Chapter 3

Chemical Process Engineering Database

Computers have been being a major tool in specific aspects of the design of chemical plants for over 30 years. Originally most of the available computer-aided design packages were dedicated to solving isolated problems such as distillation column or heat-exchanger design.

In an actual project, however, all individual tasks have to be coordinated to produce a competent and consistent design. Essentially, most firms still rely on slow paperwork procedures: An engineer finishes one task with one package, prints out the results, has them approved, and passes them along to the next engineer, who feeds them into another package (Winter and Rosen, 1986). One factor that is clearly evident is the volume of data is considerable, growing in a typical project from just a few hundred items at the conceptual stage to several hundred thousands on completion of process design, to a few millions during project engineering, with the constructed plant being defined by tens of millions data items (Angus and Winter, 1985). Published data indicates that a large project, for example, involves a million engineering man-hours and possibly 2 million input operations and 24 million output or access operations during the normal 3-year life of such a project (Waligura and Motard, 1977 quoted in Benayoune and Preece, 1987). The problem is compounded by the fact that most of this data is duplicated. An engineer's time spent on handling program data is up to 60% (Cherry et al., 1982; James, 1985 quoted in Blaha, 1989). Because manual data transfer is time-consuming and error-prone, a more efficient way of working is sorely needed.

The solution to the *data problem* would be integrated system with automated data-handling facilities. Several attempts at integration of different levels have been reported. The database management system (DBMS) approach is the real solution to this problem. A DBMS is a computer program for managing a permanent repository of data, called a database. The database is stored in one or more files and can be shared by many users and many application programs (Blaha, 1989).

3.1 Database Models.

In engineering design, there were three existing commercial database models often presented: hierarchical, network, relational. However, these models have many weaknesses in chemical engineering application, so that a novel object-oriented database model has been purposed as a good modern form of engineering database.

3.1.1 Hierarchical Database Model.

The hierarchical is the oldest approach which data are represented as a simple inverted tree. Blaha, Yamashita and Motard, (1985) presented the samples of this model in figure 3.1 and 3.2. The four trees in figure 3.1 contain utility demand data. Two different types of records are used. The top record contains utility name and utility units. The bottom records have equipment name, equipment type, and quantity.

Figure 3.2 shows an alternate hierarchical view of the same data. Figure 3.1 is utilities oriented; figure 3.2 is equipment oriented. The asymmetry of many-to-many relationships is a major drawback of hierarchical systems.

The hierarchical model may be viewed as a restricted model where each descendant can only be connected one parent, and where cross links between descendants are not allowed. Although This restriction is efficient for processing, it is too rigid for logical modeling and leads to redundancy. Any schema defined in terms of this model unresponsive to changes and difficult to map to any other model (Buhmann, 1980).

50 lb Steam	lb/hr	
C 101	Distillation Column	5000
C 102	Distillation Column	9000
E 201	Heat Exchanger	5000
F 101	Furnace	5000

250 lb Steam	lb/hr	
F 101	Furnace	-9000

Condensate	gpm	
E 201	Heat Exchanger	-4
F 101	Furnace	28

Cooling Water	gpm	
E 101	Heat Exchanger	25
K 101	Compressor	2

Figure 3.1: Sample data in hierarchical form, utility-oriented.

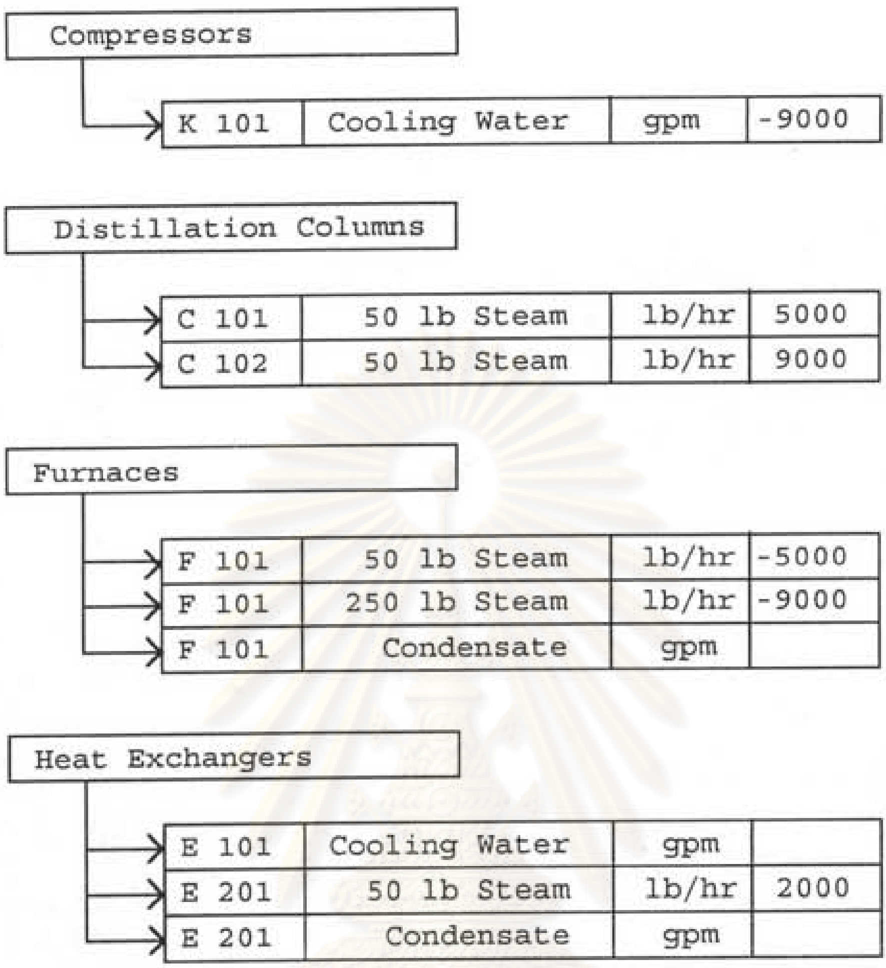

Figure 3.2: Sample data in hierarchical form, equipment-oriented.

3.1.2 <u>Network Database Model.</u>

Blaha, Yamashita and Motard, (1985) presented: The network model is a more general structure than a hierarchy. Networks directly represent many-to-many relationships with records and links. Figure 3.3 casts the utility demand data into network form. The prime flaw of this approach is undue complexity.

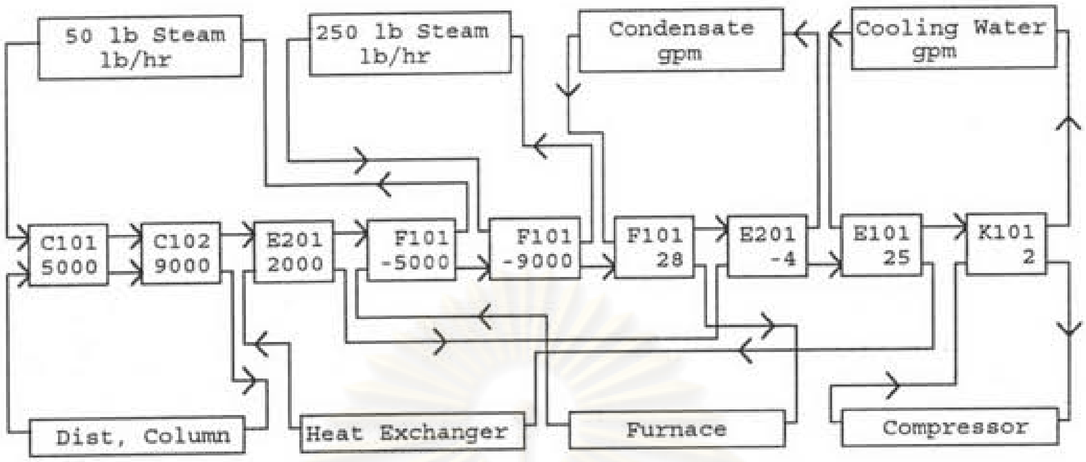

Figure 3.3: Sample data in network form.

The major shortcoming of the network model is the classification of data in term of entities and attributes, which is often conflicting when modeling different applications that will be integrated. Therefore, the network model is not readily adaptable to changes in the object system, and semantic information may be lost or misinterpreted when modifying the global view in response to changes in the object system (Buhmann, 1980).

3.1.3 <u>Relational Database Model.</u>

The relational database (RDB) has been being the conventional and standard systems widely used in most software development. It can be defined as a collection of tables (called relations) and commands for manipulating the tables. Blaha, Yamashita and Motard, (1985) illustrated this model by Figure 3.4.

Relations surpass hierarchies and networks for several reasons.

Utility Relation

Utility Name	Utility Unit
50 lb Steam	lb/hr
250 lb Steam	lb/hr
Condensate	gpm
Cooling Water	gpm

Equipment Relation

Equipment Name	Equipment Type
K101	Compressor
C101	Distillation Column
C102	Distillation Column
F101	Furnace
E101	Heat Exchanger
E201	Heat Exchanger

Quantity Relation

Utility Name	Equipment Name	Quantity
50 lb Steam	C101	5000
50 lb Steam	C102	9000
50 lb Steam	E201	2000
50 lb Steam	F101	-5000
250 lb Steam	F101	-9000
Condensate	E201	-4
Condensate	F101	28
Cooling Water	E101	25
Cooling Water	k101	2

Figure 3.4: Sample data in relational form.

On surface, commercially popular relational database model has been widely used for engineering applications. However, the relational database systems have been designed in response to the needs of typical business applications. These systems are well equipped to store and manipulate flat data that can be represented as series of two dimensional tables, accessed, and modified by high-level query languages and presented in attractive forms. However, the set of structures, operations and constrains in relational data is limited and fixed. They are often not good at modelling the relationships among data when any structure and operation needed by the applications become more complex (Ahmed et al. 1992; POET SoftwareGmbH, 1993). It is worth noting that these same deficiencies are found in other conventinal database models, such as the hierarchical and the network models, when employed in designing an engineering database (Huang and Fan, 1988).

3.1.4 Object-Oriented Database Model.

Objected-Oriented database (OODB) is a newly powerful new tool to manage large amounts of data and to model the relationships among the data. As an Objected-Oriented programming language, it uses *classes* and *objects* to provide its features. Blaha (1989) stated: OODB modeling technique encourages the kind of evolutionary development that often occurs with complex engineering problem. The OODB model proved to be a valuable tool for stimulating and documenting design discussions. The model deals with *objects* and *relationships*. An object is a concept, abstraction or thing with crisp boundaries and meaning for the problem at hand. Purification feed pump, reactor feed preheater, and the most recent simulation run are objects. A group of simular objects form an *object class*. Pump, heat exchanger and simulation run are object classes. The classes are described by *fields*, such as cost, discharge pressure and diameter. Equipment and utility at the top of figure 3.5 are example of object classes. Class names are shown in the top of the box; field names are listed in the bottom.

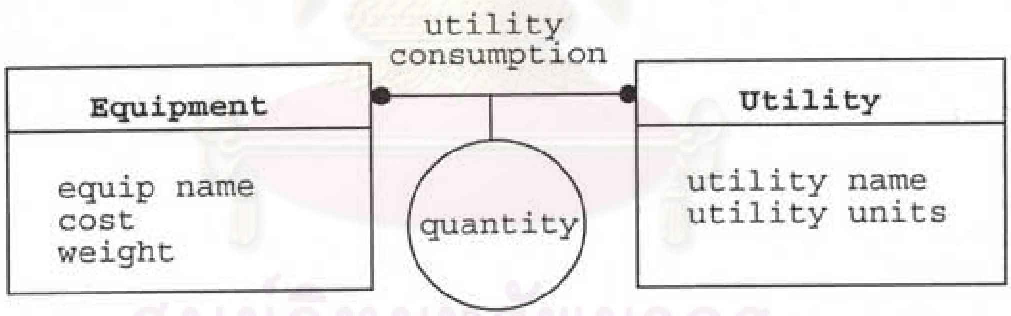

Figure 3.5: The data modeling process.

A relationship is a logical binding between objects. There are three types of relationships: *association*, *aggregation* and *generalization*. The model indicates a relationship with the line between object classes.

1. Association is the most general type of relationship. It relates two or more independent objects. The utility consumption in figure 3.5 is an association. Each piece of equipment consumes many utilities. Each utility is consumed by many pieces of equipment. Equipment and utility are distinct classes that interact. A heat exchanger may consume steam and produce condensate. Steam is consumed by many exchangers, perhaps a reactor, and live steam distillation columns. Associations may have properties, which are circled.

2. Aggregation combines low-level objects into composite objects. Aggregation is often referred to as assembly-component or "a-part-of" relationship. Aggregation often occurs in bill-of-materials or parts explosion type of problems. As shown in figure 3.6, a frame is a part of a motor; a shaft is a part of a motor; and many bearings are a part of a motor. Arrows point towards the composite object.

3. Generalization provides hierarchical structure for classes. Generalization partitions a class into mutually exclusive subclasses. The triangle in figure 3.7 is the symbol for generalization.

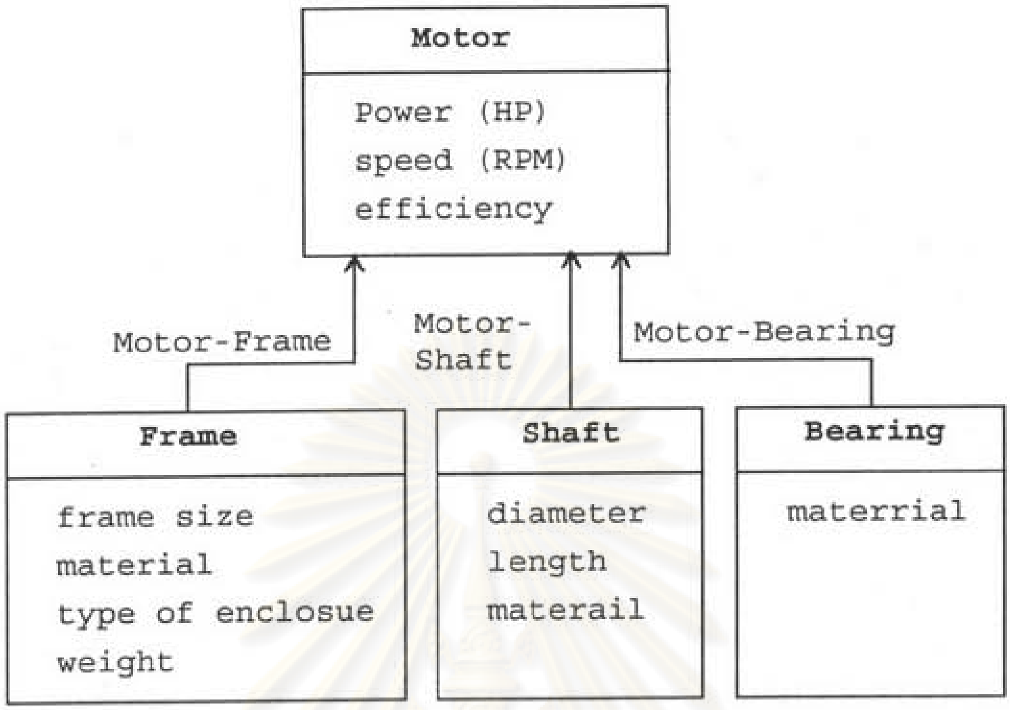

Figure 3.6: Aggregation relationship.

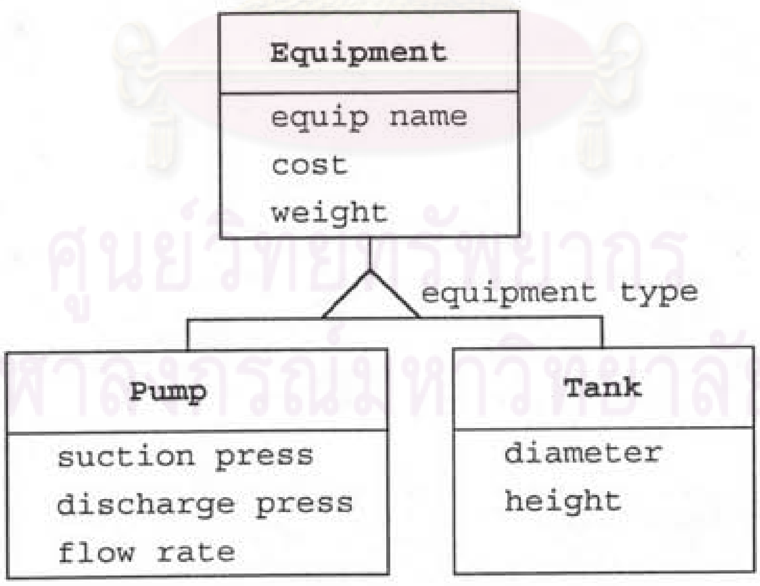

Figure 3.7: Generalization relationship.

In this example, a piece of equipment may be a pump or a tank. The equipment object class stores general information. Pump and tank store specific information. Each pump has a name, cost, weight, suction pressure, discharge pressure and flowrate. Generalization may have an arbitrary number of levels. Thus one may further refine pump into centrifugal pump, diaphragm pump and plunger pump. Each level of a generalization represents one aspect of the same object. An object simultaneously exists at each level of a generalization hierarchy. The equipment type field next to the generalization triangle is the discriminator field which indicates the appropriate subclass for each superclass object.

3.2 Advantages of OODB Model Over RDB Model.

For engineering applications, the advantages of object-oriented database model over the relational database model can be presented as follows (Ahmed, 1992).

1. *A more realistic data model.* Design classes and objects in OODB represent real-world design entities. They give a better feel for the mechanics of the problem than do a set of flat tables as in RDB.

2. *A more powerful data model.* Object data representation is highly flexible. It allows data to be inheritance and polymorphic. Problems that normally

occur in an RDB, such as normalization and unneccessary duplication of data, do not exist.

3. *Easier schema development.* Generalization and inheritance allow the schema to be better structured, more intuitive, and capture the semantics of the application. The schema definitions are likely to be modified as designers arrive at a better understanding of their problem.

4. *Object identity.* The RDB model is value based, which expresses relationships between two objects by embedding the same (or equal) value in two or more related objects. The unique keys are often defined to distinguish between similar entities, frequently resulting in data duplication. The OODB model is identity based, and can relate two or more objects independently of their embedded values.

5. *Powerful unified knowledge representation.* OODB representation of information provides a uniform framework for integrating data (attributes) and knowledge (procedures).

3.3 Industrial Database Background.

The first implementations of the database systems for chemical process engineering can be attributed to Tsubaki and his co-workers at Chiyoda Chemical Engineering & Construction Co. Ltd., Japan in 1973 and this team had actived until about 1979. (Angus and Winter, 1985; Benayoune and Preece, 1987).

In the late 1970's, ICI devoted considerable attention to database technology and one project involved a detailed examination of data use and data flow within the company. Another project was known as PEDB having a large number of programs which could be beneficially linked to the system. Since September 1984, a major contribution to this field has come from ICI whose product, DesignMASTER, commercialized by ChemShare, was one of the leading integrated process design systems on the market. The bulk of 15 man-years' development with in ICI and 6 man-years within ChemShare has been spent in achieving these ends (Craft, 1985). The DesignMASTER system embodied the PEDB technology (Angus and Winter, 1985).

One of the database CAD systems on the market was the Piping Design and Management System (PDMS). Because its database was specific application, it would be very difficult to reorganize to serve the whole project. Another organization to make significant investments in this area was CADCenter, whose PEGS and CHES programs use a database approach. PEGS was a schematic drafting system for PIDs which includes a project engineering database to accommodate both the data captured from the PIDs and additional alphanumeric data. Davy McKee Ltd. (London) developed CHES as PEDB, it interfaced to a database manager operating on the host computer (Angus and Winter, 1985). Similar systems have been developed by other software vendors. The Intergraph System from the Intergraph Corp. and the PLANTMAN System from Quest Genesys offer similar graphics and database capabilities to those of PDMS and PEGS.

Prosys Technology Ltd. launched its database system, Prodabas. It was the one of the first commercial system to be specifically developed for chemical engineering and was based on relational database technology. As DesignMASTER, described above, Prodabas was mainly intended for process design.

Several other engineering companies such as Bechtel, E. I. du Pont de Nemours, Exxon, Kao Corp. (Tokyo), Mitsubishi and Monsanto were known to be working on project engineering databases. However, most of this work was rather empirical and done with the usual secrecy and short range goals. No details were available in the literature about these projects, however they were believed based on commercially available database tools (Angus and Winter, 1985).

3.4 Academic Experience.

Benayoune and Preece (1987) presented: The academic contribution to the application of database technology in chemical engineering is quite modest compare to that of industries. Most university research has been aimed at analyzing the design activities involved in chemical plant design and understanding the problems involved with handling engineering data.

One of the first university research projects to be undertaken was that of Cherry (Cherry, 1975 quoted in Benayoune and Preece, 1987) carried out at the Department of chemical Engineering, University of Cambridge, in the early 1970s. Having arrived at the conclusion that none of the existing DBMS satisfied the requirements for engineering data. Cherry went on to develop a new data-management system but his system was not classified as a DBMS.

On the other hand Buchmann (1980), who did his work at the Department of Chemical Engineering, University of Texas at Austin, in the late 1970s, selected the binary data model to support his system. He divided the database into several levels:

- a project database to hold approved project data;
- a family of design workspace databases for each design activity;
- a set of catalogue databases for standard non-project specific data.

Buchmann claimed that the layering of the database together with his data dictionary based design methodology reduces the complexity of handling the plant design data. Buchmann pursued his work at the University of Mexico to develop a database interactive design system.

One of the major projects addressing the integration of CAD packages around a database system was that of Motard at the University of Washington. The research was backed by a consortium of CAE vendors, users and consultants.

Blaha (1984) researched to apply database management to chemical process engineering. His work included:

- Identify features needed by an engineering DBMS.
- Develop a conceptual data model for interfacing CADRE to many process engineering application programs.

Huang (1988) presented a case study of knowledge engineering in process engineering. In preserving the flowsheet, a database model, amalgamating the notion of objects and relations, is employed.

Patakas (1989) studied to understand the design data and identify a data model which will assist the engineering process design.

Shenoi (1989) presented the equivalence class model which is implemented within the object-oriented framework. The ability to effectively deal with imprecision and abstraction in the relational structure is a significant feature of the present model.