Korean J. Chem. Eng., 18(5), 612-622 (2001)

Multi-Dimensional Design Process Management by Extended Product Modeling for Concurrent Process Engineering

Seung-youp Han, Young-soo Kim, Tae Sung Yoon* and Tai-yong Lee†

Department of Chemical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 305-701, Korea
*Engineering Research Institute, Faculty of Engineering, The University of Tokyo, Tokyo 113-8656, Japan

(*Received 8 June 2001 • accepted 20 August 2001*)

Abstract−Conventional product and process models have focused on static features. That means product models are mainly based on structural decomposition of products, and process models are also often described by activity decomposition such as work breakdown structure. From the view of design process management, it is difficult to describe dynamic features of design processes appropriately through conventional methodologies. In this paper, a multi-dimensional approach for design process management was explored to manifest characteristics of design processes for chemical plant design. Parallelized design process for concurrent process engineering should be managed by two-dimensional design activity flows. The process management makes it possible to guide progress of design processes in a helix structure by horizontal and vertical activity control simultaneously. They stand for teleological and causal relation between design activities, respectively. That can be achieved based on an extended product model, which represents various design perspectives explicitly from a conventional design activity model. The extended product model is composed of product data, design activities, and activity drivers. Dynamic features of the extended product model are expressed by an activity chain model. These concepts will support the realization of concurrent process engineering for chemical plant design in the sense that they provide design process management strategies.

Key words: Concurrent Process Engineering, Product Modeling, Design Process Management

INTRODUCTION

The demand for higher quality and lower cost with shorter development lead-time in chemical plant design has forced engineering industries to focus on new strategies for efficient design process management. Many conceptual methodologies have strived for the last decade to minimize development cost and to maximize development efficiency through whole lifecycle from project planning to disposal. One remarkable attempt is Concurrent Process Engineering (CPE) by CAPE.NET supported by EU process industries and research centers. CAPE.NET emphasizes that chemical process design should be performed under concurrent consideration of various design perspectives in order to achieve process flexibility, radically improved integration, rapid prototyping, and so on [Bogle and Perris, 1999].

Many kinds of methodologies should be implemented appropriately in order that a wheel for CPE rolls on successfully, but what plays a role as a shaft in the wheel is an integrated information model [Krause et al., 1993]. The most important part of the integrated model is the product model because it may be a static structure for other data models. CAPE.NET suggests a global framework to integrate a whole design process; however, it does not contain rigorous representation related to product data and design activities.

So far, many generic product models that have a neutral format have been developed including those for a chemical process [Owen, 1993]. However, most of them have focused on a standard description of product data to share design information among heterogeneous design environments. That makes it difficult for the product models to contain characteristics of the design process corresponding to various design perspectives. In addition, the product models are limited in describing design intent, histories and rationales. Thus, product data should have an explicit relationship with design processes. The relation should make it possible to expand design activities systematically with logical meaning. In this paper, we will propose an extended product model that can satisfy such a condition.

1. Concurrent Engineering

In general, Concurrent Engineering (CE) is defined as a systematic approach to the integrated and concurrent design of products and their related processes including manufacture and support. This approach is intended to cause the developers, from the outset, to consider all elements of the product life cycle from conception to disposal including quality, cost, schedule, and user requirements [Bullinger and Warschat, 1995]. Generally, three possible strategies can be identified as CE guiding principles: parallelization, integration and standardization. Parallelization in the product development process implies the cutting and optimization of time. The first step is to remove existing float time in the development process. This means that processes that do not have any dependencies on other processes are carried out simultaneously. Accelerated execution of linked processes through this approach proves to have an advantage, but it makes higher complexity in design process management. The complexity is caused from an increased amount of information transfer between departments or individuals, and inconsistent management of the information. Integration is a measure to overcome these interface problems. Integration demands working in interdisciplinary teams, thinking and behaving in a process-oriented way, and realizing a common objective instead of several depart-

---

†To whom correspondence should be addressed.
E-mail: tylee@mail.kaist.ac.kr

ment-specific objectives. In a narrow sense of information management, integration can be achieved by making inter-activity relationships in the parallelized design process. This paper focuses on this topic among three guiding principles for CE. Finally, standardization of process is needed so as to avoid repetition and needless work as well as to learn from existing experience of the company. Standardization of product data is related to technical/structural aspects such as the usage of modules or components in the final products and it can be supported by ISO10303 STEP (STandard for the Exchange of Product model data).

2. General Features for Product Modeling

Product data models that can support various computer-aided engineering applications have been developed to achieve domain specific problem solving. Even though every product model has a specialty to describe its own characteristics, it has been defined considering extensibility, conceptuality and integrity for the model to be used as a general product data model. There are some general features that most product data models intended to accomplish. The features will be basic guidelines for developing the extended product model proposed in this paper.

- Most product models proposed currently have a tendency to be defined by a definite form to increase reusability of the models. Object-oriented data modeling has gained great popularity. The main reason for the popularity is that object-oriented data modeling provides database designers with high-level abstractions to represent information in the manner close to the designers' conceptual view of the information [Chung and Fischer, 1994]. Product data can be described by simple repeating pattern if object-oriented approach is used [McKay et al., 1996].
- Product data models should be defined with multiple perspectives if data management through life cycle is required and the product data model is intended to be used as frame structure of a data warehouse [Inmon et al., 1997]. That means a product data should be managed under the consideration of design processes [Peltonen et al., 1996]. Some researchers proposed product models combined with design activities or a framework of data model relations to show how a product can be realized by mapping design processes one another [Gorti et al., 1998; Kjellberg and Schmekel, 1992].
- One of the important functions for product data management is to describe design histories and rationales. They can be managed by additional description in product data based on design process. From the description, product data can be retranslated in a view of design processes [Taura and Kubota, 1999; Shah et al., 1996; Chandrasekaran et al., 1993].

There have also been many researches to make a product data model for chemical process industries. Product data management to support recording design rationale using a way of knowledge representation was proposed [King and Banares-Alcantara, 1997]. Integration of data model for process design using ISO standard, STEP was attempted from the view of global product management [Bayer et al., 2000]. In addition, product models confined to specific perspectives or life cycle activities will be useful because they can be applied to real systems more rigorously. Information models for planning and scheduling of batch processes and for plant operation were proposed [Book and Bhatnagar, 2000; Lu et al., 2000]. Both information models were also based on ISO standard, STEP.

3. Chemical Process Design Activities

Chemical process design activities that we intend to focus on are parts of life cycle activity for the chemical process industry. The activities can be broadly divided into process design activity and engineering design activity according to who mainly performs each activity and what kind of information is dealt with. It is difficult to share information between process and engineering design activity owing to their different characteristics. For example, the former used to be represented by PFD or P&ID, which includes 2D topological information and its attributes expressed by documents or text, while the latter consists of physical and geometrical information to perform design equipment, plant layout, safety evaluation and so on. Therefore process and engineering activities are separated each other from the viewpoint of information management. The scope of the extended product model covers both design activities simultaneously.

4. Public Product Database

There are several public databases for standard product data proposed by ISO, POSC/CAESAR, etc. Most of the product databases are provided in a type of class library. For ISO, there have been attempts to make Application Protocols (AP) to support life cycle activities of process engineering. They are functional data and their schematic representation for process plant focused on P&ID [ISO10303-221, 1997], plant spatial configuration [ISO10303-227, 1997], and process engineering data for major equipment [ISO1003-231, 1998]. POSC/CAESAR has provided a full set of class classification for gas and oil industries [POSC/CAESAR, 1997].

PRODCUT MODELING FOR CHEMICAL PLANT DESIGN

The product model is extended to treat the specific features of chemical plant design. The extended product model consists of three parts. One is a slightly modified product data model from conventional product data models, another is a design process model based on activity model, and the other is a functional requirement that provides a functional relationship to represent design dependency according to perspectives among design activities. The main objective of the extended product model is to construct a comprehensive product model based on design process, which can be an essential kernel in the design process management system. In addition to these extensions, two critical features are also considered, namely, multi-dimensional aspects of managing design processes for chemical plants and methodology for describing design intent. The proposed product model will be able to support not only integration of design process for CPE, but also data driven approach to capture design intent.

1. Extended Product Modeling

As mentioned in the previous section, the product model should have a very close relationship with the design process. The design process is generally represented to be a sequential procedure of design tasks. When we intend to reorganize the sequential design process to be an overlapped form using the concept of parallelization for CPE, we are faced with two problems. One is how to represent logical relationship between parallelized design tasks, and the other is how to deal with the design space network caused by the relationship. In this paper we focused on the former problem. The latter problem will be left for another scope of work [Han et al., 2000].

In conventional product modeling, product data is defined separately with design process model. Even though a design activity has some product data as input or output of the activity, product data are referred or generated only following a fixed design activity sequence. That means product data cannot control design process directly although the product data makes some requirements to evaluate feasibility or predicted problems that may happen in other coming design activities. For example, suppose a designer has a few alternative design results through his design activity. He can choose a preferred one within his design heuristics or knowledge, but his decision-making may bring about a design constraint with other product data and increase design load and cost in subsequent design activities. In this case he may want to evaluate his alternatives in other design points of view that are not working yet. It may be possible to combine the related design activities by temporal modification of the design process. That, however, makes it difficult to manage design process consistently and to record why the design activities is interacted with each other due to the absence of formal description method for dependency among design activities. In addition, as complexity of design dependencies increases, subtasks of an activity are liable to be redundant.

The main reason why these problems cannot be solved using conventional product modeling environment is that causality among design activities is not represented properly. In general, the causality is included in design process implicitly. The conventional description of a design process such as activity model supported by Process Industries STEP consortium (PISTEP) was developed with optimality of chemical plant design from the view of teleology, but the activity model seems to be scattered without coherence from the view of causality. In general, it is very difficult to define causality as a definite form on fixed design processes because the causal relationship can be changed on occasion. For example, causality between tank design activity and safety evaluation activity does exist or does not exist according to its situations such as what kind of material will be contained, where the tank will be located, and so on. Therefore, it seems to be natural that the conventional design process has been described from teleological viewpoint in order to express design processes in a definite form.

The main purpose of an extended product model that we present is to represent the causality among design activities explicitly. The extended product model is classified into three parts: product data model, design process description represented by activity model, and functional requirements as one of the design activity drivers. Most conventional product models are composed of product data model and activity model. In the extended product model, functional requirements are supplemented because the causality cannot be expressed properly through activity model as mentioned before. The basic concept of the classification is originated from Object Modeling Technique (OMT). OMT suggests three kinds of views: static, dynamic and functional ones for general system analysis. They can be mapped to each part of an extended product model, respectively [Rumbaugh et al., 1991; Han et al., 1999].

1-1. Product Data Model

Product data model plays the role of static structure in the extended product model. As mentioned above, many kinds of conventional product models have been developed. The product data model presented in this paper is basically based on the conventional models such as STEP and POSC/CAESAR. The classification of the product data model is, however, a bit different from that of conventional models. Design data of chemical process can be classified into two major groups according to characteristics of design processes. Most conventional activity models show that the whole design process for chemical processes is divided into process and engineering design activity depending on someone who mainly performs each activity and what kind of information is involved. In the view of data management, product data should be considered separately to avoid ontological confusion. Therefore, a *product* object is classified into a *behavioral* and a *physical* object at the top level as shown in Fig. 1 to represent product data for process and engineering design activity respectively [Han et al., 1999; Batres et al., 1999].

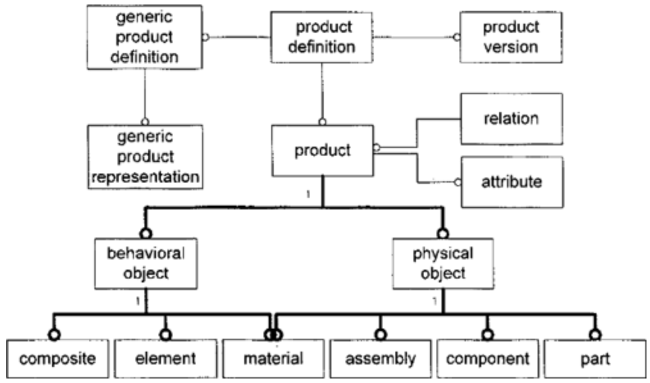


Fig. 1. Meta model for product data.

The figure stands for a meta definition for product data model represented by EXPRESS-G. A tree relationship that indicates supertype and subtype relation shall be displayed as a thick solid line and all other relationships shall be displayed as normal width solid lines. Relationships are bi-directional, but, following the EXPRESS style, one of the two possible directions is emphasized. For example, if an entity A has an explicit attribute to entity B, then the emphasized direction is from A to B. In EXPRESS-G, the "to" end of a relationship shall be marked with an open circle [ISO10303-11, 1991].

A *Product definition* is an abstract object to describe a product. It has two attributes as multiple identifier, *product* and *product version*, because instances of a *product* class should be distinguished according to its version as well as the *product* instance itself. The *product definition* also has *generic product definition* and *representation* as additional attributes to describe product data without loss of generality. A detailed description of them is beyond the scope of this paper because they can be referred from Part 41 [ISO10303-41, 1997]. The *product* object has its own attributes that can be referred from various standard product data in order to represent physical or behavioral characteristics of the *product* object. Then, physical or logical relationships between *product* objects can be expressed by *relation* objects. The relation object can be described many kinds of associations based on natural language expression such as '*is connected to*', '*is part of*', and so on. We can also refer to the rigorous associations from AP 221. Product data representation using *product* and *relation* makes it possible to improve a data-managing

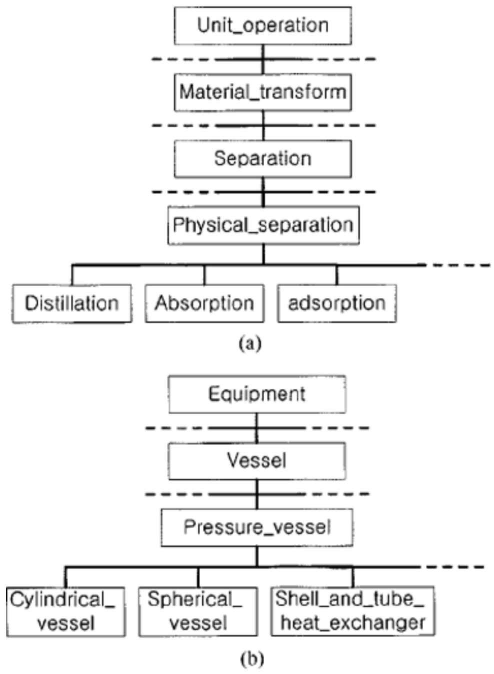


Fig. 2. Subparts of behavioral and physical object.
(a) Behavioral objects (b) Physical objects

environment from document and drawing driven environment to a data driven one.

Product objects can be classified into *behavioral* and *physical* objects as mentioned above briefly. That implies classification not only for primary usage of product data according to design process, but also for characteristics of information contents contained in each object. *Behavioral* objects are used to define a capability to perform process function, independently of the physical structure. A *behavioral* object is concerned with the ability to do something in contrast to the thing that might actually do it. The *physical* objects are something that have consisted or that consist of matter, that is, actual, specific materials which can be touched. Typical example of *physical* objects is equipment. An example subpart of *behavioral* and *physical* objects is shown in Fig. 2.

In this figure we will see that the same design object in a practical view can be described differently in aspects of design perspectives. For example, a *distillation* object means one of the separation processes that contains two kinds of mixture flow sustaining equilibrium status on each stage. But a *vessel* object by which the *distillation* object might be realized is regarded as something assembled by *shell* and *heads* including accessory component for fluid guidance. Besides, someone who designs a *heat exchanger* in a conceptual process design stage may describe it as a facility where two kinds of flow whose temperatures are different from each other are guided. The heat exchanger, however, may be described similarly to the *vessel* for *distillation* except internal flow guidance type. That is the reason why *behavioral* and *physical* objects should be dealt with separately. It is also for efficiency of data management.

Behavioral objects have two kinds of subtypes classified into *composites* and *elements*. The basic criterion of the classification is a representational extent of the objects. *Composites* stand for process functional units at unit operation level. Some top-level objects of *composites* are *material transform*, *heat transform*, *material transport*, *storage*, etc. They can be classified more and more rigorously by defining their subtypes as shown in Fig. 2. *Composites* objects, however, do not contain whole data for a functional unit. We may extract some sub-units that can be commonly used in several *composites*. The sub-units can be defined as *element* objects separately from *composites*. Typical instances of *elements* are *port type*, *fluid characteristics*, *phase*, etc. Consequently, a *behavioral* object is completed by adding aggregation of *element* objects. The relationships between *composite* and *element* objects are described by *relation* object shown in Fig. 1. The main purpose of this classification is to reduce redundant data definition as much as possible.

Physical objects can be classified in a similar standpoint to *behavioral* objects. *Physical* objects are classified into *assemblies*, *components* and *parts*. *Assemblies* include general *equipment* and aggregated modular systems such as *fire protection system*, *electrical power system*, etc. *Components* are decomposed objects of *assemblies* up to manufacturing level. Typical *components* are *enclosure*, *end*, *plate*, *valve*, etc. *Parts* are the smallest units of *physical* object such as *gasket*, *flange*, *bolt*, *nut*, etc. Detailed classification of *parts* can be referred from the Parts Library [ISO13584, 1995]. *Material* objects also should be defined as a subtype for *physical* and *behavioral* objects with multiple inheritances. *Material* objects are classified into subtypes, *process material* like *water* and *structural material* like *iron*.

So far, we have represented the basic structure of the product data model including its classification. Even though the top-level description of the product data model is defined somewhat differently from conventional product models to satisfy requirements for CPE, it is not necessary to construct full contents of a product data model in detail. Instead, it is recommended to use various standard product databases. Actually, detailed product classifications of this product data model have been referred from some of them. The mainly referred product data based on various product models are as shown in Fig. 3. The extended product model should be referred from various public databases because, in general, the conventional product models have been developed for specific scopes and purposes.

1-2. Design Process Model

While product data models are emphasized from a static view, design process models provide one of the system analysis methodologies in a dynamic view. Design processes are basically a systematic representation of procedural problem solving activities. De-

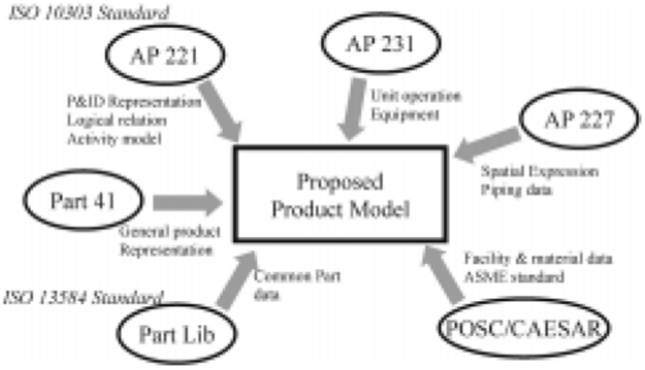


Fig. 3. Relationship with public databases.

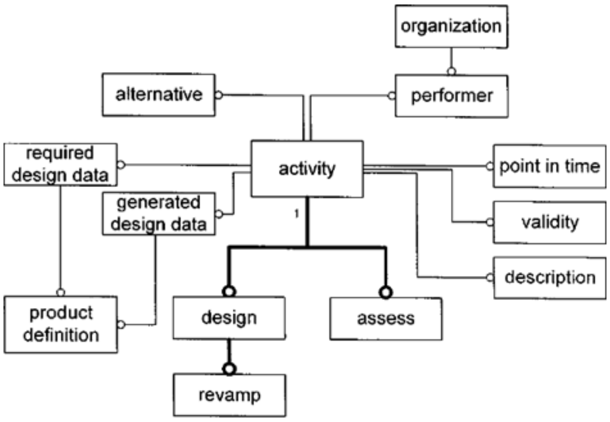


Fig. 4. Meta model for design activity.

sign processes can be regarded as a set of sequential units simply called activities. A meta model for design activities is as shown in Fig. 4.

Activity objects have some basic attributes such as *performer*, *point in time*, *status*, *related design data and alternative* in order to describe a design activity in aspects of design intent management. The *performer* means someone who performs the design activity, and he/she may be involved in an organization. In this paper, rigorous descriptions for *organization* are beyond the scope. The *point in time* attribute stands for when the activity is performed. The *validity* attribute notifies whether the design activity is currently valid or not. In the whole design process, all of design activities are not valid because design activities can be propagated simultaneously following paths for various alternatives. More comments for this situation will be shown later in detail. *Activity* objects also should have a relationship with *design data* because results from performing design activities are eventually represented by the design data. There are two kinds of design data. One is *required design data* that have to be referred to perform a design activity. Actually, they may not be represented explicitly because performers of the activity want to refer to previous design data as much as possible for better design. The other is *generated design data* by the design activity. In contrast to the *required design data*, the *generated design data* can be related to product data explicitly to avoid authority confusion when modifying design data. The design data attributes are associated to *product* objects because most design data can be represented by product data. In general, performing design activities often make several alternatives as results for the activities. Then, one of them will be determined by decision-making. However, all alternatives should have design data even though some of them are not selected in order to manage design histories or rationales. A decision made in a point of time can be changed to other alternatives by design constraints or change of external circumstances. Therefore, an activity object has multiple identifiers, *activity* and *alternative* like those of product object. Activities are also classified into two types, *design* and *assess*, as remarked by AP221. *Design* objects create product objects directly, and *assess* objects evaluate product objects to fit for a purpose and create approval object.

The proposed meta model stands for the basic constitution that a design activity should have. Actual design processes have been represented by Integration Definition (IDEF0), generally called activity model. IDEF0 is one of the most popular expressions for process analysis [Colquhoun et al., 1993]. IDEF0 is used to produce a kind of function model, a structured representation of the activities. It can also represent the information and the objects that interrelate those activities. A basic unit for IDEF0 representation is like Fig. 5. The box stands for a unit of tasks defined by the activity meta model. It can be divided into sub-tasks through analyzing dependency among them. Process decomposition can be done at various levels of abstraction with hierarchical structure. Relationships among activities are described in a uniform format by input, output, control and mechanism as shown in Fig. 5, and what they mean are as follows:

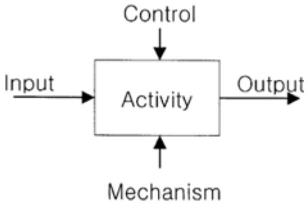


Fig. 5. IDEF0 definition.

- Input: Something transformed by the activity
- Output: Something produced or modified by the activity
- Control: Something that constrains how the activity is undertaken
- Mechanism: Something that does the activity

A basic structure of an activity model was published by PISTEP. Application protocols of STEP such as AP221, AP227 and AP231 adapted activity models expressed in IDEF0 on the basis of the PISTEP's activity model. The activity models can be referred to as standard design activity models if necessary. We make use of AP221's activity model for activity class definition because it covers wide design processes not being too specific.

1-3. Functional Requirements

In general, design is not single objective problem. We can consider many kinds of design perspectives in chemical process design even though the main objective is to design a chemical plant that can produce chemical products to satisfy planned quality and quantity. Frequently commented perspectives in chemical process design are safety, maintainability, operability, manufacturability etc. The perspectives must be considered in a design process simultaneously. The need for considering the perspectives in design processes has been emphasized by a well known methodology called Design For X (DFX). The trade-off barriers in concurrent engineering should be solved by synthesizing different DFX principles to provide a well-rounded outcome [Liu et al., 1999]. It is, however, very difficult to construct a design process including all of the perspectives. If we intend to represent the perspectives on an activity model, it will be too complicated owing to so many interactions among activities. That may cause loss of generality of activity models, and therefore, in general activity models, the perspectives are usually included in design activities implicitly, or a perspective is blocked as an activity. Functional requirements are defined in this paper in order to deal with the design perspectives explicitly. Functional requirements provide clear representation of activity relations by dealing with design

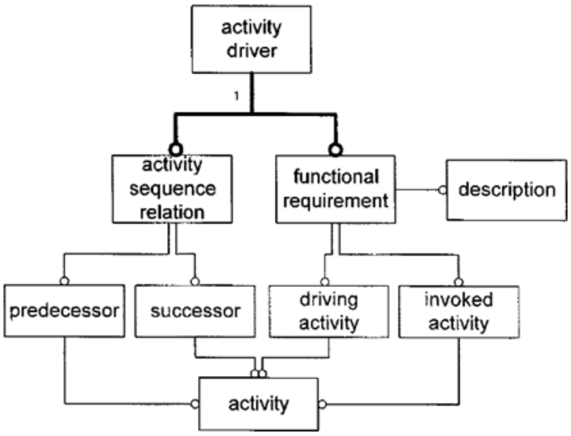


Fig. 6. Meta model for functional requirement.

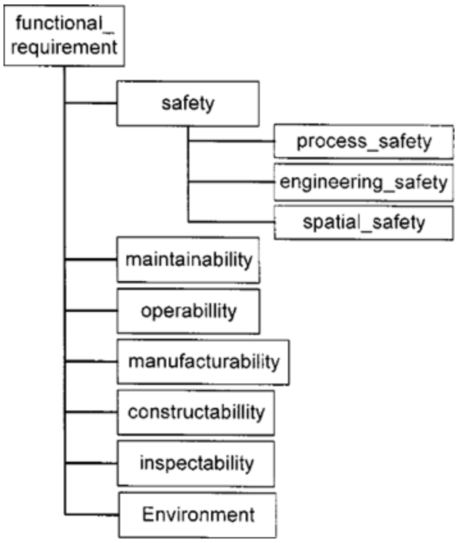


Fig. 7. Top-level classification for functional requirements.

perspectives independently of activity sequences.

Functional requirements are key concepts of the extended product model. To accomplish CPE, one of the most important problems is to integrate inter-related design activities through design perspectives in parallelized design processes. It was addressed that conventional product models based just on product data, and activity models are not sufficient to represent relationship between activities, which are requested to be processed successively regardless of activity model sequences. That function may be required in case that a designer wants to evaluate his/her design alternatives in early design stage or that some later activities can proceed independently of results of intermediate design activities. In brief, it is a problem whether design processes can be controlled explicitly by requirements arbitrarily caused in design processes.

Actually, there are two kinds of something that drive design activities. We can call them *activity drivers* as proposed in Fig. 6. The activity drivers can be classified into *activity sequence relations* and *functional requirements*. *Activity sequence relations* stand for explicit expression of activity relations defined in activity models. An *activity* instance can work immediately if all conditions for the *activity* such as inputs, controls and mechanisms are completely prepared. *Activity sequence relations* have two attributes, *predecessor* and *successor*. They are identified by *activity* instances. Therefore, design processes make progress by *activity sequence relations* without any other explicit requirements. That means a procedure of ordinary design processes guided by fixed process model.

Functional requirements play a role of describing causal relationship among activities so as to satisfy the functions mentioned above. *Functional requirements* also have two attributes, *driving activity* and *invoked activity*, identified by *activity* instances. Besides, *descriptions* for the *functional requirements* are needed to explain more rigorously why the *invoked activity* should be followed at that time.

Classification of *functional requirements* is proposed in Fig. 7. The figure contains only top-level classification based on general perspectives in chemical process design, and therefore, any other perspective can be defined according to characteristics of a target plant or design environments. In addition, a designer can define *functional requirements* based on the meta structure as a user defined type if necessary. Basically, *activity drivers* are determined at each design stage by the activity performer. That may, however, make an incomplete relationship between activities because requests of design performers at each design stage are not unified or consistent; therefore, the final management of *activity drivers* should be left to a project manager or someone who can control the project as a whole.

One of the important purposes of using functional requirements except describing causal relationships between activities is management of design intent explicitly. A work itself to describe functional requirements among design activities is able to contain reasons why the design activities should be performed. The details about design intent will be discussed later.

2. Multi-Dimensional Design Process Management

The extended product model composed of product data models, activity models, and functional requirements was established as basic components to support CPE environment at the abstract level. In this section we will present how they can be aggregated and applied in order to acquire functionality for design description at the concrete level.

Design processes can be described more rigorously by analyzing chained structure of design activities in that activity sequences stand for design intent and histories implicitly in themselves [Taura et al., 1999]. Conventional activity models can also be regarded as one of the activity chains. They, however, have focused on managing a whole project rather than considering various design perspectives concurrently. As mentioned in the previous section, it is nearly impossible to make an activity model that can cover all kinds of design perspectives in a definite form because causal relations required from design perspectives strongly depend on design conditions. Consequently, an actual design process should be described by its own design activity sequence although the design process can be guided globally through a fixed activity model not to deviate from a central project management.

This paper proposes an activity chain model by functional requirements to provide consistent description of design processes. The activity chain model stands for a basic expression to be repeating units as shown in Fig. 8. Design activity sequences are re-

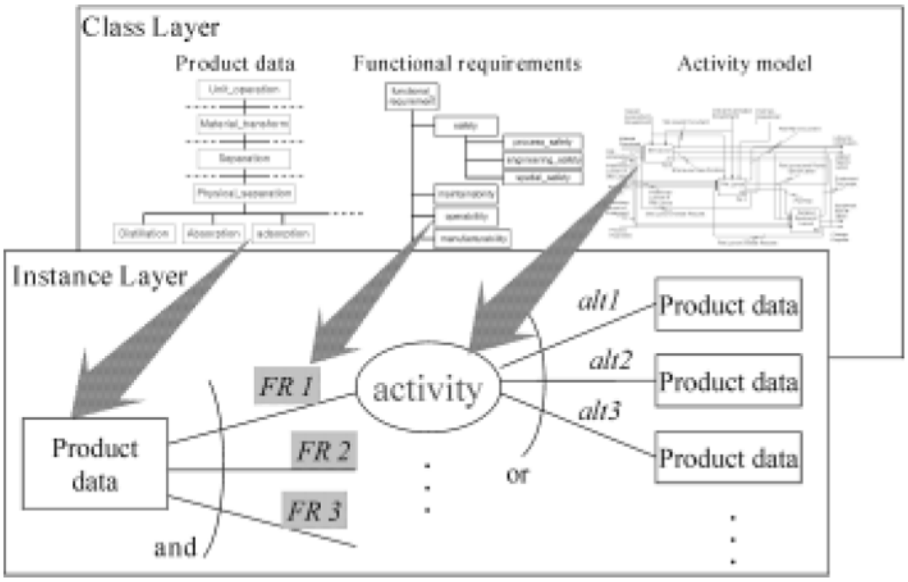


Fig. 8. Activity chain model.

organized by using a set of the repeating units. Product data located on the left hand side of this figure stands for a set of design results produced by the corresponding design activity. In other words, that means a state in a design process represented by the product data. At the design state, several design activities can be invoked by functional requirements to reflect various design perspectives. The invoked activities can be regarded as sub-goals of the activity that produced the previous product data. For example, suppose a designer gets reactor data in the conceptual process design stage for unit operation design. Then, he/she would want to verify whether the design results are feasible and adequate even from other design perspectives such as safety, controllability, manufacturability, etc. with respect to characteristics such as reaction material, temperature, pressure, flow characteristics, etc. If there are certain requirements to evaluate or to perform additional design, he/she can set up activity drivers as functional requirements. The drivers will invoke the corresponding activities. Functional requirements for the product data should be inserted with 'AND' relation because all of the selected functional requirements have to be satisfied. There may also be some other activities derived from the product data as following activity models without any specific functional requirements. They are represented by activity sequence relations defined in a meta model for activity drivers, and can be treated in the same way as functional requirements.

The activity invoked by functional requirements or activity sequence relations may make several design alternatives as the design results. For example, *identify safety* activity invoked by a *safety* requirement may request redesign of the reactor because the reactor cannot satisfy preliminary safety requirements, or request additional equipment design to mitigate hazardous factors. In the latter case, the activity for the safety evaluation may make another activity chain to be propagated over again. The alternative product data are related to the design activity with 'OR' relation; then, only one of them should be selected in the real design processes. In Fig. 8, the behind layer shows that the activity model works based on object-oriented concepts. The activities, product data and functional requirements in the front layer are represented by instances of classes defined in the extended product model.

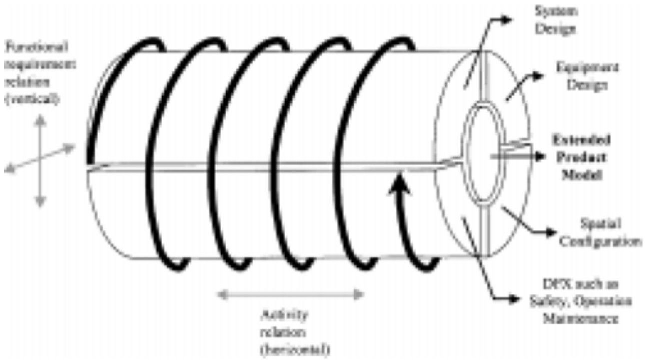


Fig. 9. Progression of design process based on extended product model.

Although the activity chain model as a repeating unit is very simple, the model provides a fundamental structure for multi-dimensional process management. Fig. 9 shows how design processes proceed in an environment supported by the extended product model. Design processes progress in a helix type, through parallelized design processes by major divisions for design processes in order to accomplish the main purpose of CPE. As shown in the figure, the helix goes forward by two-dimensional driving forces caused by two subtypes of activity drivers, respectively. One is horizontal dimension controlled by activity sequence relations. Direction for global design process is governed by this dimension for the design processes in order not to wander away owing to complicated functional requirement relations. The other is vertical dimension controlled by functional requirements. Activities in heterogeneous design processes can be linked to each other by using this dimension. The two-dimensional approach for activity management may leave a trade-off problem sometimes. In general, vertical relations between activities lead to network structure among activities, and increase distresses in managing a design process due to the complexity, compared with the process management by deterministic activity models. It is, however, believed that the two-dimensional approach can manifest requirements which may occur in real situations more intensively. In addition, a design activity guided by the vertical dimension can be considered as an activity which has higher priority with which the activity should be verified in the design process rather than every other activity instance within the activity class which should be performed at a time when the time for the activity class comes in activity models.

3. Design Intent Description

When a facility or a system is designed, there must be intent to design it. Traditionally, design intent has been described with text format in product data management system. The designer is forced to write intent such as *when, where, who, what and how*. Then, the design intent is managed with related product data. Design intent related with '*why*' is probably the most important thing to capture and describe. So far, capturing design intent for the '*why*' has not been completed in that it has been managed based on documents where product data are contained. Design intent description without considering design process can describe design rationales for the decision by which product data should be designed, but cannot describe why the design activity should be performed. There exist many cases when a design result should be revised, and additional design activities should be followed owing to the revision in general design processes. For these cases, intent description based on product data cannot describe why the design result should be revised or why the additional activity should be followed.

Design intent description using the extended product model will lessen the problems. The six categories for design intent can be described appropriately by using three types of the extended product model. Activity classes can include design intent such as *when, where, who* and *how* with their attributes. In this paper, the description is defined as simplified form because it is not the main focus; however, it can be extended more rigorously like a traditional product management system if necessary. Design intent corresponding to *what* is product data linked to activities. Finally, reasons why a design activity should be driven are described by functional requirements that make associations between design activities. Since functional requirements are instantiated with an explicit form in a data management system, the intent description can be achieved by data-driven approach, not by document-driven one. The characteristics are also one of the important advantages of using the extended product model.

IMPLEMENTATION AND DISCUSSION

The extended model equipped with multi-dimensional design process management and design intent description is realized as a Window-based design support system. Information for the extended product model is managed by a commercial database. Product, activity and functional requirement data merged by activity chain method have data relations with one another like Fig. 10. The figure is written in Entity Relationship Diagram (ERD), which is commonly used for database design. Entities represented by a rectangle may be regarded as a table in a database, and lines between entities stand for relationships with each other. Relationships should be defined with cardinality. Numbers in diamond-shaped boxes and both ends of a relationship mean maximum and minimum cardinality, respectively. As shown in the ERD, activity entities can contain more than one design alternative, and an alternative instance selected by decision-making is able to propagate one or multiple activity drivers, functional requirements or activity sequence relations. Finally, successive activities invoked by activity drivers are also defined as a type of activity entities. The loop shows how a repeating unit defined by the activity chain can be implemented in a database. On the other hand, a decision among alternatives can make multiple product data because we could not say that a design activity should be mapped into a product data instance classified through product data modeling. Product relations describe geometrical and logical connections among product data using associations defined in AP221. Since product data and activities are actually classified into so many classes with hierarchy, product data and activity entities are assigned to one of the classes as expressed in the ERD.

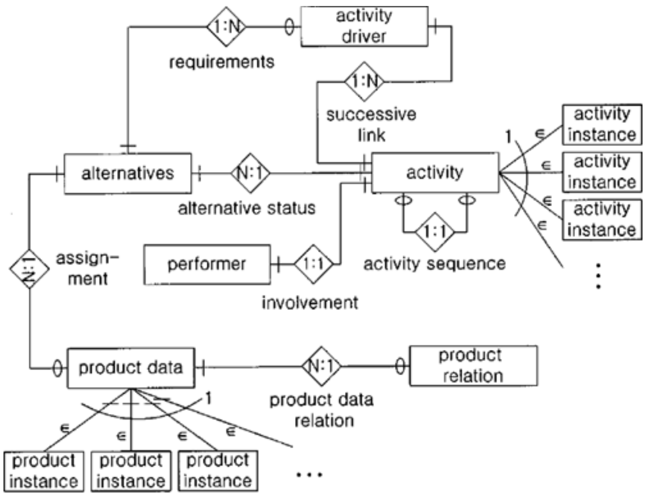


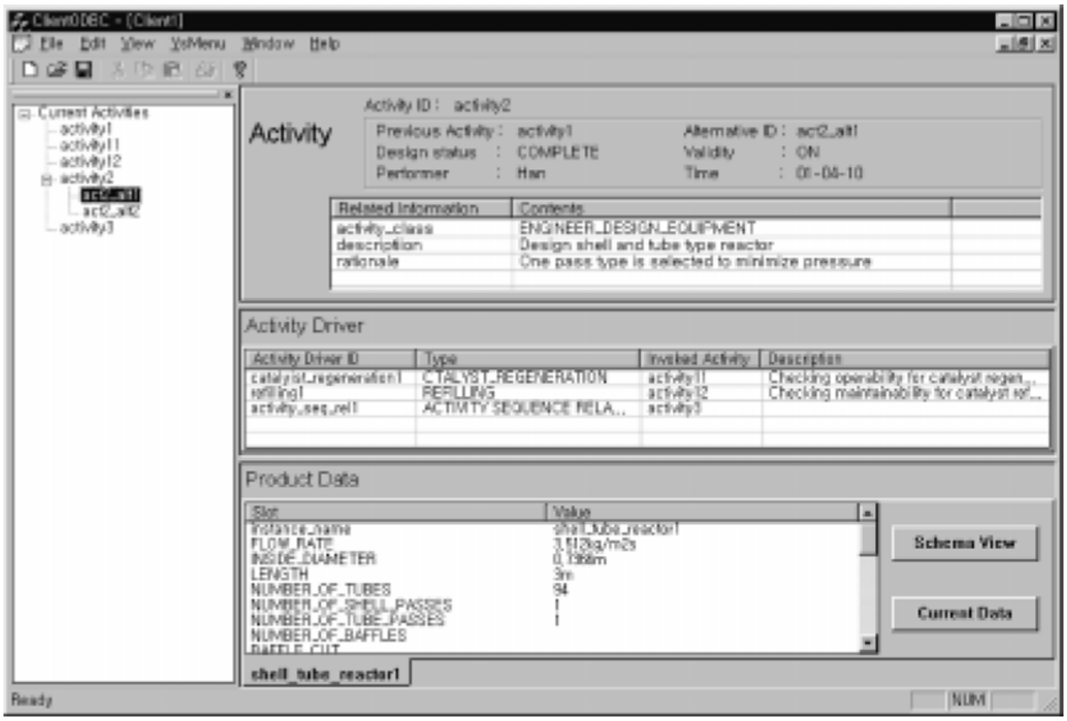
Fig. 10. Entity relationship diagram.

The data management system was implemented as shown in the following screen views. Fig. 11 shows the main window for the system where data management can be performed focused on activity data. It contains general activity data description defined in the meta model, a set of product data as results of the design activity, and activity driver information that describe which design activities can be induced from the results of the design activity.

The interfaces where an agent can access are classified into two types. One is for design agents (we call them just agents) and the other is for a design process manager (we call it just a manager). They can be regarded as clients and a server, respectively, in an aspect to manage a design process. We assume that the manager controls negotiation and approval processes related to decision making.

The figure shows a screen view when a design agent connects the system. The upper tree structure of the left side stands for all current activities and the lower one means specified design activities for the agent connected at present. The activities provided in the lower one can be managed only, that is, the agent can create alternatives and product data produced by his/her design activity. Other activities in the upper tree can be seen only as references.

The right hand's frame is classified into three parts to show each dimension presented in the activity chain model. The upper one contains information for activities. The information shown in the frame



Activity ID : activity2

Previous Activity : activity1
Design status : COMPLETE
Performer : Han
Alternative ID : act2_alt1
Validity : ON
Time : 01-04-10

Related Information	Contents
activity_class	ENGINEER_DESIGN_EQUIPMENT
description	Design shell and tube type reactor
rationale	One pass type is selected to minimize pressure

Activity Driver ID	Type	Invoked Activity	Description
catalyst_regeneration1	CTALYST_REGENERATION	activity11	Checking operability for catalyst regen...
refilling1	REFILLING	activity12	Checking maintainability for catalyst ref...
activity_seq_rel1	ACTIVITY SEQUENCE RELA...	activity3	

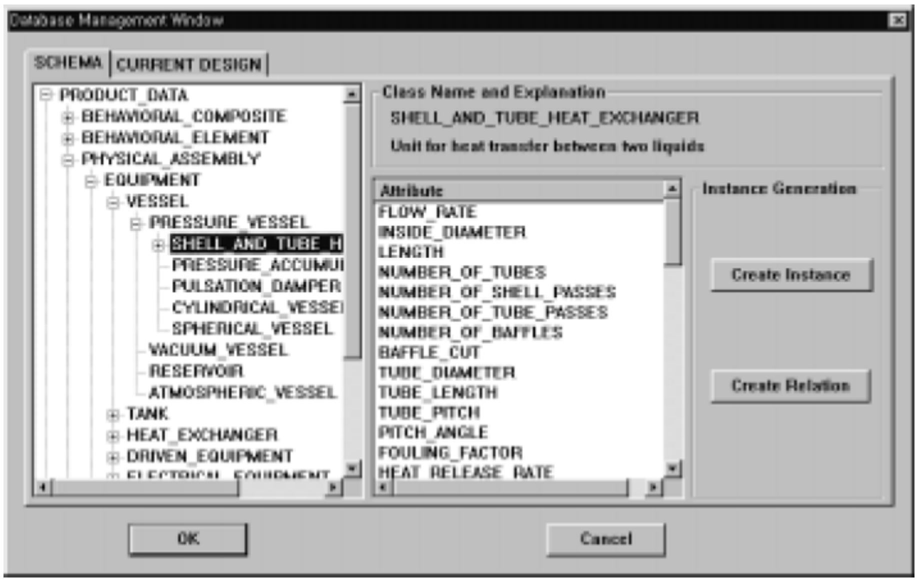
Slot	Value
instance_name	shell_tube_reactor1
FLOW_RATE	3.512kg/m2s
INSIDE_DIAMETER	0.1368m
LENGTH	3m
NUMBER_OF_TUBES	94
NUMBER_OF_SHELL_PASSES	1
NUMBER_OF_TUBE_PASSES	1
NUMBER_OF_BAFFLES	
BAFFLE_CUT	

Fig. 11. Screen view of activity manager.

contains attributes defined in the activity meta model. Among the attributes, the design status means all of the alternatives for the activity are proved by a manager and there is no design activity being performed currently. The validity means the selected design alternative is feasible and believed at current design status. The value may be changed whenever a decision revision occurs. Thus, if design status is COMPLETE and validity is ON, results of the design activity are acceptable, and can be the basis for other design activities currently.

Activity drivers such as FR and AR shown in the middle frame stand for relations between activities. They may be determined by the design agent who performed the activity or a manger. The number of relations can be added during design processes if needed. The relations will be effective when a manager approves them, then they are used for agents to trace design flow.

Finally, product data made by the activity are shown in the lower frame. Attributes to describe a product instance have been defined in product data classes and the corresponding values are determined by the agent. Creation of a new product instance is executed by clicking the *Schema View* button. If an agent wants to see all current product data and product relations designed through a whole design process, the lower button is used.



SHELL_AND_TUBE_HEAT_EXCHANGER

Unit for heat transfer between two liquids

FLOW_RATE
INSIDE_DIAMETER
LENGTH
NUMBER_OF_TUBES
NUMBER_OF_SHELL_PASSES
NUMBER_OF_TUBE_PASSES
NUMBER_OF_BAFFLES
BAFFLE_CUT
TUBE_DIAMETER
TUBE_LENGTH
TUBE_PITCH
PITCH_ANGLE
FOULING_FACTOR
HEAT_RELEASE_RATE

Fig. 12. Screen view of product schema manager.

When a designer wants to make product data for his/her design purpose, standard product data can be referred to from the class libraries shown in Fig. 12. The window supplies a great deal of the product data class with the classification defined in this paper. A designer can create instances of product data corresponding to the class definition, and the created product data can be managed with the designeris design activity in the main window. Based on this environment, product data and information of the design process can be dealt with simultaneously. More important is the fact that the system makes it possible to represent an integrated design process very explicitly and describe design intents and histories through the design management itself.

CONCLUSIONS

We have presented a methodology for multi-dimensional design process management in order to accomplish CE in chemical process design. One of the main obstacles to achieving CPE has been that there is no appropriate methodology to integrate various design processes characterized by different design perspectives.

In this paper, we proposed the extended product model modified from conventional approaches focused on product and process models by adding the concept of functional requirements. Functional requirements are considered as another view of a conventional design activity model. That means functional requirements are defined by extracting causal reasons that exist inherently in design processes. Design processes can be controlled at last in multi-dimensional aspects by the functional requirements. One dimension is a general direction of design processes following the activity model, and the other one is the direction to enable crossover control among design processes that have different characteristics from the view of data and organization. We called them horizontal and vertical dimension, respectively.

The concept of activity chain was proposed to describe the multi-dimensional process management coherently. As activity chains make design branches using alternatives and activity driversí representation, the entire design processes are gradually completed in a parallel manner, not sequential one. In addition, the product data model was reformulated a little to make clear classification by two criteria. One is a view of data usage in dominantly different design processes. Product data can be classified into physical and behavioral ones from this viewpoint. The other is a view of scale that classifies product data into composites and elements or assemblies and components to reduce redundant definition of data. Finally, we made an environment for product data management based on a commercial database as a prototype. The system enables a user to refer to previous activities and product data, and to record design results in the same manner. The system will also play the role of a basic frame for a collaborative design environment.

ACKNOWLEDGEMENT

This work was partially supported by the Brain Korea 21 Project and by the Korea Science and Engineering Foundation (KOSEF) through RRC/NMR Project #01C04.

REFERENCES

Batres, R., Naka, Y. and Lu, M. L., "A Multidimensional Design Framework and Its Implementation in an Engineering Design Environment," *Concurrent Engineering*, **7**, 43 (1999).

Bayer, B., Schneider, R. and Marquardt, W., "Integration of Data Models for Process Design ñ First Steps and Experiences," *Computers and Chemical Engineering*, **24**, 599 (2000).

Bogle, D. and Perris, T., "CAPE.NET Responding to the Business Challenges," *Computers and Chemical Engineering*, **23**, S779 (1999).

Book, N. L. and Bhatnagar, V., "Information Models for Planning and Scheduling of Chemical Processes," *Computers and Chemical Engineering*, **24**, 1641 (2000).

Bullinger, H. J. and Warschat, J., "Concurrent Simultaneous Engineering Systems," Springer (1995).

Chandrasekaran, B., Goel, A. K. and Iwasaki, Y., "Functional Representation as Design Rationale," *IEEE Computer*, **January**, 48 (1993).

Chung, Y. and Fischer, G., "A Conceptual Structure and Issue for an Object-Oriented Bill Of Materials (BOM) Data Model," *Computers Ind. Engng*, **26**, 321 (1994).

Colquhoun, G. J., Baines, R. W. and Crossley, R., "A State of the Art Review of IDEF0," *Int. J. of Computer-Integrated Manufacturing*, **6**, 252 (1993).

Gorti, S. R., Gupta, A., Kim, G. J., Sriram, R. D. and Wong, A., "An Object-Oriented Representation for Product and Design Processes," *Computer-Aided Design*, **30**, 489 (1998).

Han, S. Y., Kim, Y. S., Lee, T. Y. and Yoon, T. S., "A Framework of Concurrent Process Engineering with Agent-Based Collaborative Design Strategies and its Application on Plant Layout Problem," *Computers and Chemical Engineering*, **24**, 1673 (2000).

Han, S. Y., Kim, Y. S., Yoon, T. S. and Lee, T. Y. "Modeling of Process Equipment by Object Modeling Technique for Integrated Design System," Proc. APCChE99, 207 (1999).

Han, S. Y., Lee, T. Y., Yoon, T. S. and Naka, Y., "Information Sharing Between Process and Engineering Design Activity in CAD Environment," *Computers and Chemical Engineering*, **23**, S573 (1999).

Inmon, W. H., Zachman, J. A. and Geiger, J. G., "Data Stores, Data Warehousing, and the Zachman Framework," McGraw-Hill (1997).

ISO10303-11, "EXPRESS Language Reference Manual," ISO TC184/SC4/WG5 (1991)

ISO10303-221, "Functional Data and Their Schematic Representation for Process Plant – Committee Draft," ISO/TC184/SC4/N592 (1997).

ISO10303-227, "Plant Spatial Configuration – Draft International Standard," ISO/TC184/SC4 (1997).

ISO10303-231, "Process Engineering Data: Process Design and Process Specifications of Major Equipment – Committee Draft," ISO TC184/SC4/WG3 N745 (1997).

ISO10303-41, "Fundamentals of Product Description and Support," ISO/TC183/SC4 N (1997).

ISO13584, "Industrial Automation Systems and Integration – Parts Library," ISO TC/184/SC4 N290 (1995).

King, J. M. P. and Banares-Alcantara, R., "Extending the Scope and Use of Design Rationale Records," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, **11**, 155 (1997).

Kjellberg, T. and Schmekel, H., "Product Modeling and 'Information-Integrated' Engineering Systems," *Annals of the CIRP*, **41**, 201 (1992).

Krause, F. L., Kimura, F., Kjellberg, T. and Lu, S. C. Y., "Product Modelling," *Annals of the CIRP*, **42**, 695 (1993).

Liu, T. H., Trappey, A. J. C. and Shyu, J. B., "ISO 10303 Based PCB Assembly Data Model for Assembly Analysis," *Concurrent Engineering*, **7**, 159 (1999).

Lu, M. L., Yang, A., Li, H. and Wada, T., "Application Driven Approach for the Development of a Data Model Standard for Process Plant Operation," *Computers and Chemical Engineering*, **24**, 463 (2000).

McKay, A., Susan, M. and Pennington, A., "A Framework for Product Data," *IEEE Transactions on Knowledge and Data Engineering*, **8**, 825 (1996).

Owen, J., "STEP An Introduction," Information Geometers Ltd. (1993).

Peltonen, H., Pitkanen, O. and Sulonen, R., "Process-Based View of Product Data Management," *Computers in Industry*, **31**, 195 (1996).

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F. and Lorensen, W., "Object-Oriented Modeling and Design," Prentice-Hall International, Inc. (1991).

Shah, J. J., Jeon, D. K., Urban, S. D., Bliznakov, P. and Rogers, M., "Database Infrastructure for Supporting Engineering Design Histories," *Computer-Aided Design*, **28**, 347 (1996).

Taura, T. and Kubota, A., "A Study on Engineering History Base," *Research in Engineering Design*, **11**, 45 (1999).

Taura, T., Aoki, Y., Takada, H., Kawashima, K., Komeda, S., Ikeda, H. and Numata, J., "An Activity Chain Model and Its Application to Global Design," *Concurrent Engineering*, **7**, 245 (1999).