12
Architectural Knowledge Structures as "Design Shells": A Knowledge-Based View of Design and CAAD Education

Rivka Oxman

Faculty of Architecture and Town Planning
Technion: Israel Institute of Technology

The concept of a knowledge based design shell is proposed as a basis for teaching design. The significance of the concept of design shell is discussed with respect to formalization, implementation, application and operation. GPRS-a generative prototype refinement design shell-is proposed, defined and elaborated. A plan type is introduced as one significant kind of structure of knowledge in architectural design is introduced. A method for representing syntactic and the semantic content to he used in design refinement is proposed. The method exploits the characteristics of both rules and frames, and integrates them in a prototype-based design system. This is demonstrated in a system called PRODS. Finally, the significance of such an approach in teaching is discussed.

Knowledge-Based Design Shells as a Vehicle in Design and CAAD Teaching

Teaching computer-aided design should be coupled with teaching design. It has been shown that teaching programming can be mapped directly into certain types of architectural knowledge (Mitchell, Liggett and Kvan, 1987). The *Topdown* system has demonstrated its use in design teaching (Mitchell, Liggett and Tan, 1989) by providing programming tools to encode the rules of a kind of parametric shape grammar. Emerging tools from the field of artificial intelligence, such as expert system shells, may serve as a step forward in the adaptation of this approach to CAAD and design education. The promising idea of a shell can be further developed in order to accommodate the characteristics of knowledge in design. A design shell would be an extension of the traditional concept of a shell which could be adapted to the structure of a particular body of design knowledge. Knowledge-based design should provide means for encoding architectural knowledge as pre-defined structures which are delivered to the CAAD user by means of design shells.

The purpose of this paper is to develop the concept of empty design shells, to present a generative shell for prototype refinement in design, and to describe its potential in teaching and research on the following four levels:

-Teaching about structures of design knowledge
-Teaching about implementing these structures as empty design shells
-Developing specific domain applications for these structures
-exploring different designs with these applications.

GPRS: A Generative Prototype Refinement Shell

Design knowledge should be available in a form which utilizes its generic characteristics. The use of terms, such as schema, and type in the design professions suggests the cognitive value of such generic forms of knowledge. One such form of generic structures is the design prototype. Such prototypes encode knowledge of the salient features of a generic solution type. With respect to knowledge-based design systems, the use of the prototype as a knowledge structure, provides a basis for the organization of design knowledge. A prototype may be defined (Gero, 1987) as parametrized design description generator, a set of interpretations, a set of design elements which are the basis for design description and knowledge relating design description and its interpretations. A prototype is selected and specialized to the task at hand by providing values to parameters. The significance of such generic knowledge structures in design can be seen in the possibility of multiple derivations from high-level abstractions of the prototype.

GPRS-A generative prototype refinement shell proposed by the author, provides a structure for refinement processes by representing formal knowledge of a generative structure, its generic vocabulary, and syntactic operations, as well as their meanings, and a design description generator, all within a single schema. It is based on the idea that the shape grammar formalism (Stiny, 1980) can form the basis for a design description generator within a prototype. It contains knowledge of pre-defined rules which can be applied for refinement of designs in a top-down fashion. There are various examples (Stiny and Mitchell, 1978), (Flemming,1981) which treat specific designs, or a style, as a knowledge structure with generative capabilities. The distinction between GPRS representation and shape grammar representation is that the latter supports formal refinements in which goals and constraints are embedded implicitly within the syntactical rules while the generative prototype provides ways to represent and use explicit constraints and goals in a reasoning process for guiding the refinement. It deals with formal knowledge in which the design generator is controlled by the interpretations of its refinement steps.

In the following section the structure of the generative prototype refinement shell will be described in relation to the following four issues:

-The formalization of design structure
-Implementation of the design shell
-Development of applications for the shells
-Providing interfaces for the applications.

Knowledge Formalization for GPRS

An assumption of this work is that designers have the ability to typify both situation and solution types. Typification is the recognition of characteristic and recurrent types of constraints in a design domain. Design can be seen as a matching process relating typified design situations and design solutions. A design situation is given by a set of conditions and requirements which are then typified as overriding goals and constraints. olution refinement can be seen as a process of matching typified design conditions to design steps of formal generation and refinement. The formalization of knowledge in GPRS, which is described below, reflects these assumptions.

Figure 1 indicates the successive stages of a refinement process. At each refinement stage conditions are typified and provide knowledge for instantiations of attributes of a subtype at a particular level of refinement. Various types of knowledge are employed in a generative prototype refinement process. GPRS represents a way of structuring all these types of knowledge in a single schema. These may be defined as follows:

a. *Typological knowledge of the prototype, subtype and their properties:* typological knowledge is the knowledge of the hierarchical relationship between the prototype and its subtype.
b. *Topological knowledge of the type:* topological relations between components of the type. Each component can have its own typological hierarchy.
c. Interpretive knowledge of design constraints: knowledge of constraints that effect decisions about refinement steps. These are typified constraints that capture experiential knowledge related to meanings in certain situations.
d. *Generative knowledge of the type:* knowledge about successive steps in a refinement process. Each stage of refinement causes instantiation of a subtype in the typological hierarchy. The refinement process is executed by operations which produce parametrized design descriptions in the subtype.

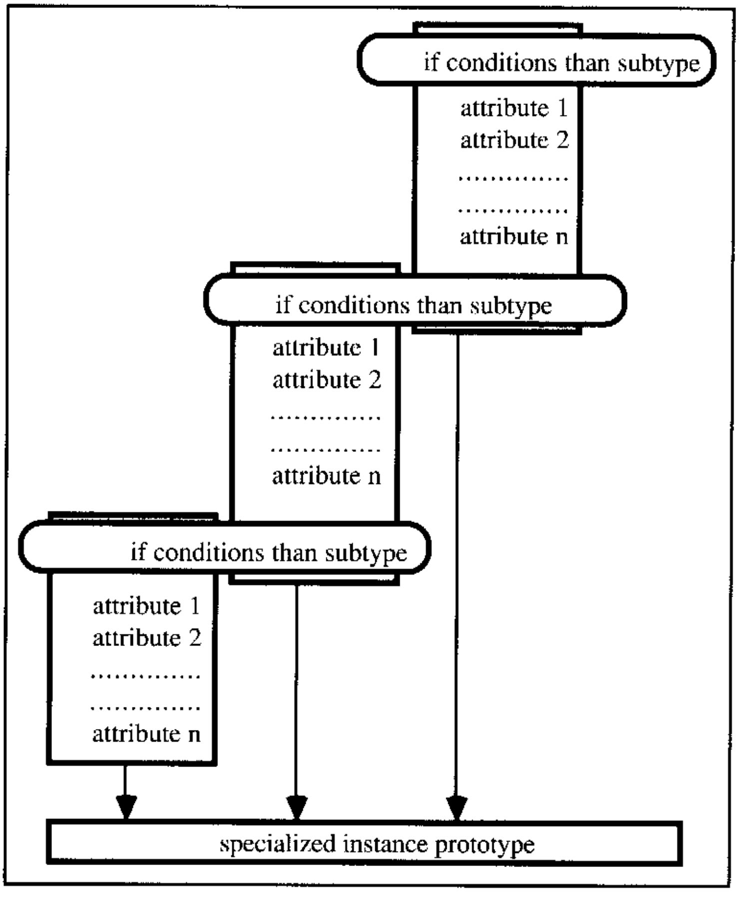

Figure 1 Description of refinement and instantiation process of instance-type

implementation Level of GPRS

Various approaches to encoding knowledge have been proposed by artificial intelligence researchers (Minsky,1975; Schank and Abelson,1977). They have developed structures such as frames and scripts. GPRS is a holistic structure capable of containing various types of knowledge. It provide means to generate solutions by reasoning from relevant knowledge bases in order to activate design operators for producing design descriptions. Representational methods were chosen according to the following criteria:

-Suitability for rendering the knowledge explicit
-Ease of encoding
-Ease of maintenance

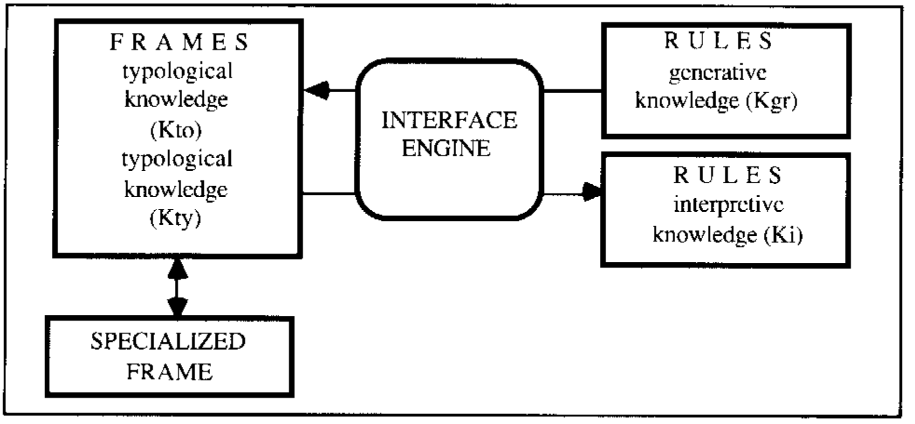

Figure 2 Refinement and instantiation process of instance-type using rules and frames in GPRS.

A hybrid representation which combined different methods according to their suitability to specific types of knowledge was selected. A special hybrid representation was developed which combined the advantages of frames as structured representations for objects or classes of objects production rules as a formalism for describing generative patterns (Gips and Stiny, 1980) and a rule-based formalism for describing causal heuristic knowledge. Figure 2 demonstrates a diagram of the different kinds of knowledge represented as rules and frames in the proposed schema. An interface module supports the flow of knowledge between rules and frames. The following section elaborates the representational method which was selected for each of the types of knowledge and the manner in which it operates as a design shell.

Rules Represent Generative Knowledge
Generative knowledge of a prototype =can be represented by production rules. A rule-chaining mechanism such as a shape grammar can provide a medium for successive refinement. This process can be guided and controlled by interpretive knowledge of the refinement stages.

Frames Represent Typological and Topological Knowledge
Typological knowledge and topological knowledge of a prototype are handled by the frame in its traditional use. The frame provides a scheme which enables a generic description to support hierarchical inheritance of attributes in the process of typological refinement. It describes its attributes and passes them, through the inheritance mechanism, to its subtypes. It also supports the notion of specifying default values in the frame. Thus, it is unnecessary to

specify all the information at each level. We assume that an instance inherits the properties of its superclass unless specified otherwise. Frames enables us to use *abstraction levels* in representation of attributes.

Rules Embedded in Frames Represent Interpretive Knowledge

Most systems dealing with objects obtain their values by means of procedures which specify the attributes. Usually a system must perform some diagnostic tasks in order to accomplish this. In a prototypical design situation, certain heuristic knowledge is associated with the situation. The system described here may consult relevant experiential knowledge in the form of situation-conclusion pairs in order to establish the instance attributes. The basic advantage of combining rules and frames in this case is that the frame structure provides explicit representation of the context in which production rules can be used for reasoning. In this way, each prototype represented in the frame formalism can direct the use of a set of rules to a specialized sub-class.

Frames Supply Generative and Interpretive Knowledge

Frames are used in a way that provides descriptions of the objects being reasoned about by the rules (Rosenman, Manage and Gero, 1986), in the context of a generative refinement process. They are used to encode a model of the prototype described by the generative rules. They provide a structure storing and supplying information associated with the abstract concepts reasoned about the rules. Interpretive knowledge is associated with the frame slots in a way that enables the frame to link relevant supplementary knowledge bases to its attributes. These knowledge bases, which are invoked by the frame, employ a rule mechanism and provide information back to the frame structure. This mechaninism operates with the generative model which specifies stages of refinement at different levels of abstraction in a top-down fashion, Each hierarchical level is then described by a sub-type frame. Each sub-type frame contains and invokes a specific knowledge base which is relevant to that particular refinement stage. At any step, the instance type contains the specific values of its attributes which are added at each stage.

System Operation of GPRS

GPRS uses a system controller in the form of an expert system rather than an object-oriented programming environment. BUILD-an existing rule-based expert system (Rosenman, 1986)-was used. It includes a frame system for representing the prototypes, and an interface engine which controls all system interactions. It passes control between the rule-based and the frame-based inference engines and provides communication between rule and frame representations.

Application Level of GPRS: A Row House Plan Type

A design prototype, such as an architectural plan type, can be formulated by establishing stages of refinement. These start with high level concepts, each of which provides associated refinement decisions which lead to formal operations.

Generative Know/edge of a Row-House

The generative knowledge of a row-house plan type is represented by rules that indicate the order in which decisions are taken in a typical design refinement process. This application is based on an existing typology (Sherwood, 1978). The first classification level of a row-house type is based on decisions about its number of floors. This results in a formal decision about the row house section. The next level deals with the zoning of the plan. There are two options for making a decision about zoning. These are defined as: exterior-interior-exterior zoning, and exterior-exterior zoning. The first type of zoning, does not provide an outside orientation for the interior zone. The functional implication of such a zoning diagram is that the wet functions, such as toilets, are usually located in that zone. In the second type of zoning, the exterior-exterior type, the two zones each have an outside access. The next decision level is built about vertical and horizontal circulation, followed by kitchen core location, etc. Figure 3 demonstrates a generative representation of a row-house plan type in GPRS.

Typological and Topological Knowledge of a Row-House

Typological knowledge (knowledge of the salient characteristics of the type organized hierarchically into types and sub-types) and topological knowledge (knowledge of the relationships between the components of the type) of the row-house are described by the frame. For example, the class Row House is a superclass of Double_Storey_Row_House and Double_Storey_Row_House_1 is an instance within the class. The attributes such as number of floors or the basic orientation of the walls etc, have been specified for the class, while no information about the instance has been specified directly. In this way, each level contains only the relevant attributes of that level.

Interpretive Knowledge of Row Houses

The basic advantage of combining rules and frames in this case is that the frames provide a generic model of the row-house as well as the knowledge by which interpretations for formal decisions are made. For example, A row house prototype, contains relevant rules for decisions about number of floors, zoning

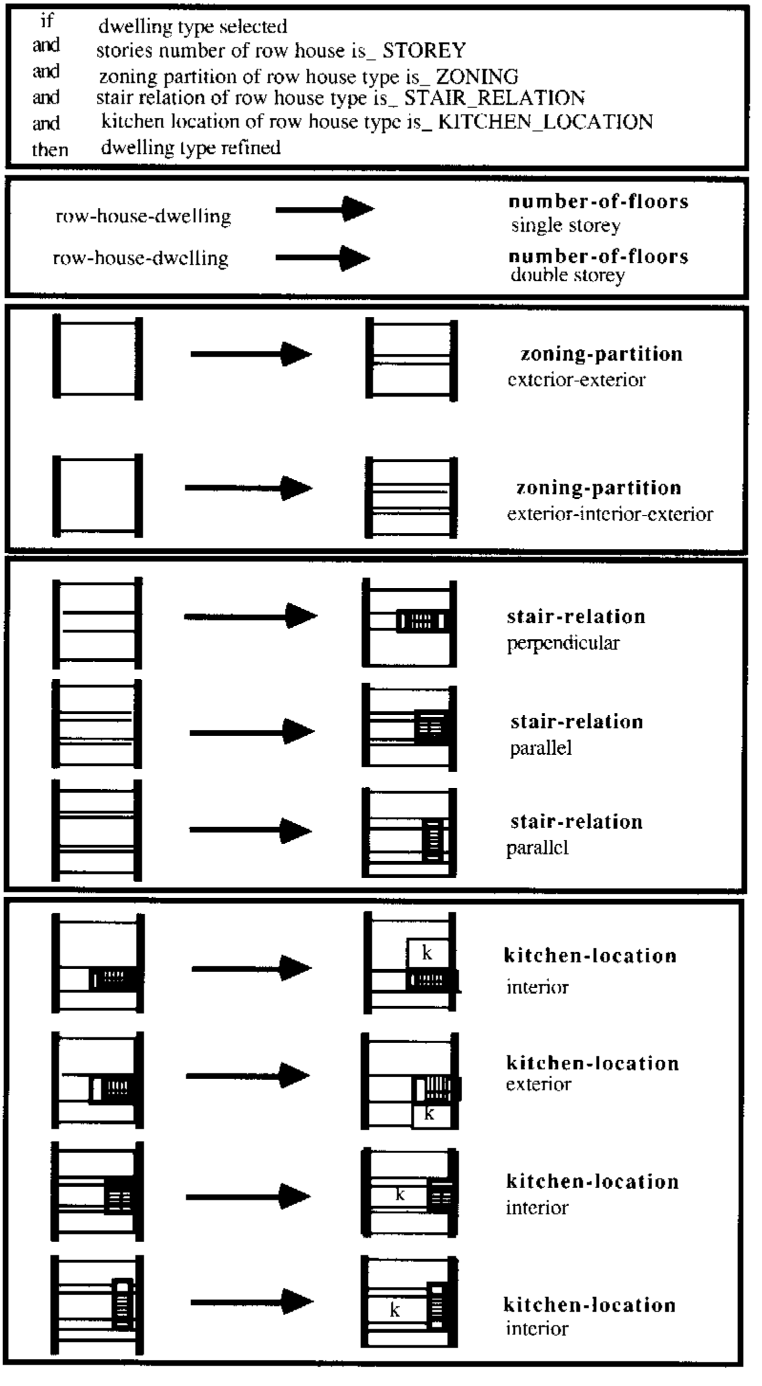

Figure 3 A generative representation of a row house plan type in GPRS

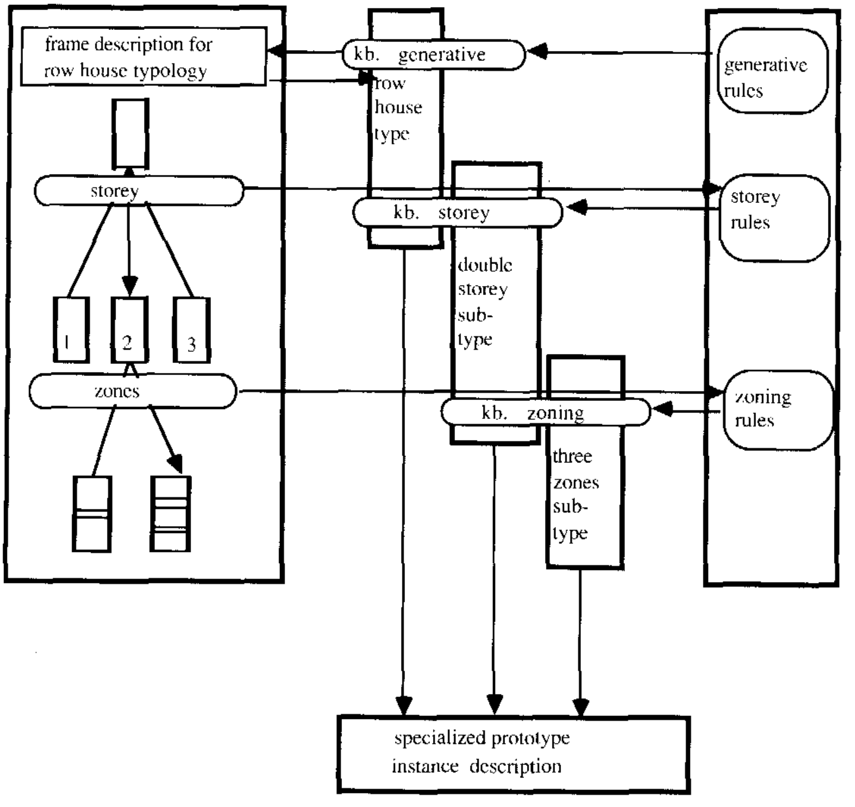

Figure 4 Knowledge flow between rule bases and frame bases in GPRS

partitioning, stair location, kitchen location, etc. These knowledge bases are activated by the 'if needed mechanism which is provided by the frame inference engine.

The knowledge bases which are embedded in the frames are rules about meanings of formal decisions in the refinement process of a sub-type. For example, in order to decide about attributes such as number of floors, a special reasoning process about the available width condition takes place by consulting a specific rule base called 'knowledge-base-storey'.

Flow of Knowledge Between Rules and Frames

The row house frame is used to describe a model of the plan type to which the generative rules refer. For example a double storey row-house, a three-zone row-house type, etc., eachhave their own plan characteristics. Each of them is described by a sub-type-frame. At any step, the instance type contains attributes which are added at each stage of the refinement process. Figure 4 demonstrates the flow of knowledge between the rules and frames of a row-house prototype. The rules are activated according to the frame-model.

Using the Application Level: Demonstration of the 'GPRS' in a Design Context
PRODS (Prototype-Based Design System) is an application system which demonstrates design by prototype refinement in architecture (Oxman and Gem, 1988) PRODS utilizes GPRS in a design context. A design situation defined by conditions specifications and requirements can provide a contextual environment for specific instantiations of a design solution type. The design process consists of the following stages:

-Context definition by input of data through which site specifications and design requirements can be specified
-Typification of design situation as intrinsic constraints
-Typifications of design solution types as extrinsic constraints or design goals
-Prototype selection by matching between constraints types
-Refinement process as successive steps of generation of the solution type.

Figure 5 shows a screen display of a design session carried out with the system in which instantiations of the prototype instances are made by consulting kb. zoning-rule based-knowledge which supports decisions about zoning.

```
The dwelling organization typification is_ ? - enter value
(how/why/explain)
the options for dwelling organization typification are:
1. compact organization, wetcore is not provided by outside
orientation
2. loose organization, wetcore is provided by outside orientation
? 1.
------------------------------------------------------------------------
dwelling organization typification is_ compact
------------------------------------------------------------------------
*****************************************************
dwelling zoning typification is_ exterior-interior-exterior
*****************************************************
zoning refinement is_true
*****************************************************
kb.zoning returned
```

Figure 5 Design session *refinement stage* reasoning with a zoning knowledge base

Modes of Interaction with a Design Shell

Once a knowledge structure is defined another practical question is raised: how does the designer interact with a design shell in a design process? Three modes of interaction have been suggested (Oxman, 1988). The first one, termed a design critic, allows the user to interact with the structure, explore formal variations, and derive a semantic description from the system. The second is termed the design generator mode. This mode allows the user to interact with the system, give a semantic description of a design, and have the system operate as a design generator. The system then produces a formal result. There is a combined mode of interaction with a system, which was termed, critic-generator. This mode allows the user to suggest a schematic formal design as a partial design, and lets the system, at any stage, check it and accomplish the detailed design automatically. This approach was demonstrated by the PREDIKT system (Oxman and Gero, 1987).

Conclusions: Some Educational Implications of the Design Shells Approach

A major research effort is required to connect the field of architectural theory and the field of CAAD theory.

The four levels of developing and interacting with design shells as described previously are: knowledge formulation, implementation, generic application, and design usage. These represent four levels and types of knowledge which are of interest in CAAD education. At the lowest level, that of usage, we learn through employing the system without modifications. The capacity to convey principles of design or computation derives from the way in which the system interacts with the user. With a tutorial kind of user interface the design decision making process, effect of design variables upon design, etc. become transparent to the user. Thus, even at this lowest level, there is considerable potential educational value.

The next level is that of applications. In a shell system, this implies the personalization of the shell through the 'filling in' of specific design content. This requires the ability to understand the concept of a structure of levels of knowledge. The user formalizes a specific level of knowledge within the higher level of knowledge of the shell. From a conceptual point of view, the personalization of shells provides exposure to generic knowledge, its role in design, and dealing with it computationally.

Implementations require a high level of knowledge of computer programming and knowledge-based systems technology. Knowledge formalization requires a high level of knowledge of design theory. Systems are time-consuming to design and develop.

The knowledge provided in low-level drawing systems is basically knowledge of pre-defined primitives and operations. The degree of freedom left to the user is very high, but, the user is not supported with knowledge by the system. It is, however, in the usage of a knowledge based view of design in which the major educational advantages of the design shell exist. To interact with stylistic or typologica knowledge bases is to interact with an expert teacher in a specific design realm. The computer acts as design tutor and design critic, with the advantage of being transparent in exposing the design reasoning processes. Thus we might consider the shell as providing knowledge-based design tutorials as a fringe benefit to using them in design applications. In a knowledge-based approach to design education, the shell concept offers a new environment for CAAD education.

Acknowledgments

This work was undertaken in the Design Computing Unit, Department of Architectural Sciences, University of *Sydney and at the Faculty of Architecture and Town* Planning, *Technion. It was supported by a continuing grant from the Australian Research Grants Scheme, and by the VPR Fund, Technion.*

References

Flemming, U. 1981. "The secret of the Casa Guiliani Frigerio". Environment and Planning *B*. Vol.8. pp.97-114.

Gips. J., and Stiny. C. 1980. "Production systems and grammars: a uniform characterization". pp.399-408. Environment and Planning *B*. Vol.7,

Gero, J.S. 1987. "Prototypes: a new schema for knowledge-based design". Working Paper. Architectural Computing Unit. Department of Architectural Science. University of Sydney.

Minsky, M. 1975. "A framework for representing knowledge". pp.211-277. in PH. Winston (ed.). The Psychology of Computer Vision. McGraw-Hill. New York.

Mitchell,W.J., Liggett,R.S., Kvan,T. 1987, The Art of Computer Graphics Programming. New York: Van Nostrand Reinhold.

Mitchell, W.J., Liggett, R.S., Tan,M. 1989. "Top-Down Knowledge-Based Design". This volume.

Oxman, RE. and Gero, J.S. 1987. "Using an expert system for design diagnosis and design synthesis". *Expert Systems* 41:4-5

Oxman, RE. and Gero,J.S. 1988. 'Designing by Prototype Refinement in Architecture". In Gero.J.S. ed.. Artificial *Intelligence in* Engineering Design. Amsterdam: Elsevier.

Oxman, RE. 1988. "Expert System for Generation and Evaluation in Architectural Design. D.Sc". Thesis. Faculty of Architecture and Town Planing. Technion. Israel.

Rosenman, M.A. 1986. *The BUILD Manual*. Sydney: Architectural Computing Unit. Department of Architectural Science. University of Sydney.

Rosenman, M.A.. Manago. C. and Gero. J.S. 1986. 'A model-based expert system shell". IAAI'86. pp.c:1 :1-15.

Schank, R.C.. and Abelson, R.P. 1977. Scripts. Plans. Goals and Understanding. New York: Lawrence Erlbaum Associates.

Sherwood, R. 1978. *Modern Housing* Prototypes. Cambridge: Harvard University Press.

Stiny, C., and Mitchell, W.J. 1978. "The Palladian Grammar". pp.5-18. Environment and Planning *B*. Vol.5.

Stiny, C. 1980. "Introduction to shape and shape grammar". pp.343-35l. Environment and planning *B*. Vol.7.