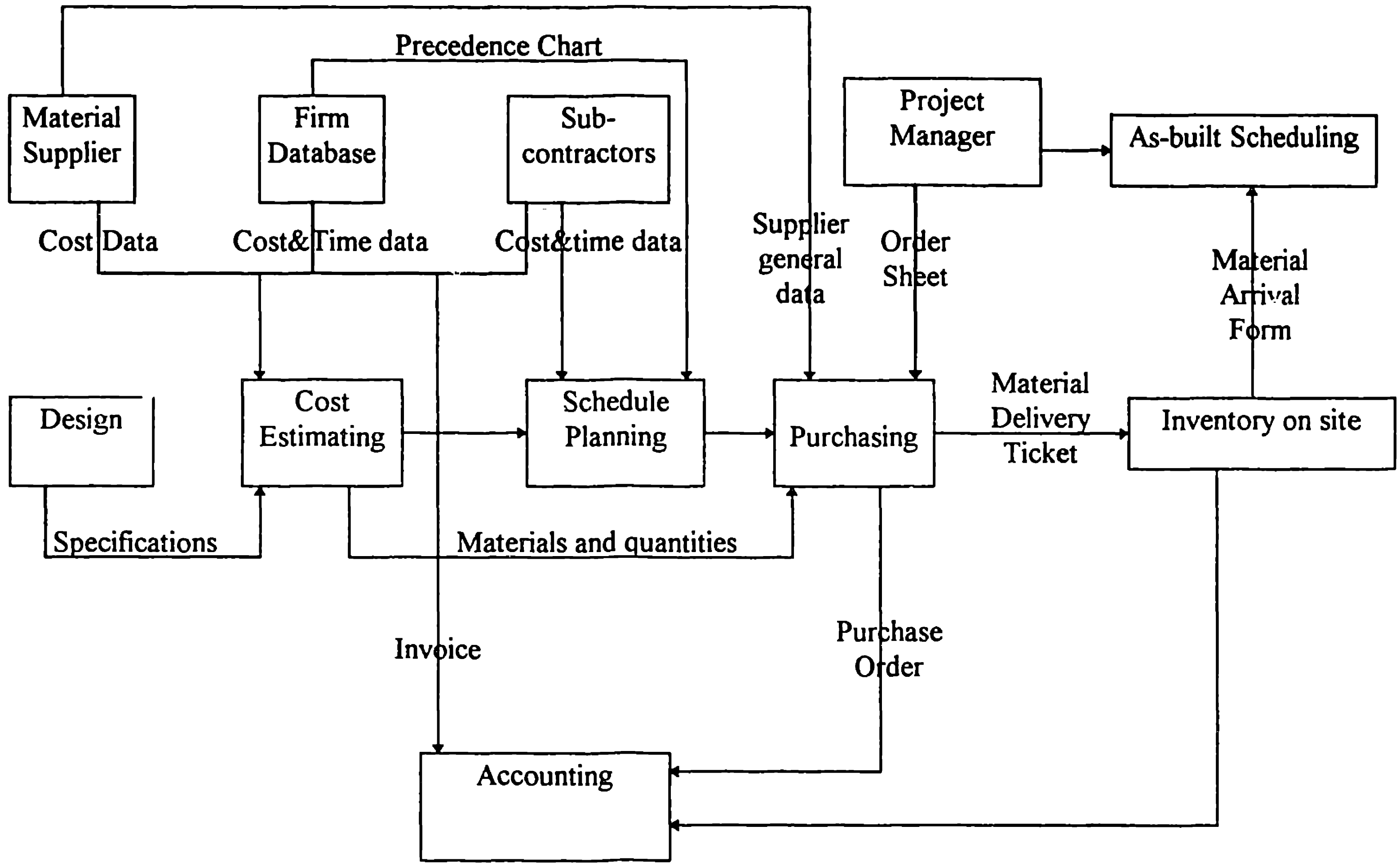


Figure 5.1 - Information Flow

Table 5.2 Construction Document Description

Document	Data
Cost Analysis	- Unit cost of material, labor and equipment - Quantities take-off
Supplier Bid	- Supplier Data - Unit cost of material - Days required for shipping and handling - Delivery date - Availability
Subcontractor Bid	- Subcontractors data - Unit cost of material, labor and equipment - Quantities take-off - Delivery date - Delivery duration
Blueprints	- Quantities take-off - Specifications - type of material and labor
Estimate	- Definition of Labor, Material and equipment - Units of measure - Quantities - Unit price - Cost.
Material Delivery Ticket	- Delivery date - Type of material delivered - Quantity delivered
Purchase Order	- Supplier Data - Type of materials - Quantities of the materials - Date of the order - Delivery date
Material Arrival Form	- Delivery date - Types of material delivered - Quantity delivered
Order Sheet	- Type of material needed

Table 5.2 - Document description - continued

	- Quantity needed - Delivery date needed
Invoice	- Type of material delivered -Quantity bought (delivered) - Delivery date - Unit cost of material and S&H
Subcontractor schedule	- Service performed (Type of production) - material, labor and equipment used - Duration and float of the activities
Job Cost Report	cost account related with activities
Transaction Journal	- Record of transaction - Accounts payable
Precedence Chart	- Activities - Activities preceding - Duration/Early start/Late start of activities - Early finish/Late finish/Total float of activities
Schedule	- Definition of the activities - Activities successor and predecessor - Duration/Early start/Late start of activities - Early finish/Late finish/Total float of activities
Daily Log	- Report of activities completed

value to the data already present into the system and enhance the advantages of the integration between the applications.

5.3 An object oriented integrated system

In the examples given in this chapter, different types of objects will be illustrated. It is not the purpose of this research to define all the objects, classes, attributes and methods of an integrated construction PMS. But, on the other hand, to fully understand the examples and the role of the single objects in the scheme of the system, it is important to give some principles of how a similar system could be structured.

Objects are usually designed to represent the actual objects (physical or conceptual components) of the domain. The identification of the objects required in the system and the definition of the requirements of each objects constitute a design for the system. This thesis assumes that there are three main elements of the object model: the libraries of general construction objects, libraries of specific project objects, and database applications objects. The objects of the general construction library represent categories of things like a resource type - labor, equipment and materials - construction activities and construction methods. The typical level of granularity is approximately that at which companies might keep productivity and unit cost data. The objects of the specific project library are the basic elements of each individual project. These include the project's specific physical components, the resource used, accounting data and the specific activities involved. These objects would be organized into various breakdown hierarchies, the depth of which is based on the detail available from or required by applications. The objects of the database applications have a behavior that can be triggered by time based events instead of accessing data in the object. These include control procedures to check consistency, monitoring other objects, etc.. This type of trigger might be used to schedule ordering of materials, for example.

In this approach, the domain primitive schema includes primitive classes that come from primitive entities identified in the domain analysis. The primitive classes are organized into separate class hierarchies, where each one involves a single concept about a component, a generic activity or a method. Users can combine primitive classes from these hierarchies to customize composite classes representing the objects needed for the specific project.

In an integrated system, the objects can represent many different things, like:

- a component of the project, like *window2# 23*, a window of type 2 (1.20x1.20m), on eastern wall of the second floor of the main building of the project.
- a category of components, the *window* itself, as a project-independent class.
- an activity, like *Place_Window_2.1*, describing the procedure, resources needed, location and other data regarding the actions for installing the windows of the second floor, eastern wall.
- a category of activity, like *Install_window*, identifying a general construction goal.
- a construction method, or a particular technique for performing an activity, *like pouring concrete with the pump*, with indication of the crew and equipment needed and productivity ratios.

and many others, like a company, a worker, a piece of equipment, and so on. These object will be organized in a taxonomy, where they will be divided in the product, process, resource and organization model.

5.4 Examples

The main idea in these examples is that the input of a data into the system will trigger a procedure that will invoke, through message passing, a specific event for an another object. That event may receive or create new data for that object and may, in turn, spawn another message to another object, that also can react accordingly. The objects can send more than one message to different objects interested in that specific events. The system must be designed carefully to avoid closed loops, or it should be provided with a security feature that identifies them.

The examples will represent different stages of the information flow of the project. Even if they are specific, they clearly identify general areas in which the automatic data interpretation is feasible and useful.

5.4.1 A scheduled activity can create a P.O.

In this example, the system will create a Purchase Order for a material starting from the data contained in the activity that utilizes that material. To better understand the benefits brought by an integrated system with an OODBMS, it is useful to review the current process. To make the example simpler and clearer, it will first be assumed that the materials used in this activity are not used in any other activity (a 1:many relationship between the activity and the materials) and that a single purchase order is needed for every material. Then the case of a many:many relationship, where the same material is used by many activities and the case of a purchase order with more than one material will be discussed.

After reviewing the specifications and the firm database, the estimator will decide on the construction method and design the work packages for installation of every material. Then he will contact the material supplier (or the subcontractor) asking for bids. Finally, he will elaborate an estimate. The scheduler, after having acquired from the estimator the quantity and the type of material needed and defined the crew productivity, will establish the time required for that activity. Then, knowing the activity constraints and timing for all preceeding activities, he can schedule the activity. Later, the project manager would look at the network plan and decide which activities he wants to order material for. Then he will contact the purchasing officer to order the materials, using an order form. At this stage, the information passed from the project manager to the purchasing department is:

a) type of material needed

b) quantity of material needed.

c) the date when he wants to have the material on site.

The purchasing office orders the material from the supplier, using the following additional data:

a) time required for shipping of the material

b) the unit cost of the material to be purchased

c) general data of the supplier.

If we refer to the Table 5.1, it is possible to identify the data needed to create a purchase order. Note that the quantity take-offs usually come from a combination of the knowledge captured in the database from previous jobs (sometimes structured in work packages), with similar activities, and the specific needs identified by the designer and the estimator for the specific project:

Responsible party:	**Source**	**Data**
Estimator:	Firm Database	Unit cost of material, labor and equipment
	Firm Database	Quantities take-off
	Material Supplier	Unit cost of material
	Designer	Quantities take-off
	Designer	Materials types and quality.
Purchaser:	Material Supplier	Days required for S&H
	Material Supplier	Delivery Date
	Material Supplier	Supplier Data
	Material Supplier	Unit Cost
	Estimator	Definition of labor, Material and equipment
	Estimator	Quantities
	Estimator	Unit Price
Scheduler:	Estimator	Estimate
	Firm database	Precedence Chart
	Subcontractors	Subcontractor schedule
	Scheduler	Activities constraints, Start date, duration,

In this example, the system will require, for every type of material, data on how many working days we want to have it on site before the start of the activity, and how many days the suppliers requires for processing the order and sending the material. So, every object representing a material will have already defined the attribute: *Days_for_supply* and every activity the attribute: *Days_prior_to_start*. The data needed for the creation of the purchase order will be in the following objects:

Activity

Name:	Place_Window_1.2:	
Attributes:		Start Date = 6/9/94
		Days_prior_to_start = 0
		Date_Site = 6/9/94
Relationships:	uses:	Windows2=32
		Frame1=32
		Ledge = 20
	controlled by:	CA_Place_Window_1.2

Cost Account

Name:	CA_Place_Window_1.2:	
Relationships:	uses:	Windows2=32
		Frame1=32
		Ledge = 20
	controls:	Place_Window_1.2

Material:

Name:	Window#2	
Attributes:	Days_for_supply =	3 (working days)
Relationships: used by:		Place_Window_1.2
made by:		Windows for All
Methods:	create_PO	

Vendor

Name: Windows for all

Attributes:	Supplier data (address, SSN, etc..)
Relationship:	makes: Window2

Purchase Order:
Name: PO_Window#2
Attribute: Delivery_Date = 6/9/94
PO_Date = 6/6/94
Material = Windows2
Quantity = 32

The procedure will be the following:

1. Once the estimate is complete, the system, with the constraints and the other data needed taken from the schedule, will calculate the *start_date* of every activity.
2. The system will accept the schedule as target and it will initialize the object *Install_windows_1.2,* an activity for installing the window on the first floor, with the indication of its *start_date*. The order from of the project manager will also indicate how many days before the materials have to be on site, with the attribute *Days_prior_to_start*. This attribute is set for the activity but can be overrun by a more specific one, defined in the material.
3. The object *Install_windows_1.2* will create the cost account *CA_Install_Windows* with material associated with the activity
4. The object will automatically send a message to the materials needed: *Window#2*, *Frame#*1 and *Ledge* with indication of:

   the quantity needed, ad example 32 windows, 32 frames and 20 Ledges.

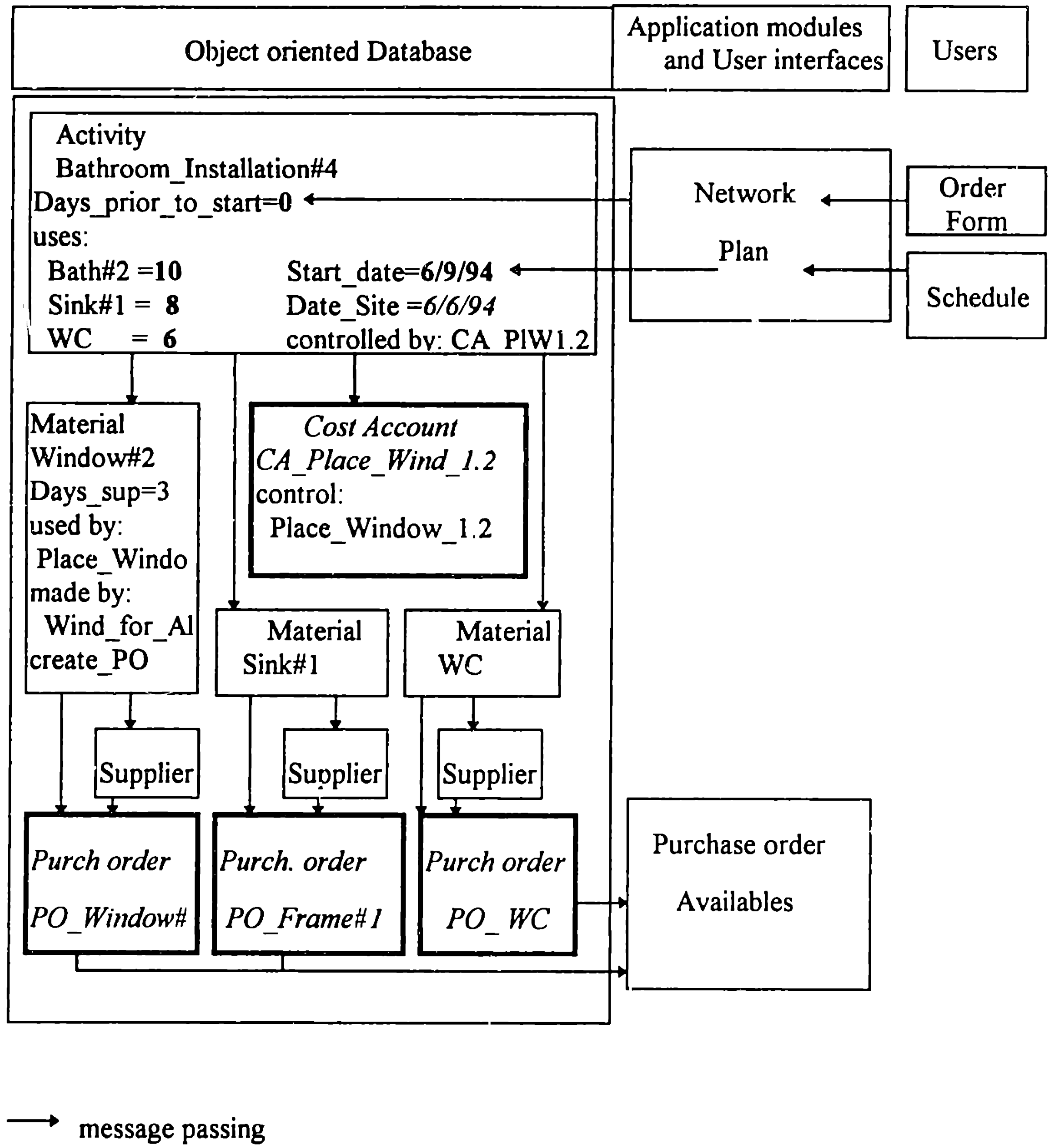
   the date wanted on site, *Date_Site* (obtained from the *start_date* minus *Days_in_inventory,* expressed in working days)
5. Each one of *Bath#2, Sink#1* and *WC* will create its own Purchase Order, with the procedure *create_PO*, using the data received from the activity, and its own, with the date to send the purchase order, *PO_Date* (obtained from the *Date_Site* minus *Days_for_supply,* expressed in working days), and the supplier data from the vendor object.

6. A purchase order will be created for every supplier, with the indication of the supplier data, type of material, the quantity, the date of on site delivery and the date when the purchase order must be sent.

The information flow is described in Figure 5.2. The procedure *create_PO* can be created in the class Material (a superclass of these materials). This approach will also allow an efficient control tool for material management, with the possibility of sending the purchase order as late as possible, reducing the possibility of schedule changes in the meanwhile. Moreover, setting the *Days_in_inventory* to zero, the system provides the management of a just in time inventory.

Other feature can be added. For example, if the material is used in more than one activity, the system can set a preferred number of units to order for every material, check the different activities that require the same type of material at a short distance between themselves and order these materials together for more activities. The object representing the material will use a linked list with direct links to all the activities that use that material. A similar list can be used between materials of the same supplier that can be included in the same purchase order.

Another example could be found in the class ConcreteComponent (subclass of Component), where the instances can be grouped per type of concrete needed (for foundation, structures in elevation, different characteristics). With a simple query on their attribute *Concrete_type*, the system will provide the total quantity of concrete, required by the project for the different instances in a specific day, and, with the same procedure discussed above, create a purchase order for the total amount of concrete, divided per type.



The data created by the user are in bold characters. The data created by the system during the procedures are in italic.

Figure 5.2 - A scheduled activity can create a P.O. - information flow

5.4.2 Arrivals of window on site

In this example, the arrival of the windows on site can automatically achieve two objectives:

- initialize the activity for which the windows are needed in the as-built scheduling.
- update the job cost account of that material and approve the payment of its invoice.

Even if the two actions happen simultaneously, to better understand the process involved in each of the two transfer and creation of data inside the OODBMS, the two procedure will be analyzed separately.

In the current process, when a material is arrived on site, a person on site (or the project manager) will record the delivery on a material delivery ticket. When an invoice arrives, the Accounts Payable person in accounting will send it to the job site for approval. After approval, the accountant will record the transactions in the Transaction Journal, placing the invoice into an accounts payable account. Finally, after the receipt of all the necessary materials, the project manager notes the actual start date and report this for the as-built schedule. The additional data needed are:

Responsible	Document	Data
Purchaser	Material Delivery Ticket	- Delivery date - Type of material delivered - Quantity delivered

5.4.2.1 Initialization of the activity

To initialize the activity, the following objects are examined:

Material:

Name:	Window#2	
Attributes:	status =	arrived
	Delivery_Date =	6/9/94
Relationships:	used by:	Place_Window_1.2

Activity

Name:	Place_Window_1.2:
Attributes:	Actual_Start_Date = 6/9/94
	status: active
Relationships: uses:	Window#2=32
methods:	resource_checking
	start_activity

The system will initialize automatically, the Activity *Place_Windows_1.2* in the as-built schedule, on the arrival of the last resource still not available: the *window2#*. Let's assume, at the beginning, that the *window2* are used only for that specific activity. The process will go through the following steps:

1. On 6/9/94, the material delivery ticket will confirm the arrival of the windows. The system will set the *Delivery_date* of the object *window2#* to the current date and change its status to *arrived.*
2. Upon the input of the date of the arrival, the object window2# will automatically send a message to the activity *Place_Windows_1.2* to notify that the material has arrived.
3. The activity *Place_Windows_1.2* will then activate the procedure *resource_checking.*
4. With this procedure, the activity will check if every other material (and resource in general) is already available.

   If this condition is true, then it will activate herself in the as-built status with the procedure *start_activity.*

   If the condition is not true, the activity will wait for the arrival of the next resource.
5. The procedure *start_activity* will update the *actual_start_ date* of the activity and change its status to *active*, having access to the as-built schedule.

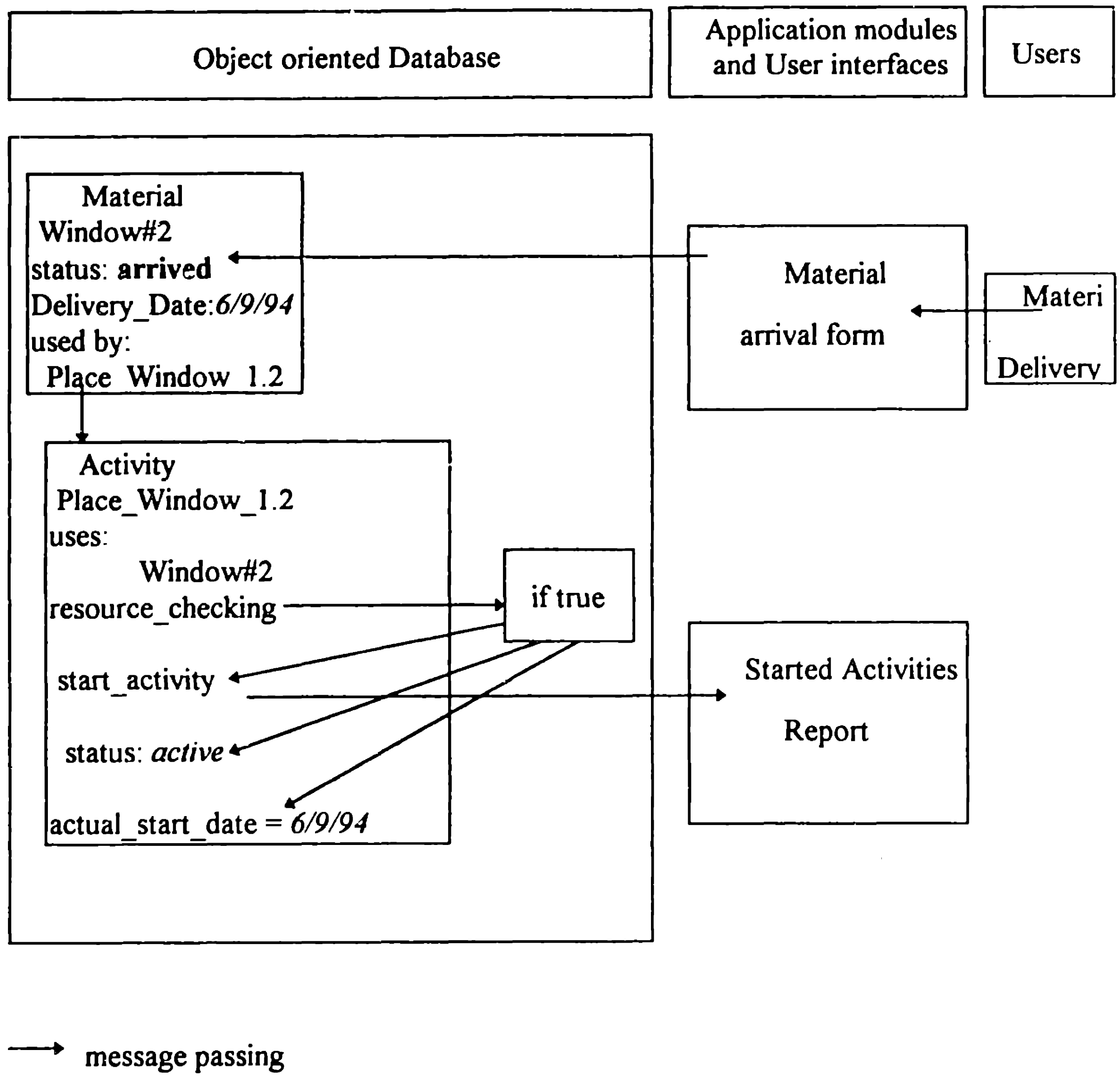
The information flow is described in Figure 5.3. In the case the materials are for more than one activity, it will send the same message to all the activities in its linked list for which that material is required. Other procedure can enhance this capability of the system. For example, the system can have a procedure checking if there is any activity, planned for a specific date, that is not yet activated, and if for the activities still inactive, alert the system with, eventually) the indication of the materials still needed.

5.4.2.2 Update of job account

On the arrival of every resource on site, the system will also update the transaction journal and the specific cost account. The process will go through the following steps:

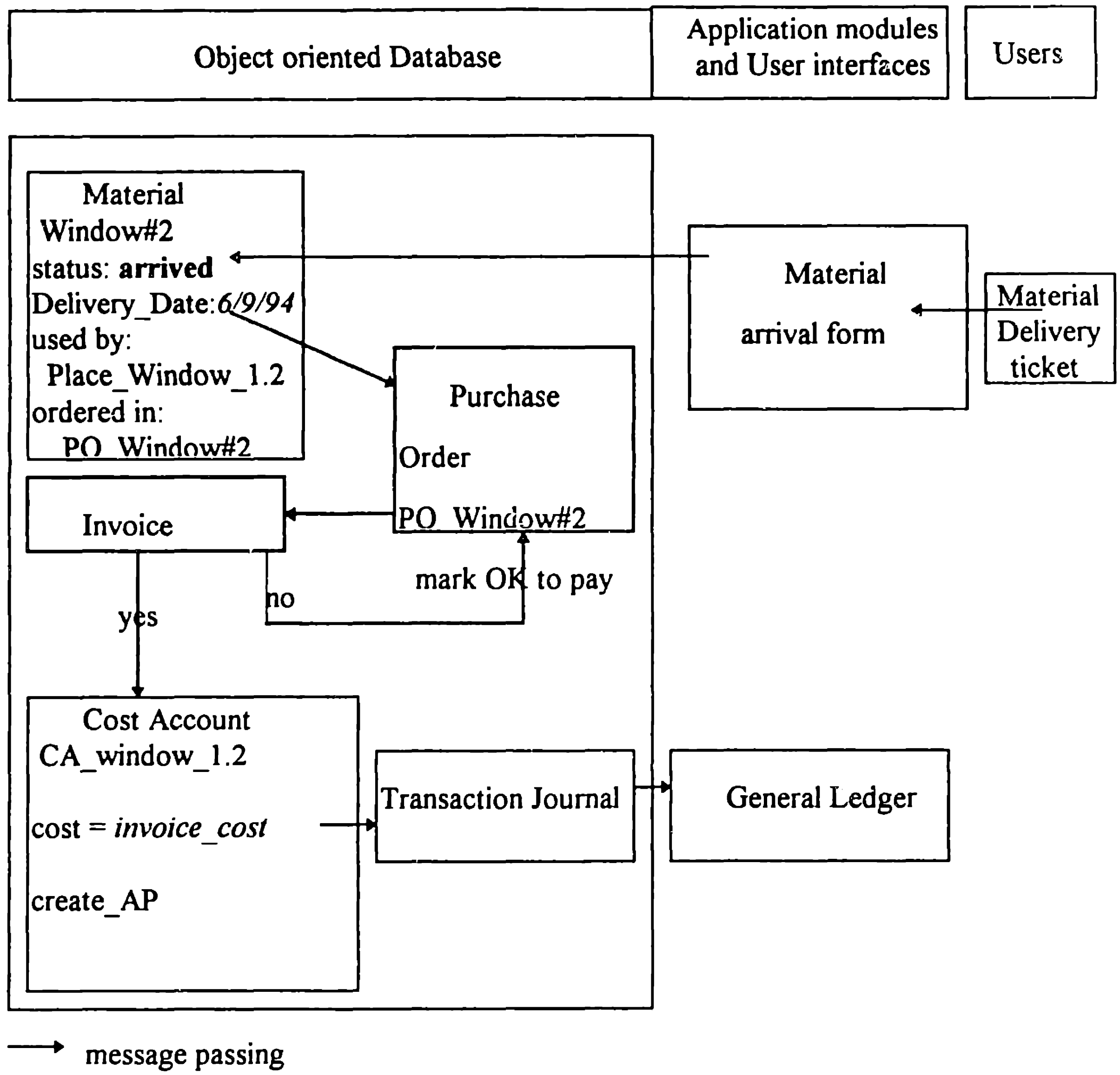
1. On 6/9/94, the material delivery ticket will confirm the arrival of the windows. The system will set the *Delivery_date* of the object *window2#* to the current date and change its status to *arrived*.
2. Upon the input of the date of the arrival, the object *window2#* will automatically send a message to the *PO_Windows2#* to notify that the material has arrived.
3. If the invoice has been received, *PO_Windows2#* will send a message to the cost account *CA_Windows_1.2*. It will update itself, and use the procedure *create_AP* to provide an Account Payable for the windows. Otherwise the PO will mark itself to "OK to pay".
4. If the cost account *CA_Windows_1.2* has been update, it will send a message to the *Transaction_Journal*.
5. The *Transaction_Journal* will update itself and send a message to the *General_Ledger*.
6. The General_Ledger will receive the message and update itself.

In the case the material is for more than one cost account, it will send the same message to all the cost account of the activities for which that material is registered. The information flow is described in Figure 5.4



The data created by the user are in bold characters. The data created by the system during the procedures are in italic

Figure 5.3 - Initialization of the activity - information flow



The data created by the user are in bold characters. The data created by the system during the procedures are in italic

Figure 5.4 - Update of job account - information flow

5.4.3 An activity that has been completed can close the cost account

Once an activity is finished, the project manager will mark the activity as completed. The accountant, after having received the daily log, will close the cost account related to that activity.

In this example, the system will automatically close a cost account as soon as the activity is completed. The additional data needed is:

Responsible	**Document**	**Data**
Project Manager	Daily Log	- Report of activities completed

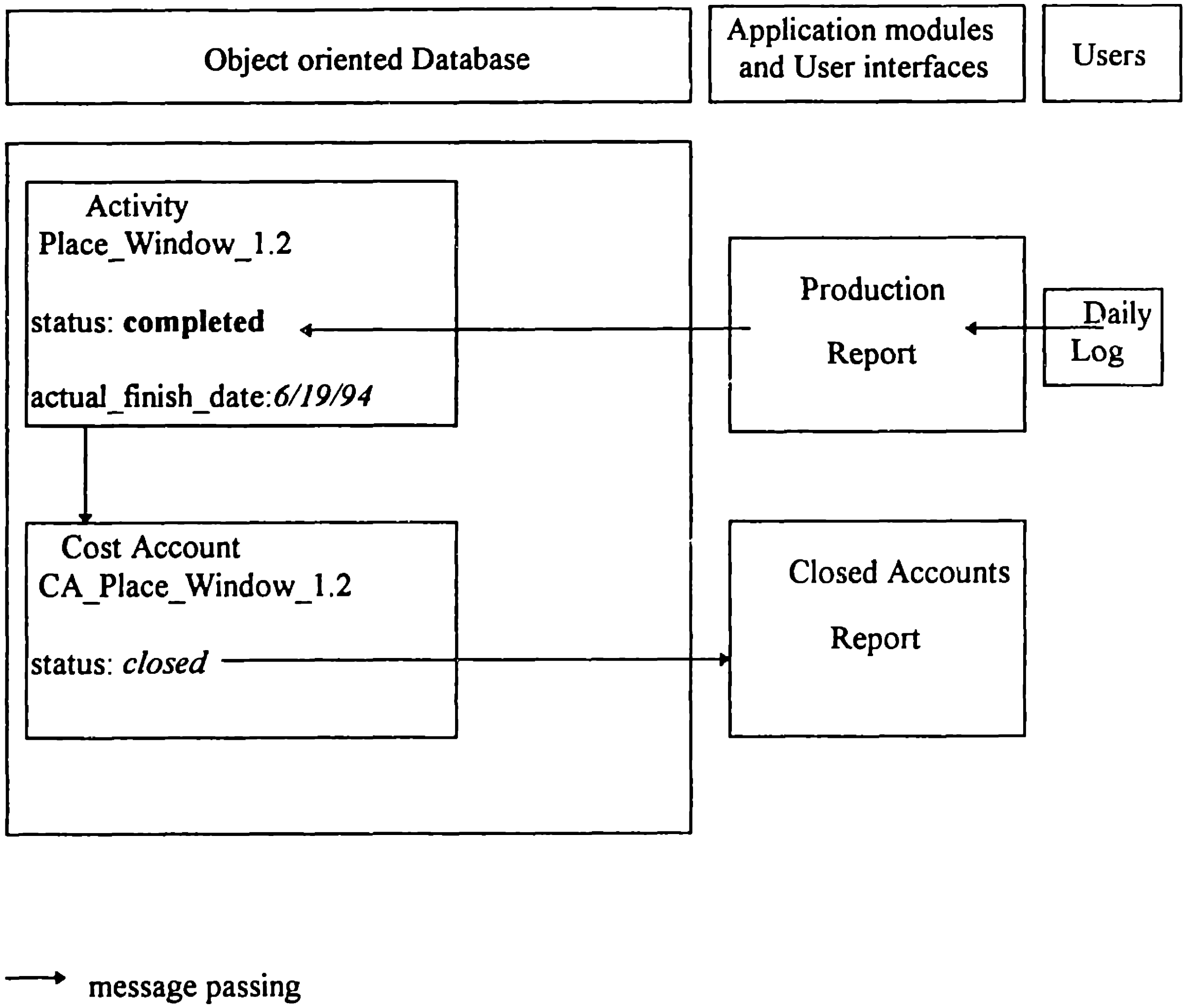
The procedure will be the following:

1. The user will input into the object *Place_Window_1.2*, the completion of the activity, changing its *status* attribute to *completed.*
2. The object will automatically send a message to the cost account *CA_Place_Window_1.2*, to change the status to *closed.*

   Additional features can include:

3a The object will automatically send a message to all the human resources and equipment it was using, to change their status to *available*.

3b The object will automatically send a message to the objects representing the materials it was using, closing the relationships with them (to indicate there is no further need from that activity).



The data created by the user are in bold characters. The data created by the system during the procedures are in italic

Figure 5.5 - An activity that has been completed can close the cost account - information flow

The four examples, that have been shown, want to demonstrate that the system can interpret the data, that has been input from the user, to create other data, used in different application. In this way, the data will go far beyond the sole purpose for what it has been input for, giving "for free" useful information to different department. The examples do not want to be exhaustive, as they have presented only few of the opportunities that the OODBMS can offer.

The benefits of this approach go far beyond the improvement in the productivity, by not requiring the collection of some data (created instead from the system), but allow, with the speed and the accuracy typical of the system, new capabilities. In fact, the close of a cost account after the completion of an activity not only saves some management time, but also improves the accuracy of cost data by not allowing inappropriate charging to an account. This helps reduce game-playing, which is so common with such systems. The slowness of several normal process, in terms of transfer and interpretation of data, hamper the development of time-based control procedure that need consistency and reliability of the data in the system.

Chapter 6
Conclusion

6.1 Summary

This thesis has started from an analysis of object oriented technology and the state of the art in computer aided project management in the construction industry. The analysis of information technology use in construction industry has revealed some important issues. The diffusion of the PC's has, at the same time, increased the number of project managers using information system and hampered the trend of integration inside the company, creating small islands of automation. On the other hand, integration clearly presents an economy of scale, for implementation expenses and efforts, and for the productivity of information: the broader the information gathered, the greater are the possible uses of any piece of information.

This thesis discussed a survey showing different utilization ratios between computer applications. It showed that some isolated applications have low utilization, implying a lower return for their data entry burden. An interpretation of this survey implies that integration might improve utilization. The current database technology offers powerful tools for data storage and transfer, but, for interpreting data gathered or processed by another application, human interpretation is still required. This barrier strongly constraints the benefit from integration in the current DBMS.

The implementation of an integrated PMS can bring several advantages to a construction company in terms of both productivity and quality. One of these advantage is information sharing among the different application (and different functions in the company) that allows single data entry. Moreover, there is more opportunities for interaction between different applications with the possibility of combining data. In particular, an integrated PMS, with the introduction of object oriented technology and knowledge based systems, can automatically interpret and create new data useful to the system.

The examples in the last chapter have shown some procedures that an OODBMS can utilize, starting from some data present in the system, to create new data that otherwise would have to be collected separately. The objects in the database present a behavior that imitates a knowledge process that might be used by the employees to interpret data created by other department, create new documents, transfer data and create summaries. In addition, some objects can have a behavior triggered by time-based events able to check consistency and to monitor other objects.

6.2 Significant contributions

The purpose of this thesis has been to identify the role of OOBDMS in the CAPM in construction. An analysis on the information system role in the construction industry has identified the importance of information monitoring, information sharing and single data entry. After having examined the main barriers to the implementation of an integrated PMS, some possible solutions have been highlighted, in terms of redefinition of the job tasks, early capture of information, incentives for information gathering and sharing and data interpretation. In this process, the work has identified some key points:

1. A piece of data, gathered or created by one user, once put in the system can have value to other users of the system.
2. Data combined from multiple users is more useful than separate groups of isolate data. They can enhance existing applications and create new system capabilities.
3. Data sharing is more complicated than just data storage. There is a strong need for interpretation.

The thesis has demonstrated how the use of an OODBMS can provide two extremely powerful features:

1. automatic data interpretation, able to create data useful to an application different from the one in which the initial data has been collected.
2. automatic control procedures for problem recognition, able to analyze data from different application quickly.

This research has shown that the integration is feasible and useful. Though it can provide guidance for future research in the interaction of OODBMS and CAPM in construction, none of these contributions constitutes a final or definitive solution. Additional research and development is required, as described in the following section.

6.3 Recommendations for further research

This thesis has pointed out several areas of further research and development that are required to fully understand the integration of an OODB in a PMS. This section will highlight such areas

6.3.1 Definition and standardization of the classes and objects in the construction CAPM

The examples shown in the thesis have provided some typical objects that should be a part of an OODB used in construction project management. To aid the development and use of OODBs and to foster the information sharing between different companies, it is extremely useful to define a taxonomy for the classes to utilize in a PMS. This work has already begun in the e-mail conference IRMA-tica 93 (an international e-mail conference on the IRMA model: an Information Reference Model for AEC), held from November 8th to December 3rd 1993. While the conference has suggested some general principles, it has not established a usable object model. Some of the topics that further research should address are:

- Definition of an object model for the AEC industry
- Definition of the principal classes in an OODBMS, for both the general construction library and the project library.
- Mechanisms of multiple inheritance to develop the objects of a specific project from the general construction library and the objects, illustrating specific project information.

While the object model should be compatible with the standard object model currently in use, the classes used by a construction project are unique to this industry and need to be analyzed specifically. Hopefully, this research should be able to establish a standard to be used in the whole industry.

6.3.2 Reengineering of a construction company around the integrated PMS.

The thesis has already outlined how the use of an integrated PMS requires significant organizational changes. At the beginning, the implementation of the system will require the involvement of senior management and strong incentives. Some tasks will no longer be useful, other tasks have to be created, while others reviewed. Further research can study the impact of the system on the organization structure. The opinion of the author is that the result, for the reasons explained in sections 3.2.6 and 4.3.1, will be a more active interdepartmental interaction and a flattening of the organization itself.

6.3.3 Early problem recognition and project control

The thesis also highlights the great potential of an OODBMS in the monitoring phase. The complexity of a typical construction project and the close interaction of different parties and contingent factors in the same activity make it difficult to recognize the causes of the problems, especially in the early stages. On the other hand, it is extremely important to identify the nature of the problem of cost or time overruns as soon as possible to recover the situation. The slowness in data gathering and transfer of the traditional PMS does not allow the project manager timely access to a good information support for problem solving. At the present state, he must repeatedly recognize the causes of the problems without any help from the database.

The early capture of the information is one of the key point of an integrated PMS for problem recognition. Having the information on a daily basis allow the identification of high variances much earlier than in the current process. Once the problem are identified, and all the data are in the system, an OODBMS can easily execute some control procedure to identify the high variances and eventual similarities.

For example, in the OODB, the objects representing the specific activities can have three attributes indicating the estimator that evaluates the activity as well as the supervisor and the foreman responsible for it. The combination of queries on these attributes, message passing and rule-based reasoning (that can be implemented in the behavior of database applications objects) will provide the OODBMS with activity patterns for evaluation of the performance and the deficiencies of the parties involved. Finally, with the correct information support, the project manager will be able to easily identify the causes of the variance and possible solutions.

Further research could analyze in detail the possible control procedures and how they interact with the OODBMS. Finally, a new scheme of information flow for project control could be designed, highlighting the new characteristics acquired with early problem recognition.

6.3.4 Knowledge Packages

One of the features of object oriented technology is modularity. An OODB can be built through different steps and different groups of objects can be linked through relationships and methods in a later stage. One of the possibilities of a construction

OODB is to combine knowledge from different sources. The knowledge domain can be separated into different packages, each package resolving a specific issue of a construction project.

The concept of a *knowledge package* is based on this idea and on the standardization of classes and objects in the AEC industry. The *knowledge package* is a group of object with all the knowledge and the objects required to solve a specific problem. For example, if we consider the issue of the analysis of the interaction of the different parts of the HVAC system and the structural system. The *knowledge package* will contain the standard objects of the HVAC and the structural system with all the methods and relationships able to check the interaction between the two systems, in terms of interference, building codes, safety rules. Specific objects can initialize the package, transferring all information contained in the package into the same objects in the DB which already contains other necessary information (geometric, cost, scheduling and other types of data). If the object is not already defined, the packages will create it as an instance of a class already present in the DB.

It is apparent that, for effective integration of the package into the existing system, standards are required. With this approach, the database can be implemented with a smaller effort and then grow through different steps to face specific issues. The knowledge package can be also considered as a commercial software, elaborated by specialized engineering groups with broader definition and then customized by the specific user.

Bibliography

Ahmed Shamim, Wong Albert, Sriram Duvvuru, and Logcher Robert, *Object Oriented database management systems for engineering: A comparison.*, July 2, 1992, , Technical Report 91-03, IESL, MIT, Cambridge, MA.

Ahmed Shamim, *Transaction and Version Management in Object-Oriented Database Management Systemss for Collaborative Engineering Applications*, January 1991, M.S. Thesis, MIT, Cambridge, MA.

Atwood Thomas, *ODMG 93: The Object DBMS Standard* in Object Magazine, Sep-Oct 1993.

Booch Grady, *Object Oriented Design with Application*, 1991, Benjamin/Cummings, Reedwood City, CA.

Date C.J., *An Introduction to Database Systems*, Third Edition, 1982, Addison-Wesley Publishing Company, Reading, MA.

Derrick Peter, *CENTRAK: New Computer System at MTA Aims to Get New York's Subway, Bus and Commuter Rail System Rebuilt On-time and Within Budget*, July 27, 1982, Mass Transit.

Faulkner Ed, *Computer use in the construction industry*, December 1988, Constructor.

Froese Thomas *Integrated Computer Aided Project Management Through Standard Object-Oriented Models* - July 1992- Technical Report 68A - Center for Integrated Facility Engineering - Stanford University - Stanford, CA.

Froese Thomas, Paulson Boyd C. Jr., *An Object-Oriented Approach for Integrated Project Management Software* - March 1991 - Working Paper 11 - Center for Integrated Facility Engineering - Stanford University - Stanford, CA.

Froese Thomas, Waugh Lloyd, *Project Management and Computers in the Year 2010* - March 1991 - Technical Report 46 - Center for Integrated Facility Engineering - Stanford University - Stanford, CA.

Grimaldi Massimo, Logcher Robert D., Marks David H., *Micro-computers packages for construction management*, Proceding of Conference held in Venice - Italy, April 1984, in Engineering Software for Microcomputers, ed. by B.A. Schrefler, R. W. Lewis and S.A. Odorizzi, Pineridge Press, Swansea, UK, 1984, pp. 619-630.

Kim Won, Lochovsky Frederick, *Object-oriented Concepts, Databases and Applications*, 1989, ACM Press, New York, NY.

Kim Won, *Introduction to Object-oriented Databases*, 1990, The MIT Press, Cambridge, MA.

Kotanchik James J., Logcher Robert D., *Project Mangement and the computer*, June-July 1975, National Development.

Ito Kenji, Ueno Yasumasa Ueno, Levitt Raymond E. and Darwiche Adnam, *Linking Knowledge-Based Systems to CAD Design Data with an Object-Oriented Building Product Model* - August 1989 - Technical Report 17 - Center for Integrated Facility Engineering - Stanford University - Stanford, CA.

Levitt Raymond E., Kunz John C., *Using Knowledge of Construction And Project Management for Automated Schedule Updating*, December 1985, Project Mangement Journal, *

Levitt Raymond E., *PMAPM: An Object Oriented Project Model for A/E/C Process with Multiple Views* - July 1990 - Technical Report 34 - Center for Integrated Facility Engineering - Stanford University - Stanford, CA.

Logcher Robert D., *Adding Knowledge Based Systems Technology to Project Control System*, in Project Controls: needs and solutions, edited by Ibbs William and Ashley David, 1987, ASCE, New York, NY.

Logcher Robert D., Pena Feniosky, *Phase I Report on Cooperative Research on Construction Production Monitoring and Control*, August 31 1990, Technical Report 90-04, IESL, MIT, Cambridge, MA.

Nasrallah Walid, *Practical issues in Message-passing style CPM Scheduling*, February 1989, M.S. Thesis, MIT, Cambridge, MA.

Natsushara Tatsuhiko, *Strategic Information Systems for Standardized Building Business*, February 1992, M.S. Thesis, MIT, Cambridge, MA.

Phan Douglas H., Howard Craig H., *The Primitive-Composite (P-C) Approach - A Methodology for Developing Sharable Object Oriented Data Representations for Facility Engineering Integration* - August 1993 - Technical Report 85 - Center for Integrated Facility Engineering - Stanford University - Stanford, CA.

Russell Alan D., Triassi Emmanuel, *General Contractor Project Control Practices and MIS*, September 1982, Journal of the Construction Division, Proceeding of the American Society of Civil Engineers, ASCE, Vol. 108, number C03, New York, NY.

Russel William, Teicholz Paul, *Barriers to Eletronic Data Exchange in the A/E/C Industry* June 1993 - Technical Report 86 - Center for Integrated Facility Engineering - Stanford University - Stanford, CA.

Sriram D., Logcher R. D., Groleau N., *An Object-Oriented Programming Enviroment for Communication, Coordination and Control in Computer Integrated Design and Construction: Phase I*, December 1987, Technical Report 87-29, CCRE, MIT, Cambridge, MA.

Sriram D., Logcher R. D., Groleau N., and Cherneff J., *DICE: An Object-Oriented Programming Enviroment for Cooperative Engineering Design*, April 1989, Technical Report 89-03, IESL, MIT, Cambridge, MA.

Sriram Duvvuru, Logcher Robert, Ahmed Shamim and Wong Albert, *An Object Oriented Framework for Collaborative Engineering Design*, in Computer Aided Product Development, Proceedings of the MIT-JSME Workshop held at MIT Nov 21-22, 1989, Springer-Verlag, Cambridge, MA.

Sriram D., *Object Oriented Systems: an Overview*, June 6, 1993, IESL, MIT, Cambridge, MA.

Swenson, E. and Logcher Robert D., *Knowledge Acquisition in Conceptual Project Scheduling*, August 1992, Research Report No. 92-25, IESL 92-01, MIT, Cambridge, MA.

Taylor David A., *Object Oriented Information Systems: planning and implementation*, 1992, John Wiley & Son, New York, NY.

Teicholz Paul, *Technology Trend and Their Impact in the A/E/C Industry*, January 1989, Working Paper 2 - Center for Integrated Facility Engineering - Stanford University - Stanford, CA.

Teicholz Paul M., *Integration of Microcomputers Applications - Current and Future Approaches* - January 1989 - Technical Report 7 - Center for Integrated Facility Engineering - Stanford University - Stanford, CA.

Thisner Anders, Teicholz Paul M. and Havas George, *PM nd the computer: the year 2001*, August 1987, number 3, Volume XVIII, Project Management Journal.

Wang Ming-Teh, *Knowledge Processing for a Construction Management Database*, 18 January 1989, Ph. D. Thesis, MIT, Cambridge, MA.

The Stone & Webster PMS has been analyzed through the following articles:

Badger, A. A., Reinschmidt, K. A. and Gandt Richard A., *Project Control System Integration*, 1987, Stone & Webster, Boston, MA.

Griffis, F. H., O'Brien William J. and Bronner Patrick L., *Construction research at Columbia*, in Architectural & Engineering Systems, May 1990.

Humphrey, Charles F., Zabilski Ronald J., *Environmental system retrofit using integrated graphic system*,1991, Stone & Webster, Boston, MA.

IBM Information System Group, *Construction and management at Stone & Webster: Advanced graphics and data base boosts productivity*, 1989, New Yrok, NY.

IBM Information System Group, *Graphic-data base system at Stone & Webster Engineering*, 1990, New Yrok, NY.

Nevins, Daniel P. and Zabilski, Ronald J., *An integrated system for fossil power plant construction and retrofit projects*, August 1989, Stone & Webster, Boston, MA.

Reinchmidt, Kenneth F., *Integrated graphics and database for utility operations and management applications*, 1990, Stone & Webster, Boston, MA.

Reinschmidt Kenneth F, Zabilsk Ronald J., *Applications of computer graphics in construction*, 1989, Stone & Webster, Boston, MA.

Stone & Webster, *COMANDS: a better way to build*, Stone & Webster, Boston, MA.

Zabilski, Ronald J., Hall Herschel E., *Presented in 3-D*, June 1989, Civil Engineering.

Zabilski, Ronald J., Hall Herschel E., *An intergated system for plant lifecycle management.*, ni Engineering/CAD Symposium I, April 1989, Louisville, Kentucky.