1987

Spatial and functional representation language for structural design

Steven J.(Steven Joseph) Fenves
Carnegie Mellon University

Nelson C. Baker

Carnegie Mellon University.Engineering Design Research Center.

Spatial and Functional Representation Language for Structural Design

by

S. Fenves and N. Baker

EDRC-12-18-87^

Table of Contents

1 Introduction 1

- 1.1. Motivation 2
- 1.2. Purpose 2
- 1.3. Organization 3

2 Background 3

3 Description of Approach 6

- 3.1. Problem Description 6
- 3.2. Terminology 6
- 3.3. Uses of the Representation 9
  - 3.3.1. Generators 9
  - 3.3.2. Critics 9
  - 3.3.3. Operator Options , 10
- 3.4. Planned Development 12

4 Illustrative Example 13

- 4.1. Spatial Grammar 13
- 4.2. Functional Grammar 15

5 Summary 18

References 19

List of Figures

Figure 3-1: Design Object Description 8
Figure 3-2: Operator Overview 11
Figure 3-3: Transformation Description 12
Figure 4-1: Structural Frame Spatial Grammar 14
Figure 4-2: Structural Frame Spatial Language 15
Figure 4-3: Structural Frame Functional Grammar 16
Figure 4-4: Context Usage for Framing Development 17
Figure 4-5: Grammar forming Structural Systems 18

Spatial and Functional Representation Language for Structural Design[1]

Steven J. Fenves[2]
Nelson C. Baker[3]

Knowledge-based systems for structural design developed to date have used simple geometric representations which have not provided adequate spatial reasoning. Shape grammars are suggested as a representation for a knowledge-based system capable of performing spatial and functional reasoning. The representation needs to serve all disciplines involved in the design process, where different semantics of each discipline are associated with the same spatial information about design objects. The representation is demonstated in the building design environment, where possible structural systems can be generated dependent upon the building's spatial layout

1 Introduction

The study of preliminary structural design is a continuing research topic in the Department of Civil Engineering at Carnegie-Mellon University. Studies to date have resulted in knowledge-based systems which are pioneering new directions in computer applications to preliminary design. However, the systems developed are first products of a new era and fall short of addressing all issues associated with preliminary structural design. This study is expected to extend past research by addressing the critical issues of spatial and functional representation in preliminary design.

---

[1] **Also appeared in Gero, J.S. (Ed.) *Expert Systems in Computer-Aided Design*, 1987IHP WG5.2, February 1987.**

[2] **Sun University Professor of Civil Engineering**

[3] **Graduate Research Assistant**

1.1. Motivation

The motivation for the study comes from the advantages and shortcomings of the current knowledge-based systems (KBS) for structural desiga Some observations of previous work are as follows.

Advantages of KBS are:

- KBS provide a knowledge representation for solving design problems using computers which previously could be solved only by humans;
- preliminary structural design heuristics have been collected and successfully used in KBS;
- KBS allow symbolic and well as numeric manipulation of information; and
- current KBS have begun to show traits of human structural designers when performing preliminary design.

Shortcomings of present KBS are:

- knowledge-based design systems, in general, lack reasoning for dealing with shape and spatial constraints [Stefik 82];
- improved spatial representations for use in the structural design process are needed which would allow the design process to reason with the representation [Fenves 83];
- present structural design systems do not have reasoning capabilities based on the architectural function of enclosed or spanned spaces;
- there is no common representation to deal with the interactions between the load-carrying function and composition of structural systems and the architectural function and use of spaces; and
- present knowledge-based structural design applications deal only with predefined structural systems, extending the full height of the building, and with predefined placement locations, without recognizing other structural systems already located.

The above motivation has prompted this study, so that a better representation can be developed, enabling the design process to reason with the spatial as well as functional attributes of structural systems.

1.2. Purpose

The primary purposes for the study are threefold:

- To develop a common representation of structural systems, architectural spaces, and their relationships.
- To develop a KBS operating on the representation by reasoning with preliminary building design grammars.

- To understand and represent the design process and terminology more formally.

In the design of a building, there are at least two concurrent goals: to provide a functional and aesthetically pleasing building (typically the architect's domain); and to provide a structurally safe building (typically the structural engineer's domain). Since the two issues play a dominant role in design, the representation developed is intended to serve as a common ground for both architectural planning and structural system design. As will be described later, each of the two tasks is to include its own definition of semantics and context of the design, while sharing common spatial attributes. The end product of this formulation should be a methodology to aid designers in performing preliminary design.

1.3. Organization

This paper begins with a survey of existing literature on computer applications in structural engineering and architecture. Following the review, a discussion of the characteristics, terminology, and conceptual parts of a representation needed for preliminary structural design is presented. A description of the areas of research needed to develop a complete system and those items explicitly selected for this study follows. An example application of the shape grammar is given in the fourth section, and finally, a summary concludes the paper.

Throughout the paper new terminology will be indicated in *italics.* Several definitions are now presented so as to provide a basic understanding of the ideas in the paper.

- **Structural System:** A *structural system* is composed of structural elements and their connections, developed to transfer lateral and gravity loads. Associated with each structural system is a "type" (e.g., a truss, braced frame, rigid frame, etc.). These types describe the load carrying function and composition of structural systems.
- **Architectural Layout:** An *architectural layout* is composed of architectural elements configured in various positions. Each element has a purpose, such as a living room or kitchen. The layout is described by the function and composition of the rooms and their enclosures.
- **Preliminary Design:** The first attempt during the design process at satisfying initial constraints in order to synthesize architectural layouts and structural systems. The resulting potential design solutions are evaluated in order to produce one system that will be elaborated during detailed design.

2 Background

In recent years there has been a great deal of excitement over the developments in expert systems and artificial intelligence. With new techniques such as AI emerging from computer science research, researchers in other disciplines are finding new approaches to old problems, and the usefulness of computer applications is further increased.

Many applications have been started in disciplines outside of computer science using new AI techniques. Some of these include: Akiner in Architecture [Akiner 86]; Brown and Dixon in Mechanical Engineering [Brown 85] [Dixon 85]; Kim in Electrical Engineering [Kim 84]; Maher and Sriram in Civil Engineering [Maher 85] [Sriram 86]; Moore in Process Control [Moore 85]; and Zumsteg in Aerospace Engineering [Zumsteg 85]. As AI applications are emerging, researchers are trying to establish common techniques and are looking to each other for guidance.

This fresh approach to problem solving is the impetus for the methodology illustrated in this paper. The following paragraphs give a perspective and description of past work that predicates the next step in developing preliminary structural design systems.

It has become clear that proper use of computer applications can reduce the cost of design of structures, as well as reduce material cost as designs become more refined [ASCE 86] [Smallowitz 86]. However, the majority of designs done by computer have only used analysis programs. These analysis tools are not very good for design, and are essentially useless for preliminary design as they require the structure's topology, geometry, and member properties as input. It is the purpose of preliminary design to determine the topology and geometry, and to develop estimates of member properties. For detailed design, when the topology and geometry of the structure have been determined, analysis tools must be used iteratively so that preliminary values of member properties may converge to acceptable values satisfying strength and serviceability requirements. New developments in artificial intelligence techniques are expected to help make the design process easier to perform by computer.

Knowledge-based approaches, one class of AI techniques, attempt to capture some of the design experience of the structural engineer. The first major application of knowledge-based techniques to structural design was HI-RISE [Maher 85]. This system was the first attempt to represent structural engineering knowledge for preliminary building design in a knowledge-based system. The system contains knowledge suitable only for tall buildings, in that it performs the design of lateral load resisting systems first This constraint was removed in ALL-RISE, a formalism for a more general approach [Sriram 86]. Both of these systems use a geometric representation presented by Fenves and Liadis [Fenves 76], allowing only orthogonal bays, without setbacks, missing bays, and the like. While these KBS have taken a great step forward in representing the designer's knowledge for preliminary design, their flexibility for the representation of the building geometry has been greatly reduced from that incorporated in algorithmic analysis programs.

Issues of geometric representation have been dealt with extensively in space planning. Most of the space planning applications have used shape grammars, developed by Stiny and Gips [Stiny 75] [Gips 74].[4] Several authors have generated spatial configurations defining floor plan arrangements. Flemming has used a shape grammar for generating an international style of architecture

---

[4] A history of shape grammars can be found in an article by March [March 85].

[Flemming 81], Koning and Eizenberg represented the floor layout configurations of Frank Lloyd Wright's Prairie Houses [Koning 81], Knight showed shape grammar use in Japanese tearoom **plans** [Knight 81], and Stiny represented Mughal Gardens [Stiny 80], among others.

An analogy between shape grammars and production systems has been presented by Gips [Gips 79], [Gips 80]. All production systems have three components:

- a method for storing and representing the objects on which they operate (working memory);
- a representation to store productions (production memory); and
- a mechanism to apply productions which transform objects to other objects (inference engine) [Charniak 85].

The objects that production systems operate on have typically been strings of symbols. However, objects have also been defined for arrays, trees, graphs, shapes, and list structures [Gips 80].

Significant work in applying AI techniques to shape grammars has been done by Coyne and Gero at the University of Sidney [Coyne 85a]. Their work has involved reasoning about architectural configurations and design formulation. The applications have been developed using Prolog, so that predicates could be developed to operate on a fact base of design knowledge. Gero indicates that there are two types of knowledge that must be represented [Gero 85]. These are:

- knowledge about individual objects (such as their connectivity, geometry or dimensions, and desired attributes such as color or type); and
- knowledge about groups of objects.

The knowledge required for groups of objects are topological relationships (i.e. above, below) which are usually transitive; geometric relationships such as A touching B; and attributes which are usually heuristic in nature, such as A should be to the left of B. Much of this knowledge was demonstrated by Akiner in TOPOLOGY-1 [Akiner 86] [Gero 83].

Following the work of Stiny and others, the grammars shown in this paper are bottom-up or constructive grammars; that is, the grammar rules use simple objects to form more complex objects. In contrast, top-down grammar rules decompose objects into smaller constituents, as in Backus-Naur Form (BNF) rules. Design KBS such as HI-RISE implicitly use a top-down approach.

From the literature reviewed it is apparent that there is a need for a much better formalism of spatial concepts that can be used in preliminary design, as well as a need to have generative abilities to create designs that can be guided by semantic information. These issues will be defined below.

3 Description of Approach

The previous section has given a background of studies related to the proposed representation formalism. This section integrates the previously independent concepts of shape grammars and structural design, and addresses more clearly the spatial issues in structural design.

3.1. Problem Description

Structural engineering design is characterized by tight coupling between spatial and functional attributes. In achieving its function of providing structural support, a structural system is strongly dependent on its spatial location. In addition, the placement of structural systems impacts architectural functions such as the movement of people within a building or the aesthetics of the building. The structural components must perform their load carrying function without imposing on the architectural functions of the enclosed space.

Previous design KBS have depended on the human designer to place the structural systems, or permitted the placement to be selected from apriori defined locations. The next generation of KBS should automatically perform location generation and structural system selection. Before a KBS of this nature can be implemented, a general representation must be developed which allows for reasoning about the location and structural function of the structural systems and its components, the location and architectural function of the enclosed spaces, and the interactions between the two. This representation must be accessible to all disciplines in the design process so that all parties can communicate through a common representation.

Therefore, the central issue is to develop a representation that explicitly incorporates *spatial attributes* (topological, geometric, and spatial relations between design objects as described below) and *functional attributes* (structural and architectural), and then to demonstrate the use of this representation in preliminary structural design. As will be shown, the semantic and contextual aspects of the attributes will help to determine the set of design solutions which can be generated.

3.2. Terminology

In natural language, sentences are better understood when one uses the semantics and context of word phrases, in addition to their syntax. The same is true in structural design. The spatial aspects (similar or on the same level as syntax in sentences, as shown below) of a structural system do not give the. total picture of its behavior. The concepts of function (similar to semantics) and design context must also enter the representation and design process. The following paragraphs attempt to define design syntax, design semantics, and design context.

Design **Object.** There are two types of symbols in the current study: structural and architectural. The *structural symbols* are structural components (e.g., beams and columns), assemblies of components such as bents, and structural systems. The *architectural symbols* are spaces and partitions, and their combinations. These symbols will be collectively referred to as ***design objects.***

Topology. Topological (or connectivity) relations exist on many levels in a building. Topology represents the connectivity of design objects; for example architectural symbols connecting partitions to rooms, rooms to floors, and floors to buildings, as well as structural symbols connecting members to bents, bents to structural systems, and structural systems to buildings. It also provides the connectivity between the two types of design objects by connecting structural members to walls/rooms, structural systems to wings of buildings, etc. Topology can represent both the internal connectivities of objects (i.e., their vertices, edges, and faces) termed *internal topology,* and the connectivity of objects to other objects, termed *external topology.* External topology will be the meaning of topology throughout this paper.

Spatial Relations. *Spatial relations* provide a mechanism for describing relative positions of objects which are not necessarily connected. These spatial relations may assign descriptions such as "East_of\ "North_of\ "Below", "Above", etc. between design objects.

Geometry. In addition to topology and spatial relations, there are *geometric attributes* of objects which consist of numeric properties such as the object's dimensions, location, and orientation in space.

Spatial Attributes. Topology, spatial relations, and geometry are collectively referred to as the ***spatial attributes*** **of a design object.**

Configuration. The purely spatial aggregation of design symbols forms a *configuration.*

Syntax. In natural language, the syntax of a sentence is defined by transformation rules indicating what words or word phases can be adjacent (or connected) to others. Using individual words, phrases and sentences are developed which satisfy a given grammar. Topology is the connectivity relation defining legal assemblages of design objects. Geometry and spatial relations further define legal shapes and combinations of design objects. In this manner, *spatial grammars* dealing with spatial attributes are similar to syntactic grammmars for natural languages.

Semantics. Semantics provide the ability to understand the function (i.e., "meaning" and/or ''purpose") of the symbols used in a grammar. In the current context, two types of semantics are present: structural and architectural. These two types can be described by illustrating the different semantics of an object "shear wall". *Structural semantics* define its function as providing lateral and gravity load-carrying capabilities. In order to perform its structural function, the shear wall occupies a vertical plane. The *architectural semantics* denote the architectural function of a wall providing a separation between spaces in a vertical plane. In this case, the separation may also provide a noise barrier between adjacent spaces. The presence of structural and architectural semantics (described as context below) provides meaning for the interaction of structural symbols in relation to the architectural symbols. For example, a shear wall needed to resist structural loads should not be placed through the middle of a room as it inhibits the movement of people.

Functional Attributes. Together, the architectural semantics and the structural semantics compose *the functional attributes* of design objects. The ability to represent and match (search) both functional and spatial attributes of design objects in *semantic grammars* is the first step toward spatial reasoning.

Design. The assignment of semantics to a syntactically correct configuration yields a *design*, either an architectural layout or a structural system.

Context. The *context* describes a current situation by relating other facts to a current fact. A significant portion of the context consists of *constraints.* Constraints bind relations between facts and other facts which either prohibit or force some action to occur as a result of the present context. In this study the context provides the ability to apply structural and architectural functions of a structural system or architectural layout to its symbols, in much the same way that words and phrases have different meanings in various sentences. For an example of structural context, the existence of a structural system in the x-direction can impose constraints on a structural system in the y-direction which would not occur if only the y-direction system were present. Architectural constraints exist when a room requires another room to be present (i.e., the presence of a dining room requires the presence of a kitchen). The architectural purpose of a room can also impose interaction constraints on a structural system, e.g., an auditorium requires a column-free space. These constraints have been previously represented along with the topology in the knowledge base, or embedded in the control structure for the application. ALL-RISE [Sriram 86] begins to look at the structural constraints by using interaction constraints between structural subsystems. However, the interaction constraints do not deal with spatial attributes of objects.

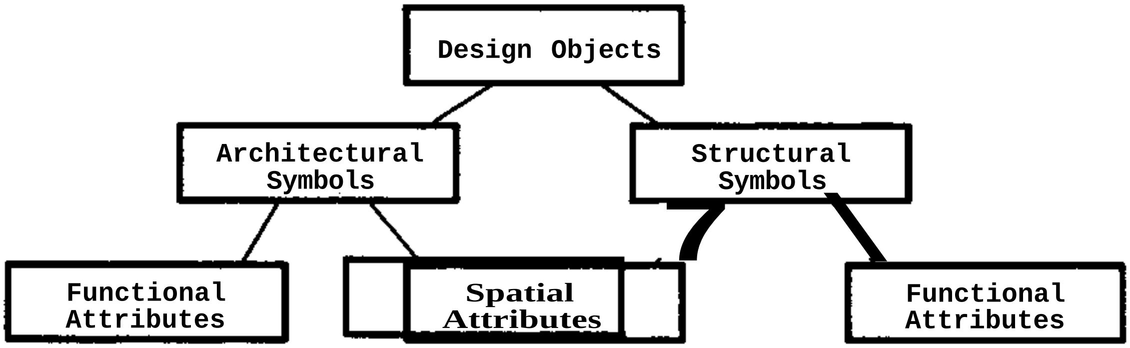


Figure 3-1: Design Object Description

The architectural and structural symbols used in building design can be visualized as shown in Figure 3-1. The information consists of the spatial (syntactic) and functional (semantic) descriptions of the appropriate design objects. However, any object may have information in both sides of figure. A wall can represent a noise barrier (function of an architectural symbol) as well as provide lateral load resistance (function of a structural symbol), but is one object with one set of spatial attributes. In other words, an object has one description of its spatial attributes, but retains separate, specific information about its semantics.

3.3- Uses of the Representation

The representation presented must support two types of operators (components of the design process), and provide common infonnation to them. These operators and their uses are discussed in this section.

33.1. **Generators**

A generator is a operator which executes a grammar in a forward direction. With the grammars considered in this paper, the generator takes design objects and combines them to produce designs. The following two generators are suggested.

Architectural Generator. The architectural generator uses architectural design objects to produce an architectural layout (both syntactically and semantically). The architectural generator may use a fixed structural configuration (syntax) and function (semantics) as input to develop a compatible architectural layout, spatially and functionally correct according to the architectural layout grammar while satisfying the configuration and function of the given structural system.

Structural Generator. The structural generator develops a structural configuration (syntax) and function (semantics) by combining structural design objects. The optional use of a fixed architectural syntax and semantics as input causes the structural generator to produce a compatible structural system.

3*3.2.* Critics

During the design process, it often becomes necessary to evaluate a candidate solution to see whether it is acceptable. This situation develops when a potential solution is generated by a different kind of generator (e.g., the structural generator for the architectural critic) or if the design process is initiated with a candidate solution. A critic is an operator for taking a candidate design and decomposing it by executing the grammar in reverse, to verify the presence of design objects (or absence of undefined objects). The following two critics are suggested.

Architectural Critic. The architectural critic evaluates a potential architectural layout by verifying that it satisfies the spatial and architectural function constraints required by the design. The architectural critic requires that the structural configuration and semantics be given and remain constant. When given a candidate architectural layout and the constant structural system, the critic determines the acceptance (or rejection) of the architectural layout. The rejection of a candidate occurs when the critic's grammar is not capable of reproducing the candidate. When the rejection occurs, the grammar can show the rules that caused the rejection, thus illustrating why the potential candidate failed.

Structural Critic. The structural critic evaluates a potential structural system candidate as a solution to a design situation. The critic uses a fixed set of constraints provided by the architectural layout (both spatial and functional), to test the acceptance of the structural solution. If the candidate does not satisfy the grammar, the grammar can indicate why the potential solution does not woik by showing the rules causing the rejection.

3.3.3. Operator Options

The design development of a major structure may occur in many ways. Traditionally, the architect prepares the architectural layout, culminating in floor plans and elevations of a proposed structure, and then the structural engineer provides a structural framing system to support the design loads, trying to "live" within the constraints imposed by the architect. However, the process could be reversed. Regardless of who the initiator of the design process is, there will be iterations between the architect and structural engineer until a solution is achieved which satisfies both parties. The proposed representation must allow a control mechanism to run the representation in any order and in any direction so that the individual operators can be used in all sequences.

The two generators can operate in either of two modes:

- exhaustively enumerate possible combinations of layout and function (pruned only by the context); or
- be controlled by a planner for more directed generation which reduces branching and is more efficient.

The second mode is the desired approach for developing acceptable solutions. Exhaustive enumeration comes "for free" as a product of the grammar, but there exist so many possible combinations that the process can be computationally prohibitive. Control of the process could be provided by a planning grammar as suggested by Coyne and Gero [Coyne 85b], or could use planning knowledge represented in some other format

The two critics discussed above can also operate in one of two modes:

- start with candidate solution and apply the transformations backwards; or
- start by generating solutions and see if the candidate is encountered.

The first process is the desired method of execution for the critic. This method could indicate when the suspect solution first strays from the grammar, and thus indicate a potential change in the candidate to make it acceptable. The second process is like that of the generators, except a matching process is perforated at the end of the generation to check for the candidate solution.

An overview of the two selected modes for the two operators is shown in Figure 3-2. This figure shows two types of global control: generator and critic. The generator control drives the grammar forward while the critic control drives the grammar backward. Both the generator and critic require certain pieces of information. For instance, to execute the structural generator, the external constraints and the structural transformations (described below) are required and the

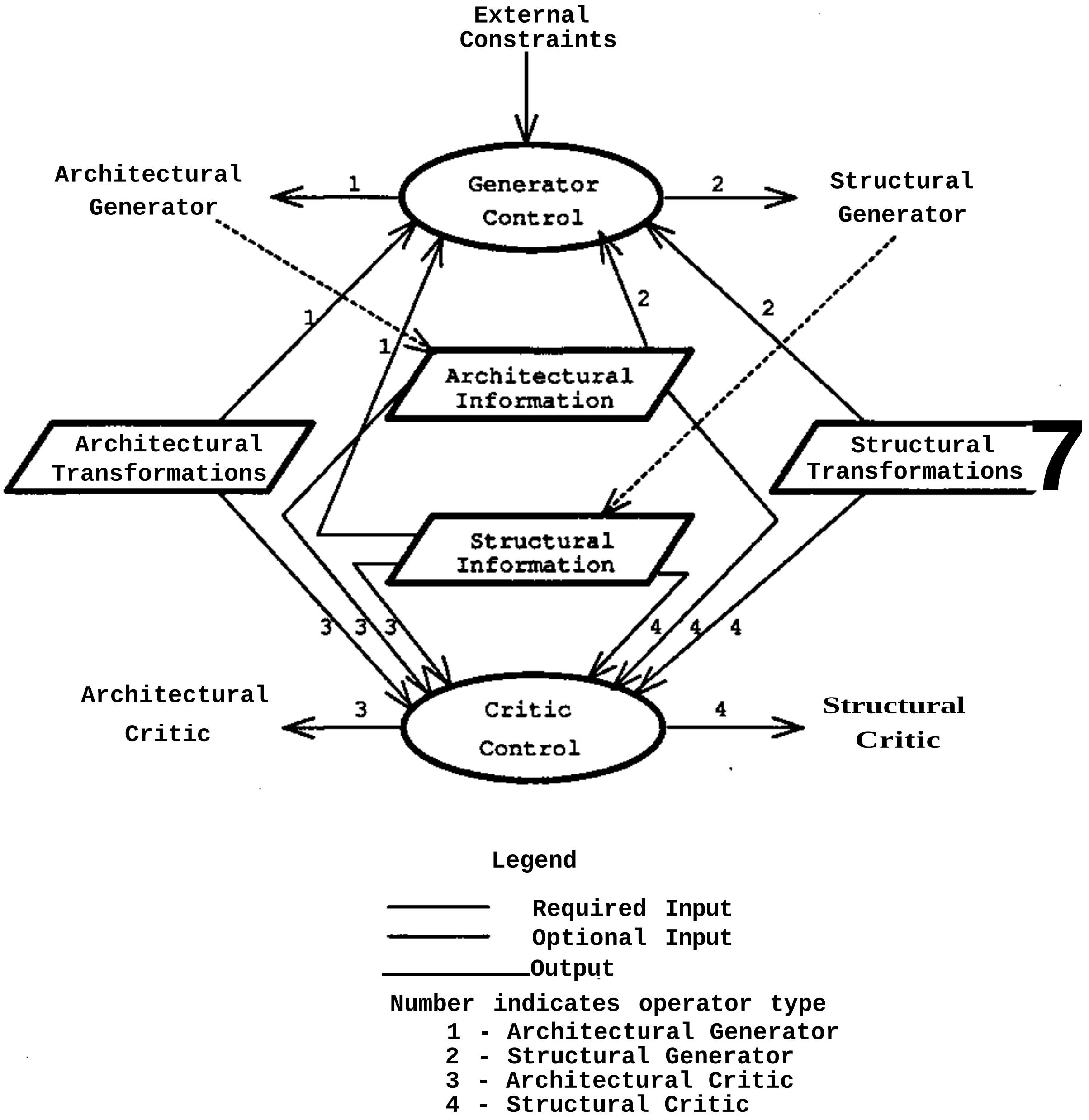


Figure 3-2: Operator Overview

architectural information (the architectural symbols shown in Figure 3-1) is optional. The execution of the structural generator produces structural information which can be used by other operators. Each of the critics uses the input to evaluate the information by testing for conformance with the corresponding set of transformations. The input associated with an operator is indicated by matching the number of the operator with the numbers adjacent to the arrows.

Structural Critic. The structural critic evaluates a potential structural system candidate as a solution to a design situation. The critic uses a fixed set of constraints provided by the architectural layout (both spatial and functional), to test the acceptance of the structural solution. If the candidate **does not** satisfy the grammar, the grammar can indicate why the potential solution does not work by showing the rules causing the rejection.

3*3.3. Operator Options

The design development of a major structure may occur in many ways. Traditionally, the architect prepares the architectural layout, culminating in floor plans and elevations of a proposed structure, and then the structural engineer provides a structural framing system to support the design loads, trying to "live" within the constraints imposed by the architect. However, the process could be reversed. Regardless of who the initiator of the design process is, there will be iterations between the architect and structural engineer until a solution is achieved which satisfies both parties. The proposed representation must allow a control mechanism to run the representation in any order and in any direction so that the individual operators can be used in all sequences.

The two generators can operate in either of two modes:

- exhaustively enumerate possible combinations of layout and function (pruned only by the context); or
- be controlled by a planner for more directed generation which reduces branching and is more efficient.

The second mode is the desired approach for developing acceptable solutions. Exhaustive enumeration comes "for free" as a product of the grammar, but there exist so many possible combinations that the process can be computationally prohibitive. Control of the process could be provided by a planning grammar as suggested by Coyne and Gero [Coyne 85b], or could use planning knowledge represented in some other format

The two critics discussed above can also operate in one of two modes:

- start with candidate solution and apply the transformations backwards; or
- start by generating solutions and see if the candidate is encountered.

The first process is the desired method of execution for the critic. This method could indicate when the suspect solution first strays from the grammar, and thus indicate a potential change in the candidate to make it acceptable. The second process is like that of the generators, except a matching process is performed at the end of the generation to check for the candidate solution.

An overview of the two selected modes for the two operators is shown in Figure 3-2. This figure shows two types of global control: generator and critic. The generator control drives the grammar forward while the critic control drives the grammar backward. Both the generator and critic require certain pieces of information. For instance, to execute the structural generator, the external constraints and the structural transformations (described below) are required and the mars. These need to be extended to context-sensitive grammars. Coyne and Gero have begun work in this area [Coyne 85c].

Three levels of design object interaction are proposed in this study. These levels will be used as implementation steps for the representation. These levels are:

- Using a shape grammar to generate syntactically correct configurations or critique configurations for syntactic correctness.
- Applying structural semantics to the structural symbols and architectural semantics to the architectural symbols, but neither critics and generators use interaction between the two symbol types.
- Applying interaction constraints between architectural and structural semantics to the design objects.

The study will formally define the structural and architectural syntax, semantics, and context of design objects for two domains: preliminary design of buildings and preliminary design of automobiles. By using the representation for preliminary design of buildings and automobiles, the flexibility of the representation will be demonstrated.

4 Illustrative Example

The ideas presented in the previous sections can be applied to any spatial problem as long as a grammar can be developed. The proposed grammar will deal not only with spatial information, but also with the semantics and context for the new domain.

To illustrate the use of the grammar formalism, an example of a potential application is presented. This example is for the placement of structural systems in a building.

As described previously, the grammar will separate the representation of the spatial attributes from the functional attributes of the structural system. Therefore, this example shows the beginnings of a grammar which will be further formalized during the study.

4.1. Spatial Grammar

The spatial attributes of the grammar consist of the topology, geometry, and spatial relations of the structural symbols composing the structure. The symbols start as abstract components (e.g., lines in space) forming bents, which form frames, and which when combined compose a *structural configuration.* A structural configuration is the syntactic representation of a structural system before any semantics about its load-carrying function are added. A simple grammar which develops the spatial attributes for a structural frame can be seen in Figure 4-1. In this example, V_T denotes the set of terminal symbols, V_M represents the set of markers, R defines the transformations, and I is an initial symbol for starting the process. In each transformation, the non-

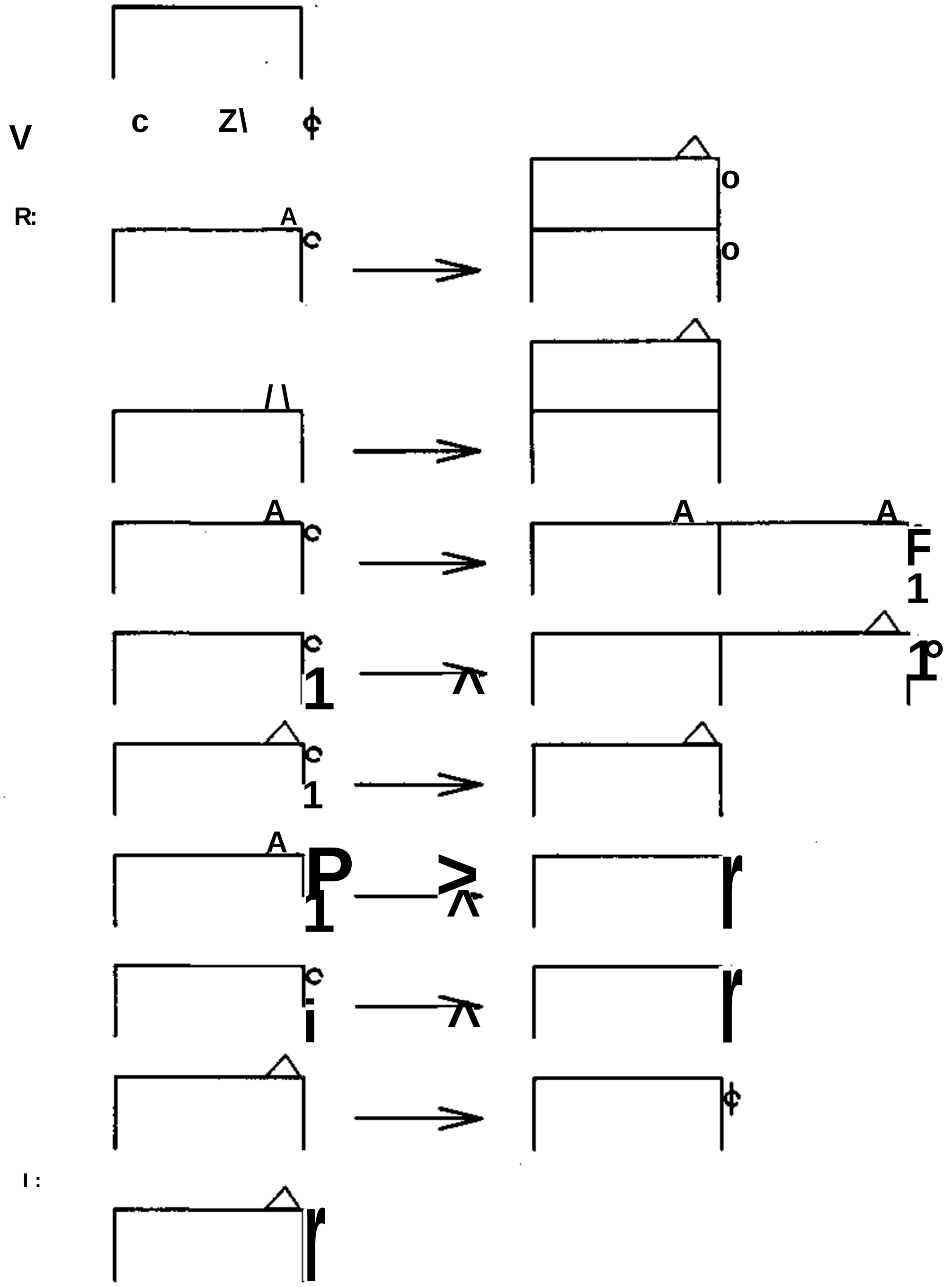


Figure 4-1: Structural Frame Spatial Grammar

terminal symbols on the left hand side are replaced by the symbols on the right hand side. The

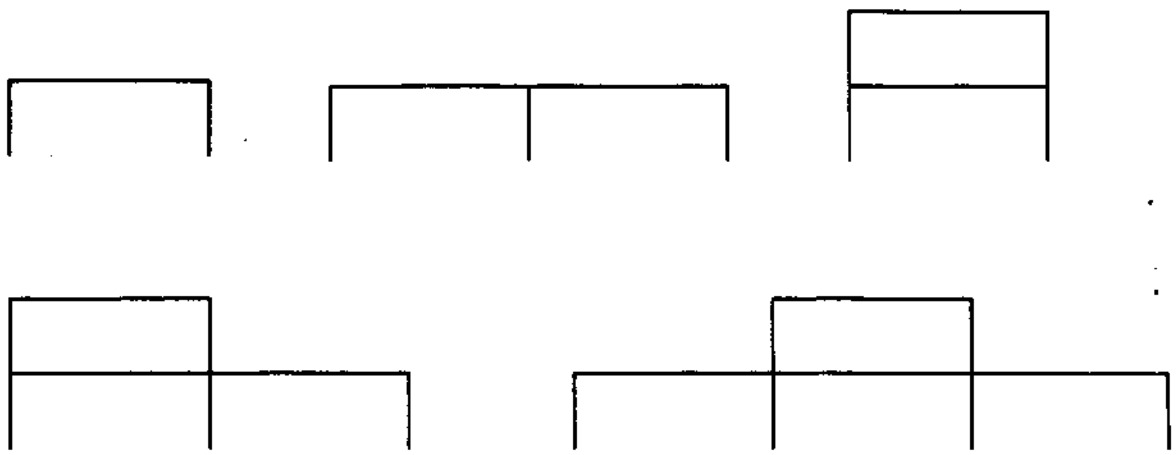


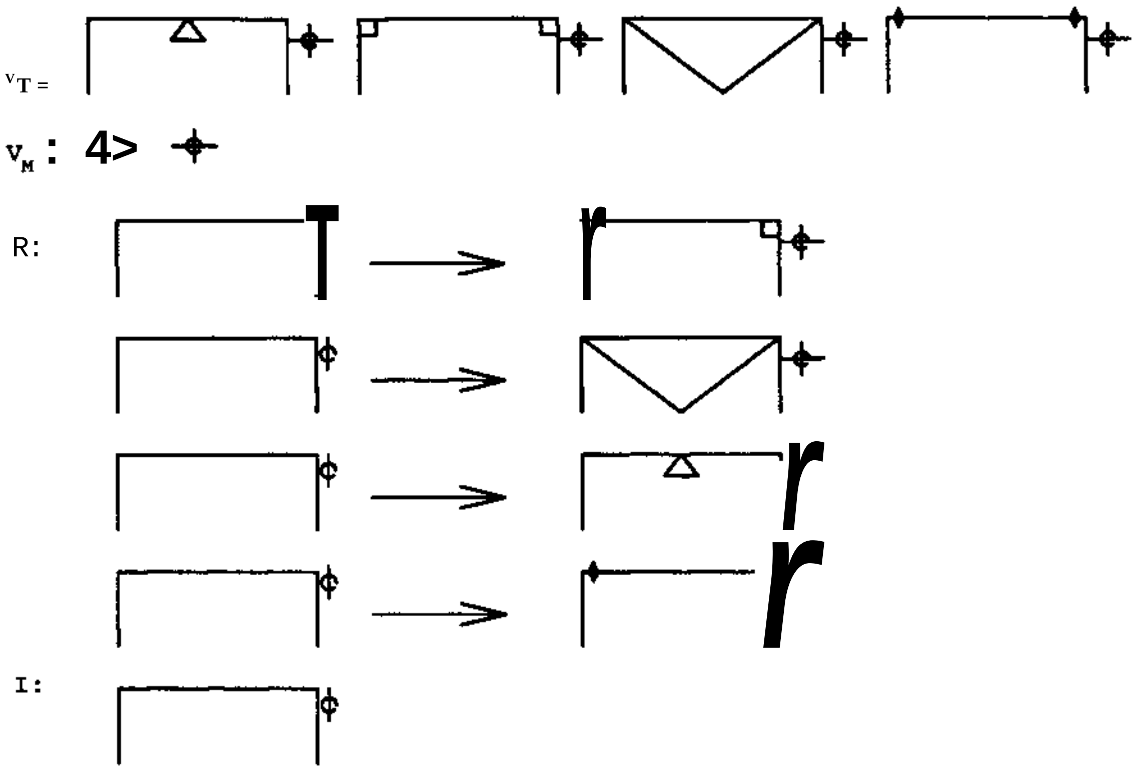
Figure 4-2: Structural Frame Spatial Language

markers used in this example provide for horizontal (circle marker) and vertical (triangle marker) growth of the structural configuration. After a syntactically correct configuration has been generated, the bents are marked with the slashed circle marker (last two rules) so that the semantics can assign structural function to the configuration. The spatial language that this grammar defines is partially represented in Figure 4-2. This language is already a departure from previous representations in HI-RISE and ALL-RISE in that it allows for varying number of stories in adjacent bays.

4.2. Functional Grammar

Using the spatial attributes generated above, a semantic grammar can give functional meaning to the structural frame. An example of structural semantics is *the frame type,* which describes its load carrying function and composition. Examples are: a braced frame, rigid frame, shear wall, etc. Figure 4-3 shows a grammar which generates frame type given the spatial attributes. Each of the rules allows a bent, described only by its spatial attributes, to acquire structural properties of a certain frame type. For example, the rule which transforms the bent into a rigid frame assigns rigid connections between the girder and its columns. The doubly slashed circle marker indicates that the bent has acquired structural semantics.

The combination of these frame types form subsystems which are dependent on the context, as illustrated in Figure 4-4. The two rules shown in Figure 4-4 allow a frame type to grow vertically dependent upon the presence of a frame type beneath. The first rule allows any frame type to be extended in the vertical direction without change. The second rule shows one possibility for changing frame types in the vertical direction: a simple framing type can be developed in the bent



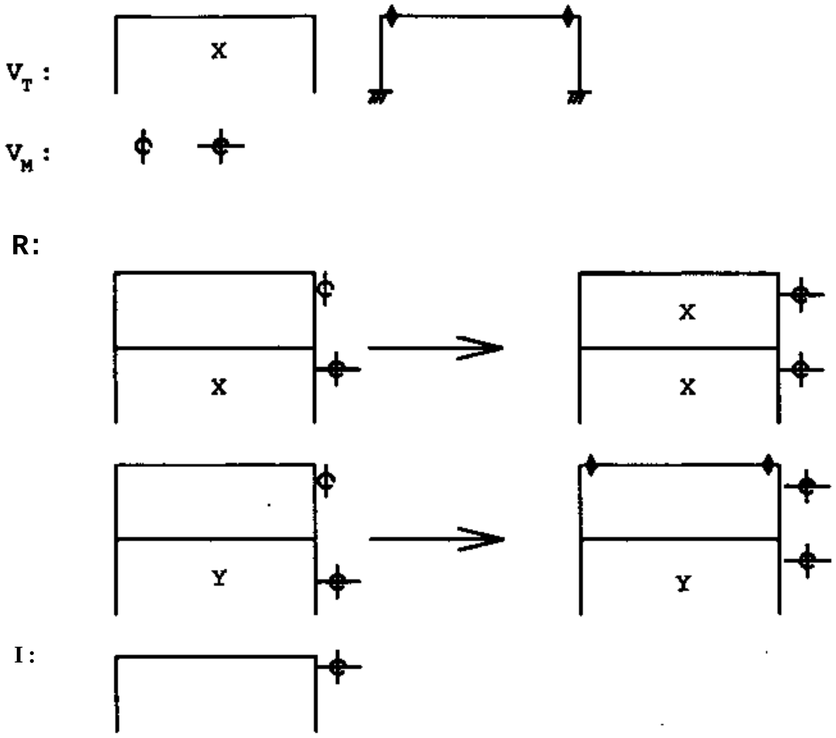
Where represents a rigid frame

Where [ZS represents a shear wall

Where represents a simple frame

Figure 4-3: Structural Frame Functional Grammar

as long as the bent starts with a type in which columns are fixed at their base. Other rules will be developed providing different combinations of frame types in a structural subsystem.



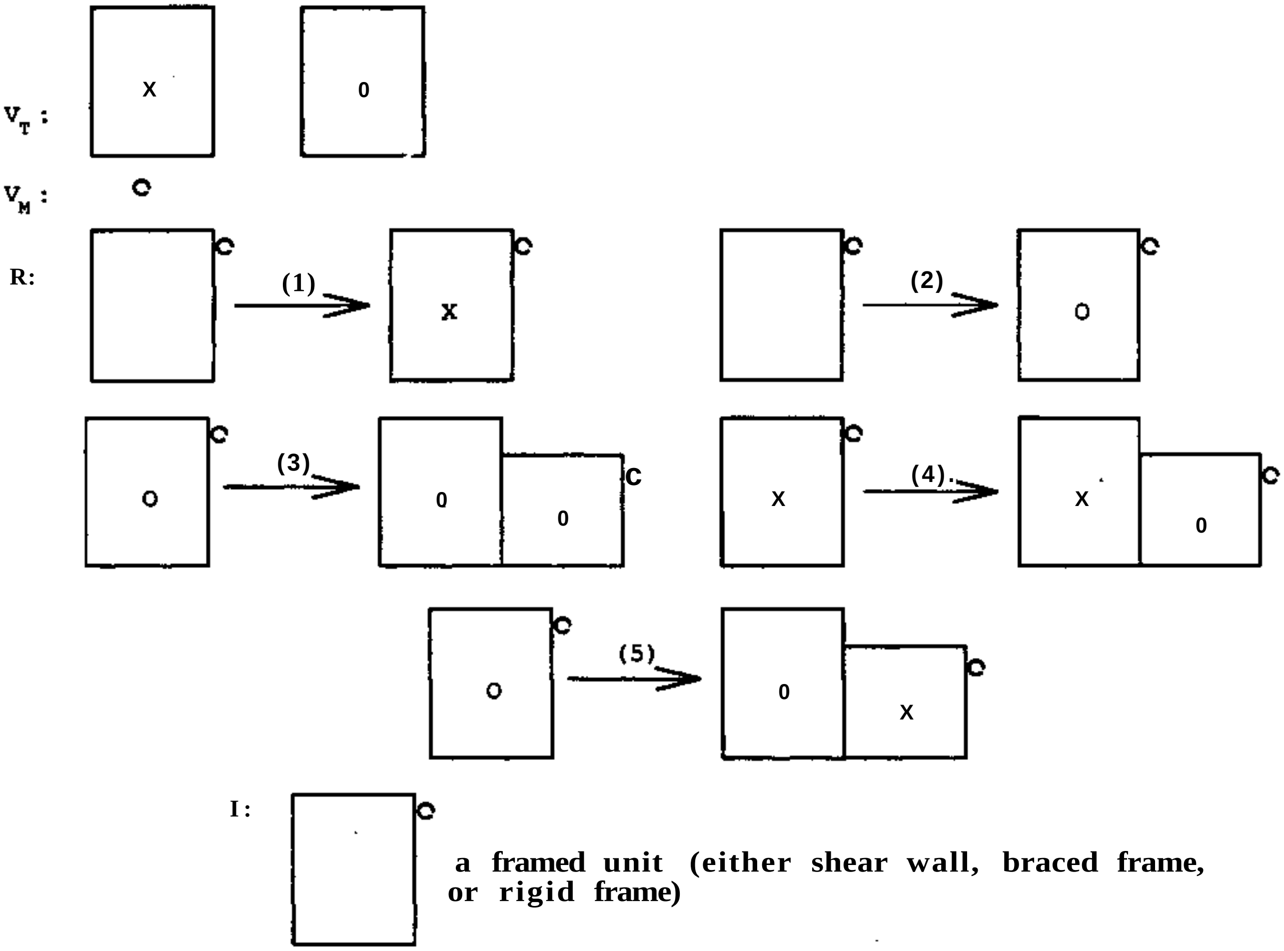
Where  represents a hinged connection

Where x represents a shear wall , braced frame, rigid frame ,or simple frame with continous (fixed) columns

Where Y represents a braced frame or rigid frame

Figure 4-4: Context Usage for Framing Development

The subsystems combine to form structural systems as shown in Figure 4-5. A given subsystem (a vertical structural unit of the same frame type) can be combined with like subsystems, or mixed with other types. The first and second rule (rules are denoted by the numbers over the arrow) characterize a vertical unit as a specific framing type; they can be viewed as high-level representations of the grammars of Figures 4-3 and 4-4. The third rule allows rigid frame units to be combined together while the fourth and fifth rules allow shear walls and braced frames to be combined with rigid frames. The rules shown in this figure do not reflect the requirements imposed by the functions of adjacent spaces. Also not shown are the termination rules which would remove the markers.



a framed unit (either shear wall, braced frame, or rigid frame)

Where O represents a rigid frame
Where X represents a shear wall or braced frame

Figure 4-5: Grammar forming Structural Systems

5 Summary

According to Buhl, there are two parts to a problem solving process [Buhl 60]. First the problem itself must be broken down and studied. Secondly, the material out of which a solution must come should be defined. This statement supports the two primary purposes of this approach: structural and architectural preliminary design must be studied to identify their elements; and the design objects must be defined allowing a representation for creating a design to be developed. The formalism of grammars is expected to provide a method for organizing the solution space, and their generative capability to produce a set of possible solutions.

Because of these capabilities, the use of a shape grammar is used for representing architectural and structural design. The use of grammars can be supported by recent work in cognitive science. Stemberg says that planning strategies differ between experts and novices [Sternberg 86]. Experts seem to use a "working forward" strategy, whereas novices seem to use a "working backward" strategy. In the study he describes, experts went from the givens and generated equations that could be solved from them, while novices started with an equation containing the unknown of the problem and successively tried equations, until one was found that worked. Grammars provide the "forward" strategy via formal transformations defined in the grammar. Sternberg also says that experts and novices represent the problem differently. Novices tend to represent the problem based around dominant objects, while experts organize based on fundamental principles. Again, grammars provide a formalism to represent the fundamental principles in design so that objects can be created.

The following considerations justify the chosen approach.

First, it is evident that syntactic, semantic, and contextual knowledge about the design objects must be represented. This knowledge is symbolic and some of it is highly heuristic in nature. Therefore, the use of artificial intelligence techniques arc applicable.

Second, the major design objective is to generate feasible alternatives. Therefore, production systems as defined in knowledge-based techniques are an applicable approach. As the discussion of grammars shows, they have close ties with production environments and thus present an attractive approach.

Lastly, it is clear that shape grammars can be used to generate spatial and functional designs using design objects. This facility has to be expanded by the use of design semantics to assign functions to the design objects, and of context to control the allowable transformations.

From the above discussion, it is clear that structural design can be represented as a grammar defining the way existing information can be combined and transformed to generate solutions for preliminary structural design.

References

[Akiner 86] Akiner, V. Tuncer, "Knowledge Engineering in Object and Space Modeling," *Expert Systems in Civil Engineering,* Kostem, C.N., and Maher, M.L., Eds., Symposium sponsored by the Techinical Council on Computer Practices of ASCE, Seattle, WA, April 8-9, 1986, American Society of Civil Engineers (ASCE), 1986.

[ASCE 86] ASCE, "Skyscrapers Predicted as Fewer, Better," *Civil Engineering,* Vol. 56, No. 3, pp. 15, March 1986.

[Brown 85]	Brown, D. and Chandrasekaran, B., "Expert Systems for a Class of Mechanical Design Activity," *Knowledge Engineering in Computer-Aided Design,* Gero, J.S., Ed., IFIP WG 5.2 Working Conference on Knowledge Engineering in Computer-Aided Design, Budapest, Hungary, September 17-19,1984, International Federation for Information Processing (IFIP), North-Holland Publishing Company, pp. 259-290,1985.
[Buhl 60]	Buhl, Harold R., *Creative Engineering Design,* The Iowa State University Press, 1960.
[Chamiak85]	Chamiak, Eugene, and McDermott, Drew, *Introduction to Artificial Intelligence,* Addison-Wesley Publishing Company, 1985.
[Coyne 85a]	Coyne, R.D., and Gero, IS., "Design Knowledge and Sequential Plans," *Environment and Planning B,* Vol. 12, No. 4, pp. 401-418,1985.
[Coyne 85b]	Coyne, R.D., and Gero, J.S., *Semantics and the Organisation of Knowledge in Design,* Working Paper, Computer Applications Research Unit, Department of Architectural Science, University of Sydney, September 1985.
[Coyne 85c]	Coyne, R.D., and Gero, J.S., "Design Knowledge and Context," *Environment and Planning B,* Vol. 12, No. 4, pp. 419-442,1985.
[Dixon85]	Dixon, J.R., et al., "Expert Systems for Mechanical Design: Examples of Symbolic Representations of Design Geometries," *Applications of Knowledge-Based Systems to Engineering Analysis and Design,* C.L. Dym, Ed., The Winter Annual Meeting of the American Society of Mechanical Engineers, Miami Beach, FL, American Society of Mechanical Engineers, pp. 29-46,1985.
[Fenves 76]	Fenves, S.J., and Liadis, S.A., "A Data Structure for Computer-Aided Design of Buildings," *APEC Journal,* pp. 14-18, Fall 1976.
[Fenves 83]	Fenves, S.J., "Computers in the Future of Structural Engineering," *Proceedings of the Eighth Conference on Electronic Computation,* Nelson, Jr., James K., Ed., American Society of Civil Engineers (ASCE), pp. 1-8, February 21-23,1983.
[Hemming 81]	Hemming, U., "The Secret of the Casa Giuliani Frigerio," *Environment and Planning B,* Vol. 8, No. 1, pp. 87-96,1981.
[Gero 83]	Gero, J.S., Akiner, V.T., and Radfoixi, A.D., "What's What and What's Where: Knowledge Engineering in the Representation of Buildings By Computer," *PARC 83,* Online Publications, pp. 205-215,1983.
[Gero 85]	Gero, J.S., Radford, A.D., Coyne, R., and Neiner, V.T., "Knowledge-Based Computer-Aided Architectural Design," *Knowledge Engineering in Computer-Aided Design,* Gero, J.S., Ed., IFIP WG 5.2 Working Conference on Knowledge Engineering in Computer-Aided Design, Budapest, Hungary, September 17-19, 1984, International Federation for Information Processing (IFIP), North-Holland Publishing Company, pp. 57-81,1985.
[Gips 74]	Gips, James, *Shape Grammers and their Uses,* unpublished Ph.D. Dissertation, Stanford University, March 1974.
[Gips 79]	Gips, James, "Artificial Intelligence," *Environment and Planning B,* Vol. 6, No. 3, pp. 353-364,1979.

[Gips 80] Gips, J., and Stiny, G., *'Production Systems and Grammars: A Uniform Characterization," *Environment and Planning B,* Vol. 7, No. 4, pp. 399-408, 1980.

[Kim 84] Kim, Jim H., McDermott, John, and Siewiorek, Daniel P., "Exploiting Domain Knowledge in IC Cell Layout," *IEEE Design and Test,* Institute for Electrical and Electronics Engineers (IEEE), pp. 52-64, August 1984.

[Knight 81] Knight, T. Weissman., "Languages of Designs: from Known to New," *Environment and Planning B,* Vol. 8, No. 2, pp. 213-238,1981.

[Koning 81] Koning, H., and Eizenberg, J., "The Language of the Prarie: Frank Lloyd Wright's Prarie Houses," *Environment and Planning B,* Vol.8, No. 3, pp. 295-323,1981.

[Maher85] Maher, M.L., *HI-RISE: A Knowledge-Based Expert System For The Preliminary Structural Design of High Rise Buildings,* Technical Report R-85-146, Department of Civil Engineering, Carnegie-Mellon University, Pittsburgh, PA, January 1985.

[March 85] March, L., and Stiny, G., "Spatial Systems in Architecture and Design: Some History and Logic," *Environment and Planning B,* Vol. 12, No. 1, pp. 31-53, 1985.

[Moore 85] Moore, Robert L., "Expert Systems for Process Control," *TAPPI Journal,* Vol. 68, No. 6, pp. 64-67, June 1985.

[Smallowitz 86] Smallowitz, Howard, "Computer Cuts Tower Steel," *Civil Engineering,* American Society of Civil Engineers (ASCE), Vol. 56, No. 3, pp. 72-74, March 1986.

[Sriram 86] Sriram, D., *Knowledge-Based Approaches for Structural Design,* unpublished Ph.D. Dissertation, Department of Civil Engineering, Carnegie-Mellon University, February 1986.

[Stefik 82] Stefik, M., et al., "The Organization of Expert Systems, A Tutorial," *Artificial Intelligence,* Vol. 18, No. 2, pp. 135-173, March 1982.

[Steinberg 86] Sternberg, Robert J., "Inside Intelligence," *American Scientist,* Vol. 74, No. 2, pp. 137-143, March-April 1986.

[Stiny 75] Stiny, George, *Pictorial and Formal Aspects of Shape and Shape Grammars,* Biikhauser Verlag Basel, 1975.

[Stiny 80] Stiny, G., and Mitchell, W.J., "The Grammar of Paradise: on the Generation of Mughul Gardens," *Environment and Planning B,* Vol.7, No.2, pp. 209-226,1980.

[Zumsteg 85] Zumsteg, J.R., and Haggs, D.L., "Knowledge-Based Analysis and Design Systems for Aerospace Structures," *Applications of Knowledge-Based Systems to Engineering Analysis and Design,* C.L. Dym, Ed., The Winter Annual Meeting of the American Society of Mechanical Engineers, Miami Beach, FL, American Society of Mechanical Engineers, pp. 67-80,1985.