Chapter 1

Any progress in systematic design?

A. H. Bridges

1.1 Introduction

In order to discuss this question it is necessary to reflect awhile on design methods in general. The usual categorization discusses 'generations' of design methods, but Levy (1981) proposes an alternative approach. He identifies five paradigm shifts during the course of the twentieth century which have influenced design methods debate. The first paradigm shift was achieved by 1920, when concern with industrial arts could be seen to have replaced concern with craftsmanship. The second shift, occurring in the early 1930s, resulted in the conception of a design profession. The third happened in the 1950s, when the design methods debate emerged; the fourth took place around 1970 and saw the establishment of 'design research'. Now, in the 1980s, we are going through the fifth paradigm shift, associated with the adoption of a holistic approach to design theory and with the emergence of the concept of design ideology.

A major point in Levy's paper was the observation that most of these paradigm shifts were associated with radical social reforms or political upheavals. For instance, we may associate concern about public participation with the 1970s shift and the possible use (or misuse) of knowledge, information and power with the 1980s shift. What has emerged, however, from the work of colleagues engaged since the 1970s in attempting to underpin the practice of design with a coherent body of design theory is increasing evidence of the fundamental nature of a person's engagement with the design activity. This includes evidence of the existence of two distinctive modes of thought, one of which can be described as cognitive, modelling and the other which can be described as rational thinking. Cognitive modelling is imagining, seeing in the mind's eye. Rational thinking is linguistic thinking, engaging in a form of internal debate. Cognitive modelling is externalized through action, and through the construction of external representations, especially drawings. Rational thinking is externalized through verbal language and, more formally, through mathematical and scientific notations. Cognitive modelling is analogic, presentational, holistic, integrative and based upon pattern recognition and pattern manipulation. Rational thinking is digital, sequential, analytical, explicatory and based upon categorization and logical inference. There is some relationship between the evidence for two distintive modes of thought and the evidence of specialization in cerebral hemispheres (Cross, 1984). Design methods have tended to focus upon the rational aspects of design and have, therefore, neglected the cognitive aspects. By recognizing that there are peculiar 'designerly' ways of thinking combining both types of thought process used to perceive, construct and comprehend design representations mentally and then transform them into an external manifestation current work in design theory is promising at last to have some relevance to design practice.

1.2 Review of design methods

Most of the pioneer design theorists discussed the nature of design as a science before proceeding to their personal descriptions of techniques which, hopefully, designers would be tempted to adopt in practice. Almost without exception they took a Cartesian view of designing; breaking the problem down into fragments and solving each of these separately before attempting some grand synthesis. Although there are differences in the scale and the level of abstraction at which they treated the parts of the problem, Asimow (1962) with his design elements, Jones (1963) with his factors, Archer (1963-1964) with his sub-problems and Alexander (1964) with his misfit variables were all clearly trying to apply Cartesian methods in design. They were largely concerned with strategies for design, describing procedures or sequences which, they hoped, would enable the designer to cope with the increasing complexity of design practice. Some of their strategies derived from source disciplines such as operations research, systems analysis, ergonomics, computing and so on. These also offered a battery of formal techniques such as linear programming, transportation methods, network analysis and decision theory which appeared to offer assistance in the making of design decisions.

Other techniques had been derived from psychology (brainstorming: Osborn, 1957, and synectics: Gordon, 1961, were the most important) which attempted to stimulate creativity, and some theorists did attempt to address the interrelationship of the analytical and creative aspects of design (for example, Zwicky's, 1948, morphological analysis and Luckman's, 1969, AIDA (analysis of interconnected decision areas)). Jones (1970) provides a catalogue of a number of non-mathematical techniques.

The next generation of design theorists developed models of the problems which face the designer and tried to include people in the equation. Markus (1967) described the relationships between four systems, two of them human and two of them concerned with the building fabric. The fully developed argument (BPRU, 1972) suggests that architecture is concerned with maximizing the cost benefit of providing the building fabric (in terms of building system and environmental system) to meet the requirement of the occupants (as defined by the activity/behaviour system and organizational objectives). The BPRU model also formalized the most significant of the 1960s models which explained design as being a cyclical process of analysis, synthesis and evaluation. Analysis is the investigation of the problem, the finding and the articulating of requirements to be fulfilled and the assembling of data; synthesis entails taking the requirements and data and inventing an appropriate design; evaluation checks the design against the requirements and provides feedback for future designs. The main weakness of this model is that it does not specifically help in the development of a design. Synthesis may be either a 'black-box' process of magical insight or a 'glass-box' mechanistic process utilizing one of the formal design methods. There was also a theoretical problem with the evaluation stage: just what should be measured for appraisal? One can only measure the mundane, quantifiable elements of the design, but these may truly reflect the merits of the more innovative designs.

Design theorists therefore attempted to refine the analysis-synthesis-evaluation model and turned their attention to the history and philosophy of science in an attempt to make design theory more respectable. The main source of inspiration has been Popper's (1963) model of science as a process of conjecture and refutation. An idea is put forward and rigorously tested; if it does not fail the test we may assume it to be true. This model was put forward to overcome the theoretical problems of induction - one can never prove a theory with certainty as there may always be a counterexample - and has some attractions as a model of early stage design. Hillier et a!. (1972) presented a detailed, theoretical exposition which was later supported empirically by Jane Darke's (1979) investigations of architects in practice. However, Popper is not concerned with understanding conjecturing, only refuting it, and even in this respect is criticized by later philosophers (how much evidence is needed before a theory is refuted?). Lakatos (Lakatos and Musgrove, 1970) 'rehabilitates' Popper by defining rules for refining theories within carefully defined boundaries. Lakatos's 'research programmes' are similar to Kuhn's (1962) 'paradigms'. Kuhn observed that Popper's ideal of bold conjectures and austere refutations did not happen in practice. Old, apparently refuted theories were tenaciously kept. Better theories were ignored or held at bay until the majority of workers in the field accepted them. Kuhn calls the currently accepted concensus a paradigm, and it is only when this paradigm becomes unworkable that the community shifts to a different one. The parallels with design are clear. What designers bring to design is largely knowledge, skills and standards as shared by the design community. Hillier and Leaman (1974) describe a 'design model analogous to Kuhn's theories but using biological examples. Again though, no explanation is given as to how new ideas may arise; indeed, well-established figures even within the philosophy of science have commented on the irrational nature of discovery (Medawar, 1967).

The most radical arguments against this rationalist view of science are propounded by Feyerabend (1975). If one has abandoned trying to find logical processes for half of science (conjecture) then one should equally abandon the other half (refutation). One argument is that theories may not be conclusively disproved - is it really the theory that is wrong or just the tests? At the time one has no way of knowing. Feyerabend suggests that any theory may be useful - 'anything goes'.

It may be seen, therefore, that even in such an apparently well-ordered discipline as science the establishment of 'design methods' is fraught with difficulties. One of the severest critics of these 'scientific' models of design is Lionel March (1976) who argued:

> A scientific hypothesis is not the same thing as a design hypothesis. A logical proposition is not to be mistaken for a design proposal. There has been much confusion over these matters, hence the illusions about scientifically testable hypotheses and value-free proposals.

1.3 So what is design?

Design may be considered as a search process in a space of alternative solutions, seeking one or more solutions that satisfy certain design criteria.

We may consider design to be a special case of the general problem solving processes, which are normally characterized by the following components:

(1) A known state of being, within a single well-defined domain;
(2) Knowledge of procedures that can operate within the domain, by which a given state may be changed;
(3) A goal expressed in terms that
  (a) Specify some new state, including the conditions that have to be met by a solution
  (b) Specify boundaries to the selection of procedures for changing the existing state

Architectural design is distinguished from many other problem-solving processes by two major characteristics:

(1) The states representing candidate solutions must be generated before they can be evaluated; and
(2) The heuristics that guide the change of state rely not only on information internal to the particular problem but also on information which is external to it.

These special characteristics recognize that solution generation (synthesis) is an important feature and that architecture is an example of what Simon (1969) calls 'problem-solving in a semantically rich task domain'. The problem is further aggravated by the fact that information used in design is always incomplete and often inaccurate, and that alternative possible action sequences may lead to quite different yet acceptable solutions.

Three major problems in the architectural design process are apparent for this definition. First, it is not possible to define an adequate set of parameters to describe a state of the design process. Second, it is not known how new states may be generated from existing ones. Third, tradeoffs between dissimilar qualities are hard to make when evaluating alternative 'satisficing' solutions. Whilst representing the states of design is a difficult problem in itself, it is the generation of new states from existing ones which is the single most difficult problem of architectural design, and it is this aspect which design theorists are now attempting to refine.'

1.4 Models of the design activity

The emerging models of design activity draw on work from cognitive psychology, linguistics and artificial intelligence, and tend to view design as a series of problem transformations governed by rules or codes linking design solutions and abstract requirements, A rule in the context of design is any problem transformation linking the criteria and solution spaces; that is, some relation which reduces the size of the solution space by mapping a problem expressed in terms of abstract requirements onto some solution or class of solutions which satisfies these requirements. Rules are inherently fuzzy in defining a relation between two sets of concepts at a higher level of abstraction than either the individual requirements or cases subsumed by the rule. It therefore seems unlikely that at the level of its application (as opposed to its level of definition) any rule will be a perfect 'fit' for a given set of criteria, as the concepts involved in rule definition can be seen as labels of fuzzy sets defined by a membership function. It also implies that the alternative solutions resulting from the application of a rule will satisfy the wider context of the rule criterion to differing degrees, and evaluation of a case within this context can be interpreted as the redefinition of the membership function of a fuzzy set of solutions in the context of a particular set of problems requirements.

The rules themselves derive from a pre-existing cognitive capability encompassing both a system of social values as expressed in the designer's personal design philosophy or ideology and the manifestation of these systems in the evolution of solution types and instrumental sets. Rule systems are therefore dynamic. Rules evolve in response to changes in the social context of design, producing changes in architectural style and the social and symbolic roles of built form. Research has tended to concentrate on the more abstract and general rules, and in particular the social and symbolic codes through which architecture interacts with society, as these are the most stable and the easiest to observe. This has tended to obscure the fact that rules of differing levels of abstraction are used at all stages in the design process, from outline concept to constructional details. However, at the lower levels of details, where the context is determined entirely by the architect and the solution space is largely unconstrained by social convention, the rules are less stable and more difficult to generalize in being valid only over the very limited range defined by that context.

The application of a rule at any stage in the design process acts as a generator of one or more cases which can be evaluated in an attempt to discover what is possible both in terms of the rule criterion and the other problem requirements. Such an evaluation is, of course, context dependent in that it depends on the actual value of the criterion achieved in relation to the wider context of the other variable values, and corresponds to Simon's concept of 'satisficing conjecture'. Failure to achieve a criterion leads to either a modification of the system of constraints or an attempt at an analysis of the problem structure followed by the inference of a rule and the generation of a modified case, which has led to this process being characterized as 'analysis through synthesis'.

The design process can therefore be seen to be one of recursive conjecture-analysis operating within the framework of abduction, deduction and induction proposed by March (1976). Overall the design activity proceeds on the basis of a series of recursive paradigm shifts corresponding to modifications of the rule system (a paradigm shift occurring with the abandonment of a rule at any level) which proceeds to more detailed levels as the proposal become more specific.

Margaret Boden (1977) reinforces this view of creativity as a result of research in Al. She maintains that the potential creativity of a computer program depends on its ability to change its form of knowledge representation. This is because some forms of such representation are better suited to a particular subject than others, and by using multiple representations the level of abstraction at which actions and transformations are represented can vary, with corresponding variation in the problem-solving power of the representation. To think divergently, the program must be able to move from one level to another. By considering a problem at a 'higher' level of abstraction critical points can be identified and redundant information eliminated.

1.5 Simulating the design process

Computers are essential in accomplishing this complicated process. They are used to simulate the two major components of the goal-directed, problem-solving process of design: the states of the designed environment and the generator/test cycle that induces transitions from one state to the next. This is a fundamental information-processing mechanism known as 'conceptual inference making'. In terms of the simulation of the design states of built environments much work has been done by researchers such as Eastman and Yasky (1981), Rasdorf and Kutay (1982) and Lafue (1979), but for computers to be used more effectively in the design process their utility must be extended beyond the purely descriptive geometric and non-geometric data which are currently handled. These data may be considered as the syntax (vocabulary) of design: it is also necessary to include the semantics (meaning) of that information. This, of course, is a problem generative semanticists have been investigating for a number of years (see, for an early computer implementation, Shank, 1975)

The debate in linguistics concerns the difference between the deep structure representation of a sentence and its semantic interpretation. If there is no difference between these two types of representation (and the generative semanticists claimed that this was the case) then two things follow. First, rules 'interpreting' deep structures into semantic structures will be superfluous: if all meaning is accounted for in deep structures, and these are isomorphic with semantic structures, then semantic structure will, as it were, be deep structure, and there is no need for a distinct level of deep structure. Second, if deep structure is semantic structure, then the function of transformations will be to interpret semantic structures into surface structures: hence 'generative semantics'. The process of making deep-structure inferences about the surface structure of architectural data is of interest to Computer Aided Design (CAD) research in general. The ability to infer the meaning behind line drawings or architectural concepts is necessary for building CAD systems with both superior internal processing capabilities and adaptable, user-friendly interface capabilities. A system equipped with sophisticated inference tools would be able to relate higherlevel design concepts to the 'syntax' of an architectural database. Therefore inspection, interpretation, consistency checking, editing, criticism and synthesis of various architectural databases could be automated to a large extent. Similarly, a system able to perform these inference tasks would provide a very congenial user interface for designers.

Manipulations of such information in complex problem domains such as architectural design can be understood best from the 'knowledge engineering' viewpoint. The information which is relevant to a particular domain is known as its 'knowledge base', and the methods of using it are known as 'search' and 'inference'. Akin (1978) has identified three broad categories of knowledge used in architectural design:

(1) Representational knowledge or design symbols;
(2) Transformational knowledge or transformation rules; and
(3) Algorithmic knowledge or heuristic rules.

Design symbols encompass a number of concepts. The first is literally 'design concepts' such as 'building', 'area', 'site', etc. The second is 'design attributes', which indicate a property of a design concept by relating it to another design concept (e.g. 'site has area') or to a 'design value' (e.g. 'site is small').

The transformation rules indicate relationships between any number of design symbols (e.g. 'the BUILDING has A PART that is AN OFFICE'). Generative semanticists and knowledge engineers have both used predicate calculus to represent these logical structures. The validity of statements (such as the example above) tend to be inferred from knowledge of other predicates, much like the conditional statement 'IF predicate 1 THEN predicate 2'. This form enables the transformation of one piece of information (predicate 1) into another (predicate 2). In general terms these transformation rules tend to be:

(1) Probabilistic;
(2) Powerful when used in combination and weak individually;
(3) Able to imply several predicates (as well as single predicates) at once; (4) Able to imply specific predicates from general ones, or vice versa.

Heuristic rules are a form of meta-level knowledge which controls the system. They are used to decide which transformation rules to apply, what to do next, etc.

The Knowledge Based System (KBS) thus represents a set of resources which may be used as a component in the 'design system' or simply as a sophisticated database system. KBSs are, however, quite distinct from conventional database systems in four important ways:

(1) Knowledge bases contain explicitly represented rules as well as simple facts;
(2) Knowledge-based storage structures have low structural semantic content compared with database structures;
(3) Knowledge-base systems include components for the automatic maintenance of semantic integrity in addition to components for syntactic checking as found in conventional database systems;
(4) Knowledge-base systems include components which can make inferences over the knowledge base, thereby providing a deductive retrieval facility.

KBSs are also distinct from 'expert systems' which are typically designed for specific tasks such as mineral prospecting, medical diagnosis, fault-finding and mathematical theorem proving. They might, however, be used as components in expert systems.

This distinction between conventional DBMSs and KBSs is particularly important in the architectural design application. The state of a physical artifact is naturally representable by the objects it consists of: their form and other attributes, and the relationships between them. This is particularly true when dealing with artifacts such as buildings, and it differs from other domains of knowledge representation such as natural language or pattern recognition, where most of the knowledge is based on the inferences made from the state of many independent units of data, each of which carries a relatively small amount of information on its own. Instead, the knowledge base in the case of physical design is object centred: objects contain both the data and the operators to modify it, as it pertains to themselves.

The objects to be simulated consist of many important attributes which describe their form, composition and the information which is relevant to their operation. These attributes vary with regard to the difficulty of their simulation in computers. In particular, the three-dimensional form attribute (shape) is most difficult to represent by means of the linear symbol structures that are used in computers. Composed of topology and geometry, shape information is a complex, interlinked collection of spaces, surfaces, lines and points which, when combined in certain well-defined ways, constitute a model of volumetric solid objects. The principles which facilitate such representation of physical artifacts have been developed in the past fifteen years by a branch of computational mathematics known as 'geometric modelling'. Examples and reviews of such systems may be found in Baer et al. (1972), Requicha (1980) and Eastman and Preiss (1984).

The representation of individual objects alone is, however, adequate to represent only the static state of artifacts that are made of many interrelated objects or 'assemblies'. When viewed as an integral part of the dynamic design process, the representation of the dependencies between objects is as important as the representation of the objects themselves. Inter-object relationships, represented as links, provide the means to combine objects dynamically into meaningful systems and cause changes that are applied to one part of the system to have an effect on its other parts. For example, relocating a wall may cause abutting walls to stretch and shrink accordingly, and windows and doors to move along with the wall itself. This problem has been addressed by Eastman (1980) and Szalapaj and Bijl (1984).

In summary, viewed as a problem-solving process, design can be stimulated by means of heuristic search procedures and a network of interrelated objects. Such simulation will facilitate the transition between successive design states, maintain their internal consistency and guide the process to a recognizable solution. The obvious question is, then, how can such simulation be actually implemented to produce a useful computer-aided design system? As the origins of this particular approach to design lie in the areas of AT, most of the prototypical examples derive from there also. Integrated circuit design is a popular test field for applying new Al methodologies, and pioneering work was undertaken at MIT, Stanford and Xerox PARC. The systems that were developed made extensive use of symbolic representations of parts and assemblies and of constraint satisfaction for defining relationships between adjacent parts. Specific examples include Sussman and Steele's (1980) work on CONSTRAINTS, Bobrow and Winograd's (1977) KRL language, and SMALL TALK (Goldberg and Robson, 1983).

Architectural implementations include the VEGA geometric modelling system, developed at Carnegie-Mellon University by Woodbury and Glass (1983), and the MOLE modelling environment being developed at the University of Edinburgh (EdCAAD, 1984).

1.6 Conclusions

It has been argued that design methods are progressing (albeit under rather more esoteric names) and that the new, holistic systems will embody a number of techniques derived from work in artificial intelligence. This theoretical work is becoming practically feasible due to a number of technical advances being made in computer science, such as:

(1) Special hardware to speed up reasoning with rules expressed in languages such as Prolog and LISP.
(2) Techniques for the automatic maintenance of the semantic integrity of knowledge bases using rules expressed in languages based on firstorder predicate logic (Frost and Whittaker, 1983).
(3) The development of methods to speed up deductive retrieval by mixing theorem-proving techniques from sorted first-order predicate logic with relational algebraic operations such as division and project as used in relational database systems (Reiter, 1978; Warren, 1981).
(4) The use of logic to express and reason with knowledge involving uncertainty, assumptions, time, etc. (Mamdani and Gaines, 1981).
(5) Methods to allow multiple user-views (external schemas) of knowledge which is stored in some standard canonical form (Johnson and Martin, 1984).

The integration of all these components could result in an interactive KBS for architectural design with the structure shown schematically in *Figure 1.1*. Such a knowledge-based CAD system would assist the

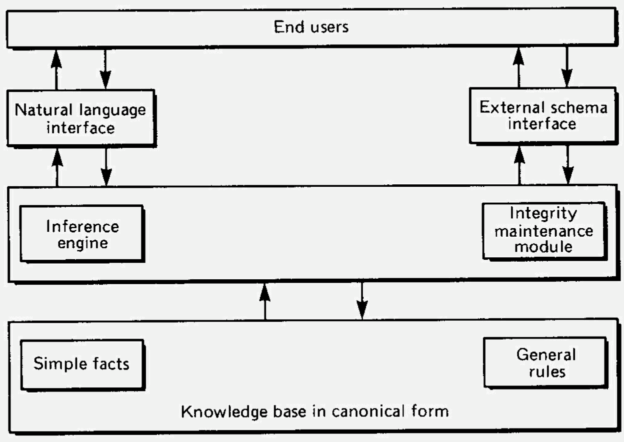

Figure 1.1. Schematic structure of the IKBS design system

designer in two main ways. First, the designer could fully model the artifact being designed and automatically maintain the semantic integrity of the model. Second, the system could provide assistance in developing a design and provide the designer with informative feedback on design decisions. The computer may thus be viewed as an 'intelligent design assistant' rather than a black (or even glass) box.

References

AKIN, 0. (1978). How do Architects Design? *Proceedings of IFIP 5.2 Working Conference on Al and Pattern Recognition in CAD*

ALEXANDER, C. (1964). *Notes on the Synthesis of Form,* Cambridge, Mass.: Harvard University Press

ARCHER, L. B. (1963-1964). Systematic Method for Designers. *Design* April 1963-August 1984

ASIMOW, M. (1962). *Introduction to Design,* Englewood Cliffs, N. J: Prentice-Hall

BAER, A., EASTMAN, C. M. and HENRION, M. (1972). Geometric Modelling: A Survey. *Computer Aided Design 11,* No. 5, 253-272

BOBROW, D. G. and WINOGRAD, T. (1977). An Overview of KRL, A Knowledge Representation Language. *Cognitive Science 1,* No. 1, 3-46

BODEN, M. (1977). *Artificial Intelligence and Natural Man,* Brighton: Harvester Press

BPRU (Building Performance Research Unit) (1972). *Building Performance,* London: Applied Science

CROSS, A. (1984). Towards an Understanding of the Intrinsic Values of Design Education. *Design Studies 5,* No. 1, January, 31-39

DARKE, . (1979). The Primary Generator and the Design Process. *Design Studies 1,* No. 1, 36-44

EASTMAN, C. M. (1980). The Design of Assemblies. *Technical Report II,* Institute of Building Sciences, Carnegie-Mellon University

EASTMAN, C. M. and PREISS, K. (1984). A Review of Solid Shape Modelling Based on Integrity Verification. *Computer Aided Design 16,* No. 2, 66-80

EASTMAN, C. M. and YASKY, x'. (1981). The Integrated Building Model and Database Scheme for the Second Phase of Integrated CAEDS. *Technical Report,* Institute of Building Sciences, Carnegie-Mellon University

EDCAAD (1984). *MOLE User Manual,* Department of Architecture, University of Edinburgh

FEYERABEND, v. (1975). *Against Method,* London: New Left Books

FROST, R. A. and WHITTAKER, S. A. (1983). A Step Towards the Automatic Maintenance of the Semantic Integrity of Databases. *The Computer Journal 26,* No. 2, 124-133

GOLDBERG, A. and ROBSON, D. (1983). *Smalltalk-80, The Language and its Implementation,* Reading, Mass.: Addison-Wesley

GORDON, W. J. J. (1961). *Synectics: the Development of Creative Capacity,* New York: Harper and Row

HILLIER, W. and LEAMAN, A. (1974). How is Design Possible? A Sketch for a Theory. *Design Research and Methods 8,* No. 1, 40-50

HILLIER, W., MUSGROVE, J. and O'SULLIVAN, P. (1972). Knowledge and Design. *Proceedings of EDRA 3*

JOHNSON, R. G. and MARTIN, N. .1. (1984). Triples as a Substructure for More Intelligent Databases. *Proceedings of Workshop on Architectures for Large Knowledge Bases,* Manchester University

JONES, J. C. (1963). A Method of Systematic Design. In Jones, J. C. and Thornley, D. G. (eds), *Conference on Design Methods,* Oxford: Pergamon

JONES, J. C. (1970). *Design Methods,* New York: Wiley

KUHN, T.S. (1962). *The Structure of Scientific Revolutions,* University of Chicago Press

LAUFE, G. M. E. (1979). Integrating Language and Database for CAD Applications. *Computer Aided Design 11,* No. 3, 127-130

LAKATOS, I. and MUSGROVE, A. (EDs) (1970). *Criticism and the Growth of Knowledge,* Cambridge: Cambridge University Press

LEVY, R. (1981). Design Research and the Evolving Design Paradigm. *Proceedings of ICSID Congress,* Helsinki

LUCKMAN, J. (1969). An Approach to the Management of Design. In Broadbent, G. and Ward, A. (eds), *Design Methods in Architecture,* London: Lund Humphries

MAMDANI, G. and GAINES, B. (1981). Fuzzy *Reasoning and its Applications,* New York: Academic Press

MARCH, L. (1976). The Logic of Design. In March, L. (ed)., *The Architecture of Form,* Cambridge: Cambridge University Press

MARKUS, T. A. (1967). The Role of Building Performance Measurement and Appraisal in Design Methods. *Architect's Journal 20* December

MEDAWAR, P. (1967), *The Art of the Soluble,* London: Metheun

OSBORN, A. F. (1957). *Applied Imagination: Principles and Practices of Creative Thinking,* New York: Scribners

POPPER, K. R. (1963). *Conjectures and Refutations,* London: Routledge and Kegan Paul

RASDORF, W. J. and KUTAY, A. R. (1982). Maintenance of Integrity During Concurrent Access in a Building Design Database. *Computer Aided Design 16,* No. 4, 201-207

REITER, R. (1978). Deductive Q-A on Relational Databases. In Gallaire, H. and Minker, J. (eds), *Logic and Databases,* New York: Plenum Press

REQUICHA, A. A. G. (1980). Representations for Rigid Solids: Theory, Methods and Systems. *Computing Surveys 12,* No. 4, 437-464

SHANK, R. C. (1957). *Conceptual Information Processing,* New York: Elsevier

SIMON, H. A. (1969). *The Sciences of the Artificial,* Cambridge, Mass.: MIT Press

SUSSMAN, G. .J. and STEELE, G. (1980). CONSTRAINTS - A Language for Expressing Almost Hierarchical Descriptions. *Artificial Intelligence* August 1980, 1-40

SZALAPAJ, P.J and BIJL, A. (1984) Knowing Where to Draw the Line. *Proceedings of IFIP Working Conference on CAD,* Hungary 1984

WARREN, D. H. o. (1981). Efficient Processing of Interactive Relational Database Queries Expressed in Logic. *Proceedings of 7th VLDB Conference,* 272-281

WOODBURY, R. F. and GLASS, G. j. (1983). VEGA: A Geometric Modelling System. *Proceedings of Graphic Interfaces,* 103-109

ZWICKY, F (1948). A Morphological Method of Analysis and Construction. In *Studies and Essays (Courant Anniversary Volume),* New York: Interscience