공학 연구논문집 제30권 1호 pp. 519~538, 울산대학교.
Journal of Engineering Research Vol. 30, No. 1, pp. 519~538, 1999
University of Ulsan.

A Schematic-to-Detailed Design Process of Buildings by the STEP-based Product Model

Kim, E-Doo
School of Architecture, University of Ulsan

<Abstract>

The Internet is the most recently growing information technology and also enables the information throughout industries to be shared on the network. The Product Model of the STEP expected to be the most popular means for representing and exchanging engineering information between computer applications, makes engineers' intentions more clearable and exchangeable than simple conventional graphical representations like drawings. In structural engineeering areas of AEC industry, only AP 230 - Building Structural Frame : Steelwork - was presented as a working draft. But, this Application Protocol needs to implement some special characteristics of building structures and a schematic-to-detailed design approach with problem decomposition which many architects and structural engineers in practice take in their design process of buildings. This paper presents a methodology for supporting the structural design process in details for the implementation of these special issues of building structures.

STEP의 프로덕트모델을 이용한 건축구조 설계과정의 구현에 관한 연구

김이두
울산대학교 건축학부

<요 약>

급속히 성장하고 있는 인터넷은 네트웍을 통해 모든 산업분야의 정보를 공유할 수 있도록 하는 정보기술로서 자리잡고 있다. 컴퓨터 프로그램들간의 엔지니어링 정보의 표현과 교환을 위한 국제표준안을 위하여 ISO에서 진행하는 STEP의 프로덕트모델은 엔지니어의 의도를 전통적인 도면을 이용하는 방식에 비해 보다 명확한 전달을 가능케 하고 있다. 그러나 건설분야에서는 아직 공인된 정보교환을 위한 국제표준안은 없으며, 철골건축구조물에 대한 시안(Application Protocol 230)이 제출되어 있다. 그러나 이 안은 건축구조물이 가지는 특성과 건축가 및 구조설계 실무자들이 수행하는 건축물의 설계과정, 개괄적인 설계에서부터 세부설계로 진행해 나가는 설계과정의 특성을 표현하지 못하고 있다. 본 논문에서는 현재 제시된 시안에 기하요소들을 추가하여 그 공간적인 관계를 활용함으로서 이러한 특성들을 구현하기 위한 방법을 제시하였다.

1. INTRODUCTION

The Internet is the most recently growing technology in society today and it is changing the way people work and communicate. This impact changes work practice in engineering and construction companies, and these changes will continue as new information technologies arise. Many experts in industries share the information about CAD and engineering data via FTP or Web browsers. The commercial applications like AutoCAD allow the users to transfer the data[1], and an internet company of CAD data bank in Japan provides engineering drawings for bridges and retaining walls[2]. The Ministry for National Development in Singapore tries to develop Construction and Real Estate NETwork (CORENET). The system is expected to allow building professionals electronically to make applications to the various government agencies, have the building applications checked and processed by using expert systems, conduct construction contracts, and make on-line inquiry about real estate information etc[3].

Although the use of information technology is rapidly growing in Architecture, Engineering and Construction (AEC), there is no product information standards. A lot of applications are dependent on the companies and even the departments in the same company, so that the data exchange between such applications can not be successfully made via network. The International Organization for Standardization issues the growing need throughout numerous industries for an integrated approach to the use of IT. The International Standards is normally prepared through ISO technical committees. ISO 10303 is an international standard for the computer-interpretable representation and exchange of product data, which provides a neutral mechanism capable of describing product data throughout the life cycle of a product, independent from any particular system. The nature of this description makes it suitable not only for file exchange but also as a basis for implementing and sharing for product databases and archiving[4].

AP230 of ISO 10303 is a Draft International Standard, which is being circulated to the member bodies for voting, specifies an Application Protocol for the representation and exchange of information on structural steel frames. Application protocols provide the basis for developing implementations of ISO 10303 and abstract test suites for conformance testing of AP implementations. The model is based on the CIMsteel Integration Standards from the Eureka 130 CIMsteel project which was aimed at improving the efficiency and effectiveness of the European constructional steelwork industry by introducing Computer Integrated Manufacture. The ISO-aligned CIS offers the industry an interim solution to data exchange, and also provides firm foundations for the development of an ISO standard. AP230 relates to the computer applications of analysis, member design, connection design, and detailing functions for the designers and constructors of buildings. The analysis model describes the connectivity of elements and nodes, the design model describes the geometry and assembly of parts and connectors, and the manufacturing model describes the physical location and properties of parts and joints[5].

This paper, at first, briefly reviews the data model of AP230 focused on the representation of structural parts from the standpoint of structural engineers in practice, and then describes the characteristics of the building design processes and design information flows that many of architects and structural engineers take in their design of building structures. In order to implement the engineers' schematic to detailed design approach, a new product model expands the data model of AP230 by introducing the geometry-based abstraction into the entity-based data model, and thus, satisfies the conformance to AP230.

2. BRIEF REVIEW OF AP230

Among the five stages of the life-cycle of a building, i.e., plan, design, construct, use, and demolish, the current version of AP230 - Building Structural Frame : Steelwork - supports the exchange of data during the first three stages, i.e., plan, design, and construct as the activities in building industries. There are a lot of activities for steel-framed building design, such as, the transfer of quality-specifications information into design, design activities, the feedback of information from designers to project planners/managers, the transfer of information into fabrication and erections, and the feedback of information from fabricators and erectors to project planners/managers. Under these design activities, AP230 addresses: structural design, loading assessment, structural scheme modelling, structural analysis, member design, connection design, and steelwork detailing.

The Application Activity Model (AAM) represents the scope and information requirements in terms of a set of definitions and a set of IDEF0 diagrams. The information requirements are specified as a set of units of functionality, application objects, and application assertions. The graphical representation of AAM indicates which processes and information flows form the basis of the corresponding application protocol. Some examples of AAMs in AP230 are shown in Figures 1 to 4. Figure 1 shows the overall processes for steel-framed structures, where the *manage total building project* (A1) initiates plan, monitor and control overall project, the *produce overall design* (A2) develops client's brief into a viable structural scheme, the *produce detail design* & the *prepare for production* (A3) adds required detailed information to allow the construction of the structure, and last, the *construct and hand over building* (A4) constructs structure in accordance to the developed design.

2.1 Units of Functionality

The brief description of the units of functionality focused on the design processes of the Building Structural Frame: steelwork AP are as follows(See Figures 2 to 4).

The *project brief* UoF in Figure 2, which is produced by the *manage overall design* (A21) activity and required by the *produce concept design* (A22) activity, comprises relevant areas of the client's brief, and it forms the basis of initial concept design. This UoF holds information about the project, the site, and the structure; and information about the basis for the design: the structure's use, its design life, number of stories, maximum and minimum dimensions, loadings, and environment.

The *structural scheme* UoF in Figure 3, which is produced by the *develop structural design* (A252) and required by the *assess loading* (A253) and the *model structural scheme* (A254), is essentially a sketch design including preliminary sizes and positions of some of the main members, and provides the basis for the idealization of the structure. It identifies key members, and the method of providing lateral stability, but it provides little detail as to how the sticks really fit together. The *structural scheme* holds structural layouts, associated member sizes, calculations and textual descriptions of the structure.

The *overall structural design* UoF in Figures 1, 2 and 3 comprises information existing after analysis but before member and connection design and it may relate to a number of structural models. The overall structural design represents a major embellishment of the initial scheme design. As part of the *develop structural design*, the information gathered from the analysis is assessed in terms of overall serviceability of the structure to see whether the initial estimates of the main members are suitable.

If they are not suitable, the sections may be changed and the analysis performed again. Structures are always simplified for analysis, and often only a small portion of a complete structure is analysed. At the *design members* activity, a typical member will be designed on the basis of analysis results for a number of elements. It holds a description of the analysed design, indicating the position, type and initial size of the structural elements required to ensure the design fits the requirements. It contains the structural concept, layout, and specification for the main structural members, required by the structural engineering designer for member and connection design.

The *design loading* UoF in Figure 3 comprises general floor/wall/roof loading - a loading schedule including characteristic loads and partial factors of safety. It holds the predicted loads on the structure which are used to create load cases. The *basic load cases* UoF, which are produced by the *model structural scheme* and required by the *perform analysis* (A255), include dead, live, and wind load cases. They holds analytical load cases, based upon the design loading, which are to be used during analysis. The *load case combinations* UoF includes serviceability or ultimate loading and most analysis applications cover them as part of a larger program. They holds multiple combinations of the basic load cases which are to be used during analysis.

The *structural models* UoF in Figure 3, which are produced by the *model structural scheme* and are required by the *perform analysis*, are wire frame models and are idealized analytical representations of the structure. Structural models are made up of nodes and elements, with associated material and geometric properties and boundary conditions. A single structure may require many structural models, and each model may be subjected to a number of load case combinations. They holds models of aspects of the structure to be used for analysis. The *analysis results* UoF in Figures 2 and 3, which are produced by the *perform analysis* and required by the *design members* and *design connections*, comprise the results of analysis of the structural models, and describe the response of the structure to the design loading. They holds the results of the analysis after a wire frame model has been created, loaded and analysed.

The *member design information* UoF in Figure 4, which is produced by the *design members* and required by the *design connections* (A33) and the *detail steelwork*, is the information required to detail the members like section size, material spec, stiffeners, requirements for restraints, assumptions made during design etc, and holds details of each member for detail design. Where members require restraints to prevent buckling, the positions and loading of these is part of the member design information. The *connection design information* UoF in Figure 4 is the information required to detail the connections (bolt groups, welds, etc.) like plate sizes, bolts, nuts washers, material spec, stiffeners, requirements for restraints, assumptions made during design etc., and holds details of each connection for detail design. Where connections require restraint to prevent premature yielding, the positions and loading of these is part of the connections design information. By this stage of design, the main members and connections have been shown to be of sufficient capacity to carry the expected loads. As they have yet to be detailed, there are no features associated with the members and connections.

The *detailed designs* UoF in Figures 1 and 4, which are produced by the *detail steelwork* and required by the *construct & hand over building* (A4 in Figure 1) activity, holds the detailed information, in various forms, necessary to construct the structure. They are required for the fabrication and erection of the steelwork. The structural detailer takes member connection design information and applies them to the stick design created in the *structural scheme*. Members and connections may be modified during the process of design. The detailer makes sure that the steelwork will actually fit together. The main difference between this detailed design stage and earlier stages is the requirement for features like notches, chamfers, holes, etc.. Detailed design information is passed on to fabricators in order that they may begin the manufacturing process. At this point, each and every piece of steel in a structure should now be fully defined and detailed with dimensions for every cutting and drilling operation required. Every bolt group, weld, end plate, stiffener, etc. should also be identified.

2.2 Application Objects and Assertions

The information requirements are also described in terms of application objects and application assertions. During the process of the units of functionality aforementioned, a lot of application objects are used as their information storage. The assertions pertain to individual application objects and to relationships between application objects and are defined in terminology of the AP domain. In the AP23, there are currently about 121 structural application objects and other entities for basic primitives like description, identifier etc; for time primitives like year_number, month_in_year_number etc; and for measuring units like length_measure, mass_measure, etc. Each application object is an atomic element that embodies an unique application concept and contains attributes specifying the data elements of the object. Figure 5 shows an example of ARM diagrams in EXPRESS-G of AP230.

Figure 6 shows a brief ARM diagram of principal application objects in EXPRESS-G, where only the key properties of individual application objects and relationships between them are represented for this study. Where, the secondary and inverse attributes of the entities was omitted for representation of its key structure.

As shown in Figures 5 and 6, The modeling process for steel structural frames begins with the *design_assembly* entity, which is a subtype entity of the abstract *assembly* and can represent a whole structure and any number of lower-level structural parts with the *struc_frame*, *struc_member*, and *struc_connection* entities. Each *design_assembly* has a set of component parts of *design_part*, a set of component connections of *connector*, and a set of analytic models of *analysis_model*. The connections are detailed by *asbly_connection* for *design_assembly* and connector for *design_part* respectively. But, both of them are finally represented in terms of *s_part* and *s_joint_system*.

3. A SCHEMATIC-TO-DETAILED DESIGN PROCESS

3.1 Characteristics of Building Design Process

Most of buildings consist of a series of vertical frame and/or shearwall substructures interconnected through horizontal floors, and they consist of the basic components that are mainly subject to these substructures. Due to these special characteristics of the building structures, many architects and structural engineers in practice take in their design process of buildings a schematic to detailed approach with problem decomposition, where they decompose a building structure into several substructures such as floors or frames at first, and then, allocate components, such as, slabs, beams, and columns, considering the interactions between the higher-level substructures. These lower-level components are then detailed one by one or all of them in groups by higher-level substructures at a time[6-8].

Therefore, the design problem can be classified into two categories, i.e., substructural- and component-level ones. The each-level design problem is subdivided into the processed of synthesis, analysis, and evaluation. At the initial stage of design, designers synthesize potential solutions for the current design problem satisfying a few critical constraints formulated from the given conditions, and then, select one or a few solutions to be pursued further in the later stage. The analysis process transforms a physical problem to a mathematical model, analyzes the model, and interprets the results of the analysis. Lastly, the evaluation is made to get the most effective and economical solution to the design conditions.

Substructural-level design

At the stage of the substructural-level design, all of the possible schematic description of the building configurations are made in terms of mainly frame and floor substructures, and appropriate ones are selected in order to provide the initial database for the detailed design. Structural engineering domains relative to this issue will be planning, and preliminary design of the selection and arrangement of lateral and vertical load-resisting structural systems, which significantly affects the construction cost and the functionality of the building. During the planning stage, the project description, such as building occupancy, material, layouts including the number of grids and their dimensions in three directions, and design codes, is made by architects, clients, and structural designers. This initial context will be a motive for the later design stages. This planning may correspond to the *project brief* UoF in Figure 2.

At the preliminary design of building structures, the only information available to the designers is the architectural specifications and a few constraints which must be taken into account, so that the creativity of engineers and their experiences obtained through long years of practices are mostly needed during this stage of the design process. All the UoFs from *produce concept design* (A22 in Figure 2) to *structural scheme* in Figure 3 may correspond to this category. The knowledge-based approach is considered as a practical scheme to aid this stage of design process[9,10].

For example, HI-RISE[10], the early expert system, showed the effectiveness of application of artificial intelligence technology in this less formalized areas of problem solving. HI-RISE utilizes plan, generate and test strategy for the synthesis process. The plan prepares design spaces in terms of several levels of abstraction of independent or loosely coupled subsystems, generates building candidates under constraints, and determines which ones are an eligible or efficient alternatives. The analysis expands the candidates to the level of structural elements by collecting whole the structural data needed for analysis. Lastly, the analytic results are used to check structural conditions such as overturing moments, deflections including story drifts, and critical member section properties based on member forces.

Component-level design

With the structural configuration and the analysis results of the building, the designers determines the physical and structural properties of members and connections. All the members and connections are detailed for the worst case of loading states in accordance with design codes and specifications. At first, the engineers synthesizes whole design information such as component configurations like geometry and topology; structural data like restraint conditions and member forces under various loading states; and other auxiliary data like material and sectional properties. Then, the key design parameters for member and connection designs are assumed and resolved, while the dependent variables are formulated according to the design theories of each component. With the details obtained, the designer checks the capacity of members and connections by applying the analytic procedures of design codes. Last included is the evaluation process for deciding which design parameters are most appropriate for the current design. These classes of problems are well defined in sequence are routine, and then it solved by step-by-step algorithmic approach. The *design members* (A32 in Figure 4) and *design connections* (A33 in Figure 4) correspond to these processes.

3.2 An Expanded Data Model for the Design Process

The design approach is a top-down hierarchical one where determinations of overall concepts precede to those of detailed ones. For example, top level description of the building consisting of frames is first made, and frame level, which consists of its components such as beams, columns, walls, and slabs, are secondly described, and last constituent detail components which are related to frame and component level are described. In order for structural engineers to perform this top-down structural design process of building structures, some conditions are required as follows.

◇ Most of the structural entities should be initiated and generated from the entities of architectural drawings, because the structural information is obtained from the architectural initial context as shown in the *produce concept design* (A22 in Figure 2) and the *produce overall architectural design* (A23 in Figure 2)

◇ The new model must incorporate the evolution property of design entities being handled during the whole design processes from the *produce overall design* (A2 in Figure 1) and the *produce detail design & prepare for production* (A3 in Figure 1) activities. Especially, some entities must be able to convert their abstraction types to another ones according to the design processes.

◇ The structural entities should be hierarchically handled with the higher aggregation entities with their geometrical representations when the design goes down from the overall to detail design levels. The lower-level entities should be automatically grouped into the higher-level entities by the designers' declarative constraints but not by their every designations.

◇ To model buildings with the top-down approach, it is required that the constituent elements in the specified frames and floors should be automatically generated from their own information when the designer describes the building structures at the substructural level. When the designer does his work at the substructural or component level, all the modeling elements should be also divided into the most basic component and merged into the larger ones as well.

To implement the requirements described above, the product model for building structures will be more proper to be structured on the basis of the geometric entities, such as, surfaces, lines, and points, and their topology than the physical components such as beams, columns, and slabs, and their assemblages of frames and floors. From the nongeometry-based abstraction, the geometry and topology comprising most of the model data and also being of major importance for structural design, are difficult to be automatically constructed. Thus, the user should specify detailed information not only on the entities that make up the structure and their parent-child hierarchies, but also on their topology. Therefore, the previous studies proposed some complex special techniques such as G. H. Powell's component-connections[11] and K. H. Law's connections[12]. However, the geometry-based abstraction provides mathematically well-defined geometric entities, so that the hierarchical and topological information can be automatically constructed from the spatial relationships of the geometric entities.

This study expands the product model by adding the geometry properties to the structural entities in AP230. The geometric and topological properties of the expanded entities for the implementation of the top-down design process are described below.

Substructural-level entity description

As shown already in Figures 5 and 6, the primitive abstraction entities for the structure or their substructures are the *design_assembly* entities, which are instantiated as one of the *struc_frame* or the *struc_member* or the *struc_connection* type. This study defines a new subtype for each of these three subtypes by adding some geometric constraints for grouping all the *design_part* entities. All three entities are defined to have more information about a set of geometrical entities like planes and lines. Hear, each plane becomes a modeling primitive for a frame or floor substructure located in the arbitrary direction in space, so that it must have the information about its normalized surface equation, local coordinates system, coordinate transformation matrix and its inversion, and so forth. If there is no bounding information like boundary edges or points, it plays a role of an architectural grid in a building structure, grouping all coplanar ones as a highest-level primitive. The line entity describes a column line in the architectural plans of buildings. There are two line types of entities; a finite and infinite one. The lines with the same equation are grouped by the infinite one as their parent.

Component-level entity description

The components, such as, joints, slabs, beams, columns, shear walls, and diagonals comprise a building structure, and they are also used for structural analysis and design. To abstract these components in accordance to AP230, this paper defines a new subtype entity of the *design_part* by including, as a new attribute, a set of topological entities like vertices, edges, and faces which may be used as elements for structural analysis. These topological entities have their own local coordinates system and also basic geometrical information by which they can be defined in space.

3.3 Implementation of the Top-down Design Process

When the user models his building with the primitives, a number of the topological entities, i.e., geometric subdivisions need to be automatically structured on the basis of their topology, because these entities, what the designers want to get for their design, become motives to model the structural elements in buildings, making it possible for a designer to describe a building structure at the high levels. That is, when a primitive is inserted, the old geometric and topological divisions need to be divided into smaller subdivisions, and when a primitive is deleted, the geometric subdivisions with lower dimensions need to be synthesized into higher ones. In addition to these subdivision or merging of design entities, the element properties must be managed during the design process. The properties also can be initiated with the same ones obtained from the topologically neighbouring entities which are sharing the bounding vertices or edges of the current entities.

Therefore, the implementation problem of the design process becomes the subdivision process of the *design_part* entities and managing them in terms of the aggregation *design_assembly* entities. The strategy is that the intersection problem of the *design_assembly* entities is converted into that of the *design_part* entities. When the designers adds a *design_assembly*, the application system must check the spacial relationships with all the existing *design_part*s of another *design_assembly*s and if required, split such *design_part*s into smaller ones. For example, in case that there are two faces; $face_i$ in a *design_part* and $face_j$ in another one, there are two cases of interactions, on the same plane and in space as shown in Figure 7[13].

Relationship on the same plane

If two faces are on the same plane, the following intersection process is performed in two-dimensional space of the surface. Figure 7(a) shows such an example case for our discussion, where most of the aspects of intersections are included.

Step 1. Intersect all pairs of the edges - Each pair of edges in two faces is checked for possible intersection. Let one pair of the edges, E_i in $face_i$ and E_j in $face_j$ as follows;

$$E_i : a_i u + b_i v + c_i = 0, \quad E_j : a_j u + b_j v + c_j = 0 \qquad (1)$$

where, both of a_i and b_i, and both of a_j and b_j are not zero. When $a_i b_j - b_i a_j$ is equal to zero, two edges are parallel. If they are on the same line, the larger is divided into smaller ones. Otherwise, an intersection point $P(u, v)$ is obtained with the following two-dimensional u and v coordinates; If these coordinates exist within the common boundaries of two edges, this point will be the intersection of the edges(see intersection points in Figure 7(a)).

$$u=\frac{b_ic_j-b_jc_i}{a_ib_j-a_jb_i},\quad v=\frac{c_ia_j-c_ja_i}{a_ib_j-a_jb_i} \tag{2}$$

If there is no intersection between two faces, either they do not intersect at all or one face completely surrounds the other. To distinguish between these cases, and determine which one is inside of the other by calculating a sum of angles. The sum of angles is calculated at a vertex of one face by traversing the boundary edges of the other one. If the angle is equal to zero, the face is outside. If it is equal to 2π, the face is inside. If one face is contained in the other, the containing face is divided into two smaller faces; one for the contained face and one for its complementary face.

Step 2. Extract the common edge lists - By traversing the edges of each face, the lists of the common edges shared by both faces are obtained as follows:

◇ If a common edge between two faces exists and its direction is opposite to each other, two edge lists for each direction are added into the edge lists as $\overline{V_{i3}V_{j6}}$ shown in Figure 7(a). If the edge has the same direction, one edge list is obtained like $\overline{V_{i8}V_{j2}}$.

◇ All the subsequent edges inside the opposite face have one pair of edge lists with the opposite direction like edges, $\overline{P_1P_2}$, $\overline{P_3P_4}$, and $\overline{P_5V_{i8}}$ of $face_i$, and $\overline{V_{j2}V_{j3}P_5}$, $\overline{P_4P_1}$, and $\overline{P_2P_3}$ of $face_j$ shown in Figure 7(a).

Step 3. Update the old faces - After all the common edge lists of two faces are calculated, the faces are subdivided. If there are no edge lists or there are only the edge lists on the boundaries of the faces, no faces are divided. Otherwise, the subdivision process is performed by traversing in the leftmost direction for each one in the edge lists. If the faces bounded by the edges only in the common edge lists are connected to both parent face of two faces(See the faces f_3 and f_5 in Figure 7(a)). Otherwise, the face is connected to its previous parent face. For example, faces $f_1$and f_6 are connected to the parent face of $face_i$, and also faces f_2 and f_4 are connected to that of $face_j$.

Relationship in space

In case of non-pararell faces, the intersection segments shared by two faces may exist. This intersection line is obtained as following procedures.

Step 1. Find all the intersection segments for each face - All the intersection points of one face to the infinite plane of the opposite face are first calculated. The intersection point is obtained by considering the distances, D_i and D_j of the starting and ending vertices, V_i and V_j of the current edge from the opposite plane. Here, each distance is obtained when its coordinates are substituted into the normalized plane equation of the opposite plane. The relations between the current edge and the opposite face are classified as follows(See Figure 7(b));

Case 1. $D_i * D_j > 0$: The edge, like $\overline{V_{i1}V_{i2}}$ exists outside the opposite face $face_j$.
Case 2. $D_i * D_j = 0$: This edge meets on the opposite plane. When D_i is equal to zero, the starting vertex V_i is inserted into the list of the intersection points. Such vertices as V_{i3}, V_{i4}, and V_{i10} are inserted into the list.
Case 3. $D_i * D_j < 0$: The edges like $\overline{V_{i5}V_{i6}}$ and $\overline{V_{i6}V_{i7}}$ pierce the opposite plane. The intersection point $P(x, y, z)$ like P_{i3} and P_{i4} is directly obtained from the following simple formula, and then inserted into to the point list.

$$P = V_i + (V_j - V_i)\frac{D_i}{D_i - D_j} \quad (3)$$

To construct the line segments, all the intersection points are ordered in the direction of from the first to secondly founded points. If zero or one intersection point is found during one face traversing, in which case one face is positioned in one half space of the other, the pair of faces is not considered as the intersected one. Otherwise, all the line segments between two subsequent points are checked whether they are inside or outside of each face. For example, three line segments, $\overline{P_{i1}P_{i2}}$, $\overline{P_{i3}P_{i4}}$and $\overline{P_{i5}P_{i6}}$ are obtained by traversing $face_i$, and one line segment $\overline{P_{j1}P_{j2}}$ is found for $face_j$ in Figure 7(b).

Step 2. Extract the common edge lists - This process excludes the parts outside of both faces, and finds the common segments shared by two faces. For example, line segments, $\overline{P_{j2}P_{i2}}$, $\overline{P_{i3}P_{i4}}$, and $\overline{P_{i5}P_{j1}}$ become the common segments of two faces, asshown in Figure 7(b). If one or more segments exist, a line containing the common segments as edges is created, and it is connected to both faces. Then, the common edge lists are also constructed with the oppositely directed edges on the intersection line.

Step 3. Update both faces - If there are common edge lists, the splitting process of the parent faces are performed for the resulting faces to be touched only at the common edges and vertices. If a common edge exists on the bounding edges of the face, a containing edge is divided into several smaller edges. For example, segment $\overline{P_{j2}P_{t2}}$ divides edge $\overline{V_{i3}V_{i4}}$ in $face_i$ into two edges $\overline{V_{i3}P_{j2}}$ and $\overline{P_{j2}V_{i4}}$. All another internal edges are intersected with the bounding edges and if any, with all the remaining segments in the face. Then, the face construction and/or subdivision process for the current line is performed by a depth-first search technique. If new faces are found, the corresponding faces are created. All the internal edges in faces are remained in each containing face. For example, two smaller faces, f_1 and f_2 are obtained by traversing with two directed edges, $\overline{P_{i3}P_{i4}}$ and $\overline{P_{i4}P_{i3}}$, when the segment $\overline{P_{i3}P_{i4}}$ is inserted into the $face_i$. However, all the common edges are contained in $face_j$ without face subdivision. The edge $\overline{P_{t5}P_{j1}}$ is also contained in $face_i$.

4. CONCLUSION

The Product Model of the STEP is expected to be one of the official means for representing and exchanging engineering information between computer applications in the future, and makes engineers' intentions more clearable and exchangeable than simple graphical representations like conventional drawings. This paper reviewed the data model of AP230 - Building Structural Frame : Steelwork, focusing on the representation of structural parts from the standpoint of structural engineers in practice.

This paper identified some special characteristics of building structures and a schematic-to-detailed design approach with problem decomposition which many architects and structural engineers in practice take in their design process of buildings, and presented a methodology for implementing the structural design process in details by adding the geometry-based abstraction to the structural entity-based data model of AP230.

REFERENCES

[1] Autodesk Inc., AutoCAD R14 Reference Manual, 1991
[2] http://www.comel.or.jp/~jcad
[3] http://www.gov.sg/corenet
[4] http://www.iso.ch/
[5] http://www.leeds.ac.uk/civil/research/cae/step/ap230/ap230.htm
[6] R. Sause, and G. H. Powell, A Design Process Model for Computer Integrated Structural Engineering, Eng. Comput.6, pp.129-143, 1990
[7] T. Y. Lin and S. D. Stotesbury, Structural Concepts and Systems for Architects and Engineers, John Wiley and Sons, 1981.
[8] ASCE, Monograph on Planning and Design of Tall Buildings, Vol. PC (1981), Vol. SC (1980), Vol. CL (1980), Vol. SB (1979), Vol. CB (1978), New York.
[9] D. Sriram, DESTINY : A Model for Integrated Structural Design, Artificial Intelligence, Vol.1, No.2, pp.109-116, 1986
[10] M. L. Maher and S. J. Fenves, HI-RISE : A Knowledge-Based Expert System for the Preliminary Structural Design of High Rise Building, 1985.
[11] G. H. Powell and H. An-Nashif, Automated Modeling for Structural Ananlysis. Eng. Comput.4, pp.173-183, 1988
[12] K. H. Law, T. Barsalou, and G. Wiederhold, Management of Complex Structural Engineering Objects in a Relational Framework. Eng. Comput. 6, pp.81-92, 1990
[13] Chang-Koon Choi and E-Doo Kim, "Top-down Geometric Modeling of Buildings on Network Database", Computer-Aided Design, Butterworth-Heinemann Ltd, UK, Vol.25, No.8, pp.468-478, Aug.1993

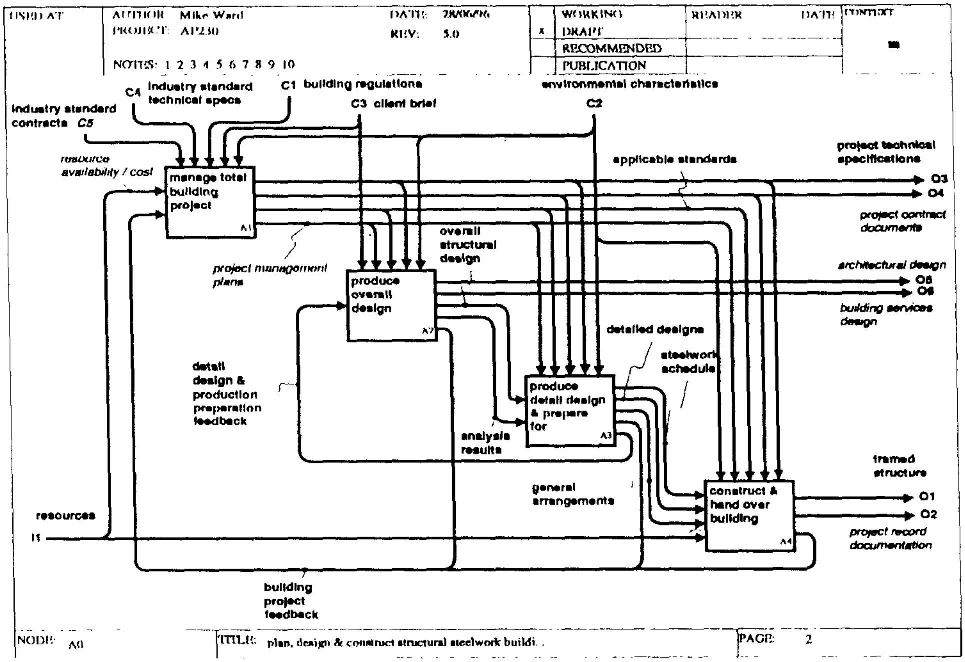


Figure 1. A0 : Plan, design & construct structural steel work building

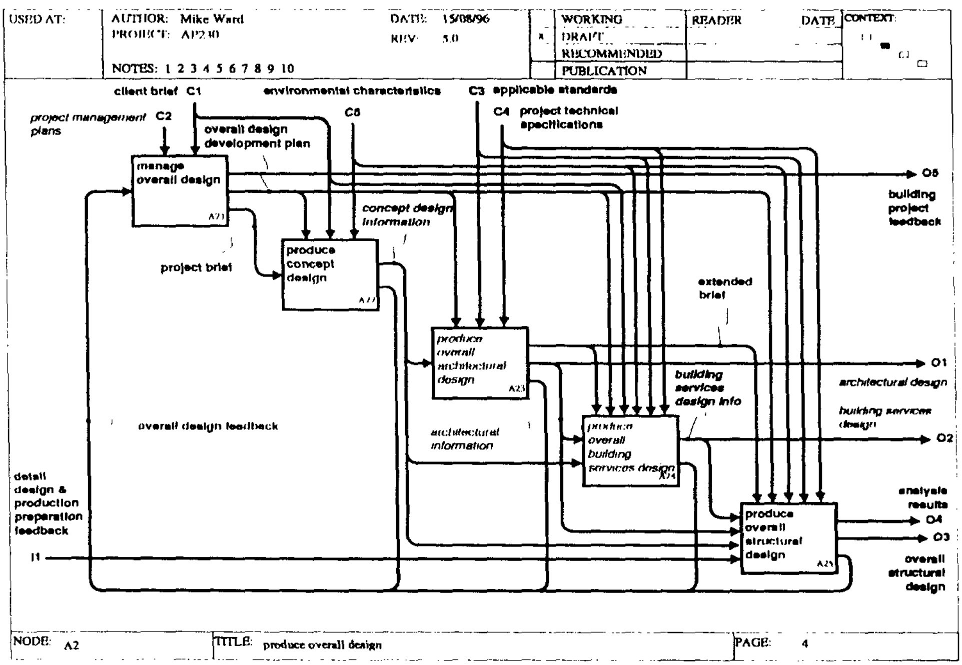


Figure 2. A2 : Produce overall design

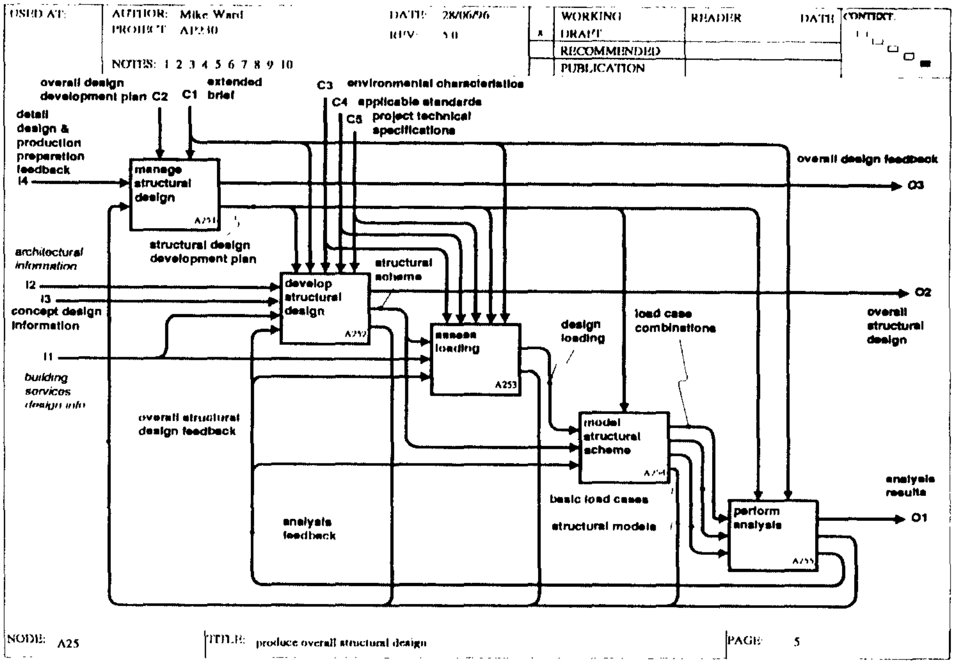


Figure 3. A25 : Produce overall structural design

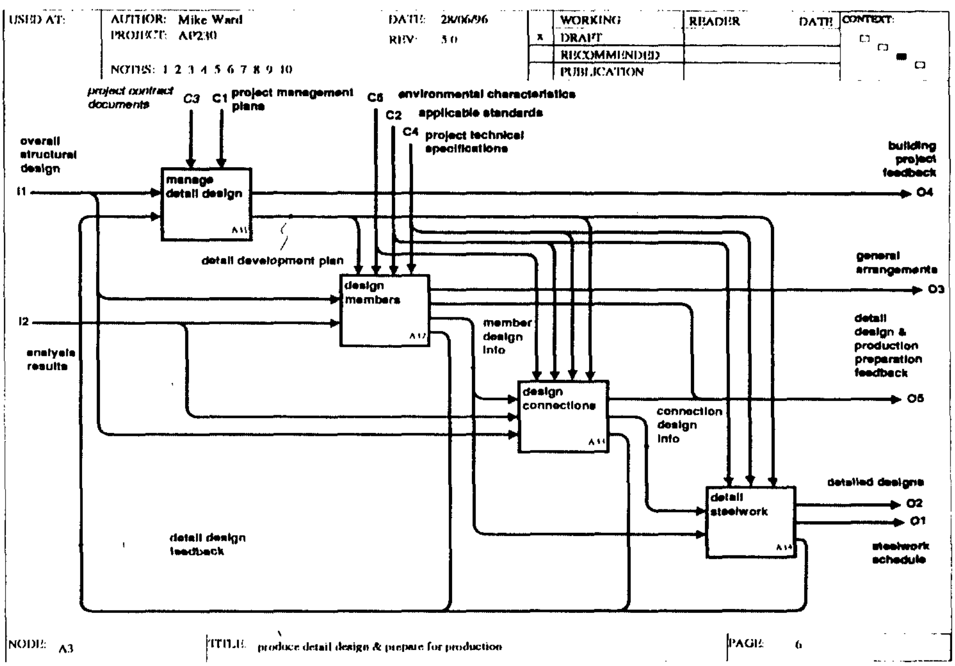


Figure 4. A3 : Produce detail design & prepare for production

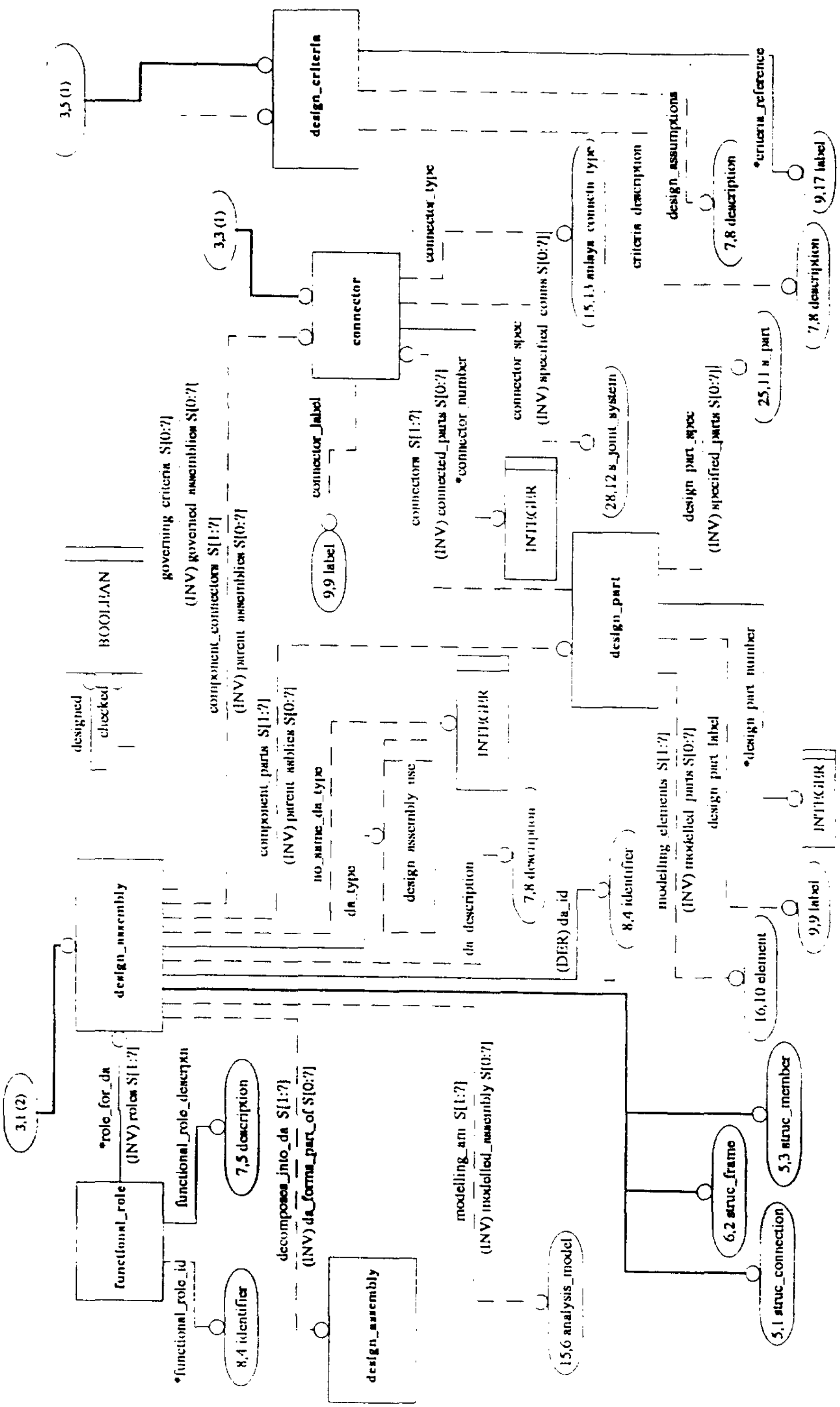


Figure 5. An example of ARM diagram 3 of 40 in EXPRESS-G

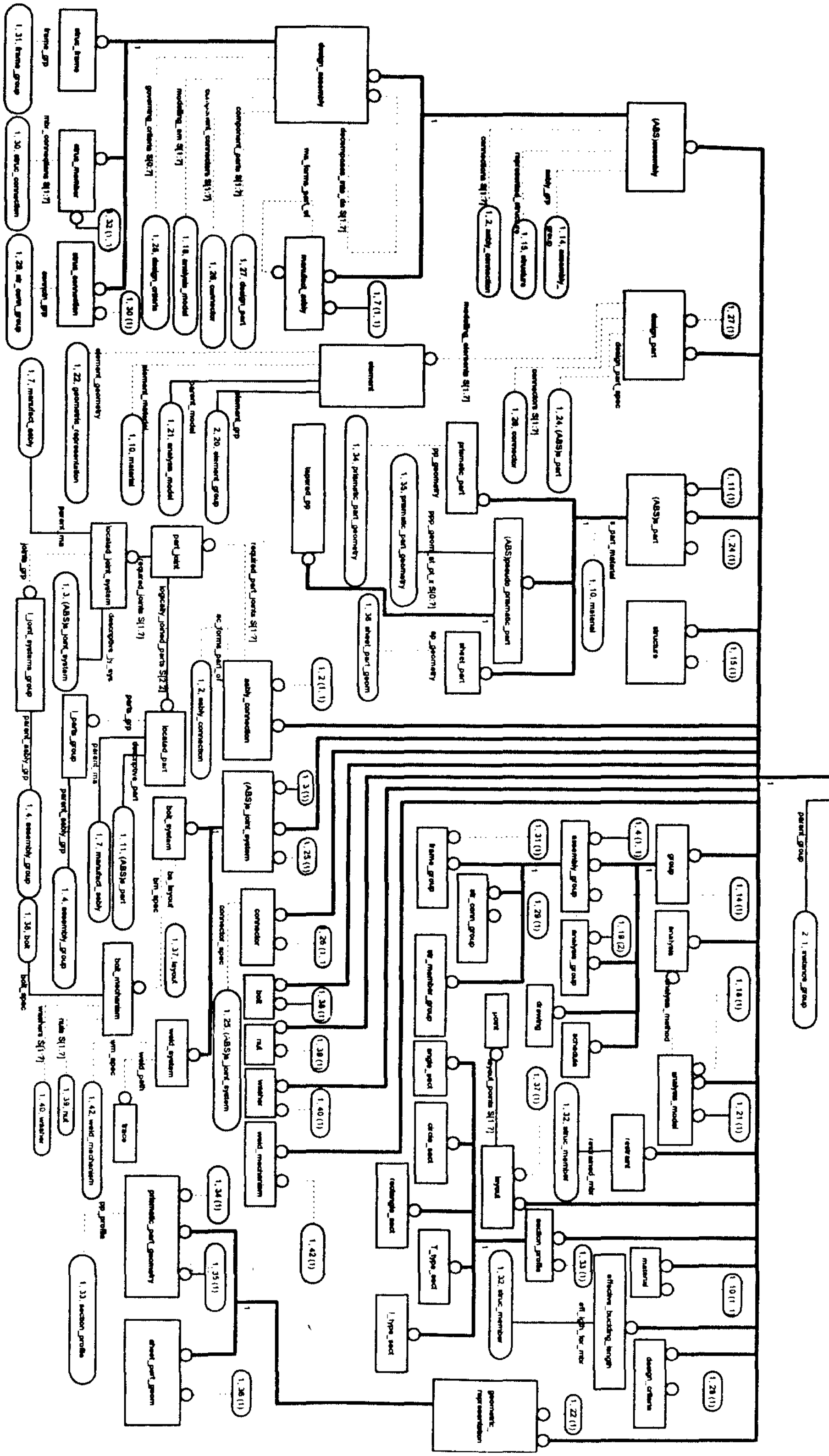
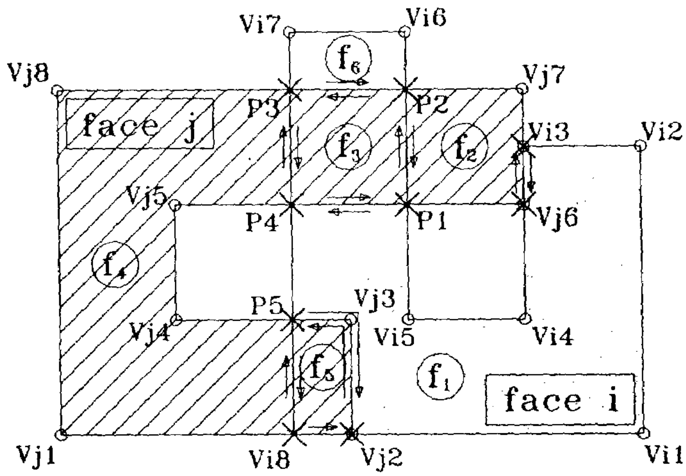
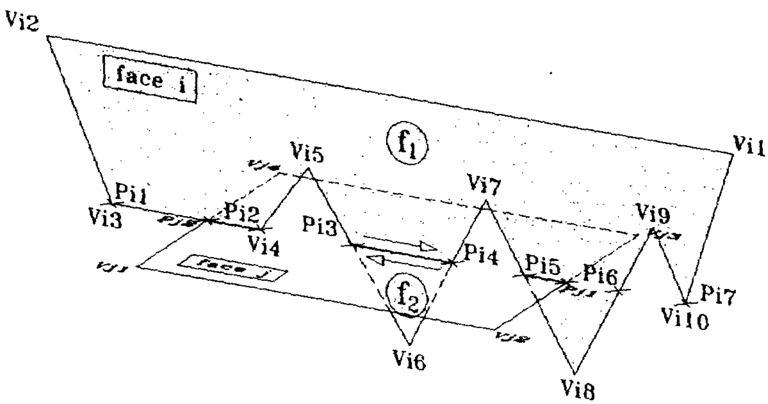


Figure 6. A brief ARM diagram of AP230



(a) Relationship on the same plane



(b) Relationship in space

Figure 7. Spatial Relationships of Design Entities