10-1-1992

Product and Process Models for Computer Integrated Preliminary Structural Design

Jerome A. Madden III

Richard Sause Jr.

Recommended Citation

Madden, Jerome A. III and Sause, Richard Jr., "Product and Process Models for Computer Integrated Preliminary Structural Design" (1992). ATLSS Reports. ATLSS report number 92-11:.
http://preserve.lehigh.edu/engr-civil-environmental-atlss-reports/179

ADVANCED TECHNOLOGY FOR
LARGE
STRUCTURAL SYSTEMS

Lehigh University

PRODUCT AND PROCESS MODELS FOR COMPUTER INTEGRATED PRELIMINARY STRUCTURAL DESIGN

by

Jerome A. Madden III
Research Assistant

Richard Sause, Jr.
Assistant Professor of Civil Engineering

ATLSS Report No. 92-11

October, 1992

ATLSS Engineering Research Center
Lehigh University
117 ATLSS Dr., Imbt Laboratories
Bethlehem, PA 18015-4729
(215) 758-3525

An NSF Sponsored Engineering Research Center

Contents

Abstract

The development of software systems for computer integrated design requires conceptual models that describe design activities and design information comprehensively. Conceptual models provide the foundation for new software systems that will benefit engineers by managing and organizing information, allowing the engineers to make more informed decisions than possible with current fragmented computer aided design tools. This report addresses the improvement and refinement of existing product and process models for design.

As an application, the preliminary design of frame structures is considered; preliminary design being the lay out of frame components, and the transition to the design of components. The problems encountered with current computer aided design tools are introduced. An overview is given to the conceptual models needed for computer integrated design systems. This overview is followed by a discussion of existing conceptual models that describe the design process and design product. Then, top-down design process and product models are applied to the preliminary design of two dimensional moment-resisting steel frames. The transition to component design is considered. Finally, the design of "typical" components is incorporated into the process and product models. The report concludes with recommendations for future research.

Chapter 1

Introduction

1.1 Background

Computer aided design tools have evolved from analysis tools to sophisticated design and drafting tools. However, most current software development, both within design firms and by software vendors, tends to focus on a narrow range of design (e.g., analysis, member proportioning, member detailing). Current software does not address the conceptual and preliminary stages of design. In these design stages structural systems, subsystems, and components are conceived, developed, and selected before detailed design is undertaken. Therefore, engineers using current computer aided design (CAD) tools are often required to perform design in a bottom-up fashion. These limitations restrict the engineers ability to manage design information as well as make informed decisions. Computer integrated design seeks to improve CAD through computer based, conceptual models of the design process and product. These improved CAD tools should allow engineers to perform design in a natural sequence and effectively manage design information, thus increasing productivity and the efficiency of designs. This research focuses on the development of design models, and in particular computer based design models for preliminary design.

1.2 Computer Integrated Design

For the purposes of this investigation integration refers to integrating the design stages (conceptual, preliminary, and detailed) and the associated design information into a cohesive process. This integration is limited to the perspective of the structural engineer, although related research efforts have focused on broader views of integration. For example, a proposed "Integrated Building Design Environment" integrates the agents of design, namely the architect, designer, fabricator, and erector (Sanvido et al, 1992). Likewise, Barone et al (1990) integrate the agents of construction, namely the designer, fabricator, and erector. Computer integrated structural design as used in this report applies to "software systems that support structural design" comprehensively (Sause et al, 1991).

1.3 Computer Aided Design Tools and the Need for Design Models

Integrated software systems are needed because current software is composed of non-cohesive CAD tools. These tools often force the engineer to design in a bottom-up fashion.

That is, many design tools do not support conceptual and preliminary design tasks (such as choosing between competing alternatives, laying-out subsystems). CAD tools often require the engineer to make premature detailed decisions. For example, the engineer may be required to place and size members at a time when very few details are available for making informed decisions. Furthermore, most CAD tools require the engineer to create a detailed geometric model and then add information about loads to that model. This procedure is often not in sequence with known information, since a structural engineer may prefer to lay out members with respect to loads, rather than loads with respect to members. Therefore, bottom-up design is contrary to the natural approach engineers would prefer to use (Zamanian et al, 1991).

Research in computer integrated design seeks to improve CAD tools through formal *design models* (CERF, 1991; Neville, 1989; Woodbury and Oppenheim, 1987). The development of these formal design models should seek comprehensive support for the design process. The models should lead to more usable engineering software that allows engineers to manage information and make more informed decisions. The use of CAD tools that are based on these models should increase productivity and result in more efficient designs. Formal, integrated design models that describe the organization, implementation, and limitations of software for structural design have not yet been developed. The research described here works toward the formal design models needed to improve current CAD software. The research focuses on the preliminary design activities for frame structures as part of an integrated approach to CAD. The research develops formal, integrated models for preliminary design that describe:

(1) the design activities structural engineers perform, and

(2) the information engineers need to know to support these activities.

1.4 Engineering Design Models

The development of design models will provide the theoretical foundation for computer integrated design. In general, a model may be thought of as “a set of general principles or categories which are used to organize, describe, and interpret an entity or process” (Martini 1990). When models focus on the design aspects of an engineering artifact (such as a frame structure) they are termed *engineering design models*. Engineering design models are used to bridge the gap between *computer implementation models* and the *real world* (Figure 1.1). Computer implementation models describe computer programming techniques for implementing engineering design models. The real world is considered as anything associated with a constructed facility. Since engineering design models are used to bridge this gap, they are usually considered computer based. However, engineering design models are developed using the terminology of an engineering domain.

The fundamental goal of engineering design models lies in the development of "general principles or categories". That is, engineering design models provide classes of engineering activities and information. Models of this type have been termed *generalization models* (Sause et al, 1991; Smith and Smith, 1977). A common example of a generalization model is the finite element analysis model. The categories of this model include nodes and elements. For a given problem, the finite element model could be used to develop any of several meshes used to predict the behavior of a structural system. Thus, the finite element model is available for analyzing structural systems in general.

Process and product models are models that formally describe the actions of engineers and the information that engineers generate. Although the terms *process* and *product* have been used traditionally in the manufacturing domain, the terms are useful in describing structural design. For example, consider an assembly line for manufacturing automobiles. At each "stop" on the assembly line some tasks are performed (e.g., welding, bolting, etc.) and this process involves adding some parts to the automobile (e.g., engine, wheels). The process model of the assembly line details the activities performed at each "stop". The product model defines the parts that are added to the automobile at each "stop".

A *design process* model for structural engineering captures the activities that the engineer performs. A *design product* model captures the information that describes a facility at the current stage of design. The models sought by this research are generalization models that provide categories for describing the design of a broad variety of structural systems. Therefore, a *design process generalization model* provides a formal, comprehensive, description of design activity categories. Likewise, a *design product generalization model* provides a formal description of design information categories that support the design process. These categories are general enough so that they can represent information generated by the design activities for a broad variety of structural systems.

When a particular entity (such as a steel braced frame for a particular project) is designed, a *design process instance model* and a *design product instance model* are used. The instance models describe the design activities associated with a particular structure (e.g., lay out bracing in Frame A) as well as the entities (e.g., W12x26 braces) that constitute the frame. In current practice, design drawings play the role of the design product instance model. This research focuses on generalization models rather than instance models. For the purposes of this report, a design process generalization model will be referred to as a process model, and a design product generalization model will be referred to as a product model.

1.5 Process Model

This investigation is not a study of the design process. However, a brief discussion of the design process provides a perspective on this research effort. A top-down structural design process model, known as the *Multilevel Selection-Development* (MSD) model (Sause and Powell, 1990) formally defines and organizes categories of design activities, phases, and tasks. These activities, phases, and tasks are discussed throughout the report, but are briefly introduced below.

The MSD model consists of two main activity categories: (1) selection activities and (2) development activities. A selection activity involves identifying, ranking, eliminating, and selecting from a number of competing design alternatives (Figure 1.2). A development activity involves evaluating a single design alternative (Figure 1.3). This discussion focuses on preliminary design, and only those tasks associated with preliminary selection and preliminary development (preliminary development I & II) are considered. The purpose of preliminary design is to "establish locations and dimensions of subsystems and evaluate subsystems. This evaluation may require design of subsystems and their components" (Sause and Powell, 1990). The multiple levels of selection and development activities for a subsystem are shown in Figure 1.4.

The research described in this report builds on the MSD model. The MSD model is applied to the preliminary design of frame subsystems, in particular to the preliminary design of two dimensional steel moment resisting frames. Furthermore, the research develops product model entities that support the MSD model activities and proposes revisions to the MSD model.

1.6 Objectives

The objectives of this research are:

(1) to propose a design product generalization model that supports the preliminary design process for frames;

(2) to elaborate and refine a MSD design process generalization model that supports the preliminary design process for frames; and

(3) to investigate the transition from frame design problems to component design problems.

1.7 Scope and Approach

This research investigates only a narrow portion of the design process. The research includes work on process and product models for preliminary design of frame structures, and for transition from a frame design problem to component design problems. Of primary importance are detailed descriptions of the product model entities used in the preliminary design of frames, and of the tasks for frame design that are broadly described by the MSD process model. Some of the results of this research are demonstrated in a prototype system that supports preliminary selection and development of frames (Madden et al, 1992; Bader, 1992).

The research began with the development of a formal product model that supported the already developed MSD process model. In particular, entity categories and relationships were developed and formalized. The development of suitable product model entities then prompted revisions to the MSD model. Finally, product model entities were studied that assist in the transition from frame design to the component design.

The remainder of the report is organized as follows: Chapter 2 provides background on related product and process models. Chapter 3 presents the original MSD process model and the revised MSD model for the preliminary design of frames. The product model entities that support these tasks are described in Chapter 4. Component activities and entities are proposed in Chapter 5. Finally, Chapter 6 summarizes the report and presents some concluding remarks.

Chapter 2

Product Model Foundations

2.1 Evolution of a Product Model

The design process is dynamic. Information about the design product is continuously refined, revised, and expanded in magnitude as well as detail (Martini et al, 1991) (Figure 2.1). The role of the product model is to describe this evolving information. The product model is composed of entities which are the various abstractions that describe a facility as it is being designed. Initially, the entities in the model are high-level abstractions that describe the facility without detail. Over the course of design, these entities are refined and new entities are added to the model. Eventually, the product model entities describe the parts of a constructed facility in detail.

A facility is a complex entity that is comprised of systems, subsystems, and components. A structural subsystem such as a moment resisting frame is referred to as a *complex alternative*. By definition, a complex alternative is an aggregation of parts. These parts may represent physical entities (e.g., floor systems, beams) or concepts (e.g., lobbies, stories). The first step in developing a product model is to develop categories of entities that describe these parts.

Product model entities from several different categories may describe real-world entities during design. For example, during preliminary design of frames, little detailed information is known about the components that constitute a frame. Thus, abstract entities are used to describe these components. Later, as more information is known about the components, more detailed entities are used. Consequently, several entity categories may be used to describe the different characteristics of the evolving real world components. Thus, the mapping between the product model and the real world is not usually one-to-one (Madden and Sause, 1992). This evolution of the product model as associated with the design process has been termed *scope* and *state modifications* (Martini et al, 1991).

Scope modifications in the product model support a shift in the design focus. For example, in the preliminary design of a frame, efficient layout and geometric stability of the frame are the focus of the design activities. Consequently, details about individual beams and columns are not considered (e.g., a wireframe type representation could be used). A scope modification of the design product model is usually accompanied by the introduction of new entities from more detailed categories. New entities are introduced because the design focus tends to become more detailed. For example, after preliminary design of the frame, entities from a new "detailed beam" category would be introduced to support the design of the beams. These entities would describe a beam's cross sectional area, shape, etc. During a particular design phase a "detailed beam" entity may undergo several revisions. These revisions reflect the iterative nature of design and are termed state modifications. State modifications are changes in an entity's attribute values.

2.2 Approaches for Developing Product Models

Design product models have been the subject of recent research. As noted by Nevill et al (1989), the "success of these models is highly dependent upon the quality of the abstractions supporting the problem solving". That is, the "abstractions" are the product model entities and the "problem solving" is the process model activities. Nevill et al (1989) have developed product and process models that are used for successive refinement of abstract assemblies. They note that scope modifications require entity categories that (a) describe the design problem and (b) provide transitioning to more detailed representations. Their approach is as follows:

(1) Create a suitable set of entity categories for representing each abstraction level.

(2) Create sets of entity categories for supporting the design problem.

(3) Create sets of entity categories for transitioning between abstraction levels.

The development of product and process models is itself a design problem. The approach used for the development of product model entity categories for preliminary design of frames is as follows:

(1) Identify types of two dimensional frames (e.g., moment resisting, braced).

(2) Identify common components of these different frames (e.g., beams, columns).

(3) Formally define entities for one type of frame (e.g., moment resisting frames).

(4) Test the proposed entities (for the type of frame) by identifying interactions and dependencies among entities (e.g., joints are formed by intersecting members).

(5) Simplify entity attributes by eliminating redundancies.

(6) Propose entity categories (e.g., beams, bays, stories, joints) which should be general for all types of frames.

Both the approach suggested by Nevill et al (1989) and the approach used in this research emphasize the importance of developing entities that support design at different levels of abstraction. The next section describes how abstraction levels for frames were identified, and this hierarchy is then compared with related work.

2.3 Decompositional Hierarchy

Several approaches have been developed for decomposing systems or assemblies. The most natural way for organizing and describing a structural system is to use entity categories that correspond to real world entities. These entity categories are organized at several abstraction levels. Although some researchers have decomposed structures according to function, the following approaches are physical decompositions. Zamanian et al (1991) decompose a building into a system of high to low level components. Sause (1989) noted that at least three abstraction levels exist, a system, subsystem, and component level. Product models have been developed for data exchange that produce similar decompositions. For example, Howard (1991) has identified system, subsystem, frame, member, element and part levels. Likewise, Bjork (1989) has divided a building into system, subsystem, part and detailed level. Regardless of how these approaches decompose structural systems, each must develop a set of entities at each abstraction level to support the design process at that level. The following section investigates two approaches for specifying entities that support these types of decompositional hierarchies.

2.4 Generic and Specific Entity Hierarchy

Two types of product model entities that are considered in this investigation are the so-called *generic* and *specific* entities as proposed by Sause (1991). These entities are associated with each of the main abstraction levels (i.e., system, subsystem, component). Generic and specific entities were identified as two types of entities needed to support the selection and development activities of the MSD model.

The primary goal of the selection activity is to formulate design requirements, identify alternatives, and rank alternatives. Problem formulation is a design task associated with each design phase of selection and is described in broad terms by Sause and Powell (1991) and in more detail for the design of frames by Madden et al (1992). The problem formulation task produces a generic entity which describes a design problem to be solved. For example, a generic entity for a frame defines the constraints (e.g., loads) and conditions (e.g., stability) that a proposed design alternative must satisfy. Since any proposed frame must meet these design requirements, this generic entity is a type of generalization. Recently, Madden and Sause (1992) describe the generic entity as a *problem entity*. This term will be used hereafter.

The primary goal of development is to propose a design alternative that satisfies the design requirements described by the problem formulation task (and represented by the generic entity). To describe the information associated with a particular design alternative an entity was created which includes the characteristics of the alternative that are important for its design. This entity, termed the specific entity, follows the generic entity in the product model hierarchy (Sause, 1991). The specific entity includes more information than the generic entity, namely:

(1) the design problem in a form that is applicable to the considered design alternative,

(2) the current representation of the design alternative, and

(3) objective indicators, which describe the merit of the design alternative (e.g., total weight of the frame could be a *measure of merit*).

Corresponding to this information are three entities, termed the *elaborated problem entity*, the *physical description entity*, and the *merit entity*. Together, these three entities will be termed a *solution entity* rather than a specific entity, as shown in Figure 2.2.

2.5 Feature Entities

Another product model entity hierarchy that has generated much interest is one composed of *feature entities*. Feature entities, or features, originated in the design and manufacturing of mechanical assemblies. Historically, features were used for identifying the parts of an already designed assembly. Once these parts were identified then decisions could be made concerning the manufacturability or manufacturing process of the assembly. Feature entities arose from a need to understand designer intent from the geometric data that is provided by computer aided drafting tools. This bottom-up technique is termed *feature extraction* (Unger and Ray, 1988).

Eventually, researchers observed that "the fundamentals of the intended manufacturing process assumptions or decisions should be known to the designer and captured by the system during design" (Dixon, 1988). Rather than using features to identify the parts of a completed geometric description, these researchers began using features as product model entities to support a top-down design process (Cutkosky et al 1988; Libardi et al 1988). Consequently, the uses of features as product model entities increased. Feature entities appear to be particularly useful in the early design stages. The need for entities that specifically support these stages of design has been noted. Several groups have noted that "few CAD systems support conceptual design of mechanical systems and assemblies (MSA). Accordingly, the designer should be able to focus on certain parts of the MSA while leaving others in an abstract or incomplete state" (Libardi et al 1988).

The first use of features as product model entities for structural design was presented by Zamanian et al (1991) who defined features as "high-level modelling primitives that facilitate the generation and modification of structural design." Zamanian et al (1991) also proposed a product model hierarchy that uses *high-level* and *low-level* feature entities. High-level feature entities appear to be abstractions of subsystems (e.g., frames) while low-level features represent components. Although specific attributes of these feature entities were not given, it was noted that feature entities must support the partial information associated with the early design stages. Consequently, "high-level features should effectively hide details about the components". For example, a high-level frame entity hides all details except the connectivity and geometry of its components. In the current investigation, feature entities are used to represent frames in the preliminary design stage. Feature entities are also used to transition from frame design to component level design problems.

Both Zamanian et al (1991) and Cunningham and Dixon (1988) have suggested that feature entities may be used to capture functional information in product model entities. Cunningham and Dixon (1988) note that in addition to reasoning about "geometry and topology, feature entities may capture designer functional intent". Similarly, Zamanian et al (1991) proposed that "the functionality of a structural feature is either specifically predetermined or entered as part of the designer intent". Thus, it is suggested that functionality attributes, like "support gravity load" or "resist transverse shear" define the functionality of product model entities.

The research presented herein proposes that feature entities may be used to identify and describe the constraints of a design problem but not to describe functional intent. For example, certain feature entities may define the constraints for a subsystem such as the loads imposed on a frame. Initially, these load constraints must be satisfied by the frame, and later similar constraints will be propagated to the components of the frame. This approach is markedly different from that proposed by Zamanian et al (1991) and Cunningham and Dixon (1988). Specifically, the product model entities do not explicitly describe functional intent but rather describe the constraints that are imposed on a frame and its components. This topic will also be discussed in relation to the process model in Chapter 3.

Zamanian et al (1991) and Cunningham and Dixon (1988) do not outline formal product and process models. However, their work suggests that features can describe several types of information during preliminary frame design. Based on their results, it appears that feature entities can play a role in a product model for frame design. In this role, feature entities are useful for:

(1) describing frame design requirements,

(2) describing a proposed frame design solution, and

(3) identifying frame components, and hence assisting in the transition from frame to component design problems.

Chapter 3

Preliminary Design Activities

3.1 Introduction

This section describes the preliminary design process model for a frame. The discussion begins with a brief example of a frame design problem, and then the goals of preliminary design are addressed. The MSD model tasks associated with preliminary design will be discussed and summarized (Sause and Powell, 1991). This section concludes with a comparison between the MSD tasks initially developed and the tasks that were developed in this research for the preliminary design of a frame.

3.2 Frame Layout Example

During preliminary design of a frame, the primary focus is on the lay out of components. The components of a frame include members such as beams and columns. Joints are regions between beams and columns. Joints are formed whenever the end of one component connects to another component. Collections of components may divide the frame into identifiable regions. Stories are regions between two vertically adjacent beams. Similarly, bays are regions between two horizontally adjacent columns. An example of a constructed facility is shown in Figure 3.1. This moment resisting frame is the prototypical example that will be used for further discussions. The components of a frame are usually layed out to support loads and are often restricted by geometric constraints. That is, loads require that a component is positioned to support the extent and direction of a load. Loads are examples of a larger constraint category termed *demand constraints* (Sause, 1989). Conversely, geometric constraints will control the positions of components in certain regions. Geometric constraints are examples of *interaction constraints* (Sause, 1989).

A structure is designed for certain loads such as vessels (V1 & V2), pipes (P1), and wind. Likewise, geometric constraints must be satisfied. Examples of load and geometric constraints are shown in Figure 3.2A. A clear space constraint is needed to keep the pump house clear of components between grid line pairs B,0 to D,1. The monorail requires a minimum distance between columns on grid lines A and B. This distance is specified by a relative position defined from grid line pairs A,1 to B,3. An offset constraint is used to specify the position of a component's surface. For example, B6's top surface is positioned 8" below grid line 4.

Components are layed out with respect to these constraints as shown in Figure 3.2B. A beam, such as B1, spans from grid lines (A,1) horizontally to (B,1). Joints are formed where B1 intersects columns C1 and C2 at grid lines (A,1) and (B,1). Similarly, a column such as C1 spans from grid lines (A,1) vertically to (A,3). The centerline of column C1 is located on reference line A. C1's joints occur at intersections with beams B1 and B3 at (A,1) and (A,3). Bays and stories are regions between components. A story, such as Story 2, has bounding beams B1 (on grid line 1) and B3 (on grid line 3). Likewise, Bay 2 has bounding columns C2 (on grid line B) and C3 (on grid line C).

3.3 MSD Tasks for Preliminary Design

As discussed earlier, every selection and development activity is associated with three design phases, namely conceptual, preliminary, and detailed (Figures 1.2 and 1.3). Within each of these design phases are a series of design tasks that describe the operations performed by the engineer. This section describes the preliminary design tasks associated with the selection and development of frames.

The purpose of preliminary design is to establish the best of the competing design alternatives. This purpose requires that each alternative is designed in enough detail so that it can be evaluated and compared with other alternatives. The following design tasks and subtasks were originally proposed as a model for the preliminary design process of any system, subsystem, or component (Sause and Powell, 1991). This discussion will eventually focus on subsystems, and in particular how these tasks apply to the selection and development of frames.

3.3.1 Selection Activity: Problem Formulation, Ranking, Elimination

The preliminary selection activity consists of three tasks, formulation, ranking, and elimination. Problem formulation involves assembling constraints in order to describe design requirements. This description of the problem is in general terms so that it is applicable to any of the considered design alternatives. Consequently, the result of this task is termed the *generic problem formulation* which is described by a so-called problem entity. Ranking involves comparing design alternatives against some established design objective (e.g., a minimum weight frame). Each developed design alternative has several measures of merit (e.g., weight) associated with it, so that the different alternatives may be ranked. Elimination involves removing the least efficient alternatives (based upon measures of merit) from the set of considered design alternatives.

3.3.2. Development Activity: Elaboration, Design Proposal, Review, Decomposition

The tasks associated with the preliminary development I of a design alternative are termed elaboration, design proposal, review, and decomposition (Figure 1.3). The elaboration task transforms the generic problem formulation so that the design requirements (constraints, conditions, objectives) reflect the considered design alternative. Elaboration consists of two subtasks, assembly and simplification.

Assembly involves assembling a *specific problem formulation*. That is, the design requirements from a generic problem formulation are re-specified in terms that are consistent with a design alternative. Simplification involves using constraint propagation to simplify constraint expressions (e.g., the variable, E, could be replaced with a known value, 29000 ksi, for steel).

The design proposal task proposes values for design variables. The design variables are the independent attributes of a design alternative that are considered during a particular design phase. Design proposal of an alternative may consider only a subset of all the design variables that have been defined for that design alternative. For example, during the preliminary design of a wide flange beam, the shape designation may be the only design variable considered. However, during detailed design, additional design variables would include the position, shape, and size of various bolt holes and copes. Design proposal is based upon a particular design strategy. A design strategy is a method or approach used for satisfying design constraints (e.g., loads) and achieving design objectives (e.g., minimum weight).

Design strategies are dependent upon how much feedback is given to the designer on the appropriateness of chosen design values. This factor is influenced by the amount of automation that is introduced into the strategy. That is, how much design "advice" or "guidance" is generated by the computer. Note that the objective of this work is to produce useful computer aided design tools not automated design tools. Several design strategies are listed below:

(1) *simple generation* is a method of proposing an alternative without consideration of the design requirements. No advice or guidance is available for the designer (or computer) to suggest the best values for the proposed alternative, except that physically impossible design variables are disallowed (e.g., specifying a negative beam distance or dimension).

(2) *redesign* is a method that first proposes an alternative, and then uses a critique of that alternative to refine the design. This method is part of an iterative cycle.

(3) *constraint satisfaction* involves proposing an alternative with respect to the design requirements from the specific problem formulation. If all constraints are considered and satisfied by proposed design values, then *direct design* is used. As noted, design proposal may not consider the complete set of design variables. When only certain design variables or only certain constraints are considered, *partial-direct design* is used. For example, when a beam is designed with respect to applied loads, the shape is usually selected based upon moment capacity. Then, other load conditions such as shear or axial force may be checked. Thus, when the shape is selected, the bending moment is assumed to control and other constraints are temporarily ignored.

(4) *optimization* attempts to find the optimal solution that both satisfies design constraints and achieves design objectives (e.g., minimum weight). Therefore, proposed design values will be reviewed against an optimal solution.

The review task evaluates how the proposed design satisfies the constraints, conditions, and objectives. Review is closely related to design proposal, since the two tasks are likely to be part of an iterative cycle. The results of the evaluation are contained in the measures of merit. That is, review produces indicators which are used for ranking how well the alternative meets the specified constraints, conditions, and objectives. Constraint satisfaction indicators, which are usually truth valued, indicate whether constraints and conditions are satisfied. The objective indicators provide the results of an evaluation of the design objective function. Review also determines when a design is sufficiently refined so that the design proposal-review iteration ceases.

The decomposition task identifies the components of a design alternative (e.g., a frame), and identifies a design problem for each of these components. The subtasks include physical decomposition, function assignment, and constraint posting. Physical decomposition identifies the parts of the frame (e.g., beams, columns, braces, joints). Function assignment requires "decomposing" the functions of the alternative and assigning these functions to the alternative's components. This assignment involves adding attributes like "bear gravity load" or "carry gravity load to the foundation" to the components. Sause (1989) has noted that "tools for representing structural functions do not currently exist". Constraint posting refers to placing the known component constraints in a location where they will be accessible to the design problems for these components.

3.4 Improved and Refined Design Tasks

This section describes the aspects of the MSD tasks that are relevant to preliminary design of frames and how the original MSD model has been improved or refined for this design problem. The main focus of preliminary design, as shown in the example at the beginning of this section, is to lay out a frame. This investigation focuses on the layout of a two dimensional steel moment resisting frames. The design variables and design strategy associated with this problem will also be discussed.

3.4.1 Selection Activity: Problem Formulation

Problem formulation (Section 3.3.1) expresses the problem to be solved in terms of constraints, conditions and the design objectives. For the design of frames, a geometric reference system was developed to describe the positions of the constraints in space. Problem formulation begins by laying out this reference system of grid lines. Next, loads and geometric constraints (e.g., offsets, clear spaces, and relative positions) are specified. The problem formulation is expressed in general terms such that the descriptions of the design requirements are applicable for any frame alternative (e.g., a reinforced concrete frame, or a steel braced frame). The preliminary design objective for frames is to lay out a cost-effective, minimum weight frame.

3.4.2 Development Activity: Design Proposal, Review

The preliminary development tasks (Section 3.3.2) that have been improved and refined for frames include design proposal and review. The elaboration task is not considered because the preliminary design of a frame does not require changes (elaboration) to the design requirements that the problem formulation task produces. For the design proposal task, a design strategy was developed and an *immediate review* subtask was added to the process model. As noted, the goal of preliminary development for frames is to establish a suitable frame geometry. This task is colloquially termed a *lay out* problem. Design proposal suggests values for design variables that establish the frame geometry. The frame geometry is established by laying out components such that all loads are supported and geometric constraints are satisfied. While developing a design strategy, the interactions between the computer implementation and the engineer are considered. The frame lay out strategy:

(1) determines that a component is needed to satisfy a constraint (e.g., to carry a load),

(2) determines the component type (beam or column), and

(3) positions the chosen component to satisfy the constraint.

That is, first determine that a component should exist, then determine its type (beam or column), and finally position the component. Therefore, a component's existence, type, and position are all design variables. During design proposal, an indefinite number of components and arrangements could satisfy the layout problem. Once a component type is chosen (e.g., beam or column) the component is then positioned on grid lines with respect to one of its surfaces (e.g., centerline, top, bottom). The design variables for the frame layout are described below:

Frame design variable:

set_of_components

Where "set_of_components" is a variable that has an indefinite number of values, each of which represents one component. Such a variable is termed *multi-entity valued*. This "set" consists of two subsets:

set_of_beams

set_of_columns

These sets hold values that represent beams or columns. Each individual beam or column has its own design variables as described below:

Beam design variable:

position

Column design variable:

position

The design variables for the frame layout include multi-entity valued attributes that are sets of components. The attributes of the components in these sets are design variables themselves. In summary, the frame layout design variables determine (1) whether a component is needed, (2) the component type, and (3) the component position. Other attributes, like connectivity, are also important for frame lay out. However, the connectivity of a frame's components can be determined automatically from their positions (Madden et al 1992), and thus connectivity is a dependent attribute.

The design strategy determines how values are proposed for the design variables. Values may be assigned by the computer or by an engineer. The goal of this research is to provide tools so that engineers can make informed decisions rather than have computers generate designs. Therefore, most of the decision making in the design strategy is performed by the engineer. For example, in the computer implementation described by Madden et al (1992), the engineer interactively lays out the components with respect to graphically displayed constraints. An alternative to this method would be for the computer to automatically lay out the components.

When the engineer is responsible for component layout, the design strategy from the point of view of the computer is simple generation. That is, the computer does not offer any advice or guidance as to the best positions of components except to prohibit physically impossible component instances. From the perspective of the engineer, the design strategy consists of (1) constraint satisfaction (partial-direct design) and (2) redesign. That is, the engineer lays out components to best satisfy load and geometric constraints. The constraints that can be directly determined are the only constraints that the engineer attempts to satisfy. If the lay out is not satisfactory then the design is revised or refined. The criteria for determining the suitability of a proposed layout is produced during review.

Although the design strategy does not offer formal advice or guidelines, a subtask termed immediate review was added. During immediate review, the computer performs a simple geometric analysis after each component is added to the model. This analysis identifies and disallows geometric impossibilities. For example, when a beam is added to the model, its position is first checked with respect to the geometric bounds established by the design requirements. If one of the beam's ends is outside these geometric bounds then the component is not added to the model. Similarly, if a new column overlaps an existing beam or column then the new column is not added to the model (Figure 3.3).

The review task evaluates the proposed frame, by determining whether constraints are satisfied, as well as determining how well the proposed frame meets the design objectives. The review task is divided into two subtasks, namely (1) constraint evaluation and (2) objective evaluation.

Constraint evaluation determines whether the component arrangement satisfies the constraints. This evaluation determines if:

(1) loads are supported appropriately,

(2) geometric constraints are satisfied, and

(3) the frame is geometrically stable.

If a constraint can be evaluated logically, then a constraint objective indicator returns a value of true or false. This type of evaluation is termed *simple evaluation.* An evaluation as to how "close" the constraint is to being satisfied would be termed *numeric evaluation.*

If the constraints are not satisfied then the process returns to design proposal and the current design variables are refined. Conversely, if the constraints are adequately satisfied then objective evaluation is performed. This iteration between design proposal and review is shown in Figure 3.4.

The objective evaluation determines how well the design objective has been met by the proposed frame. Currently, the design objective is to minimize the total weight of the frame, although other objectives should be considered. An evaluation of this design objective produces measures of merit whose values are used to indicate how well the objective is met. The measures of merit include the individual weights of the steel beams, columns, and connections. These weights are calculated and then summed for evaluating the minimum weight of the frame.

The weight of each component may be estimated by using heuristics. These heuristics require that the component loads are identified. For example, a beam's load constraints are those loads that the beam directly supports. For columns, a "tributary area" type approximation can be used to estimate the load that a column carries.

An example of these heuristics is demonstrated Figure 3.5 for estimating the weight of a uniformly loaded beam. The expression used for calculating the weight of the section is:

$W_s = 5*M/d$

where:

M is the midspan moment (kip-ft) (1 kip-ft = 1.356 kN-m)

W_s is the weight of the section (lb/ft of length) (1 lb = 4.448 N, 1 ft = 0.3048 m)

d is the depth of the beam (inches)

This heuristic assumes:

(1) a wide flanged section,

(2) a span to depth ratio of 2 (ft/in), and

(3) an allowable bending stress of 24 ksi (1 ksi = 6895 kN/m^2)

The example estimates the weight of the beam. Connection weight can also be estimated by assuming that connections weigh a certain percentage of the total steel weight. For example, common estimates are that connections account for 3% of total steel weight for petrochemical facilities, or 5% of total steel weight for fossil fuel power plants.

If the objective indicator is not satisfactory then the proposed design variables may be refined in design proposal (Figure 3.4). Otherwise, preliminary development I (Figure 1.3) is concluded and the objective indicator will be used during the selection activity to rank competing design alternatives.

One important difference between the MSD process model proposed by Sause (1989) and the process described above is that the decomposition task (Figure 1.3) and its subtasks (physical decomposition, function assignment, and constraint posting) have been removed from the preliminary development I phase. The physical decomposition subtask became unnecessary because its purpose "identifying the physical parts of a complex alternative" was already accomplished in design proposal. That is, the design variables used in design proposal identify the components of the frame. The function assignment subtask was established to identify the function of a component (e.g., carry load to foundation) and introduce a design subproblem for each component. It was mentioned earlier that many researchers believe that it is necessary to explicitly assign function to components. However, function assignment was removed from the model because:

(1) in frame design it does not appear necessary to explicitly define the function of each component, and

(2) component design problems do not have to be introduced during this phase.

The first statement assumes that each proposed component satisfies the load and geometric constraints. That is, regardless of whether the function of each component is explicitly stated (e.g., "support equipment", or "transmit load to foundation") the components must satisfy the constraints. Since the review task will not allow a load to be unsupported, the components must be positioned appropriately to serve the function of "supporting". Review also ensures that this load will be transmitted to the foundation, because a geometrically stable frame is required. Therefore, a load is supported if the position and extent of the component supports the position, direction, and extent of the load. For example, if the beam shown in Figure 3.5 is positioned to support a distributed load then the constraint is satisfied. This information does not have to be explicitly expressed as an attribute of the beam. Rather, the position of the beam simply satisfies a constraint on the frame. For the example in Figure 3.6, explicit functional requirements could be a step in defining a load carrying system. The engineer lays out a system that (1) carries vertical distributed loads and (2) transmits lateral forces. These systems could be considered as independent load carrying systems for the purposes of component layout. However, the bracing components may significantly contribute to the load carrying capacity of the beam. Therefore, although load carrying systems may be conceptually independent, their interactions make defining functional behavior difficult.

In summary, a component must support the loads applied along its length, satisfy geometric constraints, and transmit any applied end moments and forces. The design of components is performed in the preliminary development II (Figure 1.3) phase when the constraints at the frame level are propagated to the component level. This phase introduces a design problem for the components. Several of these issues are addressed in Chapter 5.

Chapter 4

Product Model Entities

4.1 Introduction

The previous chapter described the activities and tasks of the preliminary design process for a frame. This chapter focuses on the product model entities that represent the information created and used in this process. This information includes the frame design requirements and the design variables of the frame (e.g., the type and position of the components). This chapter focuses on feature entities that are used to organize the design requirements and physical description information (Figure 4.1). A description of entities that represent the elaborated frame design problem and the frame measures of merit will not be given. For the purposes of this discussion, feature entities are defined as "any entities that are necessary to arrange the parts of a frame" (Madden and Sause 1992). This chapter will discuss feature entities and how they are used in a hierarchy of problem and solution entities for frame design. Chapter 5 will discuss how feature entities are used for transitioning to the component level design subproblems.

4.2 Feature Entities of the Frame Problem

For preliminary design, both the design requirements and the physical description of a proposed solution may be thought of either as single entities or as aggregations of parts. As mentioned in the discussion of the design process, the problem formulation task assembles a problem entity that describes the design requirements of the problem to be solved. The design requirements for a frame were described in an earlier example and are shown in Figure 3.1. The grid lines, loads, offsets, clear spaces, and relative positions used in formulating the frame design requirements are the attributes of the frame problem entity (Figure 4.2). The grid lines, loads, etc. are each multi-entity valued attributes. An individual grid line, load, etc. is described by a *feature entity of the frame problem*. Five feature entity categories are used to (1) create a geometric reference system, (2) describe position, extent, and magnitude of loads, and (3) describe position and extent of geometric constraints. These entity categories (grid line, load, clear space, offset, relative position) are shown in Figures 4.3 through 4.7. The entity categories have also been described in the context of a computer implementation by Madden et al (1992).

4.3 Feature Entities of the Physical Description

As mentioned in the discussion of the design process, design proposal arranges the components of the frame. This arrangement forms a geometrically stable and efficient frame that satisfies the design requirements of a frame design problem. The parts of a frame were described earlier and are shown in (Figure 3.1). The beams, columns, joints, stories, and bays form the attributes of the frame physical description entity (Figure 4.8). The beams, columns, etc., are each multi-entity valued attributes. An individual beam, column, etc. is described by a *feature entity of the physical description*. During design proposal, the independent attributes of these feature entities establish the positions of the beams and columns. The connectivity of the components is also described so that the geometric stability of the frame may be evaluated during review.

In order to support the design process, the feature entities of the physical description have attributes which establish the geometry and connectivity of the frame. Five feature entity categories were identified in a product model that supports the preliminary design activities for frames. These entity categories (beam, column, bay, story, joint) are shown in Figures 4.9 through 4.13.

4.4 Feature Entities in the Problem and Solution Entity Hierarchy

The feature entities presented earlier (i.e., the feature entities of the problem and the feature entities of the physical description) are associated with the problem and solution entities, respectively. As discussed in Chapters 2 and 3, problem entities describe the design requirements of a frame. Similarly, solution entities describe: (1) the design requirements for a particular type of frame, (2) a physical description of the proposed frame, and (3) measures of merit for the proposed frame. The relationship between feature entities and the problem and solution entities is shown in Figure 4.14. Feature entities of the problem are aggregated to form a *frame problem entity*. Likewise, feature entities of the physical description are aggregated to form part of a *frame solution entity*. The relationships between the problem and solution entities and the selection and development activities are shown in Figure 4.15.

In summary, two types of feature entities were identified as useful in the preliminary design activities for frames, namely (1) feature entities of the problem, and (2) feature entities of the physical description. Note that the purpose of these feature entities is to describe the frame design requirements or the frame itself rather than the individual beams, columns, and joints. These entities do not completely describe the real-world components they represent (e.g., the beams or columns in the constructed facility). Rather, they describe the real-world components in sufficient detail for the preliminary design of frames. Consequently, when component design is considered in Chapter 5, some of the feature entities of the frame physical description will undergo state and scope modifications. Then the real-world components of the frame will no longer be described as frame feature entities but rather as *stand-alone* entities for the component design problems.

Chapter 5

Preliminary Development II & Component Design

5.1 Introduction

The previous chapters discussed preliminary development I for frames. The focus of preliminary development I is at the frame design level since the goal of this phase is a suitable arrangement of components. Preliminary development II (Figure 1.3) focuses on component design rather than frame design. The first part of this chapter will discuss the preliminary development II process model tasks as applied to frames. This discussion will include the tasks associated with the design of frame components. Then, the product model entities needed to support the preliminary development II tasks will be discussed. Finally, the aspects of product and process models needed to group components into "typical components" are addressed.

5.2 Preliminary Development II Tasks

The preliminary development II tasks of the MSD model consist of elaboration, analysis, component design, and aggregation (Figure 1.3). Elaboration serves the same purpose as in preliminary development I. The analysis task determines the response of a frame to its simulated environment (Sause, 1989). This response results in load and geometric constraints that are assigned to the frame's components. Component design involves designing the components in enough detail so that the frame may be evaluated. For example, the type and dimensions of each beam and column should be established in order to estimate the cost or weight of each component. Aggregation involves collecting and combining the results of the component design problems. For example, the components' measures of merit (such as cost or weight) may be combined to provide measures of merit for the frame. The following paragraphs describe the preliminary development II in more detail.

The elaboration task is not considered because it is assumed that the generic problem formulation from preliminary development I has not changed. Therefore, as in preliminary development I, the elaboration task is not required for the development of two dimensional steel moment resisting frames. However, a new *component introduction* task is needed because the decomposition task was removed from the preliminary development I phase

(Section 3.4.2). One of the subtasks of the decomposition task was physical decomposition, which identified component design problems. In the improved MSD model, the component introduction task helps introduce design problems for the components.

Analysis determines the response of the frame to its environment. This task has two primary goals:

(1) determine the load constraints and other restrictions which are imposed on the components, and

(2) formulate geometric constraints for the components.

Corresponding to these goals, two types of analyses are considered, namely structural analysis, and geometric analysis. Each type of analysis consists of two subtasks, modelling and analyzing. Modelling extracts values from product model entities (e.g., member properties from a beam entity) as input for the analysis subtask (e.g., using the stiffness method). As mentioned above, analysis produces constraints for the frame components. For example, load constraints for components include end moments, and shear and axial end forces. Geometric constraints for components include width or depth restrictions on member dimensions.

Structural analysis is often performed using the stiffness method. The stiffness method uses engineering principles (such as equilibrium and compatibility) to determine constraints on the components from load constraints on the frame. The stiffness method requires that certain properties of the components (e.g., the cross sectional area and moment of inertia) are known before the analysis is performed. Preliminary development I does not determine values for these component properties. Consequently, an initial estimate of the component properties is needed. Two approaches are suggested as a precursor to structural analysis:

(1) Perform an approximate analysis (e.g., using standard moment coefficients, or the portal method) to approximate the internal forces in the components. Then, the results from the component design task will provide a shape designation from which properties can be derived.

(2) Suggest an initial shape designation for each component (assuming each component in a steel frame will have a standard rolled or prefabricated shape). This initial shape desig-

nation will allow values of cross sectional area and moment of inertia to be derived during the modelling subtask.

The first method does not require an initial shape designation since the component forces are directly determined. The second method is commonly used in CAD practice, where the engineer uses experience to suggest an initial shape designation.

The second type of analysis, geometric analysis, determines geometric constraints for the components. Geometric analysis requires (1) identifying frame constraints that potentially affect component dimensions (e.g., relative positions and clear spaces) and (2) formulating geometric constraints for the corresponding components. As an example, consider several relative position constraints that span across several stories (Figure 5.1). From this example, given the distance from grid line 1 to grid line 3, two related relative position constraints, and two offset constraints, a depth constraint for beam B2 can be formulated. It is noted that geometric analysis, especially where constraint propagation is concerned, will require further research.

The component design task involves the selection and development of simple alternatives (the frame components). This task involves formulating and solving design problems at the component level. That is, the constraints on the components that are determined during the analysis task are assembled as the design requirements for the components. The purpose of component design is to select a shape designation (e.g., W12x26) for each component that will satisfy the design requirements. The design variable for component design is as follows:

Component design variable:

shape_designation

Where shape_designation is a variable that represents the suggested shape designation for a component. This design variable is in addition to the component's position variable established in preliminary development I. A detailed discussion of the component design subtasks is not within the scope of this report. However, component design follows the MSD process model tasks for a simple alternative (Figure 5.2).

The selection and development of the components requires proceeding through the activities and phases of the MSD process model. The MSD model suggests that conceptual, preponent designs may be used for members in the same locations (e.g., floor beams) or members in the same load carrying system (e.g., braces). Consideration of the typical component design problem requires changes to the process and product models. These changes include adding a task to the preliminary development II phase of the process model and adding an entity category to the product model.

5.4.1 Revised Design Tasks

CAD systems usually have tools for designing typical components. However, CAD systems often perform typical component design from the bottom-up. For example, the designer may separately select shapes for each component. Then, several components are grouped and then the "strongest" or "heaviest" shape is chosen as the typical component solution. This shape is then tested for compliance with design standards for each component in the group. Conversely, the MSD model follows a top-down approach, and an improved process model for the design of typical components includes the following tasks for preliminary development II (Figure 5.7):

(1) elaboration,

(2) component introduction, in which the feature entities of the frame physical description are used to identify components that need to be designed,

(3) analysis, in which constraints and conditions are found for the components,

(4) component grouping, in which the designer identifies typical components,

(5) component design, in which design tasks that select and develop alternatives for typical components are carried out (problem formulation, design proposal, etc.,), and

(6) aggregation, in which measures of merit are assembled and returned to the selection activity.

Therefore, the typical component design problem requires introducing a new *component grouping* task before the component design task. This task is placed before component design so that the results of analysis are available for making decisions about grouping. The purpose of component grouping is to identify those components that will have the same design.

Component design involves the selection and development of simple alternatives (components). The problem formulation task (from the selection activity) assembles the load and geometric constraints from the individual components and formulates a "typical" component design problem. Two approaches may be taken to formulating this typical component design problem. One approach assembles the constraints for the individual components and then formulates a controlling or "worst case". This approach is useful for geometric constraints, which might limit the dimensions of a component. For example, in Figure 5.8 two beams that might be grouped into one typical beam have different depth constraints. These depth constraints can be combined to formulate a minimum and maximum allowable depth for the beam. Another approach assembles the constraints for the individual components into a single set and tests a proposed design against this constraint set. This method is more useful for load constraints. For the example in Figure 5.8, a proposed typical beam alternative would be evaluated with respect to the dead and live load combinations for both beams B1 and B2 (Hemingway, 1991). Therefore, separate approaches are used for formulating the load and geometric constraints of the typical component design problem. Note that the formulated design problem for typical components may not be solvable (e.g., the geometric constraints could be contradictory) and further research is needed in this area.

5.4.2 Product Model Entities

The attributes of the typical component problem entity (e.g., a beam problem entity or a column problem entity) are similar to those for the frame problem entity. That is, the typical component problem entity may be considered as a single entity or as an aggregation of parts. The information included in the problem entity is derived from the design requirements entities associated with each of the grouped components. That is, the loads and geometric constraints for each component are assembled, formulated and held in the typical component problem entity (Figure 5.9).

The attributes of the beam problem entity are shown in Figures 5.10. The beam problem entity has two attributes, loads and depth_constraint. The loads attribute is a multi-entity valued attribute. This attribute describes the different loads, load cases, and load combinations for each beam that is grouped into the typical beam design problem. An individual load is termed a *beam load feature entity.* The depth_constraint attribute is a geometric constraint on the typical beam depth (e.g., depth <= 18"). This constraint is derived from the beams that are grouped into the beam problem entity. An individual depth_constraint is termed a *beam depth feature entity.* The beam load and beam depth feature entities represent the individual loads and geometric constraints associated with the beam design requirements entity or the beam problem entity. These constraints have been termed feature entities because they now describe the parts of the design problem to be solved.

Similarly, the attributes of the column problem entity are shown in Figure 5.11. The column problem entity has two attributes, loads and width_constraint. The loads attribute is a multi-entity valued attribute. This attribute describes the different loads, load cases, and load combinations for each column that is grouped into the typical column design problem. An individual load is termed a *column load feature entity.* The width_constraint attribute is a geometric constraint on the typical column width (e.g., width <= 14"). This constraint is derived from the columns that are grouped into the column problem entity. An individual width_constraint is termed a *column width feature entity.* The column load and column width feature entities represent the individual loads and geometric constraints associated with the design requirements entity or with the column problem entity.

As mentioned, a detailed description of the component solution entity (the elaborated component problem entity, the component physical description entity, and the component merit entity) is out of the scope of this research. However, the relation between the component problem entity and the solution entity is shown for a beam in Figure 5.12.

Chapter 6
Summary and Conclusions

6.1 Introduction

This report has presented conceptual models for computer integrated design. A description of the deficiencies of current software systems attempted to demonstrate the need for formal models of the design process and the design product. The anticipated application of these design models is in the implementation of computer integrated design software. The research focused on refining a narrow portion of the overall design process (preliminary design) and design product (two dimensional steel moment resisting frames). However, the role of engineering design models has been described in more general terms. The purpose of this final chapter is to summarize the contents of the report and discuss the uses of these results in further research.

6.2 Summary

This investigation has focused on the development of engineering design models that are appropriate for implementation in a computer integrated design software system. The approach to developing these models was as follows:

(1) Select a portion of the design process to study (preliminary design of frames), and select a design alternative (two dimensional steel moment resisting frames) as an example of a design product.

(2) Identify the physical parts of a design product instance model. Use these parts to propose entity categories for a design product generalization model, based on the study of a portion of the design process.

(3) Revise an existing Multilevel Selection-Development (MSD) design process generalization model.

Existing design models were reviewed in Chapter 2. Chapter 3 first described the MSD process model. This process model then was refined and improved for the preliminary design of two dimensional steel moment resisting frames.

In Chapter 4 the product model entities for frames that supported the MSD process model were presented. In particular, entities were introduced to represent (1) the information which comprises a description of a frame design problem, and (2) the information which comprises a description of a frame design solution.

Finally, design activities and entities were considered for the design of components in Chapter 5. The notion of "typical" design was introduced into the design process and design product models. In particular, tasks were added for the design of "typical" components. Likewise, product model entities were introduced (1) to transition from frame design problems to component design problems, and (2) to represent the information which comprises a description of a typical component design problem.

6.3 Recommendations for Further Research

This investigation has focused on a narrow portion of the design process and product. Specifically, the preliminary design of steel moment resisting frames was the area for which these models were developed. However, since these models were developed as generalization models, the design activities and entities they describe should be applicable to the design of systems or other types of frames. Further research could (1) expand the process (consider conceptual or detailed design phases for two dimensional steel moment resisting frames) or (b) expand the design product (develop product model entities for several different design alternatives, e.g., reinforced concrete frames, steel braced frames, etc.). In either case, the notion of "typical" designs should continue to play an important role for those systems or subsystems that are considered.

Some of the results described in this report have been demonstrated in a prototype computer implementation (Madden et al 1992). The study of engineering design models specifically for computer implementation was essential for producing detailed, explicit and formal models. Consequently, it is suggested that further development of these product and process models should be closely tied to the development of prototype implementations.

6.4 Concluding Remarks

It appears that the engineering design models developed in this research are useful steps toward computer integrated design of structures. The objectives of the research were met since the problem, feature, and solution entities satisfactorily contributed to a design product generalization model. Likewise, the MSD model was refined and demonstrated as a sound design process generalization model. The transition from frame design problems to component design problems was investigated. This investigation revised design process tasks and proposed new product model entities. In conclusion, it is hoped that the developers of computer aided design tools will be able to realize the potential benefits of using such models as the foundation for their future systems.

FIGURES

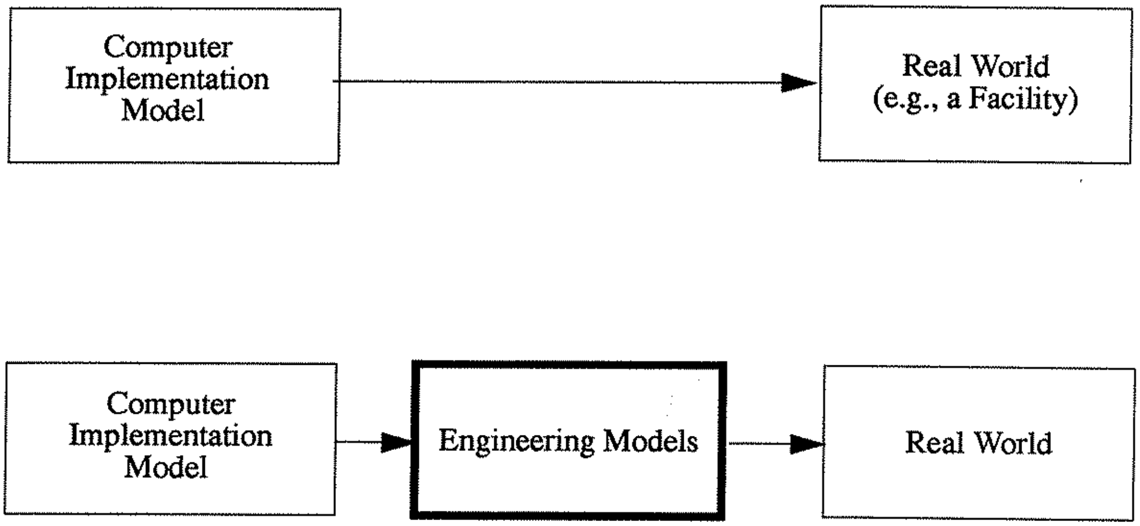


FIGURE 1.1 MAPPING BETWEEN REAL WORLD AND COMPUTER MODELS

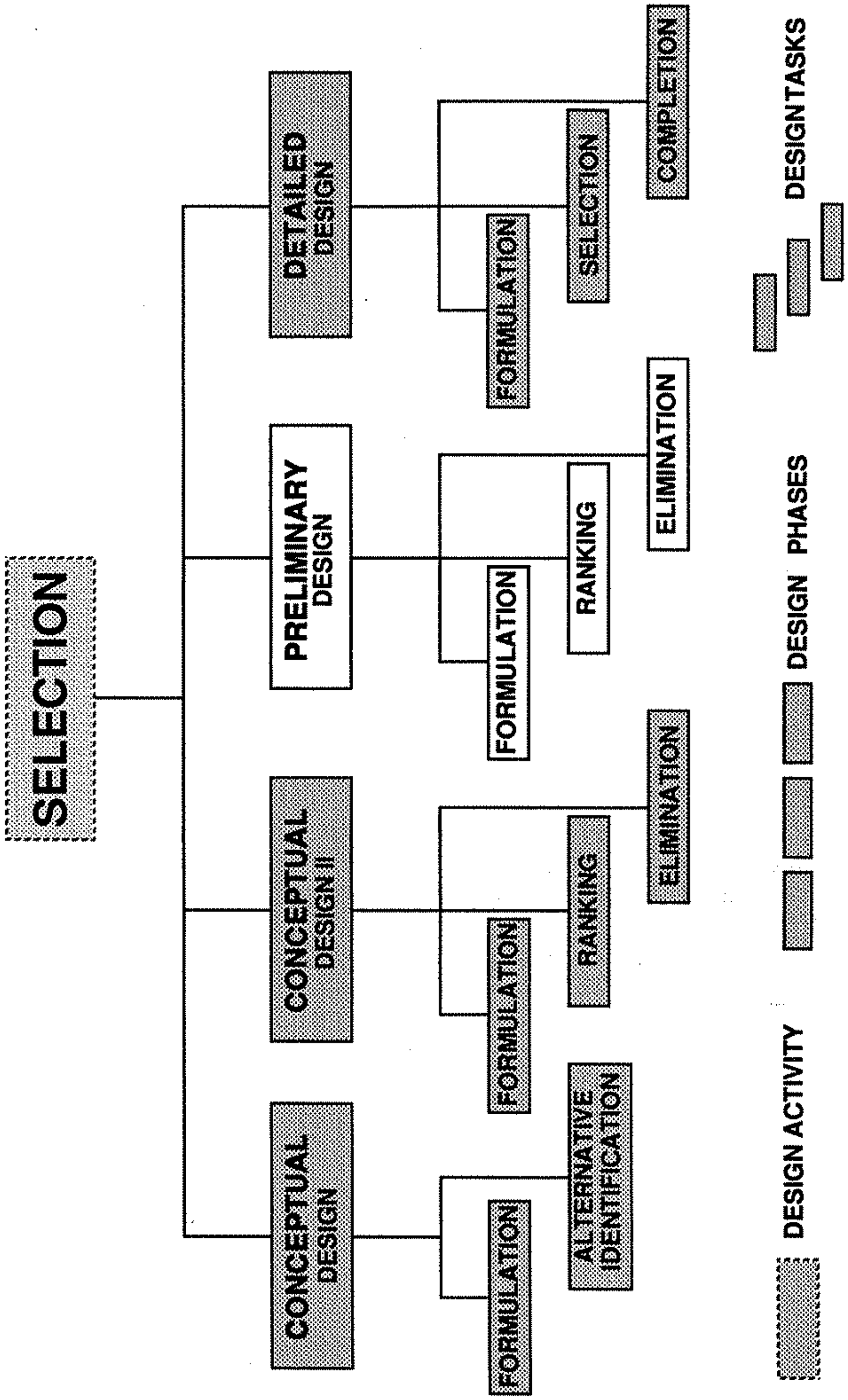


FIGURE 1.2 SELECTION ACTIVITY

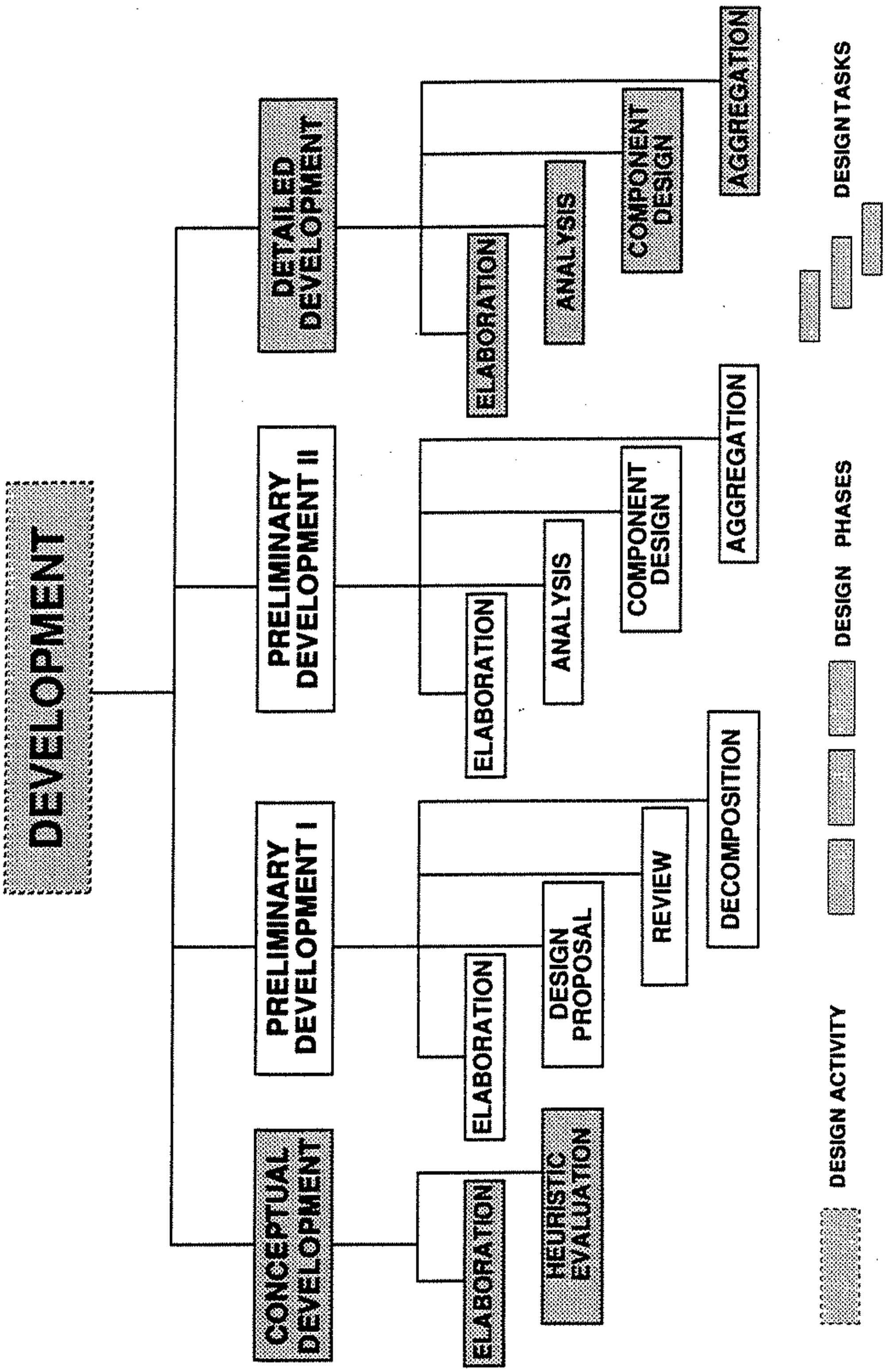


FIGURE 1.3 DEVELOPMENT ACTIVITY

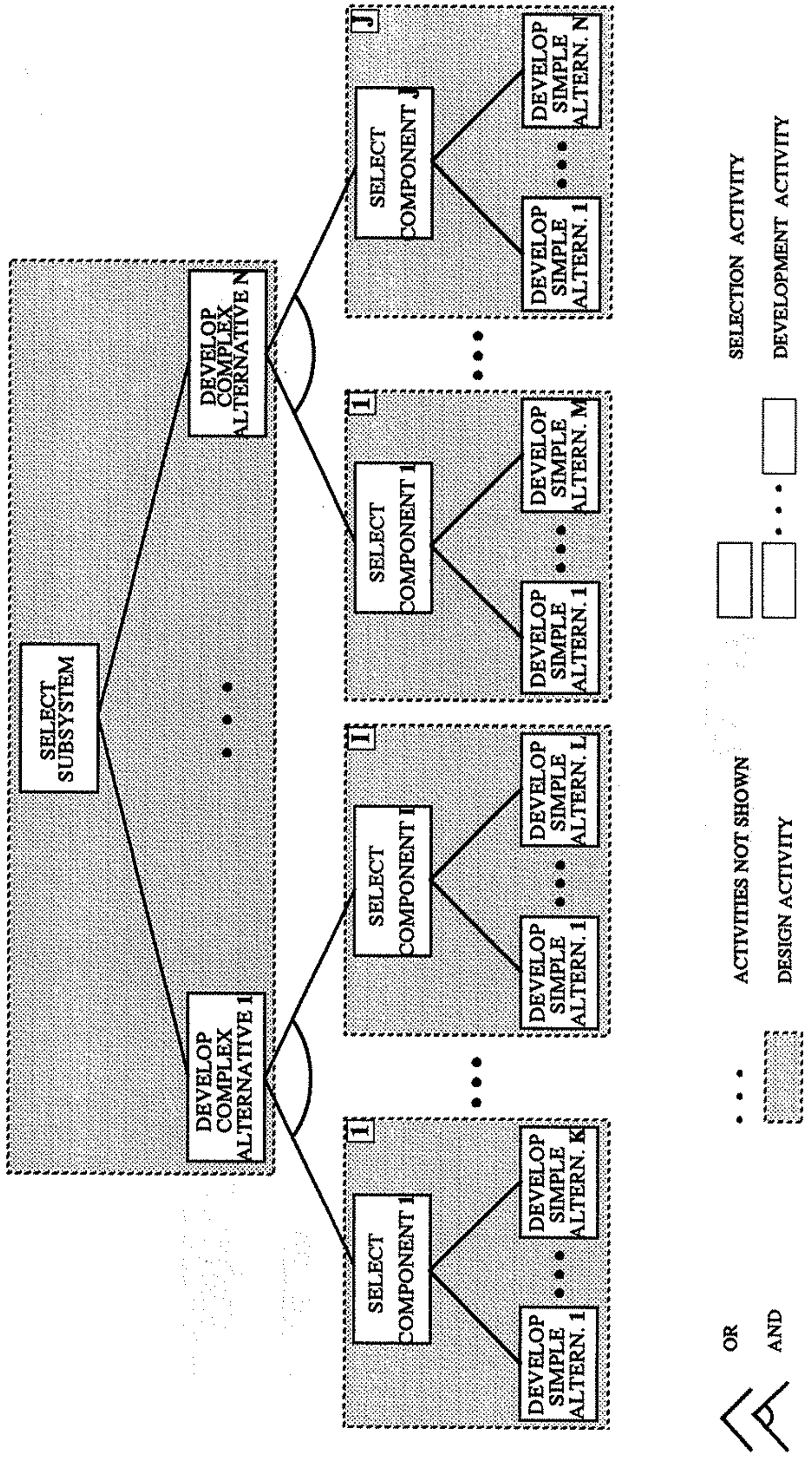


FIGURE 1.4 ACTIVITY TREE

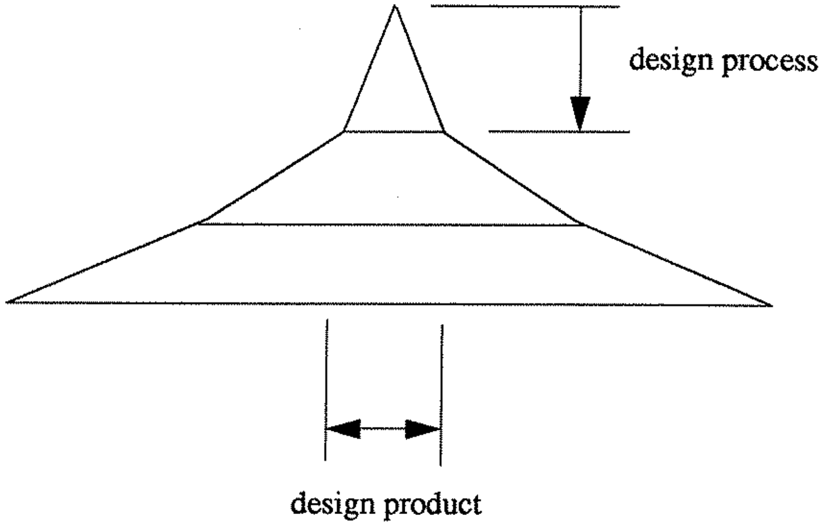


FIGURE 2.1 INFORMATION PYRAMID

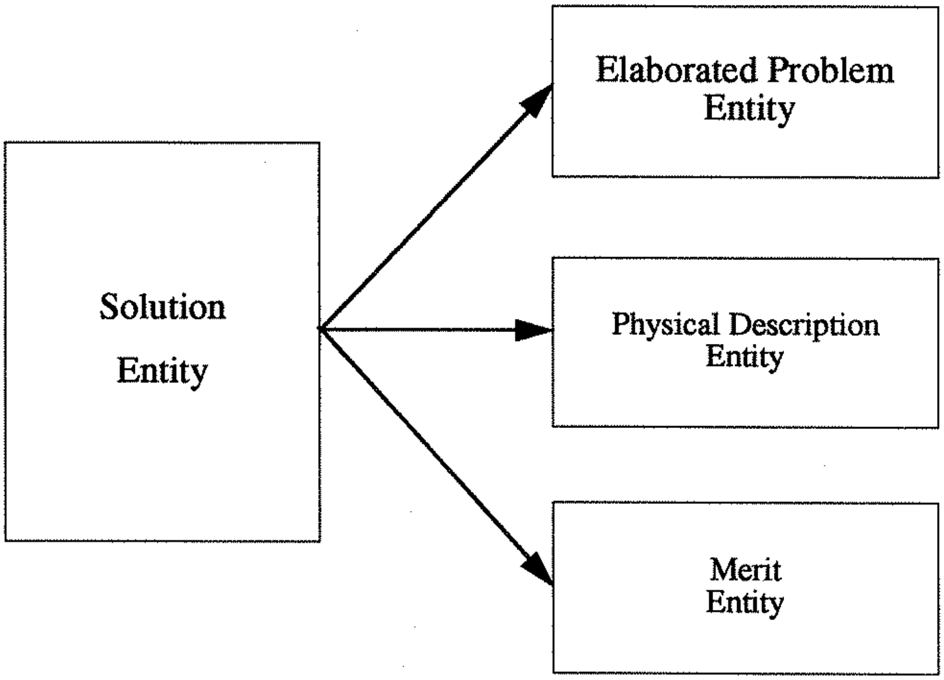


FIGURE 2.2 SOLUTION ENTITY

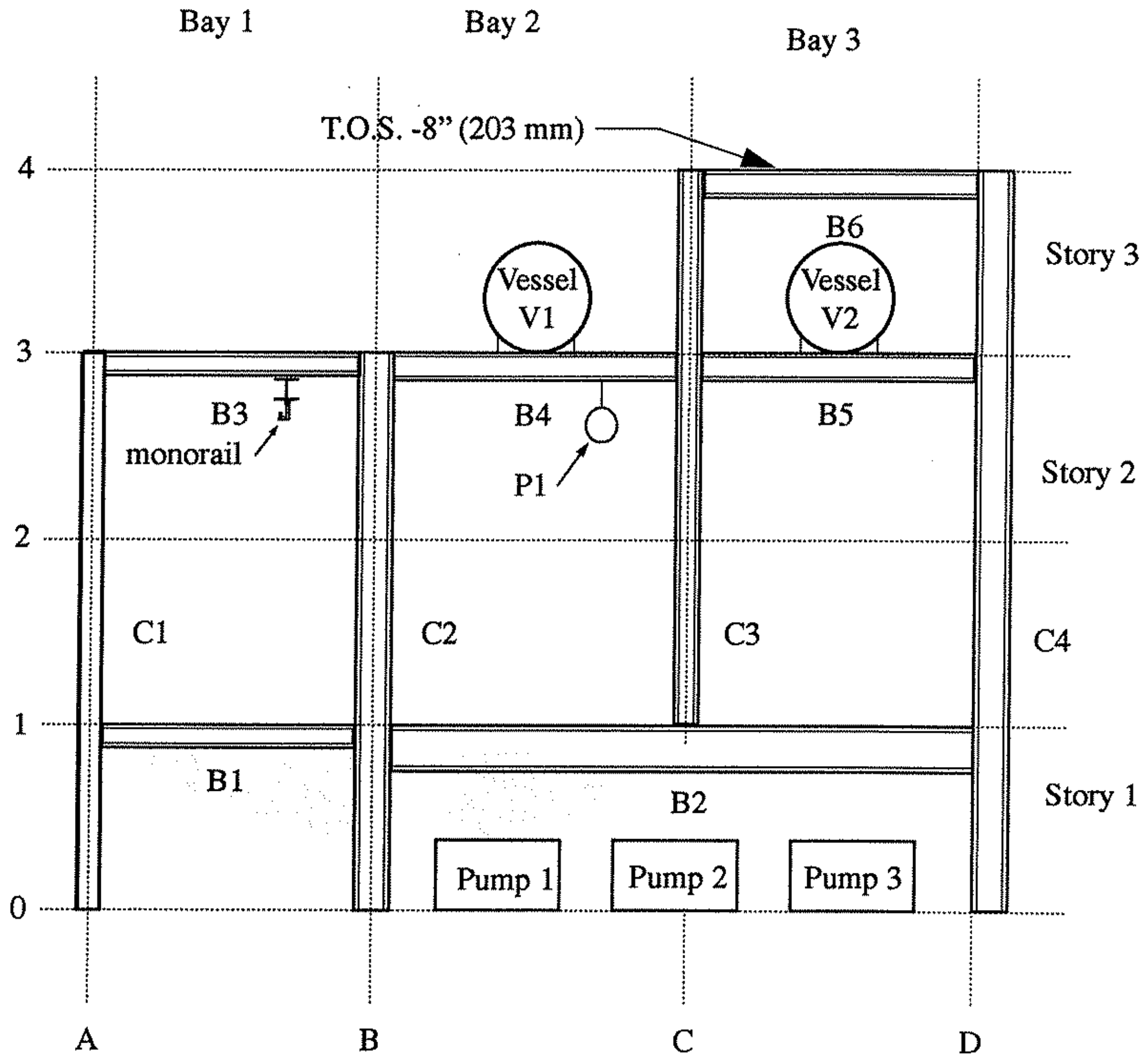


FIGURE 3.1 CONSTRUCTED FACILITY

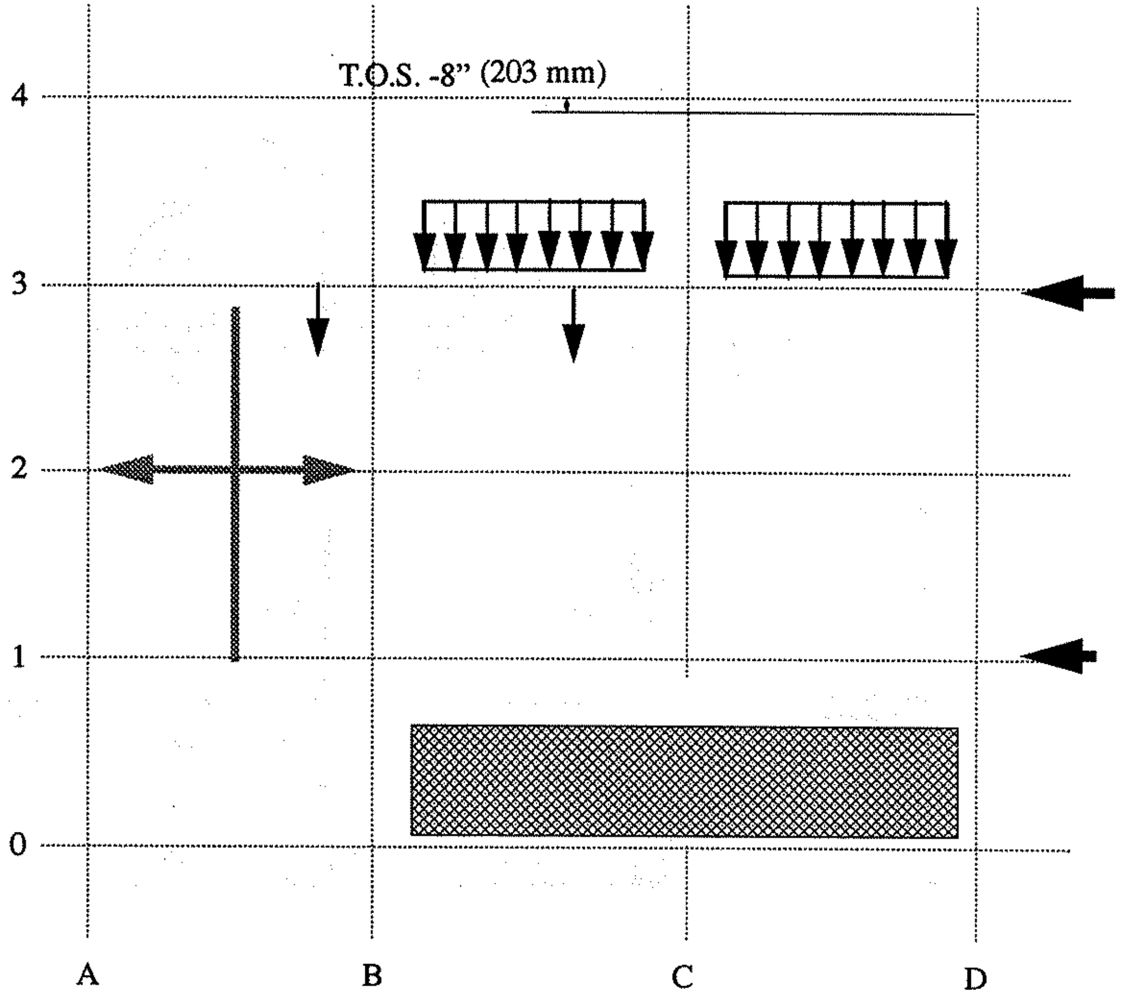


FIGURE 3.2A FRAME CONSTRAINTS

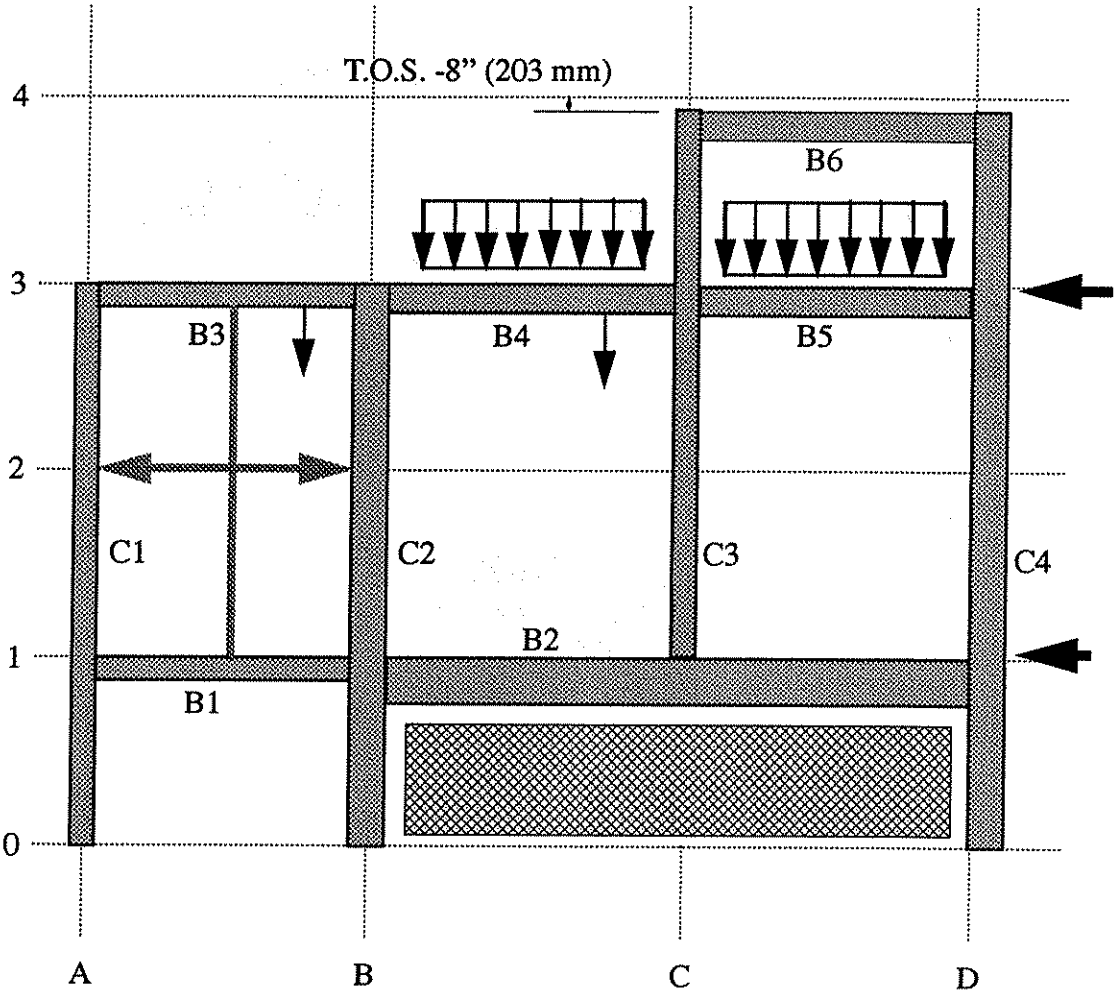


FIGURE 3.2B FRAME LAYOUT

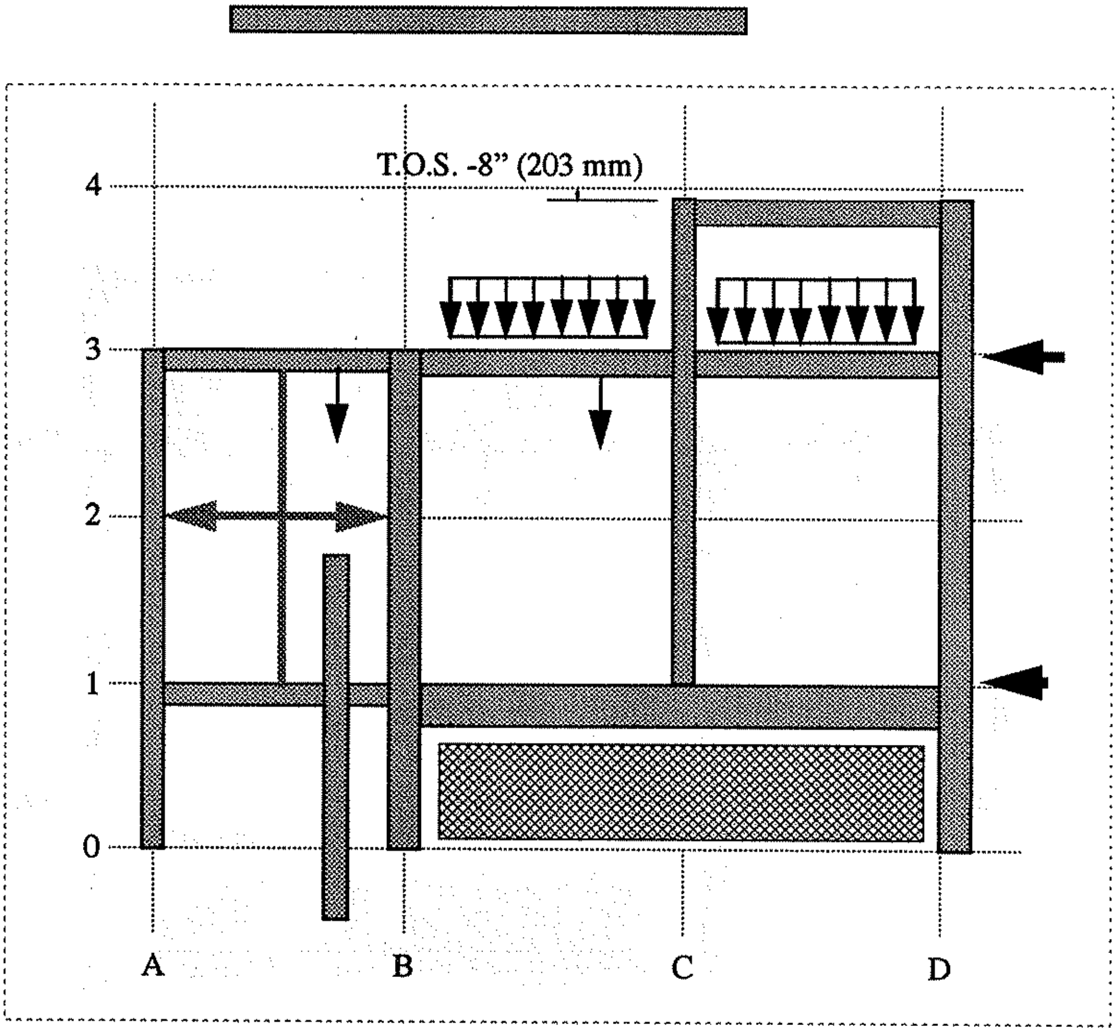


FIGURE 3.3 ILLEGAL COMPONENTS

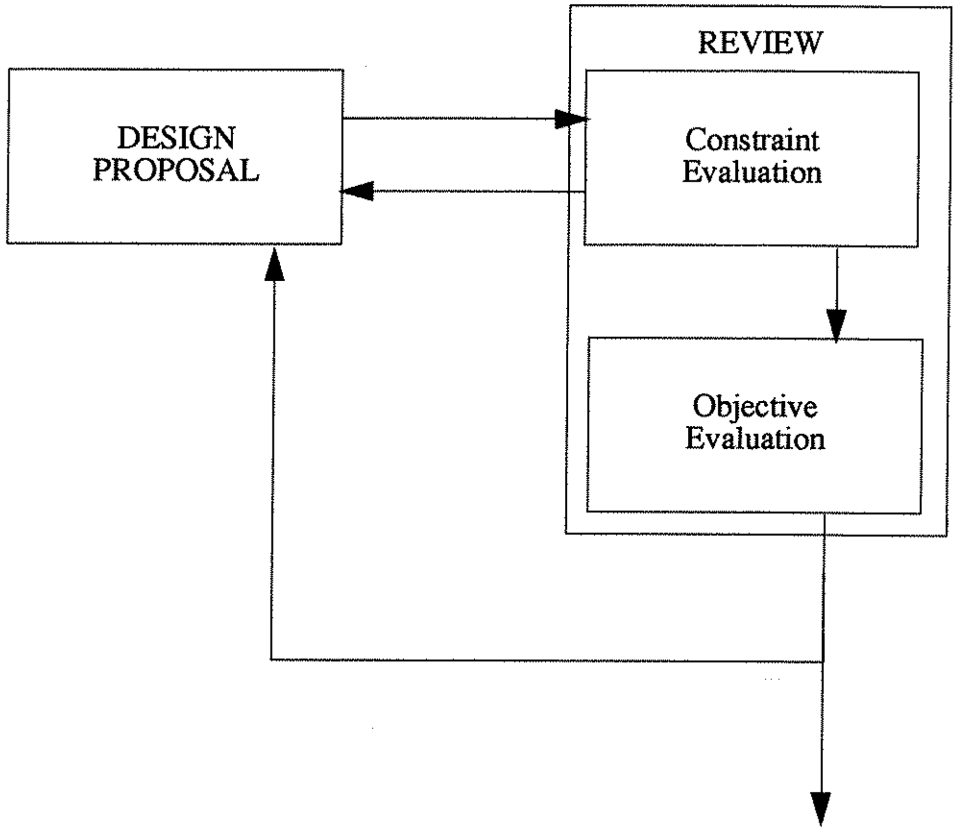
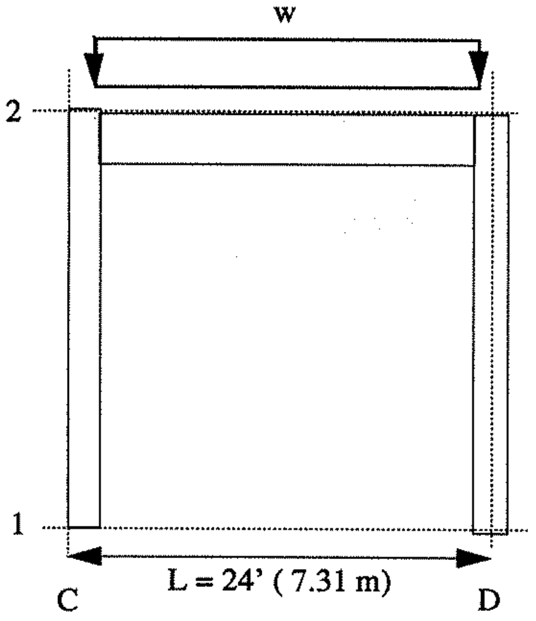


FIGURE 3.4 DESIGN PROPOSAL - REVIEW DESIGN CYCLE



LET:

w = 1.3 k/ft (18.9 kN/m)

M = 93.6 k-ft (126.9 kN-m)

d = L/2 = 12" (304.8 mm)

Ws = 5*M/d = 40 LB/FT (583 N/m)

FIGURE 3.5 OBJECTIVE INDICATORS

BEAM: Support gravity & live load (w)
BRACES: Transmit lateral load (P)

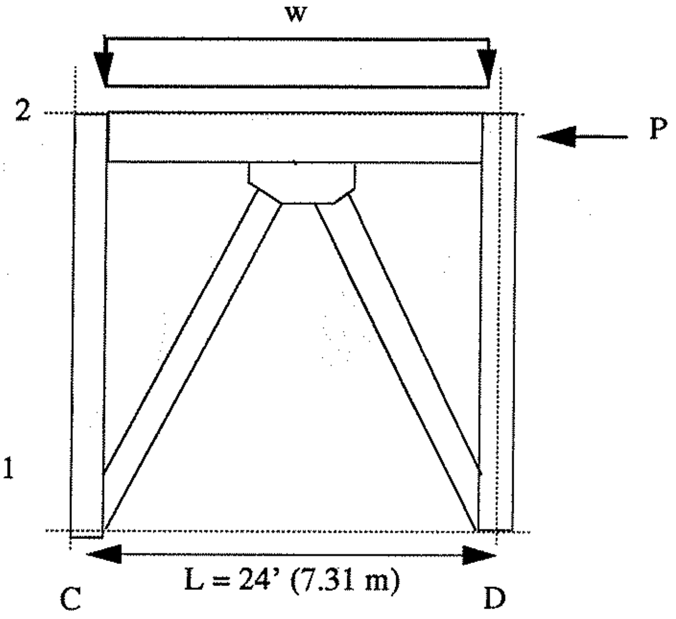


FIGURE 3.6 FUNCTION ASSIGNMENT EXAMPLE

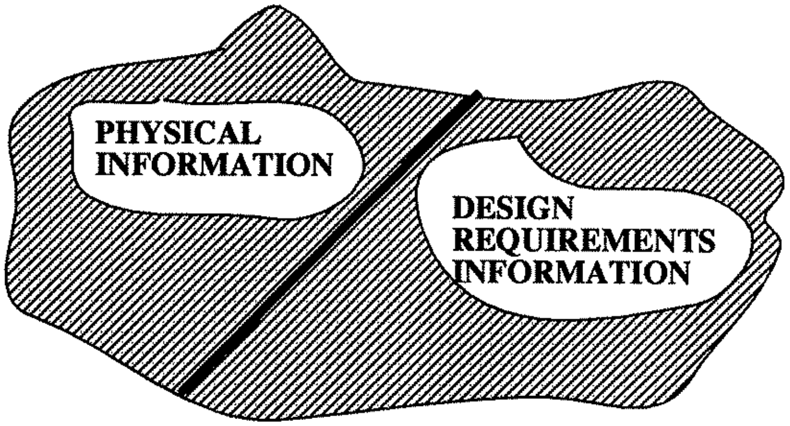


FIGURE 4.1 PRODUCT MODEL INFORMATION

Frame Problem Entity

Attributes:

grid lines

loads

offsets

clear spaces

relative positions

FIGURE 4.2 FRAME PROBLEM ENTITY

Grid Line Entity

Attributes:

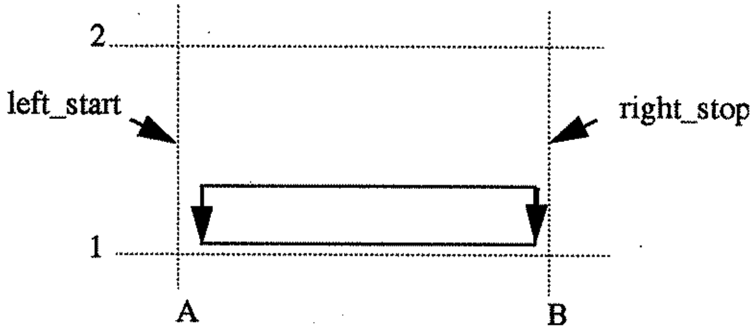
name: label

distance: distance from origin

orientation: vertical or horizontal type

FIGURE 4.3 GRID LINE ENTITY

Load Entity



Attributes:

name: description of the source of this constraint

left_start or bottom_start: bound of constraint (grid line +/- offset)

right_stop or top_stop: bound of constraint (grid line +/- offset)

supporting_surface: the component surface that is to support the load

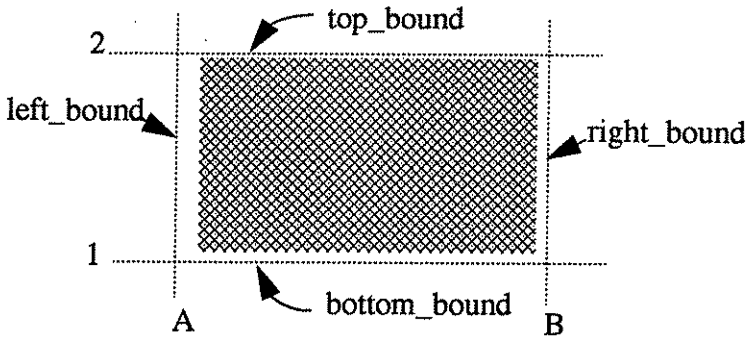
left_magnitude or bottom_magnitude: magnitude of the load corresponding to left_start or bottom_start

right_magnitude or top_magnitude: magnitude of the load corresponding to right_stop or top_stop

load_category: a nominal load category (e.g., live, dead, or wind)

FIGURE 4.4 LOAD ENTITY

Clear Space Entity



Attributes:

name: label for identifying constraint

bottom_bound: bound of constraint horizontal grid line (+/- offset)

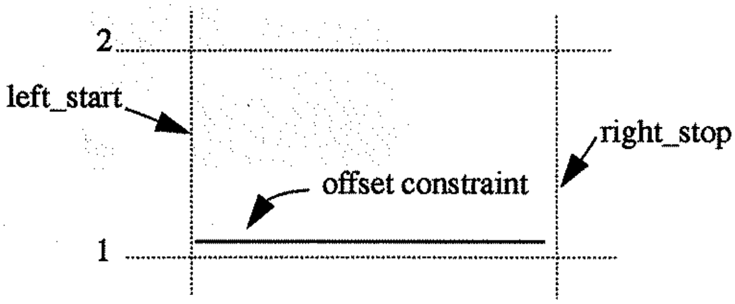
top_bound: bound of constraint horizontal grid line (+/- offset)

left_bound: bound of constraint vertical grid line (+/- offset)

right_bound: bound of constraint vertical grid line (+/- offset)

FIGURE 4.5 CLEAR SPACE ENTITY

Offset Entity



Attributes:

name: label for identifying constraint

left_start or bottom_start: bound of constraint (grid line +/- offset)

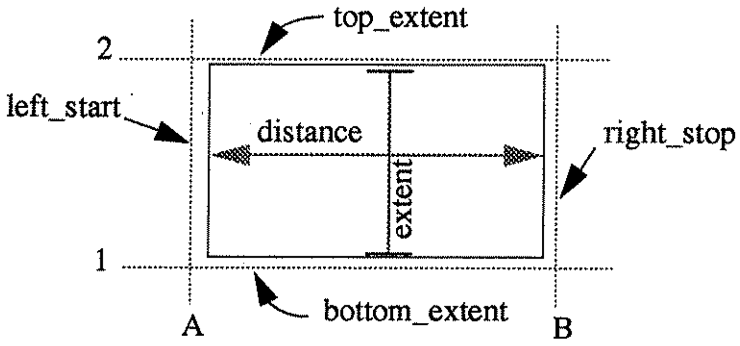
right_stop or top_stop: bound of constraint (grid line +/- offset)

offset_value: numeric value that represents the offset

component_surface: the surface of the components on the grid line that is constrained

FIGURE 4.6 OFFSET ENTITY

Relative Position Entity



name: label

left_start or bottom_start: bound of constraint (grid line +/- offset)

right_stop or top_stop: bound of constraint (grid line +/- offset)

bottom_extent or left_extent: extent of constraint (grid line +/- offset)

top_extent or right_extent: extent of constraint (grid line +/- offset)

start_gridline_surface: component surface that is constrained on the start grid line

stop_gridline_surface: component surface that is constrained on the stop grid line

distance: numeric value that may be a specified spacing between component surfaces

FIGURE 4.7 RELATIVE POSITION ENTITY

Frame Physical Description Entity

Atttributes:

beams

columns

joints

stories

bays

FIGURE 4.8 FRAME PHYSICAL DESCRIPTION ENTITY

Beam Entity

Attributes:

name: label for identifying individual beams

left_end: left end (grid line intersections +/- offset)

right_end: right end (grid line intersections +/- offset)

surface: beam surface that is positioned (e.g., top, bottom, and centerline)

left_joint: label of joint

right_joint: label of joint

intermediate_joint_list: list of joint labels

FIGURE 4.9 BEAM ENTITY

Column Entity

Attributes:

name: label for identifying individual columns

bottom_end: bottom end (grid line intersections +/- offset)

top_end: top end (grid line intersections +/- offset)

surface: column surface that is positioned (e.g., left, right, and centerline)

bottom_joint: joint label

top_joint: joint label

intermediate_joint_list: list of joint labels

FIGURE 4.10 COLUMN ENTITY

Bay Entity

Attributes:

name: label for identifying individual bays

right_bound: vertical grid line label

contained_beam_list: list of beams that are contained within the bay bounds

partial_crossing_beam_list: list of beams that partially cross the bay bounds

crossing_beam_list: list of beams that fully cross the bay bounds

FIGURE 4.11 BAY ENTITY

Story Entity

Attributes:

name: label for identifying individual stories

top_bound: horizontal grid line label

contained_column_list: list of columns that are contained within the story bounds

partial_crossing_column_list: list of columns that partially cross the story bounds

crossing_column_list: list of columns that fully cross the story bounds

Figure 4.12 STORY ENTITY

Joint Entity

Attributes:

name: label for identifying individual joints

left_member: label of beam that frames into joint

right_member: label of beam that frames into joint

top_member: label of column that frames into joint

bottom_member: label of column that frames into joint

left_fixity: fixity (e.g., either fixed or pinned) of left_member

right_fixity: fixity of right_member

top_fixity: fixity of top_member

bottom_fixity: fixity of bottom_member

FIGURE 4.13 JOINT ENTITY

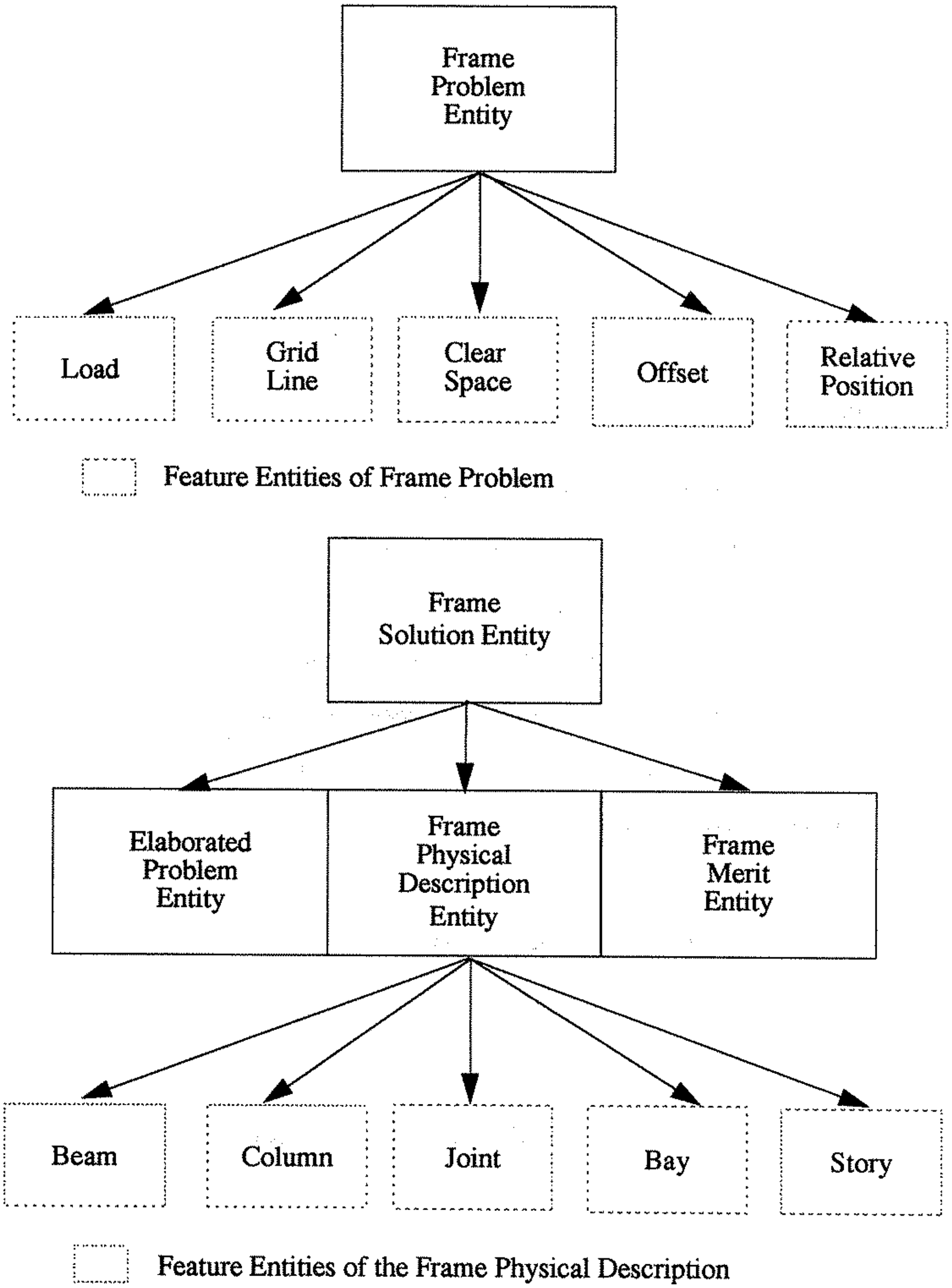


FIGURE 4.14 FEATURE, PROBLEM, AND SOLUTION ENTITIES

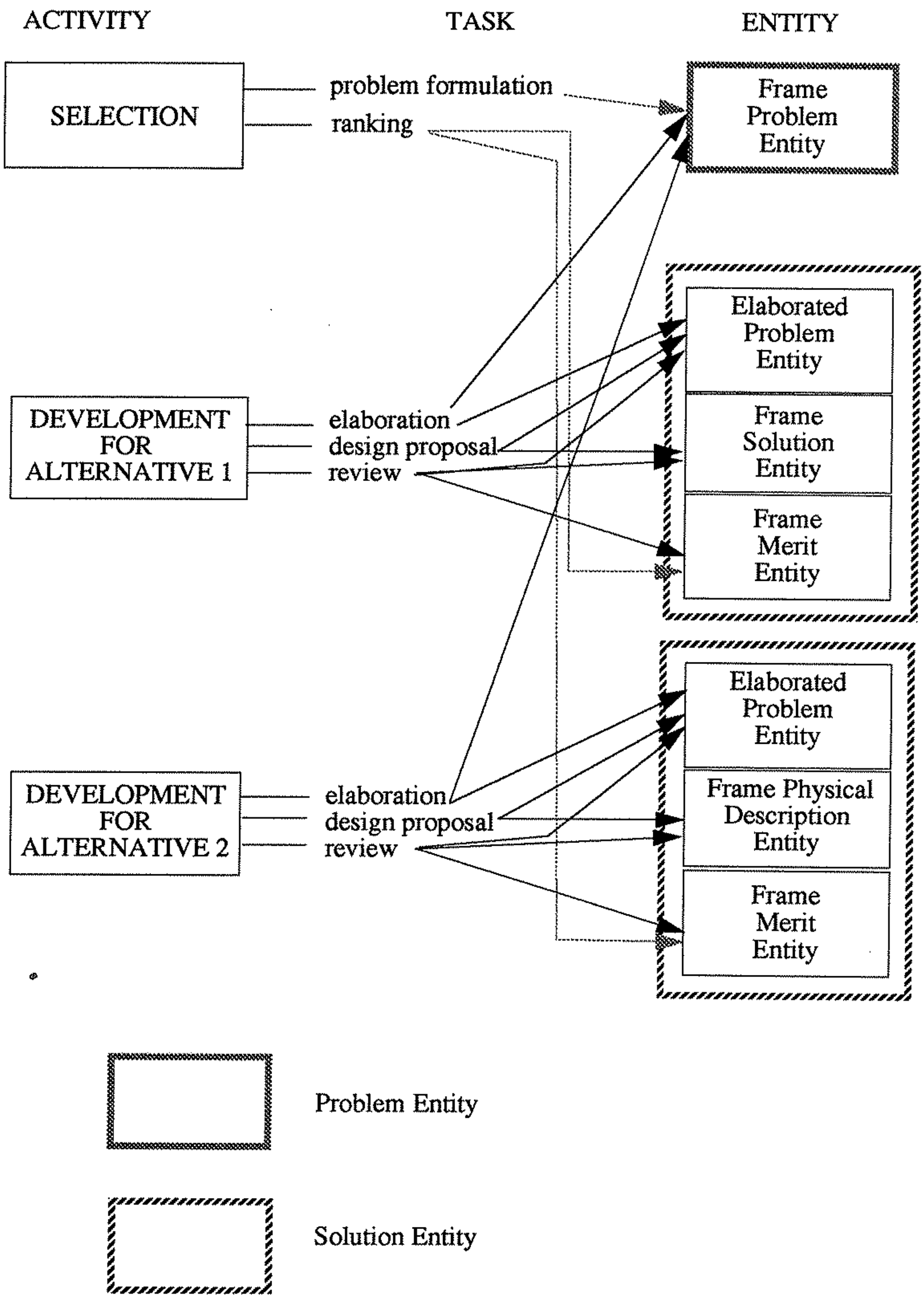


FIGURE 4.15 PRELIMINARY DESIGN - TASKS & ENTITIES

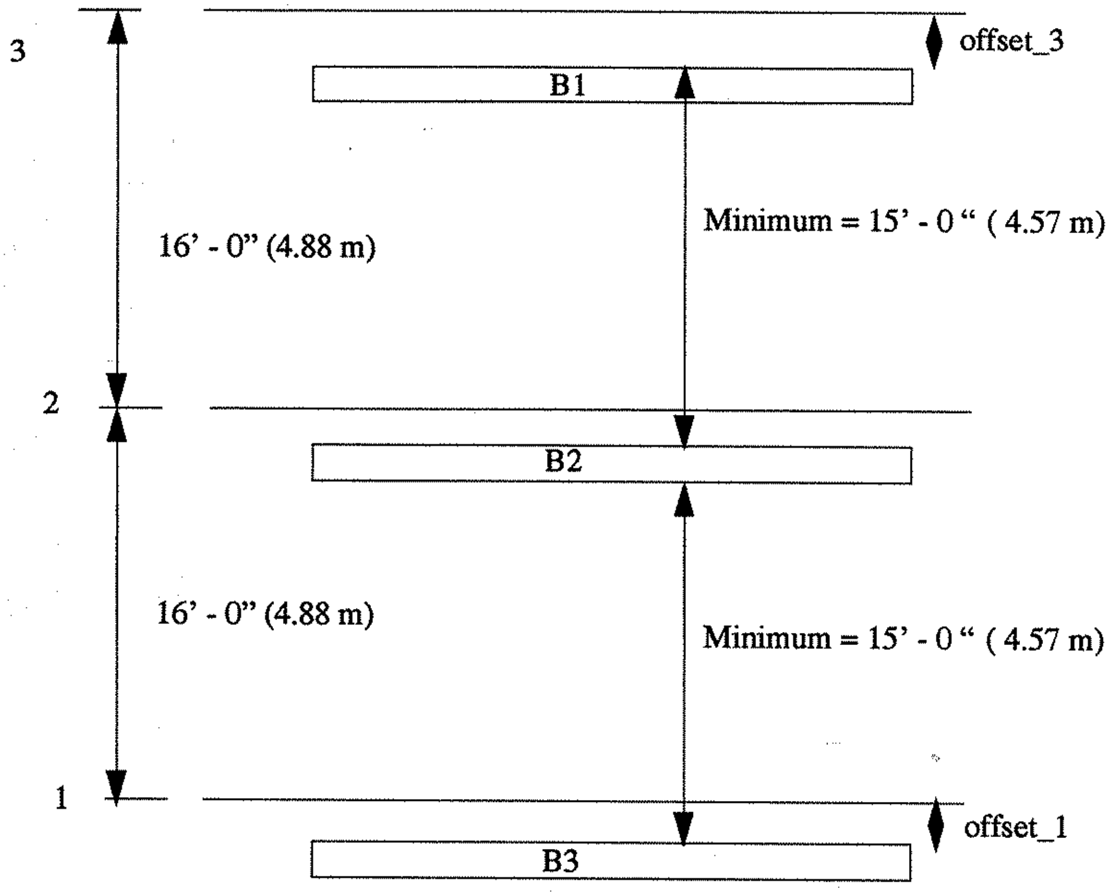


FIGURE 5.1 GEOMETRIC ANALYSIS

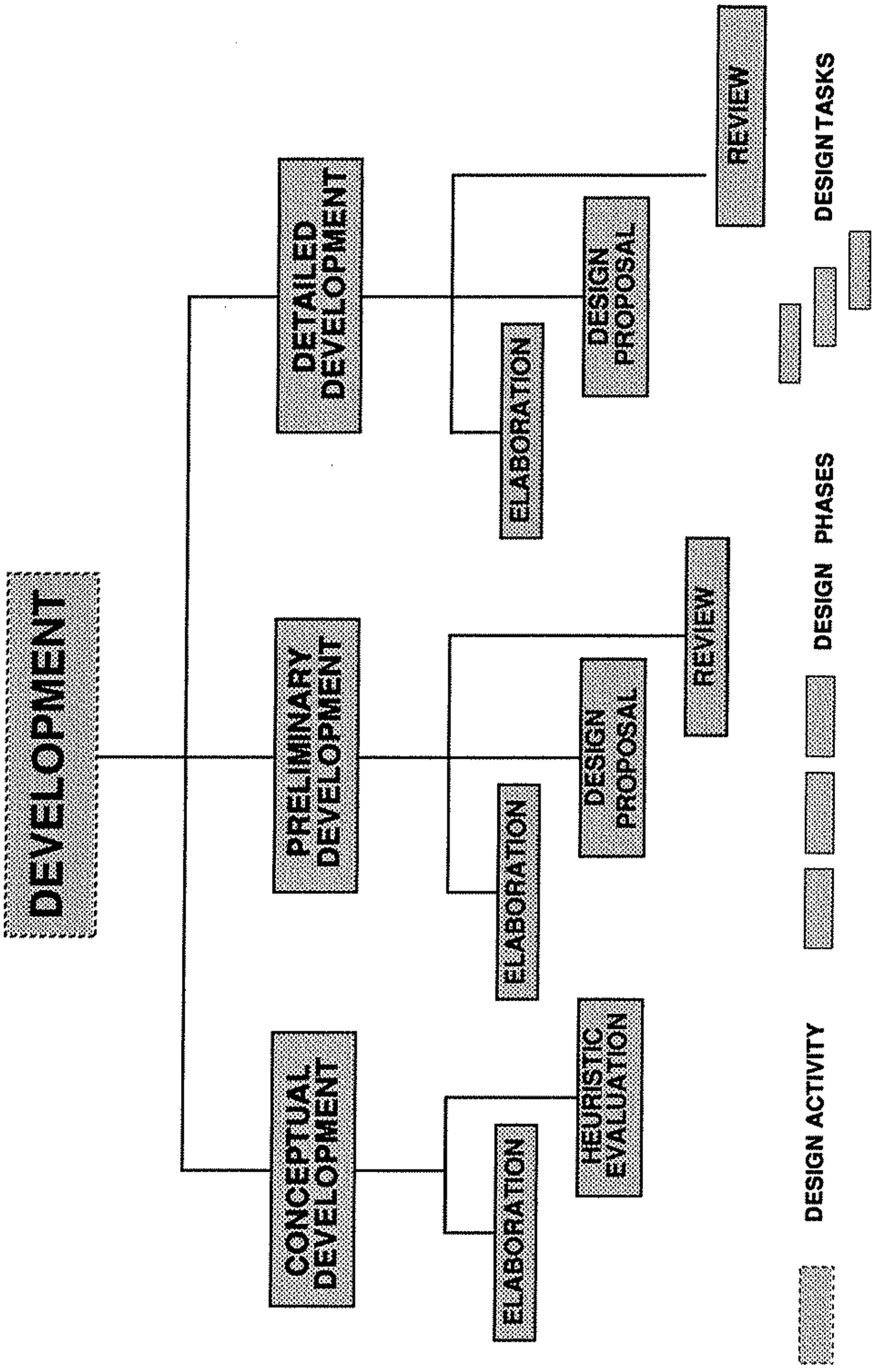


FIGURE 5.2 DEVELOPMENT ACTIVITY

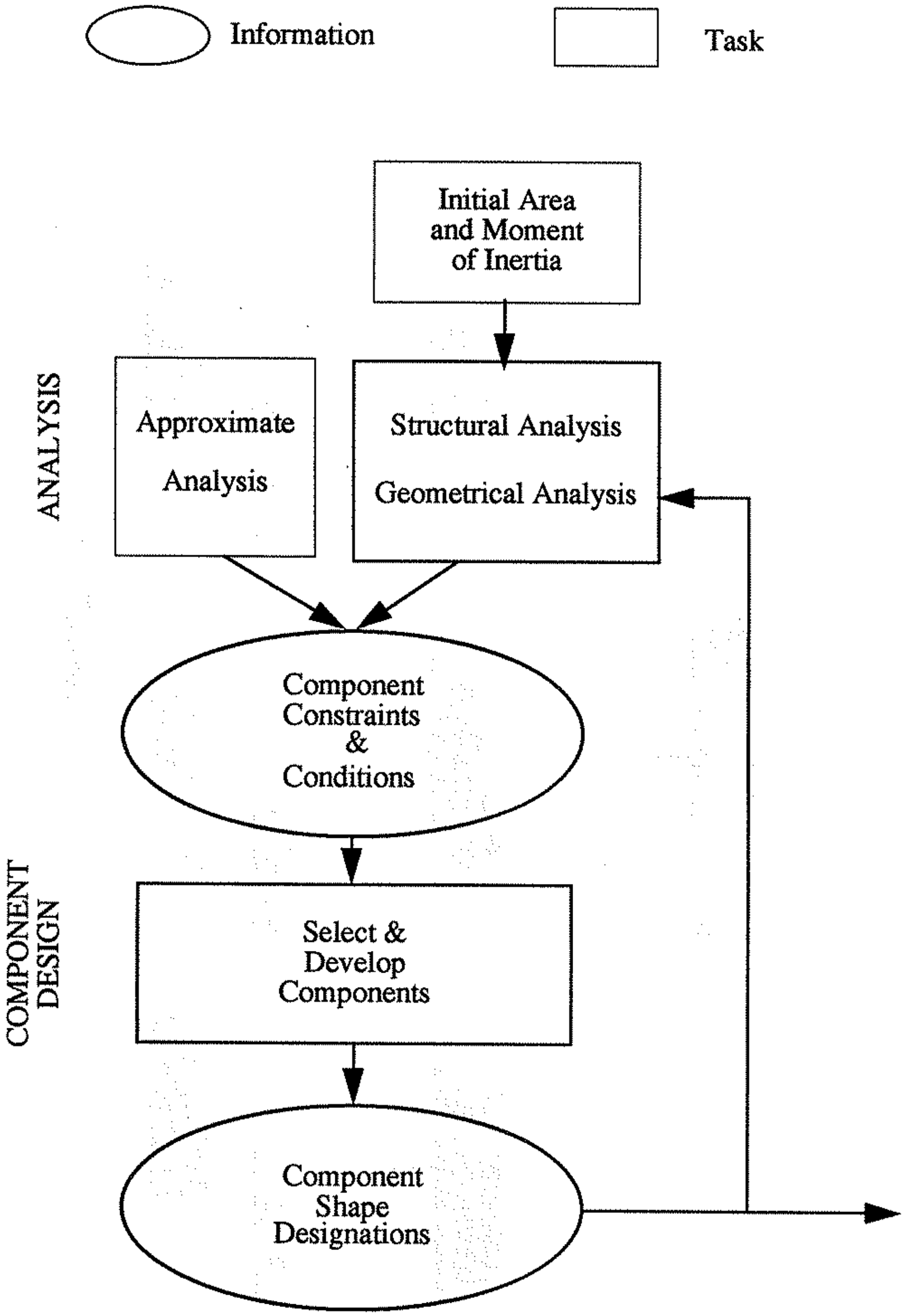


FIGURE 5.3 COMPONENT DESIGN

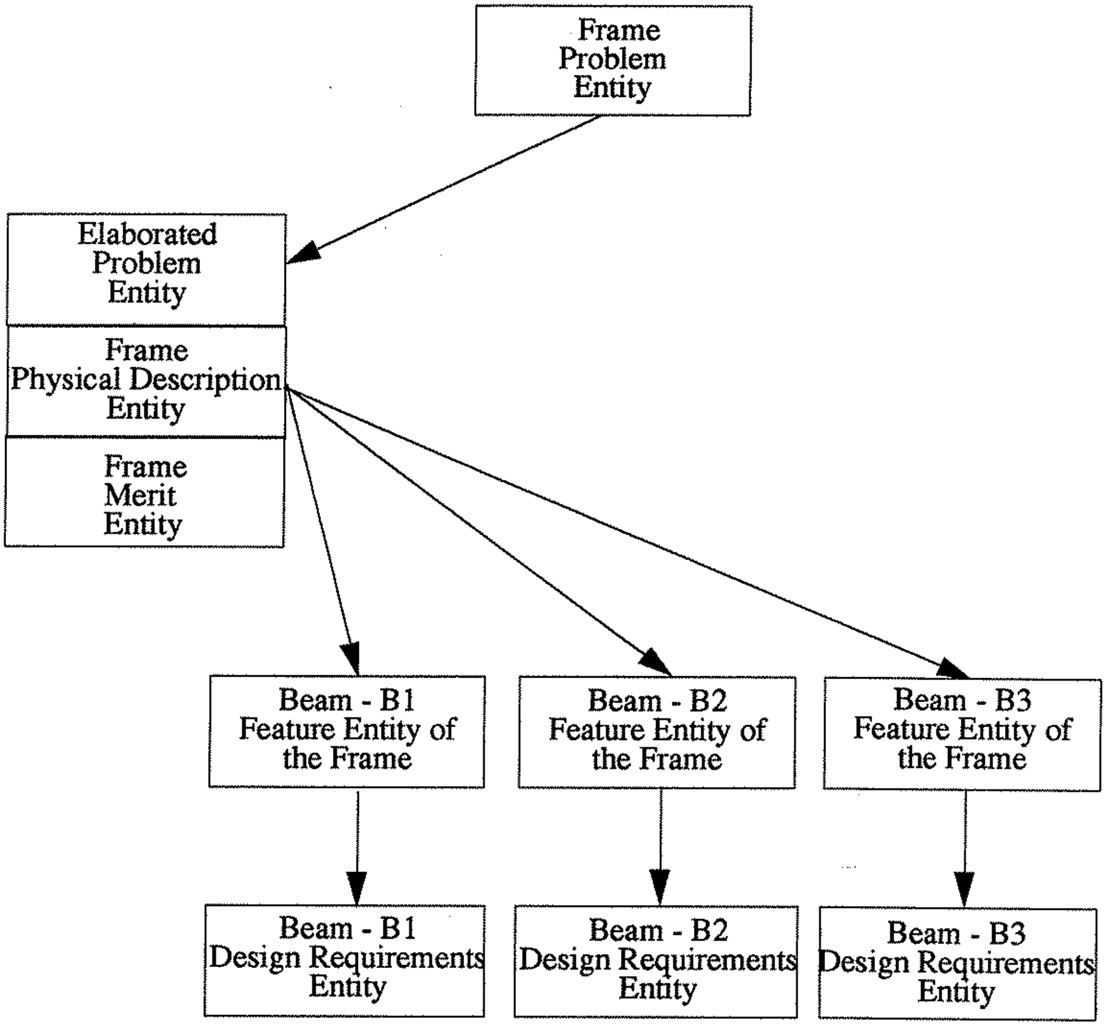


FIGURE 5.4 PRELIMINARY DEVELOPMENT II PRODUCT MODEL ENTITIES

Beam Design Requirements Entity

Attributes:

left_end_loads: moment, shear and axial forces applied to beam left end
right_end_loads: moment, shear and axial forces applied to beam right end
intermediate_load_list: list of loads between beam ends
depth_constraint: inequality or expression that restricts beam depth

FIGURE 5.5 BEAM DESIGN REQUIREMENTS ENTITY

Column Design Requirements Entity

Attributes:

bottom_end_loads: moment, shear and axial forces applied to column bottom end
top_end_loads: moment, shear and axial forces applied to columns top end
intermediate_loads: list of loads between column ends
width_constraint: inequality or expression that restricts column width

FIGURE 5.6 COLUMN DESIGN REQUIREMENTS ENTITY

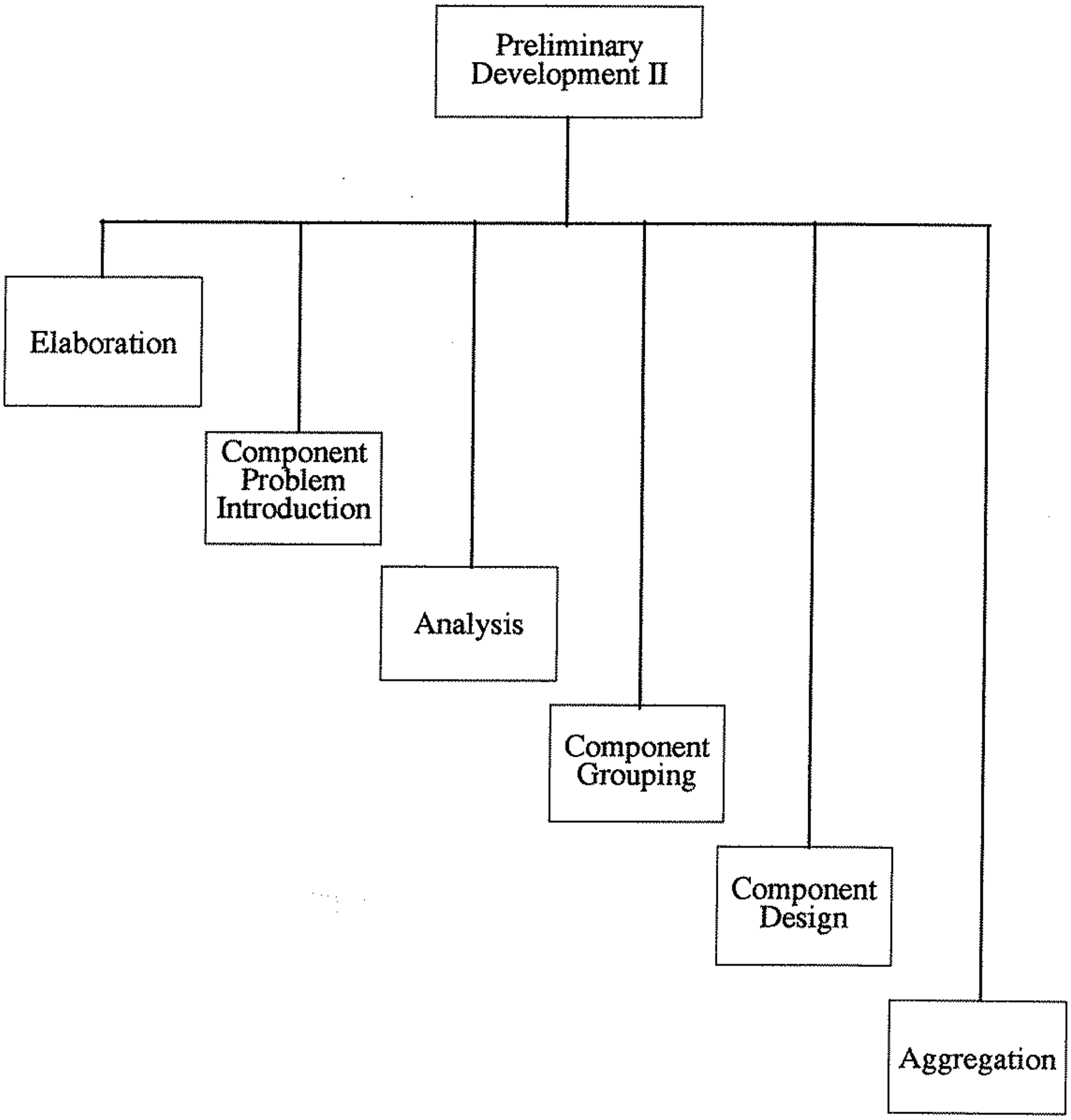


FIGURE 5.7 REVISED PRELIMINARY DEVELOPMENT II TASKS

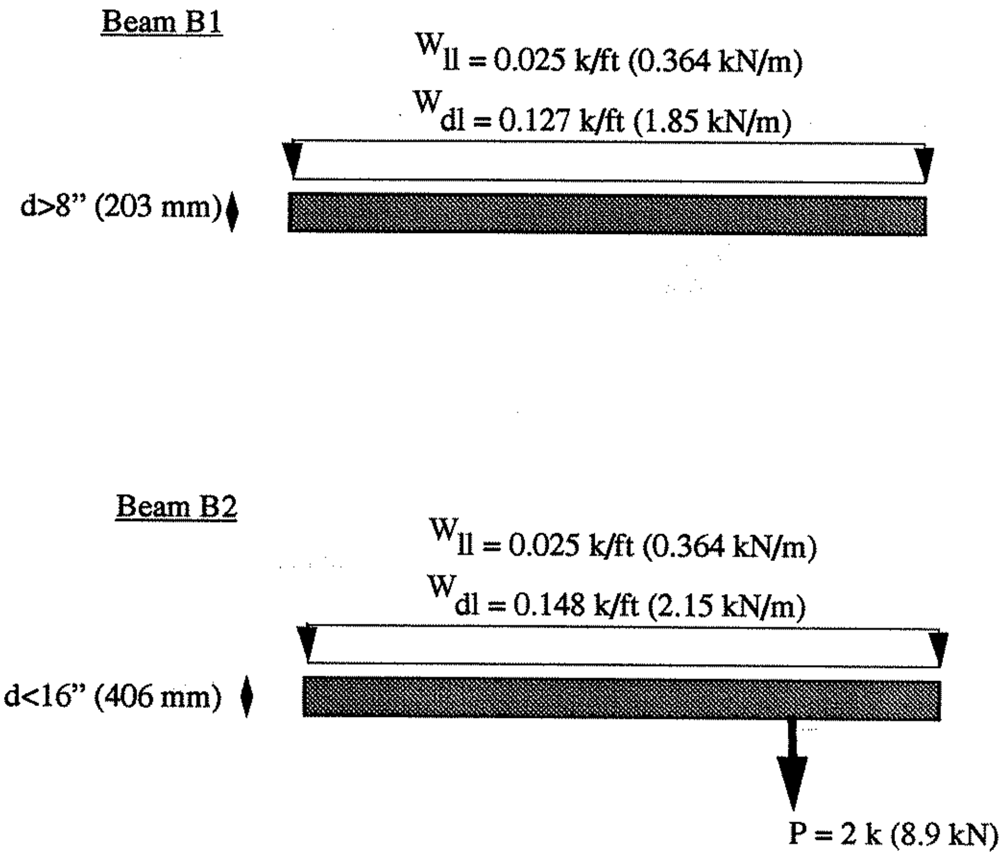


FIGURE 5.8 TYPICAL BEAM EXAMPLE

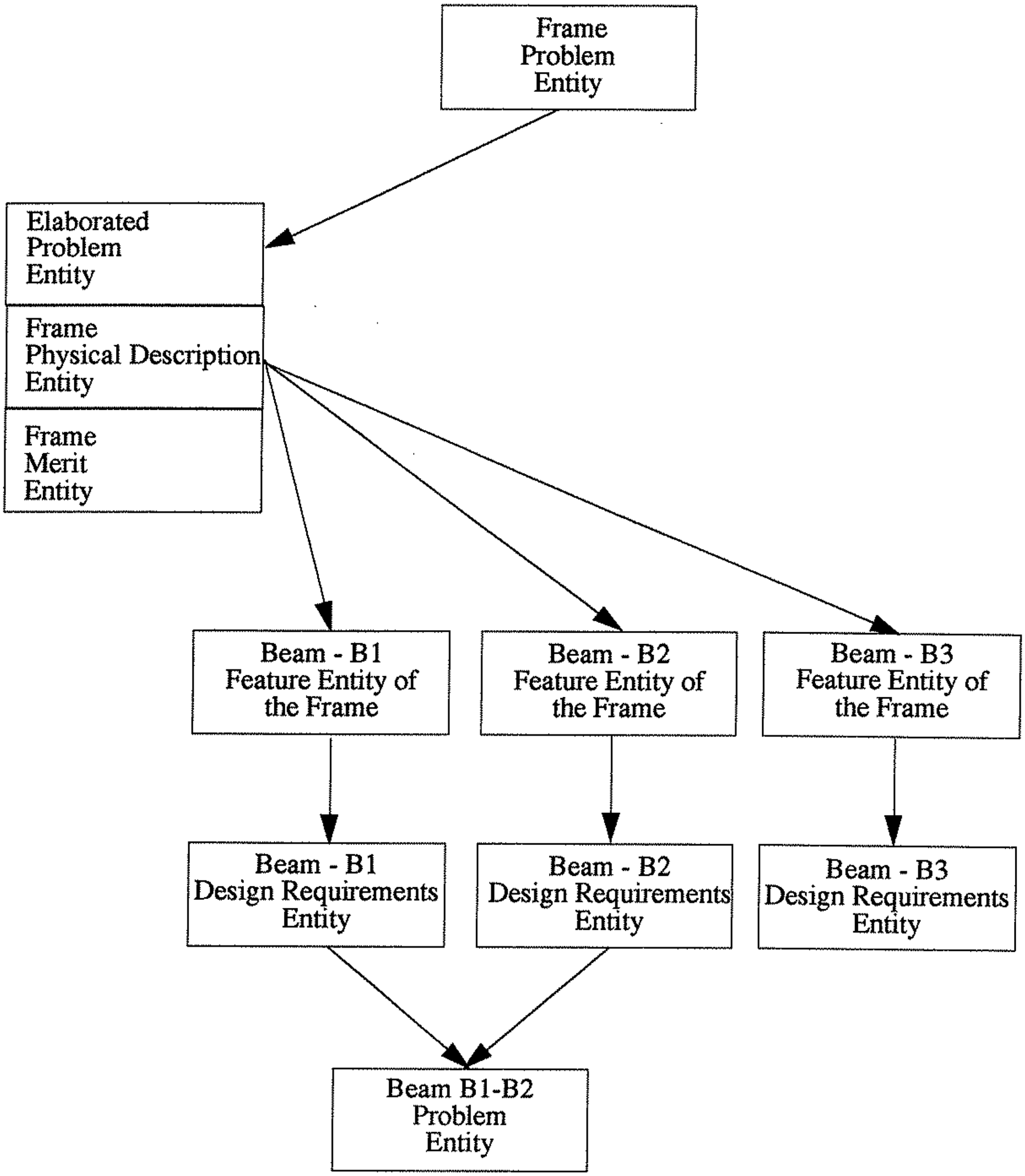


FIGURE 5.9 COMPONENT PROBLEM ENTITY

Beam Problem Entity

Attributes:

loads

depth_constraint

FIGURE 5.10 BEAM PROBLEM ENTITY

Column Problem Entity

Attributes:

loads

width_constraint

FIGURE 5.11 COLUMN PROBLEM ENTITY

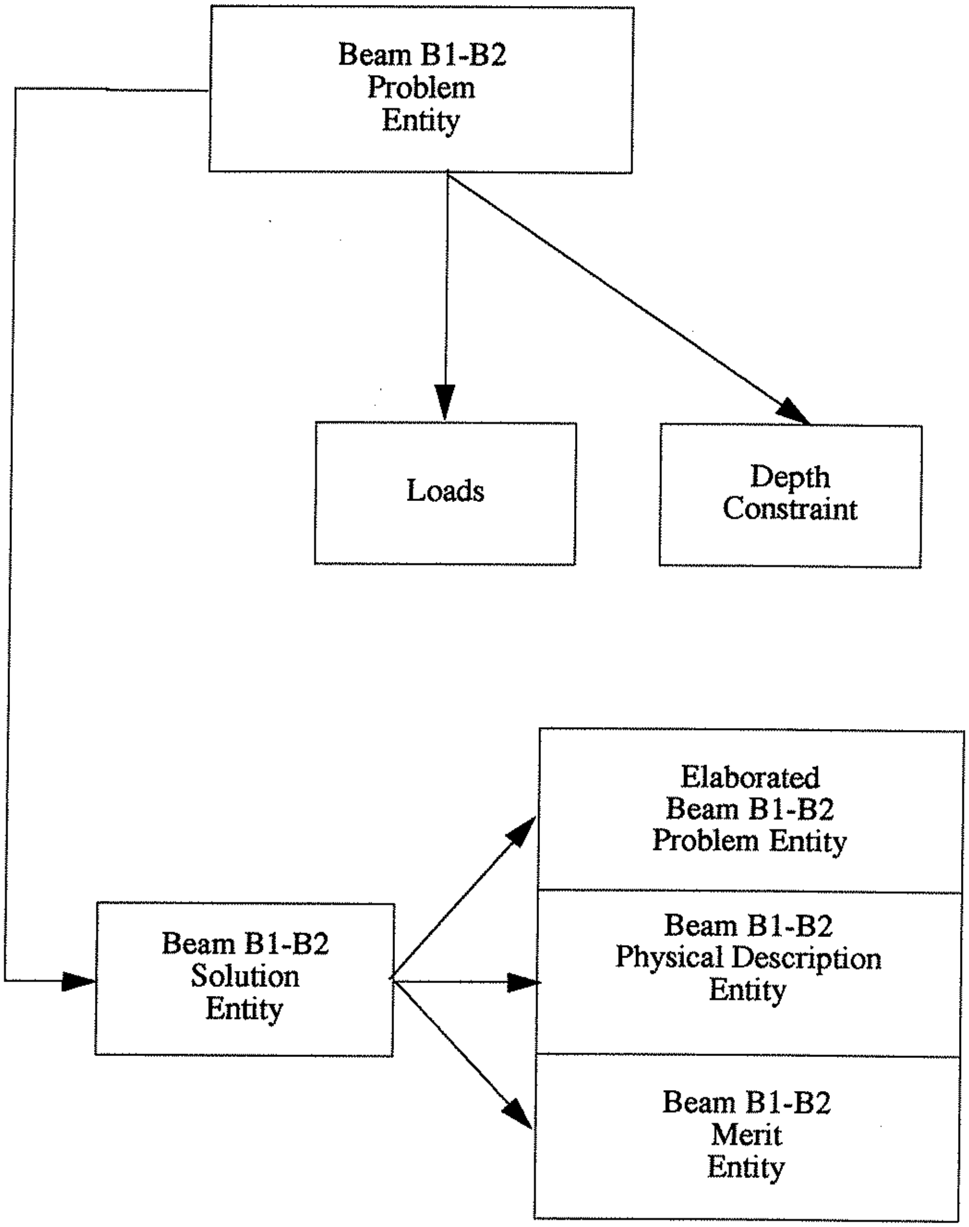


FIGURE 5.12 TYPICAL COMPONENT PROBLEM & SOLUTION ENTITY

Bibliography

Bader, D. A. (1991), "Object Oriented Simulation of Systems with Examples in Structural Design and Parallel Processing." Master's report, Lehigh University.

Barone, M., Werkman, K., Wilson, J. L., and Hillman, D. (1989), "A Knowledge-Based System for the Evaluation of Beam-to-Column Connections", NSF-ERC ATLSS Report 89-11, Lehigh University.

Bjork, B-C. (1989), "Basic Structure of a Proposed Building Product Model." CAD, 21 (2), 71-78.

CERF, Civil Engineering Research Foundation (1991), "Setting a National Research Agenda for the Civil Engineering Profession", Vol. 1, August 1991.

Cunningham, J. J. and Dixon, J. R. (1988), "Design with Features: The Origin of Features", Computers in Engineering 1988, ASME, pp. 237-243.

Cutkosky, M. R., Tennenbaum, J. M., and Muller, D. (1988), "Features in Process Based Design", Computers in Engineering 1988, ASME, pp. 557-562.

Libardi, E. C., Dixon, J. R., and Simmons, M. K., (1988), "Computer Environments for the Design of Mechanical Assemblies: A Research Review", Engineering with Computers, 4, 121-136.

Hemingway, G. (1991), "A Framework for Integrated Design", ATLSS REU Report, Center for Advanced Technology for Large Structural Systems, Lehigh University.

Howard, H. C., Abdalla, J. A., Phan, D. H., and Lavakare, A. P. (1991), "Primitive-Composite Approach for Structural Data", Computing in Civil Engineering and Symposium on Data Bases, Proceedings of the Seventh Conference, ASCE, pp. 799-808.

Madden, J. A., and Sause, R. (1992), "A Design Product Model for Computer Integrated Structural Engineering", Computing in Civil Engineering and Geographic Inofrmation Systems Symposium, Proceedings of the Eighth Conference, ASCE, pp. 113-120.

Madden, J.A., Sause, R., Bader, D., and Falkin, J. (1992) "A Framework for Integrated Design of Steel Frame Structures", NSF-ERC ATLSS Report 92-11, Lehigh University.

Martini, K. (1990), "Geometric Modelling for Structural Design", Ph.D. Dissertation, University of California, Berkeley.

Martini, K., Sause, R., and Powell, G.H. (1991), "Models for Computer Integrated Civil Engineering Design", The Fourth International Conference on Computing in Civil and Building Engineering: Extended Abstracts, Tokyo, Japan.

Nevill Jr., G.E. (1989) "Computational Models of Design Processes", Design Theory '88, Proceedings of the 1988 NSF Grantee Workshop on Design Theory and Methodology, S.L. Newsome, W.R. Spillers, and S. Finger, eds., Springer Verlag, pp. 82-116.

Nevill Jr., G.E., Garcelon, J.H., and Tambling K.B. (1989), "Creating Abstraction/Feature Sets for Top Down Mechanical Design: Experiences with Automating Preliminary Design of Structures", Computers in Engineering 1989, ASME, pp. 273-279.

Sanvido V., Fenves, S., and Wilson, J. L. (1992), "Towards a Virtual Master Builder", Journal of Professional Issues, ASCE, 118 (3), pp. 261-278.

Sause, R. (1989), "A Model of the Design Process for Computer Integrated Structural Engineering", Ph.D. Dissertation, University of California, Berkeley, 1989.

Sause, R. and Powell, G.H. (1990), "A Design Process Model for Computer Integrated Structural Engineering", Engineering with Computers, 6, pp. 129-143.

Sause, R. (1991), "Towards Management of Design Alternatives in Object Oriented Databases", Computing in Civil Engineering and Symposium on Data Bases, Proceedings of the Seventh Conference, ASCE, pp. 202-211.

Sause, R. and Powell, G.H. (1991), "A Design Process Model for Computer Integrated Structural Engineering: Design Phases and Tasks", Engineering with Computers, 7, pp. 145-160.

Sause, R., Martini, K., and Powell, G.H. (1991), "Object Oriented Approaches for Integrated Engineering Design Systems", Journal of Computing in Civil Engineering, ASCE, 6 (3), pp. 248-265.

Smith, J.M. and Smith, D.C.P. (1977), "Database Abstractions: Aggregation and Generalization", ACM Transactions on Database Systems, 6 (2), pp. 105-133.

Unger, M. B., and Ray, S. R. (1988), "Feature-Based Process Planning in the AMRF", Computers in Engineering 1988, ASME, pp. 563-569

Woodbury, R. F. and Oppenheim, I. J. (1987), "An Approach to Geometric Reasoning", IFIP WG 5.2 First International Workshop on Intelligent CAD, Cambridge, MA, October 6-8.

Zamanian, M. K., Fenves, S.J., Thewalt, C.R., and Finger, S. (1991), "A Feature-Based Approach to Structural Design", Engineering with Computers, 7, pp. 1-9.