The Design Process: Properties, Paradigms, and Structure

Dan Braha and Oded Maimon

***Abstract*—In this paper, we examine the logic and methodology of engineering design from the perspective of the philosophy of science. The fundamental characteristics of design problems and design processes are discussed and analyzed. These characteristics establish the framework within which different design paradigms are examined. Following the discussions on descriptive properties of design, and the prescriptive role of design paradigms, we advocate the plausible hypothesis that there is a direct resemblance between the structure of design processes and the problem solving of scientific communities. The scientific community metaphor has been useful in guiding the development of general purpose highly effective design process meta-tools [73], [125].**

I. Introduction

A. Motivation and Objectives

DESIGN as problem solving is a natural and the most ubiquitous of human activities. Design begins with the acknowledgment of needs and dissatisfaction with the current state of affairs, and realization that some action must take place in order to solve the problem. In this way, scientists have been designing and acting as designers (sometimes unconsciously) throughout their lives. As such, it is of central concern to all disciplines within the artificial sciences (engineering in the broad sense).

Design science is a collection of many different logically connected knowledge and disciplines. Although there is no single model that can furnish a perfect definition of the design process, design models provide us with the powerful tools to explain and understand the design process. Design has been discussed in, among others, contexts such as general design methodologies [105], [52], [108], [36], [6], [21], [22], design artifacts representation [30], [48], [94], [122], [92], [83], computational models for the design process [78], [84], [91], [96], [120], [71], knowledge-based CAD systems [32], [117], [97], and design theories [46], [112], [124], [72], [73], [13].

Our research in engineering design [13], [72], [73] has led us to believe that evolution is fundamental to design processes and their implementation by computer-aided design (CAD) and "expert" design systems in many domains. In spite of the disparity between the models, and regardless of whether one is designing computer software, bridges, manufacturing systems, or mechanical fasteners, evolutionary speaking, they are similar. As the design process develops, the designer modifies (due to bounded rationality) either the tentative (current) design, or the specifications, based on new information obtained in the current design cycle. The modification is performed in order to remove discrepancies, and eventually establish a fit between the two parts. The evolved information reflects the fundamental feature of bounded rationality. The new information determines the tentative design knowledge, stating the relation among high and low levels of design specifications. It also determines the inference rules (or inference mechanism) that specify the method for deriving new design specifications and/or design artifacts. Both the sets of design knowledge and inference rules reflect the beliefs, skills, and expertise unconsciously developed by designers through the repetitive experiences. The converging design process includes a testing stage for verifying the tentative design against the tentative specifications to establish the direction of their future elaboration. This process terminates with an acceptable design. These characteristics were arrived at from arguments based on the concept of "bounded rationality" [106].

In this paper, we present a largely philosophical discussion of our motivations. We focus our attention on how *scientific communities* solve problems. Our thesis is that design as an evolutionary problem solving activity conforms to the structure of problem solving of scientific communities. That scientific communities are successful at generating and deciding between alternative explanations for phenomena is indisputable. Scientific progress, looked at globally and with a time scale of many decades, seems coherent and purposeful. At any one time many conflicting theories and paradigms may attempt to explain the same phenomenon. Scientific communities themselves can be the subject matter of scientific research. The nature of science has been a fertile topic in philosophy from the pre-Socratic through the present day. We are particularly indebted to a number of philosophers and historians of science of this century among them Popper, Kuhn, Laudan, and Lakatos [86], [57]–[60], [66], [61]–[63]. We hope to gain insight from this research that will be useful in guiding the development of general purpose highly effective design process meta-tools [73].

B. Overview of the Paper

Section II scrutinizes the bounds and objectives of design from the perspective of the design problem. The basic characteristics as articulated in this section are: 1) generally, design-

Manuscript received July 27, 1995; revised April 16, 1996. The work of D. Braha was supported by the Department of Manufacturing Engineering, Boston University, Boston, MA 02215 USA.

D. Braha is with the Department of Industrial Engineering, Ben-Gurion University, Beer-Sheva 84105, Israel.

O. Maimon is with the Department of Industrial Engineering, Tel-Aviv University, Tel-Aviv 69978, Israel.

Publisher Item Identifier S 1083-4427(97)00135-5.

ers act and behave under conditions of bounded-rationality; 2) alternatives, options, and outcomes are usually not given in advance (ill-structured problems), and must be found and developed by some research process; 3) usually, the optimum decisions will not be sought and satisfying decisions will be fully accepted; and 4) computationally speaking, most design optimization problems (well-structured problems) are intractable. Hence, in spite of knowing that there exists an optimal solution to a design problem, the designer may still be forced to satisfice. As a result of these basic postulates, we argue in Section III that the design process can be viewed as a stepwise, iterative, evolutionary transformation process. These characteristics establish the framework within which different design paradigms are examined. Section IV gleans useful ideas from the metaphor of scientific research to define design paradigms. Having defined a design paradigm, we survey the contemporary design paradigms. All the paradigms share the characteristic of observed evolutionary phenomenon which occurs between the time when a problem is assigned to the designer and the time the design is passed on to the manufacturer.

Following our previous discussions on descriptive properties of design, especially the adaptive and evolutionary properties discussed in Section III, and the prescriptive role of design paradigms (Section IV), we pose the hypothesis in Section V that there is a direct and striking resemblance between the structure of design processes and the structure of problem solving of scientific communities. The basic correspondence is summarized as follows: 1) the counterpart of the Kuhnian paradigm or Laudan's research tradition is the designer's knowledge-base needed to generate the set of design solutions; 2) the counterpart of a set of phenomena, events, or problems are design problems that are entirely characterized by and generated as a result of measurable and nonmeasurable requirements (specifications); 3) the counterpart of a scientific theory (set of hypotheses) is the tentative design/form serving (much the same as scientific theories) as a vehicle for the designer to capture his or her thoughts; 4) scientific discovery follows the hypothetico-deductive method, or the more justifiable procedure (following Popper) of conjecture and refutation with a direct correspondence with the evolutionary nature of design processes; and 5) incremental redesign activity corresponds to the continual and incremental evolvement of scientific theories within a normal science, whereas innovative redesign activity corresponds to a transition to a new paradigm (conceptual or paradigm-shift).

Regardless of whether or not the scientific community metaphor serves as the bases for explanations for the evolutionary design process, it also has a heuristic value in explicitly carrying out the act of design. In a recent paper [73], we developed a model of the process based on double interleaved activities of *analysis* and *synthesis*, which explode the specification world (the counterpart of scientific phenomena), and the design artifact (the counterpart of a scientific theory), until a successful solution is achieved. In [73] we illustrated the application of this evolutionary design model to the design of mechanical fasteners. A powerful and comprehensive design of a wormgear reducer (gear box) has been provided in [125]. Section VI outlines a design methodology, based on the scientific community metaphor, by emphasizing the variational (or parametric) design part. Section VII concludes the paper.

II. Properties of the Design Problem

A. The Ubiquity of Design

The natural point to begin any discussion of design is to state succinctly in a single sentence what it is that one does when one designs and what the end product is. Such an endeavor has been attempted in a variety of contexts including architecture, engineering, and computer science. Clearly, an oversimplified or single-sentence definition of design will not do. One reason why definitions fail is the omnipresence of design or problem solving as a natural human activity. We have been designing and acting as designers (sometimes unconsciously) throughout our lives. Designing is pervasive in many human activities, for example, an engineer conceiving a new type of toaster or configuring a manufacturing cell, a financial manager configuring a profitable portfolio, or a cook concocting a new pizza. Underlying these design tasks is a core set of principles, rules, laws, and techniques that the designer uses for problem solving. According to common sense, design is the process of putting together or relating ideas and/or objects in order to create a whole which hopefully achieves a certain purpose [19]. Design, according to the *Encyclopedia Britannica*, "is a process of developing plans as schemes of actions; more particularly a design may be the developed plan or scheme, whether kept in mind or set forth as a drawing or model .... Design in the fine arts is often considered to be the creative process per se, while in engineering, on the contrary, it may mean a concise record of embodiment of appropriate concepts and experiences. In architecture and product design the artistic and engineering aspects of design tend to merge, that is, an architect, craftsman, or graphic or industrial designer cannot design according to formulas alone, nor as freely as can a painter, poet, or musician." In its effort to promote research in the field, the National Science Foundation defines design as "the process by which products, processes, and systems are created to perform desired functions, through specifications." These specifications include desired object features, functions, constraints, etc. Another broad definition is that design is any arrangement of the world that achieves a desired result for known reasons. The process of design itself involves some of the same constraints as diagnostic processes or planning processes. Design approaches have traditionally been subjective, that is, a standardized set of rules is not readily available which can be applied to all classes of design problems.

B. Design as a Purposeful Activity

Design begins with the acknowledgment of needs, dissatisfaction with the current state of affairs, and realization that some action must take place in order to correct the problem. Most design theorists, including [105], [4], [67], and [98], have derived a number of consequences of this ostensibly intuitive observation.

- There is a distinction between engineering science (the "science of the artificial," as Simon called it) and natural science (e.g., physics, chemistry, and biology) that can be expressed in a variety of ways. First, the aims and methodology of natural science and engineering differ, that is, natural science is concerned with "analysis" and engineering with "synthesis." Second, natural science is "theory-oriented" while engineering is "result-oriented," and third, the engineering activity is creative, spontaneous, and intuitive, while science is rational [98].
- Design is a pragmatic discipline concerned with how things should be done. Thus, the design activity is influenced by the designer's world view and values. Consequently, the recognition and identification of the design problem, the nature of the design solution, and the determination of valid research topics in engineering design are all intimately a function of the designer's perspective [22].

C. Design is a Transformation Between Descriptions

Louis Kahn, the famous architect, viewed design as a process by which the transcendent forms of thinking and feeling produce the realization of form. By form, Kahn meant the essence created by a certain relationship of elements within the whole. Thus, in practical terms, a design problem is characterized in terms of a set of requirements (specifications, goals, and constraints) such that if an artifact or system satisfies the requirements and is implemented according to the proposed design, the design problem will be solved [93], [76], [111].

D. Categories of Design Requirements

The most basic type of requirement is empirical, measurable, or well-defined in nature. A requirement is well-defined when it specifies externally observable or empirically determinable qualities for an artifact. Some requirements can naturally be stated as empirical, which means that one knows precisely what procedures to construct or use in order to determine whether or not a given design meets such requirements. Design problems that are entirely characterized by such requirements fall within the category of what Simon [102] termed well-structured problems. The most important varieties of well-defined requirements are functionality, performance, reliability, and modifiability. Functional requirements refers to the capability of the designed artifact to do certain desirable things, that is, the minimum set of independent specifications that completely define the problem. Thus, the functional requirements are the nonnegotiable characteristics of the desired solution. We distinguish between functionality and behavior as different levels of description, where the function of a piece of a system relates the behavior of that piece to the function of the system as a whole. Performance refers to the competence of the desired artifact to achieve its functionality well. In practical terms, it usually refers to economy in the use of some observable set of resources. Reliability of artifacts is defined as the probability that the artifacts will conform to their expected behavior throughout a given period of time. Modifiability refers to the ease with which changes may be incorporated in the design of artifacts. Modifiability requirements completely support the evolutionary characteristic of the design process, and the act of successive changes or improvements to previously implemented designs.

A design problem may also be generated as a result of requirements that are not measurable. Such requirements are termed ill-defined requirements (conceptual), and any reasonably interesting and complex design problem will contain ill-defined requirements. A design problem produced fundamentally as a consequence of a set of ill-defined requirements is referred to as an ill-structured design problem [102]. In order that a design can be shown to satisfy a set of requirements, including ill-defined objectives, all requirements must eventually be transformed into well-defined requirements. The process by which this information is transformed into well-defined design objectives is called the design requirements extraction process. Hence, the extraction, elaboration, or refinement of requirements is an inherent and integral part of the generation of design.

E. Bounded Rationality and Impreciseness of Design Problems

Decision-making during the design activity deals with highly complex situations. The traditional methods of decision-making are based on the classical model of pure rationality, which assumes full and exact knowledge about the decision situation being considered. In design, assumptions about the exact knowledge are almost never true. At least to a large measure, the requirements are not comparable and therefore the preference ordering among them is incomplete. The departure from "pure-rationality" based methods is needed in design because of the fact that the designer has a limited information-processing capacity and the information is vague. Generally, designers act and behave under conditions of "bounded-rationality" [104], [106]. The concept of bounded rationality was developed by Simon in the context of administrative decision-making [104], and subsequently elaborated *inter alia* to design decision-making. Such limitations may arise in several ways: the designer may not know all the alternative courses of actions, or, even assuming all the conditions are known, the designer may be unable to decide the best course of action, or, finally, the time and cost of computing the best possible choices may be beyond the bounds of the available resources.

F. The Satisficing Nature of Design Problems

The bounded rationality led Simon to postulate that, more often than not, the optimum decisions will not be sought and satisfying decisions will be fully accepted, that is, instead of requiring an optimal design, designers accept a "good" or "satisfactory" one. In Simon's terms, this attitude toward design, which allows the use of heuristic methods, is called "satisficing." The postulate of satisfying decisions is related to the psychological theory of "aspiration level" given in the classical work of [68]. Another related concept is "incrementalism" given by [69]. "Incrementalism" is also based on the limited information-processing capacity of the decisionmakers (designers) which forces them to make decisions similar to those previously made.

G. The Intractability of Design Problems

Optimization theory is applied as a recognized technique that can assist designers in the decision-making process of design. Utilizing optimization theory to solve design problems poses optimization problems which demonstrate inherent intractability. Typical instances of design optimization problems include: 1) design of mechanisms employs graph enumeration and graph isomorphism problems are known to be NP-complete; and 2) design of printed circuit boards (PCB) includes partitioning, placement, and routing problems, which are known to be intractable. Such problems are referred to as NP-complete, or nondeterministic polynomial time complete problems [31]. The CPU time required to solve an NP-complete problem, based on known algorithms, grows exponentially with the "size" of the problem. There exist no polynomial time transformations for NPC problems nor are there any polynomial time algorithms capable of solving any NP problems; therefore, these problems are considered to be "open" or unsolved problems. The potential to solve these NP and NPC problems depends on the availability of certain heuristics. Hence, in spite of knowing that there does indeed exist an optimal solution to a design problem, the designer may still resort to satisficing methods.

H. The Form of Design

Designing an artifact can be considered a transition from concepts and ideas to concrete descriptions. By form (a synonym to design) we mean the essence or ultimate output of a design process created by a certain relationship of elements within a whole. For example, the form of a piston for a model aircraft engine, is a piece of short cylinder designed to fit closely and move inside another cylinder or tube. The piston consists of a cylinder, piston rod, and pin. Despite whether it is made of plastic, iron, or steel, it is recognized as a piston as long as the cylinder, piston, and pin remain in a certain relationship to one another.

The concept of form is elusive, abstract, and complex. The design process involves conceiving of the concepts relevant to the form and the relationships between them, and representing the concepts using specific well-defined language. In the case of engineering design, such design descriptions range from specifications in formal language (such as computer-aided engineering systems, symbolic programming techniques associated with artificial intelligence (AI) and hardware design/description languages) through description in quasiformal notation (such as linguistic descriptions and qualitative influence graphs) to very informal and visual descriptions (such as functional block diagramming, flow-diagrams, and engineering drawings). The concepts underlying a design are captured in three views. The functional view describes the design's functions and processes, thus connecting its capabilities. This view also includes the inputs and outputs of the activities, i.e., the flow of information to and from the external activities. For example, in the design process of integrated circuits the functional level includes a register–transfer diagram. The behavioral view describes the design's behavior over time, the states and modes of the design, and the conditions and events that cause modes to change. It also deals with concurrency, synchronization, and causality. Good examples are constraints that components must satisfy such as timing properties. The behavioral and functional views are invariant characteristics of the design or form. The structural view describes the subsystems and modules constituting the real system and the communication between them. It also captures geometrical information. While the two former views provide the conceptual model of the design, the structural view is considered to be a physical model, since it is concerned with the various aspects of the system's implementation. As a consequence, the conceptual model usually involves terms and notions borrowed from the problem domain, whereas the physical model draws more upon the solution domain. Examples include details about materials, layout, process parameters, heat conductivity, and other physical parameters.

The design/form serves several distinct roles in the development of an artifact. First, a design/form constitutes a tangible representation of the artifact's conceptual and physical properties, and thus serves as a vehicle for the designer to visualize and organize thoughts. Second, it serves as a plan for implementation. To accomplish this, the design/form should contain a systematic representation of the functional relationships of the components. Such demarcation of form/design and implementation has not always been necessary. Jones [52] has pointed out that the craftsman of the 19th century did not distinguish between conceptualizing an artifact and making it, and that transition from craft evolution to design-by-drawing is attributed mainly to a growth in the size and complexity of artifacts. Third, the design description must also serve as a document (for instance, in the form of user manuals) that describe how to harness the final artifact by the user. Finally, the form/design serves as a vehicle for reflecting the evolutionary history that led to the emergence of the final form/design, thus facilitating the inspection, analysis, and redesign (change) of the artifact.

III. Properties of the Design Process

A. The Evolutionary Nature of the Design Process

Let us recapitulate some of the basic postulates of the design problem as addressed in previous sections.

- Generally, designers act and behave under conditions of bounded-rationality.
- Alternatives, options, and outcomes are usually not given in advance (ill-structured problems), and must be found and developed by some research process.
- Usually, the optimum decisions will not be sought and satisfying decisions will fully be accepted.
- Computationally speaking, most design optimization problems (well-structured problems) are intractable. Hence, in spite of knowing that there exists an optimal solution to a design problem, the designer may still be forced to satisfice.

Many design theorists argue that the design process can be viewed as a stepwise, iterative, evolutionary transformation process [105], [124], [112]. The concepts underlying the evolutionary characteristic of design are captured in three views: purposeful adaptation of artificial things, ontogenic[1] design evolution, and phylogenic[2] design evolution (both latter phrases are borrowed from biology [38]). Purposeful adaptation, according to Simon, can be thought of as an interface between the "inner" environment (the substance and organization of the artifact itself) and an "outer" environment (the surroundings in which it operates). If the inner environment is appropriate for the outer environment, or vice versa, the artifact will serve its intended purpose. For instance, a ship's chronometer reacts to the pitching of the ship only in the negative sense of maintaining an invariant relation of the hands on its dial to the real-time, independent of the ship's motions. Regardless of whether or not the adaptation model is a universal feature of artificial systems, it also has a heuristic value. Hence, we can often predict behavior from knowledge of the artifact's goals and its outer environment with only minimal assumptions about the inner environment.

Ontogenic design evolution refers to the design processes that share the characteristic of observed evolutionary phenomenon which occurs between the time when a problem is assigned to the designer and the time the design is passed on to the manufacturer. During this period, the design evolves and changes from the initial form to the acceptable form. In this case, we say that there is a fit between the design and the requirements. The evolutionary model of design seems to support the cognitive model of design: Yoshikawa [124] argues that the design process can be decomposed into small design cycles. Each cycle has the following subprocesses: 1) awareness—problem identification by comparing the object under consideration and the specifications; 2) suggestion—suggesting the key concepts needed to solve the problem; 3) development—developing alternatives from the key concepts by using design knowledge; 4) testing—evaluating the alternatives in various ways such as structural computation, simulation of behavior, etc. If a problem is found as a result of testing, it also becomes a new problem to be solved in another design cycle; and 5) adaptation—selecting a candidate for adaptation and modification. Protocol studies on how technically qualified people design were conducted by several researchers (e.g., [2], [37], [119], and [53]). Subjects were given problems to solve in a specified amount of time and told to talk aloud while they were developing the design. Based on these studies, the researchers formulated several models of the design process. However, in spite of the disparity between the models, evolutionary speaking they are similar: as the design process develops, the designer modifies either the tentative design or requirements, based on new evidence (information) obtained in the current design cycle, so as to remove the discrepancy between them and establish a fit between the two parts. Regardless of whether or not the evolutionary model is a universal feature of design processes, the adaptive model also has a heuristic value and serves a useful purpose in explicitly carrying out the act of design. Solving a problem by beginning with a set of goals, identifying subgoals which when achieved realize the goals, then further identifying sub-subgoals that entail the subgoals, and so on, goes by several names in the computer science, cognitive science, and AI literature. Goal directed problem solving, stepwise refinement, and backward chaining are notable jargons used [3], [18], [79], [50]. One of the most celebrated of these "weak" methods is means-ends analysis. This method was proposed by Newell, Simon *et al*. in the late 1950's and first used in the general problem solver (GPS), one of the earliest and most influential systems developed within the problem space/heuristic search paradigm. Means-ends analysis relies on the idea that in a particular task domain, differences between possible initial or "current" and goal states can be identified and classified. Thus, for each type of difference, operators can be defined that can reduce the difference. Associated with each operator is also a precondition that the current state must satisfy in order for the operator to be applied. Means-ends analysis then attempts to reduce the difference between the current and goal states by applying the relevant operator. If, however, the preconditions for the operator are not satisfied, means-ends analysis is applied recursively to reduce the difference between the current state and the precondition.

Phylogenic design evolution refers to the act of redesign, which is defined as the act of successive changes or improvements made to a previously implemented design. An existing design is modified to meet the required changes in the original requirements. A conventional instance of redesign is encountered in discussions of the history of electronic computers where it is convenient to refer to architectural families/computer generations. The members of the family/generation are related to one another through an ancestor/descendant relationship [8]. In general, the concept of computer family/generation is tied directly to advances in technology. For example, vacuum tubes and germanium diodes characterize the first generation, discrete transistors the second and so forth.

The act of redesign can be illuminated and explained by considering two modes of evolution, namely incremental and innovative. The redesign activity may be defined as incremental if 1) over a long period of time the overall artifact's concept has remained virtually constant, and 2) artifact improvements have occurred through incremental design at the subsystem and component levels and not at the overall system level. That is, there has been no major conceptual shift. The automobile is an example of an incremental redesign related to an overall artifact's concept. The design team concerned with the next new car will take it for granted that there will be a wheel approximately at each corner and that, more or less, it will have the basic attributes of the Model "T" [87]. Many other artifacts may be said to fall into the incremental redesign category, for instance, bicycles, tractors, ships, and scissors. Innovative redesign activity is concerned with innovative, novel conceptual design. Pugh and Smith [88] observed that in all probability, while many overall artifact's concepts are fixed, there is a tremendous

[1] Ontogeny: The life history of an embryonic individual.

[2] Phylogeny: The evolutionary history of a lineage.

opportunity for dynamism and innovation at the subsystems and components levels. For example, the differential gear is used in all cars today. There have been tremendous advances in gear technology, manufacturing processes, and materials improvements, but the concept is static. The innovative redesign activity is followed by incremental redesign activity. Notwithstanding, the limited slip differential is an innovation and improvement of the subsystems level—it is an innovative redesign activity. An innovative redesign is also encountered in the evolution of the ball valve. The first British patent was granted to Edward Chrimes in 1845. This artifact appears to have been conceptually static until the early 1970's with the introduction of the Torbeck valve, and later the Ve Cone valve. As another example, consider the evolution of bicycles which underwent at least seven stages of innovation and improvement of the subsystems level: 1) the pedal system was installed to replace footwork operation, enhance control of the wheels, and increase speed; 2) incremental improvements in technology led to increasing the bicycle's speed; 3) the increase in speed created difficulties in stopping with feet. Thus, breaks were installed; 4) wheel diameter was enlarged to increase speed; 5) the increase in wheel diameter led to instabilities in the bicycle. Thus, chain transmission systems were installed to increase speed and safety by lowering the need for larger wheel diameters; 6) instabilities associated with increased speed and the beating of the wheels against the roads led to the emergence of tires; and 7) in order to enable the rider to have greater control of the pedals, the free wheel system was instated which created a more dynamic connection between the pedals and wheels.

There are three additional points to note in this regard. Firstly, the artifacts in the phylogenic design evolution are mature artifacts that either have been implemented or are operational. Secondly, the time lapse for the entire phylogenic design evolution is measurable in terms of years (the first ball valve was introduced in 1845, while the first innovative emergence of the Torbeck valve was introduced only in the early 1970's) rather than days, weeks, or months as in the ontogenic case. Finally, a single cycle of redesign will, in general, by itself constitute one or more cycles of ontogenic evolution.

B. Design Process Categories

Sriram *et al*. [110] have classified the design process into four categories: creative design, innovative design, redesign, and routine design. These classifications of design are process dependent and product independent. In creative design, the domain specific knowledge (e.g., heuristic, qualitative and quantitative) that is needed to generate the solution set and the set of explicit constraints (such as functionality, performance, environmental, manufacturability, and resource constraints) may be partially specified, while the set of possible solutions, the set of transformation operators, and the artifact space are unknown. Thus, the key element in this design activity is the transformation from the subconscious to the conscious. In innovative design, the decomposition of the problem is known, but the alternatives for each of its subparts do not exist and must be synthesized. Design might be an original or unique combination of existing components. Sriram *et al*. argue that a certain amount of creativity comes into view in the innovative design process (see also [120]). Redesign is defined as the act of successive changes or improvements to a previously implemented design. An existing design is modified to meet the required changes in the original requirements. In general, two scenarios may lead to the condition of redesign. First, when the design is passed on to the implementer, the artifact may fail to satisfy one or more critical requirements, and thus must be modified so that it satisfies the requirements. Second, the environment for which the artifact had been originally designed changes (e.g., in technology or other purposes for the artifact differ from those previously assumed) and produces new requirements. In routine design, the artifact's form, its method of design, and its mode of manufacture are known before the design process actually begins. It follows that an *a priori* plan of the solution exists and that the general nature of the requirements (satisfied by this design) is also *a priorily* known. The task of the designer is essentially to find the appropriate alternatives for each subpart that satisfies the given constraints [110], [14], [15]. Sriram *et al*. explain that at the creative end of the spectrum, the design process might be spontaneous, fuzzy, chaotic and imaginative. At the routine end of the spectrum, the design is predetermined, precise, crisp, systematic, and mathematical.

IV. Survey of Design Paradigms

A. Defining a Design Paradigm

According to the dictionary, a paradigm is an example, a pattern, or a model. For instance, we refer to the process of designing finite-state dynamic systems based on four paradigms: finite-memory machine, Moore machine, Mealy machine, and combined machine. All of these paradigms are based on the assumption that one subsystem of the designed structure system is a temporary storage of states of some variables, while the remaining subsystems represent function dependencies among appropriate variables. The paradigms differ in the nature of the function dependencies, which affects the constraints imposed upon the structure of the system to be designed. Another example of the role of paradigms is the notion of functions as building blocks for computer programs which form the basis for the development of a distinct style of programming called functional programming [43].

The common notion of a paradigm was enriched by Thomas Kuhn's seminal treatise on the nature of the genesis and development of scientific disciplines [57]–[60], [75]. The concept of a design paradigm is best elucidated by the Kuhnian paradigm concept as will be illustrated in this section. To Kuhn, a paradigm in its essence comprises two related concepts:

- *A Disciplinary Matrix*—A disciplinary matrix refers to a network of beliefs, values, techniques, theories, etc. that are shared by, and common to, a given scientific community. The following components are identified within a disciplinary matrix [58], [60]:

1) Symbolic generalizations, examples of which include Newton's laws of motions and Ohm's laws in electricity; 2) beliefs (or commitment) in metaphysical and heuristic models, such as the belief that the structure of an atom resembles a tiny planetary system [44], or that logical languages are the most effective medium for expressing the declarative knowledge in artificial intelligence systems [81]; 3) values, for example, the desire for a simple theory or solution as exemplified in the principle known as Occam's razor; 4) exemplars or shared examples, that are defined as the actual problem–solution complexes encountered by students of scientific disciplines in the course of their training, education, and research apprenticeship (through laboratory work, examinations, and the solving of textbook exercises), and by scientific practitioners during their independent research careers. A particular set of assumptions upon which several different design methods may be based, is often referred to as a methodological design paradigm. A methodological design paradigm may be, like models within a Kuhnian paradigm, metaphysical in origin, or purely heuristic in nature. Similar to the role of a Kuhnian paradigm in the context of scientific discovery, a design paradigm serves as a framework or starting point for the solution of design problems. It is, thus, fundamentally an abstract prescriptive model of the design process that serves as a useful scheme for constructing practical design methods, procedures, and (computational) tools for conducting design. Design paradigms should not be confused with design methods. By definition, methods based upon the same paradigm are equivalent in the sense that they share the same set of possible solutions. This set consists of all solutions to the problem except those that violate any of the assumptions that constitute the paradigm. Hence, a given design method may be regarded as concrete and a practical embodiment of a design paradigm; it is an explicitly prescribed set of rules which can be followed by the designer in order to produce a design. A paradigm, according to the definition, will provide a framework or scheme for one or more design methods, just as it may serve as a framework or scheme for descriptive and automated tools.

Various schools of thought may become associated with a design paradigm. For example, in hardware logic design ("gate-level" design), the so called "eastern school" (a "naturalist" methodology) favored the use of block diagram in designing basic circuits, while the "western school" (a "formalistic" methodology) advocated the use of Boolean algebra. Another example, in structural engineering of bridge design, concerns the debate between advocates of mathematical analysis of structural forces as subordinate to the development of structural form, and the approach that sophisticated analysis of the structural forces has priority over (and is determined by) structural form [10].

B. Design Paradigms

There are two major approaches to increasing our understanding of design disciplines that lack sound scientific theories: case studies (the counterpart of exemplars or shared examples) and models (the counterpart of a disciplinary matrix). The case studies approach was prevalent in such disciplines as psychology, prior to the establishment of the experimental method. This technique is also predominant in engineering design which relies mostly on the situation interpretation. The second approach is to use a model to define and understand the design process. Various perspectives and models need to be considered in order to gain a better understanding of the design process. Although there is no single model that can furnish a perfect definition of the design process, models provide us with powerful tools to explain and understand the design process. Models can be classified into five major types of paradigms: analysis–synthesis–evaluation (ASE), case-based, cognitive, algorithmic, and artificial intelligence. Following is a review of each of these paradigms.

C. The Analysis–Synthesis–Evaluation (ASE) Design Paradigm

The ASE design paradigm is a very widely believed paradigm in the engineering discipline. Three basic phases of design described by [20], [7], [51], and [70] are analysis, synthesis, and evaluation. Analysis is concerned with defining and understanding what must be translated by the designer to an explicit statement of functional requirements (goals). Synthesis is involved with finding the solutions among the feasible alternatives. Evaluation is concerned with assessing the validity of the solutions relative to the original functional requirements [20]. In general, several instances of these three phases may be required in order to progress from a more abstract level to a more concrete level in the design process. A general model of design can be visualized as a feedback loop of synthesis, analysis, and evaluation. The ASE model of design process is inherently iterative; the designer repeatedly goes back to refine and improve the design until it satisfies the requirements. Analysis and synthesis are on the forward path of the design loop, while the evaluation process is on the backward path, verifying the synthesized solutions [99]. A cycle is iterated so that the solution is revised and improved by reexamining the analysis. It has been argued that these three phases of the design, which are imperative for any design irrespective of domain, form a framework for planning and organizing design activity.

Fig. 1 depicts a more comprehensive version of a commonly used model of product development and design process. The design activity is viewed as part of the total product development process. Engineering a product involves several stages [109]. The first stage involves a market survey for potential products. This is followed by the conceptualization stage, wherein a product is conceived either as the result of a need or motivated by a potential profit. In the research and development stage, the information needed for the design of the product is developed. The design stage involves configuring the product based on several constraints. This is followed by the manufacturing process which yields the actual product (often preceded by developing a prototype). The product is then tested for quality in the testing stage and marketed in the marketing stage. The maintenance stage of the product is provided as a service by most organizations. The above process is iterative; for example, problems may arise during manufacturing and the product may have to be redesigned.

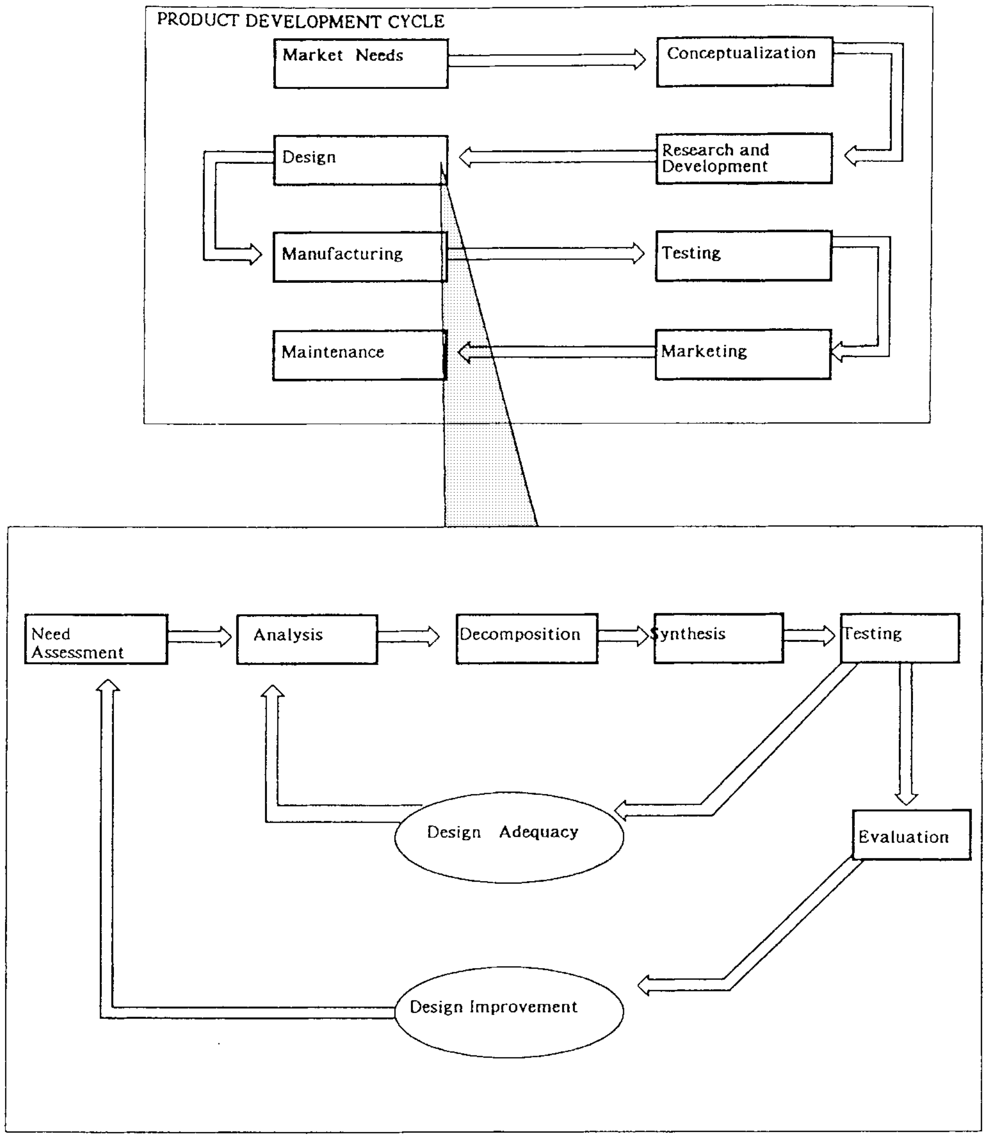

Fig. 1. Comprehensive model of the design process.

The process of solving a typical design problem involves various stages: problem identification, analysis, decomposition, synthesis, testing, evaluation, and detailed design. The first task is concerned with identifying the problem (often fuzzy in nature), resource limitation, target technology, etc. Analysis involves listing the requirements and performance specifications, as well as specifying the constraints and objectives. Synthesis is the process of selecting components to form a system that meets design objectives while satisfying constraints that govern the selection. The components may themselves be complex entities which need to be synthesized first. Decomposition is often resorted to as a means for synthesizing the artifact into smaller and smaller components [17]. The artifact is decomposed into a hierarchical assembly of systems and subsystems until terminating in a functional or physical attribute. In such a case, components, individually and through interaction with each other, meet the design goals. The testing stage involves the response of the system to external effects. This is determined by using an appropriate model of the system (such as stress and thermodynamic analysis) to check the feasibility of a design. Evaluation of such a design involves critiquing the solutions relative to the goals and selecting among alternatives. Traditionally, evaluation criteria have been represented either as production rules in production rule-based systems or as constraint rules in blackboard-based systems [27]. Other measures of performance for engineering design were constructed with the help of the theory of fuzzy sets [23], [26], [123]. The scope of these evaluators has been restricted to testing the validity of a solution rather than its degree of acceptability. In order to achieve these goals, design critiquing, which involves evaluating a design in terms of its effectiveness in satisfying a set of design objectives and constraints, was recently proposed [90], [56]. Detailed design involves the determination and evaluation of several preliminary geometrical layouts of designs. Various components of the design are refined so that all applicable specifications are satisfied. All seven stages are inextricably intertwined and are not distinct phases in the process of design. Essentially, there are three possibilities for feedback-edges from testing back to analysis and synthesis, and from evaluation back to problem identification. In the first case, the design (or form) fails to satisfy one or more of the requirements. The design must then be modified by returning to the synthesis stage. In the second case, new requirements (or constraints) emerge during testing, and the design fails to satisfy one or more of them. The new requirements must then be integrated with the "current" requirements and further analysis must be done. The outer cycle (Fig. 1) demonstrates that the evaluated solution might revise the perceived needs.

Another view (or style) of the above design stages, which is popular in many European countries, is described in [85]. This design model involves the following stages: clarification (which is similar to the above first two stages); conceptual design (which is similar to the above three later stages); embodiment, where several preliminary geometrical layouts of designs are obtained and evaluated; and detailed design (same as above). The conceptual design activity, constitutes the major part of the design process. The stage of conceptual design considerably determines the direction, flexibility, and bounds of the design. It has been shown [121] that the conceptual design stage constitutes only 3% of the total product resource costs (research and development), while it determines almost 50% of the product's features including performance, manufacturability, production costs, and other concurrent engineering factors. Hence, designers should carefully devise efficient design synthesis and evaluation tools.

The preceding section mainly described the descriptive role of the ASE paradigm. However, from the perspective of design paradigms, a more interesting issue is whether the ASE can serve as a basis for developing designs. Alexander [4] devised a method which includes a stage of extensive, detailed, and comprehensive analysis of requirements, then one or more stages of synthesis. Sriram and Cheong [109] have provided a brief description of an industrial product, supercritical fluid chromatography, which is based largely on the ASE paradigm. Finally, traditional configuration design procedures of flexible manufacturing systems (FMS) comply with this paradigm [29].

Several methods within the ASE design paradigm have recently evolved from the same foundation of competitiveness in terms of achieving high-quality and low-cost products (these include concurrent engineering, design for manufacture, quality function deployment, and robust designs techniques [100]). While it is beyond the intention of this paper to go into these methods in great detail, an indication of the relation to the ASE design paradigm is given. The most significant of these methods is quality function deployment (QFD), developed in Japan in the 1970's, and popularized by the automobile industry. QFD can be (roughly) described as a four-phase approach to design [74]: 1) customer requirements planning—translates customer expectations ("the voice of the customer") in the form of market research, competitor analysis, and technological forecasts into the desired and specific product characteristics; 2) product specifications—converts the customer requirements plan for the finished product into its components and the characteristics demanded; 3) process and quality control plans—identify design and process parameters critical to the achievement of the requirements; and 4) process sheets (derived from the process and quality control plans)—are the instructions to the operator. Thus, interpreting the QFD process in the terminology of the ASE design paradigm shows that steps 1 and 2 constitute an analysis phase, whereas step 3 constitutes a synthesis phase. The design process style that is invoked in the QFD process is a top-down method (explicitly defined in later sections).

As product designs tend to become conceptually static, QFD will tend to become a more powerful method. It can also be used as a guideline for incremental redesign activity (see Section III-A). If, however, the design implementation is in the start-up growth stage (as a consequence of innovative redesign, for example), and the customer has yet to experience the benefits of these changes, other methods may be invoked.

Although the ASE paradigm is a very widely believed design paradigm in the engineering disciplines, it bears several problems: first, with the explicit ordering of the three stages; second, with the idea of clearly separating the intellectual activities of analysis and synthesis [95], [114]; and third, with its preclusion of the role of the designer's world view (or *a priori* conceptual model) in the design process. However, if a design problem is well-structured, the design space is sufficiently small (see Section II-D), and the designer uses conceptual models (that is, the overall design of the artifact is known beforehand), then the ASE may be an appropriate paradigm (both descriptive and prescriptive).

D. Case-Based Design Paradigm

In contrast to other design domains, such as software engineering and circuit design [113], a simple and obvious correspondence between specific functional requirements of the artifact and individual components in the design does not usually exist. Due to the tightly coupled and interacting nature of mechanical designs, reasoning from prior design cases is proving to be a suitable design methodology as opposed to direct "decompose and recombine" (or "generate and test") strategies that have successfully been utilized in VLSI design [116], [111]. Cases are the primary way in which engineering students are taught to design. This is because there are no general algorithms for design. The designer activity is a consequence of his or her experience and training, much of which is based on previous exposure to similar design problems. This is particularly true in engineering design [85], [39]. Even when a novice engineer joins a design project, an important part of the engineer's training involves going through the design records of previous projects. Cased-based problem solving is based on the premise that a design (or a machine) problem solver makes use of experiences (cases) in solving new problems instead of solving every new problem from scratch [54]. Design cases reflect good design principles, such as function sharing [113] and incorporate decisions that take advantage of, or compensate for, incidental component interactions. Lansdown [65] argues that "innovation arises from incremental modification of existing 'tried and true' ideas rather than entirely new approaches ... the transformation from initial to final description is continuous and design is more like fine-tuning a set of already working ideas rather than inventing something new, although the results might not resemble anything previously imagined.” Coyne *et al*. [20] use the similar term “prototype model”: “A prototype typifies, or exemplifies, a class of designs, and thus serves as a generic design…a description of a class of designs also may be prototyped or knowledge or rules may even constitute a prototype.” A particular design can then be instantiated or exemplified from the class (prototype) of designs.

Coyne *et al*. [20] classify the case-based paradigm into three activities: creation, modification, and adaptation. Creation is concerned with incorporating requirements to create new prototypes. Modifications is concerned with developing a working design from a particular category of cases. Adaptation is concerned with extending the boundaries of the class of cases. Pugh and Morley [89] have conducted extensive design process research by interviewing successful design teams in British industry. The research indicates that embodiment design (models, prototypes, etc.) may be produced very early in the process of design, both in incremental and innovative design activities (see Section III-A).

E. The Cognitive Design Paradigm

A cognitive model is representative of how people perform a mental task or activity and the interrelationships of active intelligent human designers with computerized tools such as computer aided drafting systems. Protocol studies on how engineers design were conducted by several researchers [2], [37], [119]. In these studies, designers were given problems to solve in a specified amount of time and were asked to think aloud (protocol analysis as a technique to study problem solving behavior is discussed and used extensively in [79]). Based on these studies, the researchers have formulated several models of the design process. For example, Ullman *et al*. propose a model of the mechanical design process called the task/episode accumulation process. Their model views the design process as consisting of the conceptual design, the layout design, the detail design, and the catalog selection stages. A set of ten operators are used to accomplish these stages. They also observed that designers normally pursue only a single alternative, rather than considering multiple alternatives. Sriram and Cheong [109] indicate (based on case studies) that while designers may have difficulty retaining several alternatives in their memory, the designers feel that tools that will aid them to pursue various choices would produce more innovative designs. Many of the features incorporated into CAE (computer aided engineering) tools were influenced by the cognitive studies. CAE tools have been utilized for diverse domains. Paper path handling, air cylinders, buildings, and circuits are a few examples of domain dependent/independent frameworks developed in the mid-1980’s. These systems used hierarchical refinements and constraint propagation problem solving strategies.

Throughout the spectrum of the design process categories, the process of creation or ideation often follows a definite pattern [42]: 1) preparation—defining the situation and gathering facts; 2) frustration—struggling against mental blocks; 3) illumination—a sudden spark of insight; 4) evaluation and execution—assessing alternatives and implementing the optimal choice (contingent to the designer’s world view).

The following attributes are identified as common elements of creativity: 1) capacity for intuitive perception: the recognition of associations and similarities among objects and concepts; 2) concern for implications, meanings, and significance; 3) ability to think imaginatively without regard for practicalities; 4) open-mindedness toward, for change, improvement, and new ideas rather than rehashing old techniques and traditions. The creative designer is warned of the cost of spending too much effort researching solutions to similar problems of the past.

A number of techniques are available which appear to animate the creative process (a prescriptive view of the cognitive paradigm). Examples are: the trigger-word method in which a designer asks himself a series of active questions. The checklist method [82] that relies on a number of questions on modifications. The morphological method that analyzes the problem and determines the independent parameters involved which are then listed on a grid and evaluated systematically. The Gordon technique that attempts to identify the fundamental concepts underlying a given situation rather than emphasizing the obvious characteristics. The brain-storming technique which refers to the spontaneous generation of ideas by a diverse group of individuals, some of whom may know little about the particulars of the problem.

F. The Algorithmic Design Paradigm

The algorithmic design paradigm is the focus of a considerable amount of current and past research of design automation. The algorithmic design paradigm views the design problem as the execution of a domain-specific algorithm which, given the requirements, generates a design satisfying the requirements in a finite number of steps. The main premise of this method is the notion that design problems are well-structured. That is, the requirements are well-defined and there are precisely defined criteria for determining whether or not a design meets the requirements.

There exist a number of instances of the algorithmic paradigm which serve to optimize complex systems: exhaustive search, rapid search, and mathematical programming techniques. Within the algorithmic paradigm, exhaustive search is a lengthy process resulting in global optimization within the field of inquiry. A style is a set of characteristics that allows us to distinguish between one group of artifacts and another in the same class. A style, therefore, represents certain search modes (design process styles) on the part of the design algorithm, the result of which is the nature of the final design (see also [103]). A number of search modes exist for algorithmic paradigms: breadth-first, depth-first, greedy method, branch and bound, dynamic programming, and so on.

The price of global optimization through an exhaustive search of alternatives is tremendous. The alternative to an exhaustive search is rapid search, wherein a set of simple but arbitrary guidelines are adopted to limit the search space. For example, in serial optimization, as each stage is optimized (e.g., selecting the types of materials and method of manufacture), the selections at the subsequent stages are evaluated conditionally with the assumption that the preceding choices hold. The algorithm proceeds in this way throughout the series of stages. Serial-optimization may be the most widely used design style in conscious human decision-making. In term of effort, it is clearly superior to exhaustive search. The greatest disadvantage of any rapid search method lies in the questionable proximity to the global optimum; the rapid-search algorithms use arbitrary guidelines for optimization. Notwithstanding the global optimization potential, many instances within the algorithmic design paradigm produce satisfying rather than optimal solutions. To understand the difference between exhaustive and rapid search, consider the domain of arc welding design. Over the years, numerous researchers have studied various aspects of the welding process, such as understanding the underlying fluid mechanics, heat transfer, phase transformation, and solid mechanics of the welding process. Attempts to incorporate them into a rapid search strategy often result in a nonsystematic, somewhat random method in which designers formulate a set of decisions that ultimately result in the welding process. Most decisions are made to optimize one aspect of the process rather than the process as a whole. This often results in a suboptimal design of the welding process. A globally optimized welding process, through exhaustive search, may be achieved by requiring a complete and thorough understanding of the complex interactions among various aspects of the welding process. The difficulty with this approach is the global understanding required of an incredibly large database.

Mathematical programming techniques are a recognized topic in many engineering design courses, and are discussed at length in texts on engineering design theory [101], [24]. Mathematical programming techniques can be used to identify the potential design configuration (e.g., the physical design of electronic circuits) by optimizing the configuration based on the functional requirements. In general, in these methods the solution to the problem is developed by solving the mathematical model consisting of an objective function that is to be optimized and a set of constraints representing the limitation of the resources.

In summary, although most interesting design problems are incomplete, open-ended and ill-structured (see Section II-E), they may decompose into one or more well-structured components. In this case, the algorithmic paradigm may be successfully applied. Thus, the algorithmic paradigm may be considered as a tool that can support and be invoked by other more general paradigms.

G. The AI Design Paradigm

AI is the field that attempts to make computers perform tasks that usually require human intelligence. The AI design paradigm is based on capturing the knowledge of a certain domain and using it to solve problems [34]. In order to automate a design process, a design system must be able to differentiate between various choices and determine the best path. The AI design paradigm views design as a problem-solving process of searching through a state-space, from an initial problem state to the goal state, where the states represent the design solutions. Transitions from one state to another are affected by applying one of a finite set of operators, based on the functional requirements (goals) and design constraints (constituting the domain specific knowledge) and meta-rules (constituting the domain independent knowledge). The design process involves representing much of their knowledge about the problem declaratively. Roughly speaking, declarative knowledge is encoded explicitly in the knowledge-base in the form of sentences in some language (usually in the form of IF condition THEN action), and procedural knowledge (which typifies the algorithmic design paradigm) is manifested in algorithms.

The AI paradigm of design relies heavily on the function–structure–behavior of an artifact and their interconnections through causality. People often refer to artifacts based on the functions they provide, and that existing structures could be combined into new ones to achieve the desired function [30]. Bobrow [11] defines function as the relation between a goal of a human user and the behavior of the system. Structure is defined as the information about the interconnection of modules, organized either functionally (how the modules interact) or physically (how it is packaged). Behavior can be defined as the relationship between input from the environment and the output of affect the component usually interfaces to the environment. In summary, the proponents of the AI design paradigm claim that through a causal reasoning approach using the problem-solving process of searching and the basic physics behind behaviors, the structure and set of behaviors to achieve a given goal in as little time as possible can be accomplished.

The use of knowledge-based expert systems (KBES) has become common enough to understand their benefits in a problem such as this. An expert system is able to use previously defined rules and cases to choose through a new problem to solve it. This would enable an expert system to pick previous design structures and behaviors and match them to the design specifications. How a knowledge-based expert system should be constructed depends on where the problem lies on the analysis–synthesis spectrum [109]. In analysis (e.g., process diagnostics), the problem conditions are posed as parts of a solution description; the possible outcomes exist in the knowledge-base of a KBES. Essentially, the solution to these problems involves the identification of the solution path. In synthesis (or design) problems, conditions are given in the form of properties that a solution must satisfy as a whole; an exact solution does not (normally) exist in the knowledge-base, but the inference mechanism can generate the solution by utilizing knowledge in the knowledge-base. Fig. 2 depicts a multilevel framework for describing knowledge-based design as enunciated by [117]. KBES's are instances of automatic problem solvers that rely heavily on domain-specific heuristics, and are also often called strong methods [80]. Knowledge-based expert systems provide the support for many of the automatic or computer-aided design systems developed in recent years, such as buildings design, circuit design, paper path handling, and air cylinders [117],

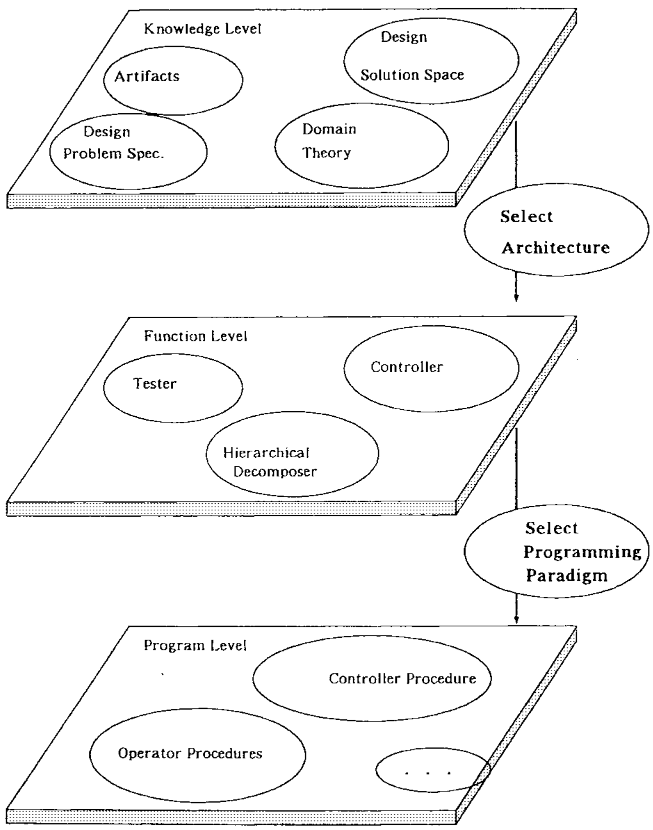

Fig. 2. Multilevel framework for describing knowledge-based system (reproduced from Sriram *et al.* [109]).

[97]. Sriram and Cheong [109] have identified the following tenets of computer-aided design systems: 1) incorporate design plans, design knowledge, and design constraints; 2) deal with evolving specifications; 3) display geometry and have the ability to associate constraints with geometrical entities (relating structure to behavior); 4) provide access to distributed knowledge/database of devices; 5) view multiple alternatives simultaneously; 6) provide access to manufacturability knowledge; and 7) generate assembly sequences automatically from geometry.

When less is known about the design task environment, domain-independent control strategies are more appropriately evoked. Problem solvers that rely heavily on domain-specific heuristics, are also often called weak methods [80]. Weak methods are used to effect and control the search through the state-space. The so-called means-ends analysis [79] lies at the center of these methods (see also Section III-A). Means-ends analysis was employed by, among others, [77] in the domain of automatic program synthesis, and by [34] in designing room configurations.

The AI design paradigm may be combined with other design paradigms to establish a "grand" problem solving strategy for the designer or design system. It is only very recently that the use of past cases is beginning to be recognized in the design automation literature [76], [32], [47]. The process of case-based design consists of the following steps that are iteratively applied as new subgoals are generated during problem-solving [115]: 1) development of a functional description through the use of qualitative relations explaining how the inputs and outputs are related; 2) retrieval of cases which results with a set of design cases (or case parts) bearing similarity to a given collection of features; 3) development of a synthesis strategy which describes how the various cases and case pieces will fit together to yield a working design; 4) realization of the synthesis strategy at the physical level; 5) verification of the design against the desired specifications through quantitative and qualitative simulation; and 6) debugging which involves the process of asking relevant questions and modifying them based on a causal explanation of the bug.

Case-based problem solving has several advantages over knowledge-based expert systems. First, cases provide memories of past solutions, failures, and repairs that have been used successfully. Secondly, from a knowledge acquisition view, there is experimental evidence that designers don't explicitly think in terms of rules [107]. Moreover, asking a designer to give examples of cases is much easier than asking him or her to give a list of rules he or she uses to design. Finally, design case studies are readily available in the literature.

Simon pointed out [105] that the design process strategies (domain-dependent as well as domain-independent) "can affect not only the efficiency with which resources for designing are used, but also the nature (form style) of the final design as well." Conversely, the designer is likely to use some special features of the problem to identify one or a small number of styles that are evoked in the process of design (that is, a form style as a determinant of a control strategy or design process style). The main three types of design process styles [105] are bottom-up, top-down, and meet-in-the-middle. Bottom-up design style involves starting with basic structures and combining these structures until the final form is accomplished. This design style incorporates search trees and optimal branch calculations. Top-down design starts with the final behavior required and subdivides this behavior into smaller behaviors which are ultimately linked to components and their respective structures. This design process style is similar to bottom-up design in that it uses search trees and optimal branch calculations. Meet-in-the-middle design process style incorporates the previous two design process styles. Either top-down or bottom-up is chosen according to the difficulty encountered in the design process and the amount of previous information available.

To sum up, the AI design paradigm is very useful in solving tightly coupled, highly integrated, and ill-structured design problems. However, when faced with an original design problem with no previous rules or past cases to help it, the expert system or case-based problem solver are incapable of original creativity. Thus the expert system or case-based problem solver are capable of helping in the design process but are not the solution to design automation. Moreover, the real problem with AI in design is in determining a way to make the computer program be creative in a manner that is manageable and not an NP hard solution. If the program just generates every possible solution, something the computer excels at, the amount of time spent generating and checking the solutions may be infinite. The design program must be able to generate a small selection of design solutions and choose between them for the ultimate choice. The human designer is able to do this quite easily but has an extensive language and other extraneous factors to assist in this process.

V. Scientific Study of Design Activities

The field of design theory is relatively new, which has been particularly stimulated by three computer-related technological advances: computer-aided design (CAD) [9], knowledge-based expert systems (KBES) [117], and concurrent engineering (CE) [100]. The main source of the slow development and confusion about design theory is that engineering design lacks the sufficient scientific foundations. Dixon [25] argued that engineering design education and practice lack an adequate base of scientific principles, and are guided too much by the specialized empiricism, intuition, and experience. Kuhn [58], [59] concluded that design is at a prescience phase and it must go through several phases before it constitutes a mature science (hence theory), which is that state of a discipline in which there is a coherent tradition of scientific research and practice, embodying law, theory, application, and instrumentation. In order to achieve this kind of maturity, designers must borrow the methodologies from other disciplines (such as AI, neural networks, logic and fuzzy logic, and object oriented methods) that have reached relative scientific maturity [20].

A design method (within a design paradigm) does not constitute a theory; theory emerges when there is a testable explanation of why the method behaves as it does [25]. Design methods do not attempt to say what design is or how human designers do what they do, but rather provide tools by which designers can explain and perhaps even replicate certain aspects of design behaviors. The major components and aims of design theories are: 1) To construct a systematic inquiry into a phenomenon which is to uncover some intelligible structure or pattern underlying the phenomenon. That is, a theory of design must be in part descriptive. 2) Theories must be in part prescriptive, that is, have the capability of specifying how design should be done, and allow us to construct more rational methods, and tools to support practical design. 3) Theories must be simple, that is, when two design theories are possible, we provisionally choose that which our minds adjudge to be the simpler, on the supposition that this is the more likely to lead in the direction of the truth. It includes as a special case the principle of William of Occam: "Causes shall not be multiplied beyond necessity." 4) Theories must be consistent with whatever else we know or believe to be true about the universe in which the phenomenon is observed. 5) The value of a design theory is determined, to a great extent, by its generality and domain-independence (including mechanical engineering, electrical engineering, civil engineering, and computer science).

While discussions of design theory from a number of different perspectives have appeared in [105], [45], [49], [21], [52], [108], and [35], in the following sections an attempt is made to demonstrate and concentrate on two interesting design theories: the axiomatic theory of design, and design as scientific problem-solving. The former theory is more grounded in the "real" environment of design while the latter theory is more abstract, speculative, or philosophical.

A. The Axiomatic Theory of Design

The axiomatic theory of design is a structured approach to implementing a product's design from a set of functional requirements that was developed by [113] and [112]. It is a mathematical approach to design which differentiates the attributes of successful product and demonstrates design that are not manufacturable. Suh defines design as the culmination of synthesized solutions (in the form of product, software, processes, or system) by the appropriate selection of design parameters that satisfy perceived needs through the mapping from functional requirements in the functional domain to design parameters in the structure domain. Suh pointed out the fact that empirical decisions often lead to superior designs (as supported by technological progress) indicates that there exists a set of underlying principles, heuristics, or axioms which govern the decision-making process. If these axioms can be formalized and their corollaries derived, then it should be possible to establish a scientific basis for guidelines in manufacturing design [113]. The axiomatic approach is based on the following premises: 1) There exist a small number of axioms which will always lead to superior decisions in terms of increased overall productivity. 2) These heuristics may be established and examined through empirical studies. 3) Axiomatic is not a mechanism for generating acceptable designs, but a tool to aid in the decision-making process.

The hypothesized principles of manufacturing axiomatic may be given as two axioms:

1) *The Independence Axiom*: In an acceptable design, the design parameters and the functional requirements are related in such a way that specified design parameter can be adjusted to satisfy its corresponding functional requirement without affecting other functional requirements (functional independence is analogous to the concept of orthogonality in linear algebra). The independence axiom does not imply that a part must be broken into two or more separate physical parts, or that a new element must be added to the existing design. Functional decoupling may be achieved without physical separation, although in some cases such physical separation may be the best way of solving the problem (recall the bicycle's evolution).

2) *The Information Axiom*: The best design has minimum information content. It coincides with the principle of "simplicity." The search for simplicity, likewise the search for beauty, is a powerful aesthetic imperative that serves as a basic component of a designer's value system. Simple in the design means, for example, being able to minimize the number and complexity of part surfaces. The simplicity principle implies that if a design satisfies more than the minimum number and measure of functional requirements originally imposed, the part or process may be overdesigned. Simple design will result in reducing product cost (such as inventory and purchasing costs), and enhancing quality.

Suh *et al*. have developed a number of theorems and corollaries which may readily be shown to be implied by the axioms. These corollaries bear strong resemblance to many design guidelines and design for manufacture (DFM) techniques (recall the ASE design paradigm) that were developed in manufacturing companies [12]. Some of these guidelines (corollaries) include minimizing functional requirements, decoupling of coupled design, integration of physical parts, symmetry, standardization, and largest tolerance.

B. *Design as Scientific Problem-Solving*

Discussions in the literature of design processes generally treats separately a category of the artificial sciences (engineering disciplines) and the natural sciences (such as physics, biology, and geology). Several criteria that demarcate the artificial sciences from the natural sciences have been identified ([105], as well as others): 1) Engineers are concerned with how things ought to be, that is, in order to attain goals, and to function while science concerns itself solely with how things are. In an ultimate philosophical sense, the natural science has found a way to exclude the normative, and to concern itself with solely with descriptive aspects of nature. 2) Engineering is concerned with "synthesis" while science is concerned with "analysis." 3) Engineering is "creative, intuitive, and spontaneous" while science is "rational and analytic." Schön [98] has proposed that design creates an entirely new epistemology (epistemology is concerned with the question of what knowledge is and how it is possible), which he terms "reflection-in-action" as contrasted to scientific discovery which concerns with "technical rationality." Cross [21] and Coyne *et al*. [20] have claimed that the aims of design and those of science differ. Coyne *et al*. summarized elegantly the demarcation between the natural science and design as "science attempts to formulate knowledge by deriving relationships between observed phenomena. Design, on the other hand, begins with intentions and uses the available knowledge to arrive at an entity possessing attributes that will meet the original intentions. The role of design is to produce form, or more correctly, a description of form using knowledge to transform a formless description into a pragmatic discipline concerned with providing a solution within the capacity of the knowledge available to the designer. This design may not be "correct" or "ideal," and may represent a compromise, but it will meet the given intentions to some degree."

That there are indeed differences in aims will be agreed in general. Unfortunately, this demarcation of the natural sciences from engineering as a result of the differences in their respective aims has led to the fictitious attitude that the methodology of science and engineering are fundamentally different. In [66] Laudan stresses at the outset that scientific problems are not fundamentally different from other kinds of problems, though they are different in degree. Indeed, we shall show that this view can be applied, with only a few qualifications, to all intellectual disciplines, and design activity in particular. Engineers wishing to construct an artifact capable of implementing a process (such as problem solving) often study naturally occurring systems that already implement the process. This approach has led, *inter alia*, many researchers to investigate psychological models of human thought as a basis for constructing the constituent methods within the AI design paradigm [79], [119]. We shall describe in the sequel those theories (which conclude, among others, [105], [55], [28], [5], [1], and [22]) that support the thesis that the engineering and

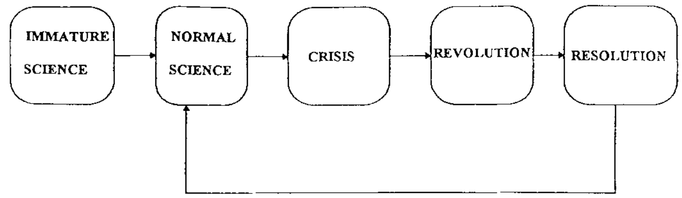

Fig. 3. Five stages in the history of scientific disciplines according to Kuhn.

natural science are methodologically indistinguishable, and that the structure of problem solving of scientific communities (scientific discovery) can be used to justify many of the decisions encountered in the design process. The relationship is one that can be seen at a high level of abstraction. The most important aspects of this relationship are forms of scientific discovery which pursue the hypothetico-deductive (H-D) or the related procedure of abductive inference, and Kuhn's model of scientific progress [58], [59] (see also Section IV-A) the primary element of which is the "paradigm" (and more elaborated ideas such as the notion of research programmes by Lakatos [61], [62], and Laudan's [66] idea of research tradition). We focus our attention on scientific progress and how scientific communities solve problems, followed by a brief outline of the parallelism to design processes. We hope to gain useful insight from this parallelism that will aid us to corroborate the thesis that design and natural science are methodologically indistinguishable.

The publication of Thomas Kuhn's *The Structure of Scientific Revolutions* in 1962 was an important milestone in the development of the historiography of science. It was the first attempt to construct a generalized picture of the process by which a science is born and undergoes change and development envisaged by Kuhn's model and further developed and refined by [61] and [66]. Their approach may be summarized as follows (see Fig. 3).

1) Immature Science: A pre-paradigm stage in which the natural phenomena that later form the subject matter of a mature science are studied and explained from widely differing points of view [58], [59].

2) Normal Science: The emergence of a paradigm (e.g., Newtonian mechanics, quantum mechanics), embodied in the published works of one or more great scientists, defining and exemplifying the concepts and methods of research appropriate to the study of a certain class of natural phenomena, and serving as an inspiration to further research by its promise of success in explaining those phenomena.

A period of normal science conducted within a conceptual and methodological framework derived from the paradigmatic achievement, involving actualization of the promise of success, further articulation of the paradigm, exploration of the possibilities within the paradigm, use of existing theory (a set of hypotheses) to predict facts, solving of scientific puzzles, development of new applications of theory, and the like. Lakatos, as well as Laudan, contend that science is seldom dominated by just one paradigm, as Kuhn claims in his account of normal science, but rather that competition between paradigms generally co-occurs with processes of development within a paradigm. Lakatos replaces Kuhn's term paradigm with the term research program (for example, we can distinguish between the flat-world and round-world research programs). The common thread linking different theories into a common research program is a "hard core" of basic assumptions shared by all investigators. This core is surrounded by a "protective belt" of auxiliary assumptions. The "hard core" which may consist of assumptions such as "no action at a distance," remains intact as long as the research program continues, but researchers can change the auxiliary assumptions in the protective belt to accommodate evidence that either has accumulated or is developed in the course of research. Laudan invokes the idea of a large-scale unit in science that he calls a "research tradition." Like Lakatos' research programs, research traditions for Laudan consist of a sequence of theories, but they lack a common core that is immune to revision. What holds a research tradition together are simply common ontological assumptions about the nature of the world and methodological principles about how to revise theories and develop new theories.

Kuhn, Lakatos, and Laudan agree that the main activity of scientists (within a paradigm, research program, or tradition) is problem solving. Scientific problem solving consists of generating hypotheses to account for phenomena, and procedures for substantiating and refuting these hypotheses. The Positivists called the former procedure (substantiation) for developing scientific theories the hypothetico-deductive (H-D) method. Popper has called the latter procedure (refutation) conjecture and refutation (C-R). The basic idea of the H-D method is that scientists begin with a phenomena (anomaly, experimental, or conceptual problem) that requires explanation. Having developed a hypotheses (for the Positivists, how hypotheses were arrived at was not a matter for logical inquiry), the task was to test and discover whether the hypotheses was true. If it was, it could provide the theory needed to explain the phenomena. The hypothesis is a general statement and so could be tested by considering initial hypothesis, and deriving predictions about what would happen under this hypothesis. If these predictions turn out to be true, the initial hypothesis would be confirmed; if the predictions turn out false, the hypothesis would be disconfirmed. In either case, a new problem (phenomena) would have been generated and the cycle begins once more. The Positivists thought that at least positive tests of a hypothesis could give support to that hypothesis. Popper [86] contended that this assumption was false; observation statements ("this swan is white") can never logically imply theories ("all swans are white"), but they can logically refute them (by providing a counterexample of just one black swan). He instead proposed that scientists should begin by making conjectures about how the world is and then seek to disprove them. If the hypothesis is disproved, then it should be discarded. If, on the other hand, a scientist tries diligently to disprove a hypothesis, and fails, the hypothesis gains a tentative or conjectural stature. Although failure to disprove does not amount to confirmation of the hypothesis and does not show that it is true or even likely to be true, Popper speaks of such an hypothesis as corroborated. The virtue of a corroborated hypothesis is that it is at least a candidate for being a true theory, whereas hypotheses that have been disproved are not even candidates. Popper terms this the process of conjectures and refutations (C-R).

Crisis: A crisis stage of varying duration precipitated by the discovery of natural phenomena that "violate the paradigm-induced (research program or tradition) expectations that govern normal science" and marked by the invention of new theories (still within the prevailing paradigm) designed to take account of the anomalous facts.

Revolution and Resolution: A relatively abrupt transition to a new paradigm (research program or research tradition) that defines and exemplifies a new conceptual and methodological framework incommensurable with the old, and continuation of normal science within the new paradigm (research program or tradition).

In addition to the foregoing properties, let us state the following important features:

- Conjectures often develop by adjusting to new problems by discovering the discrepancy between their predecessors and the phenomena (recall Newell and Simons' general problem solver scheme). This process led, for example, to better understanding of the Euler's formula and the concept of "polyhedron" (see [62]).
- When an anomaly is discovered in a theory, the outcome is often an adjustment [62] of the theory rather than a total dismissal of the existing paradigm (which defines the character and structure of the original theory), research program or tradition. Most theories evolve through a continual and incremental activity. (Recall the concept of "incrementalism" given by [69] in the context of design.)
- Although corroboration does not give us a logical basis for increasing our confidence that the hypothesis will not be falsified on the next test, it does serve to limit us to an ever-narrowing set of hypotheses that might be true. Popper has compared this process to Darwinian natural selection, for both nonadapted organisms and false theories are weeded out, leaving the stronger to continue in the competition. Evolution is often gradual, taking years or decades. Other theorists have pursued the idea that theory development may be parallel to the process of evolution by natural selection (see, for example, [58], [59], [16], and [118]).
- As pointed out above, for the Positivists how hypotheses were arrived at was not a matter for logical inquiry, since discovery was assumed to be nonrational process. Hanson [40] was one of the first to urge philosophers to redirect attention to discovery (extending the context of "scientific justification"). His proposal was to pursue what the 19th century American Pragmatist, Charles Peirce, called abductive inference, which is similar to the H-D method. The alternative to deductive reasoning is generally taken to be induction. One of the things that has brought about renewed interest in discovery is the recognition, partly motivated by work in empirical psychology, that human reasoning involves additional modes of reasoning than deductive logic and enumerative induction such as "common sense" knowledge, gestalt-like perception, analogical reasoning, or by sheer trial and error. Because scientific reasoning is simply an extension

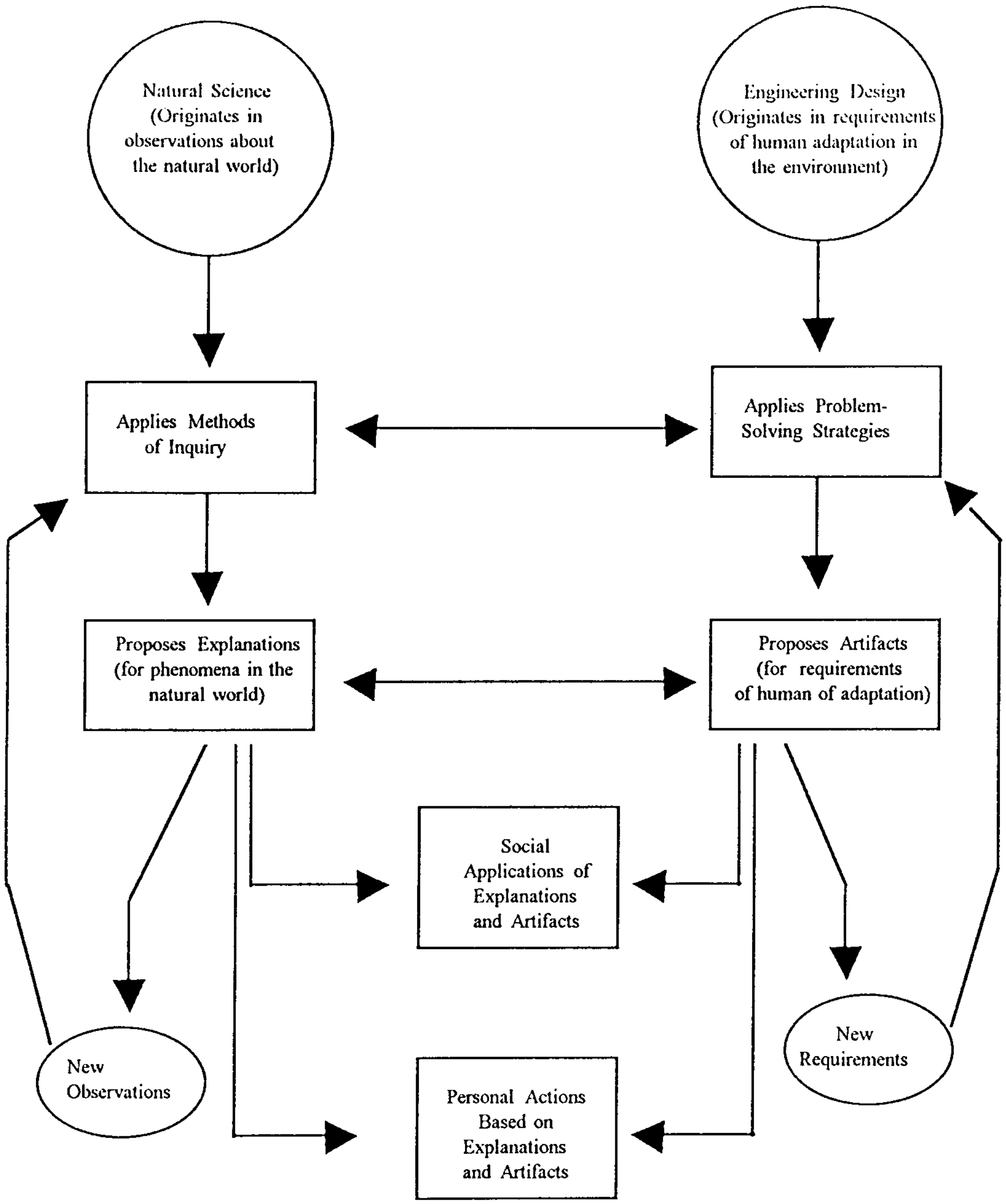

Fig. 4. The interrelationships between the science and engineering design.

of ordinary human reasoning, there is reason to think that such strategies figure also in science. Newell and Simon [79] popularized the idea that in solving complex problems we rely on heuristic principles that simplify the process through which we search for a solution. Recently there has been considerable interest by both philosophers and those in AI in using AI as a tool for studying scientific reasoning [64].

Following our previous discussions on descriptive properties of design, especially the adaptive and evolutionary properties discussed in Section III-A, and the prescriptive role of design paradigms, it is plausible to believe the hypothesis that there is a direct and striking resemblance between the structure of design processes and the foregoing structure of problem solving of scientific communities is corroborated (following Popper). Fig. 4 illustrates the parallel relationship between inquiry that is associated with science and that which is associated with engineering design. This correspondence is summarized as follows:

- The counterpart of the Kuhnian paradigm or Laudan's research tradition is the designer's knowledge-base needed to generate the set of design solutions. The designer's knowledge-base is a coherent tradition of knowledge and practice, embodying theories (such as stress analysis, thermodynamics analysis, information on materials, and the like), design paradigms, applications, heuristics (transformation operators), instrumentation, and law (such as existing zoning regulations). The most versatile designers will represent much of their knowledge about their environments declaratively. The set of knowledge underlies the designer's conceptual assumptions and world view and values serving as best approximation to what the design domain actually is; so, we really should be talking about the designer's beliefs rather than the designer's knowledge. But, following the tradition established by the phrase "knowledge-based systems," we will speak of the designer's knowledge.

- The counterpart of a set of phenomena, events, or problems are design problems that are entirely characterized by and generated as a result of measurable and nonmeasurable requirements. Laudan distinguishes two kinds of problems that can confront a research tradition—empirical inadequacies of the current theories and conceptual problems with the theories comprising the tradition. It is with a direct correspondence to the distinction (see Section II-D) between empirical, measurable, or well-defined requirements (which specifies externally observable or empirically determinable qualities for an artifact) and ill-defined requirements (conceptual).
- The counterpart of a scientific theory (set of hypotheses) is the tentative design/form (which constitutes a systematic representation of functional relationships of the components; see Section II-H) serving (much the same as scientific theories) as a vehicle for the designer to capture her thoughts, as a plan for implementation, and as a vehicle for reflecting the evolutionary history that led to the emergence of the final form/design, thus facilitating the inspection, analysis, and redesign (change) of the artifact.
- Scientific discovery follows the hypothetico-deductive (H-D) method, or the more justifiable procedure (following Popper) of conjecture and refutation (C-R). Moreover, Popper (as well as [58], [59], [16], and [118]) has compared the process of scientific discovery to Darwinian natural selection. It is with a direct correspondence with the evolutionary nature of design processes (see Section III-A): as the design process develops (due to bounded rationality), the designer test (to corroborate or disprove the hypotheses) the tentative design against the requirements. If the test fails, the designer modifies either the tentative design or requirements (the counterpart of theory adjustment), so as to remove the discrepancy (recall how Euler's formula was adjusted to a new "counterexample;" see [62]) between them, and converge or establish a fit between the two parts. Testing involves a wide range of reasoning types, such as classical and approximate logical systems (theorem provers, qualitative reasoning); experiments (e.g., simulations); knowledge accumulated from previous experience; and common sense reasoning (heuristics and "rules of thumb"). If a problem is found as the result of testing, it also becomes a new problem to be solved in another design cycle.
- Besides the embodiment of ontogenic design evolution within the model of scientific discovery, Kuhn's model of scientific progress can also explain phylogenic design evolution (see Section III-A). Incremental redesign activity corresponds to the continual and incremental evolvement of scientific theories within a normal science, whereas innovative redesign activity corresponds to a transition to a new paradigm (conceptual or paradigm-shift).
- Comparison of theories in terms of "degree of falsifiability," a concept most fully developed by Popper [86] and Newtonian world-view of "reductionism", cast considerable light on the axiomatic theory of design [113]. Popper concludes that the best reason for entertaining a theory is that it is testable (more accurately "falsifiable"), i.e., that it makes strong predictions that are readily capable of being refuted by evidence if the theory is, in fact, false. Let us, following Popper, restate the criterion in a number of other forms. Theory T_1 (e.g., "All heavenly bodies move in circles") is decidedly stronger than Theory T_2 ("All planets move in ellipses") if it is more universal and precise. Since in a theory it is desirable to obtain the maximum of prediction from the minimum of assumptions, the more universal and precise a theory, hence the more falsifiable, the better. In the case of two theories related as T_1 and T_2 we will entertain the weaker, T_2, only if the stronger, T_1, is falsified. Another form is that of simplicity. "Simple" theories are generally thought preferable to "complex" theories. Consequently, for the same design problem, a computer time-sharing system design, T_1, which was accompanied by larger amounts of experimental work, is stronger than a computer time-sharing system accompanied by poor amounts of experimental work. Similarly, as previously noted, Suh's information axiom and derivations (such as the minimization theorem and corollaries, see [113]) are derived from Popper's (and Occam's razor) simplicity principle.

What we propose to do now is briefly indicate how Suh's independence axiom and derivations, as well as various hierarchical decomposition principles of design problems represent a direct embodiment (from its paradigmatic aspect) of the 17th century Newtonian mechanics. The essential point of the Newtonian reductionistic language is that, in the Newtonian picture, the categories of causation are isolated into independent mathematical elements of the total dynamics. Indeed, the independence axiom and the principle of a nearly decomposable system (A hierarchical system of components $C_1, \cdots, C_n$ where each C_i is itself an aggregate of more primitive entities such that the interactions between the entities within the C_i's are appreciably stronger than those between the C_i's is called a nearly decomposable system, see [105]) are nothing but a paraphrase of the Newtonian language, adapted to inherently nonmechanical situation.

- Both designer's knowledge-base and Laudan's research tradition have the component of a group's shared past cases (used within the case-based design paradigm) and examples. By that it means the concrete problem–solutions and cases that students encounter from the start of their scientific and engineering education, whether in laboratories, on examinations, or at the technical problem–solutions found in the periodical literature and information about a wide variety of devices (their parts, characteristics, materials, uses, and behavior) found from commercial catalogs.

- The ASE design paradigm is based on a phase of thorough analysis of the requirements, followed by the actual synthesis of design, and then a phase of testing the design against the requirements, which is strikingly resemble to the one of the most influential methodologies of science, namely, inductivism. According to inductivism, only those propositions can be accepted into the body of science which either describe hard facts by making observations and gathering the data (the counterpart of an analysis phase of requirements) or are infallible inductive generalizations (the counterpart of a synthesis phase of design) from them. The problem with inductivism (as well as with the ASE paradigm) is that the inductivist cannot offer a rational "internal" explanation for why certain facts rather than others were selected in the first instance. Hanson [41] argued that observation itself is theory-laden, that is, what we perceive is influenced by what we know, believe, or are familiar with (which implies that designers have some conceptual models prior to gathering requirements).

We have so far tried to display the parallelism between natural sciences and engineering by illustration, and the examples could be multiplied *ad nauseam*. We have not discussed, for example, the fact that design systems as well as the scientific communities have several similar mechanisms for allocating effort (such as funding structure and peer review), nor the high degree of concurrency of activities with respect to the overall problem solving shared by scientific communities and concurrent engineering (CE). In summary, we believe that the scientific discovery metaphor can supply important insights that will aid in the interpretation (descriptive role) and construction (normative role) of design problem solving systems. Finally, we may realize that design theorists can expect no easier fate than that which befell scientists in other disciplines.

VI. A General Design Methodology

This section presents a design methodology, based on the scientific community metaphor, by emphasizing the variational (or parametric) design part. In variational design, the dimensions of a part are calculated by solving a system of constraints or specifications (typically, nonlinear equations). Let us summarize some of the very basic features of the evolutionary design model as articulated in [73] and [125]:

- In variational design, an artifact at any particular abstraction level is described in terms of part types (a group of objects which are similar but have different sizes). Every part can be described by a set of attributes. Each attribute can be described by its dimension (such as wire diameter, spring diameter, number of active coils, and modulus of elasticity).
- Specifications or constraints, at any particular abstraction level, are the various functional, behavioral, performance, reliability, aesthetic, or other characteristics or features that are to be present in the physically implemented artifact. In the case of the gear box design [125], the initial specifications were to design "a mass production device, which can be used for lifting light objects or opening a garage door in a family or small warehouse." In variational design, closed-form constraints are usually either *Euclidean* (including distance, tangency, parallelism, and so on), or *functional* (such as mass properties, forces, stiffness, strength, rating life, and so on). A higher order constraint is a property that is satisfied by lower order constraints.
- Design proceeds as a succession of cycles. In each cycle, the objects that evolve are the design/specifications complexes: the *direction* of evolution is toward the attainment of satisfied specifications, and the *mechanism* of evolution is the attempt to verify the plausibility of existing specifications, and, as a consequence, the introduction of new specifications and design parameters. If the design satisfies the specifications, then there is a fit between the two, otherwise there is said to be a misfit between design and specifications. For example, the gear box design process is terminated when the transmission parts, casing, shaft set, and accessories are fully specified, and all user-specified requirements (duration, capacity), strength constraints and heat balance are satisfied by the current solution. This evolutionary design model follows the hypothetico-deductive (H-D) method, or the more justifiable procedure (following Popper) of conjecture and refutation (C-R) of scientific discovery (see Section V-B).

The plausibility of specifications (*qualitative* design specifications as well as *closed-form equations*) as claimed by the designer is determined by the kind of evidence invoked in support of the specifications. The evidence, in turn, involves relevant *knowledge* of the design domain; tools and *techniques* of verification (finite-elements or finite-differences); and the *causal history* of the individual specifications as illuminated by a constraint dependency graph. Consider the following examples: 1) The plausibility of constraint C_1 ("to insure a long using life for the rope") is considered validated only if constraint C_2 ("the overbending of the rope is avoided") is validated. This, in turn, is considered validated only if constraint C_3 ("the minimum drum diameter should be large enough") is validated. According to the engineering recommendation value (knowledge of the design domain), for the minimum drum diameter to be large enough, a diameter 15 to 20 times the rope should be chosen. Therefore, the designer chooses the drum diameter as $d_{dr} = 3.75$ in (a design parameter). The *tool* used for support of constraint C_3 is simply to check if the selected drum diameter is indeed 15 to 20 times the rope diameter. 2) The plausibility of the constraint "select an efficient reducer" is determined by a concept selection process (the *verification tool*), which is used to measure the degree of efficiency of each alternative with respect to well-defined criteria. This, in turn, is considered validated only if the design parameter ("the reducer is a worm and wormgear type") is validated (or selected). The specifications and design parameters have dependencies among them. Dependencies between the specifications and parameters are represented by *rules*, or logical relationship the plausibility of which implies the plausibility of the specification or the design parameter. The rules are specified in the first-order calculus. For example, the plausibility of specification C ("the shear strength and compression stress of the key must be less than the allowable shear strength and compression stress, respectively ") is determined by the logical relationship: ("the shear stress should be calculated" $\wedge$ "the compression stress should be calculated" $\wedge$ "the key's material should be specified" $\wedge$ "the shear stress must be less than the allowable shear stress of the material" $\wedge$ "the compression stress must be less than the allowable compression stress of the material"). In this case, the specification part is updated by replacing specification C by its *antecedents* (C is called *consequent*). The specification "the key's material should be specified" derives a new design parameter DP "the material of the key is ASTM 40." In this case, the *design part* is updated by adding the new design parameter DP.

A most distinct feature of the design evolutionary model is that the design process is constantly *subject to revision*. For example, in the gear box design process, if the shear strength and compression stress specifications are not satisfied by the key, the designer has either to change the key material (DP_1), or change the shaft diameter DP_2 (the key size is associated with the diameter of the shaft). In that case, all specifications that dependent on DP_1 and/or DP_2 will have to be revised in the light of this new design state. The evolutionary design process is, thus, nonmonotonic in nature in accordance with Kuhn's model of scientific progress (see Section V-B).

VII. Summary

To conclude this paper, it is useful to restate that design contains a wide range of concepts. Design begins with the acknowledgment of needs and dissatisfaction with the current state of affairs, and realization that some action must take place in order to solve the problem. Design science is a collection of many different logically connected knowledge and disciplines constituting miscellaneous design paradigms. Although there is no single paradigm that can provide a complete definition of the design process, there are common characteristics that form the framework within which various paradigms are utilized.

We maintained that the demarcation of the natural sciences from engineering design, as a result of the differences in their respective aims, has led to the fictitious attitude that the methodologies of science and engineering design are fundamentally different. Alternatively, we displayed the parallelism between the natural sciences and engineering design. The resulting framework has been useful in guiding the development of general purpose design process meta-tools.

References

[1] J. Agassi, *Technology: Philosophical and Social Aspects.* Tel-Aviv, Israel: Open Univ., 1985.
[2] O. Akin, "An exploration of the design process," *Design Methods and Theory*, vol. 13, no. 3–4, pp. 115–119, 1979.
[3] S. Alagic and M. A. Arbib, *The Design of Well-Structured and Correct Programs.* Berlin, Germany: Springer-Verlag, 1978.
[4] C. Alexander, *Notes on the Synthesis of the Form.* Cambridge, MA: Harvard Univ. Press, 1964.
[5] G. S. Altshuller, *Creativity as an Exact Science.* New York: Gordon and Breach, 1984.
[6] E. K. Antonsson, "Development and testing of hypotheses in engineering design research," *J. Mech., Transmissions, Automation Design*, vol. 109, pp. 153–154, 1987
[7] W. Asimow, in *Emerging Methods in Environment Design and Planning*, G. T. Moore, Ed. Cambridge. MA: MIT Press, 1962, pp. 285–307.
[8] C. G. Bell and A. Newell, *Computer Structures: Reading and Examples.* New York: McGraw-Hill, 1971.
[9] P. E. Bezier, "CAD/CAM: Past, requirements, trends," in *Proc. CAD*, Brighton, U.K., 1984, pp. 1–11.
[10] D. P. Billington, *Robert Maillart's Bridges: The Art of Engineering.* Princeton. NJ: Princeton Univ. Press, 1979.
[11] D. G. Bobrow, *Qualitative Reasoning About Physical Systems: An Introduction.* Cambridge, MA: MIT Press, 1985, pp. 1–5.
[12] G. Boothroyd and P. Dewhurst, *Product Design for Assembly.* Wakefield, RI: Boothroyd and Dewhurst, 1987.
[13] D. Braha and O. Maimon, "A mathematical theory of design: Modeling the design process (part II)," *Int. J. General Syst.*, vol. 25, no. 3, 1997.
[14] D. C. Brown and B. Chandrasekaran, "Expert systems for a class of mechanical design activity," in *Knowledge Engineering in Computer-Aided Design*, J. S. Gero, Ed. Amsterdam, The Netherlands: North-Holland, 1985, pp. 259–282.
[15] ——, "Knowledge and control for a mechanical design expert system," *Computer*, vol. 19, pp. 92–100, July 1986.
[16] D. T. Campbell, "Evolutionary epistemology," in *The Philosophy of Karl Popper*, P. Schlipp, Ed. LaSalle, IL: Open Court, 1974.
[17] B. Chandrasekaran, "Design problem solving: A task analysis," *AI Mag.*, Winter 1990.
[18] E. Charniak and D. McDermott, *Introduction to Artificial Intelligence.* Reading, MA: Addison-Wesley, 1985.
[19] C. W. Churchman, "The philosophy of design," *ICPDT*, Boston, MA, Aug. 1987, pp. 17–20.
[20] R. D. Coyne, M. A. Rosenman, A. D. Radford, M. Balachandran, and J. S. Gero, *Knowledge-Based Design Systems.* Reading, MA: Addison-Wesley, 1990.
[21] *Development in Design Methodology,* N. Cross, Ed. New York: Wiley, 1984.
[22] S. Dasgupta, "The structure of design processes," in *Advances in Computers*, vol. 28, M. C. Yovits, Ed. New York: Academic, 1989, pp. 1–67.
[23] A. R. Diaz, "A Strategy for optimal design of hierarchical system using fuzzy sets," in *NSF Engineering Design Res. Conf.*, Univ. of Massachusetts, Amherst, June 1989, pp. 11–14.
[24] G. E. Dieter, *Engineering Design: A Materials and Processing Approach.* New York: McGraw-Hill, 1983.
[25] J. R. Dixon, *AI EDAM*, vol. 1, no. 3, pp. 145–157, 1987.
[26] Z. Dong, "Evaluating design alternatives and fuzzy operations," in *Int. Conf. Eng. Design*, Boston, MA, Aug. 1987, pp. 17–20.
[27] S. J. Fenves, U. Flemming, C. Hendrickson, M. L. Mehar, and G. Schmitt, "Integrated software environment for building design and construction," *Comput. Aided Des.*, vol. 22, no. 1, pp. 27–35, 1990.
[28] J. H. Fetzer, "Program verification: The very idea," *Commun. ACM*, vol. 31, pp. 1048–1063, Sept. 1988.
[29] *FMS Handbook*, CSDL-R-1599 U.S Army Tank Automotive Command under Contract DAAE07-82-C-4040, 1983.
[30] P. Freeman and A. Newell, "A model for functional reasoning in design," in *Proc. 2nd Int. Joint Conf. Artificial Intelligence*, 1971, pp. 621–633.
[31] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness.* San Francisco, CA: Freeman, 1979.
[32] J. S. Gero, "Prototypes: A new schema for knowledge based design," Tech. Rep., Architectural Computing Unit, Dept. Architectural Sci., 1987.
[33] *Knowledge Engineering in Computer-Aided Design.*, J. S. Gero, Ed. Amsterdam, The Netherlands: North-Holland, 1985.
[34] J. S. Gero and R. D. Coyne, "Knowledge-based planning as a design paradigm," in *Design Theory for CAD*, H. Yoshikawa and E. A. Warman, Eds. Amsterdam, The Netherlands: Elsevier, 1987.
[35] *Methodologies for Computer Systems Design.* W. K. Giloi and B. D. Shriver, Eds. Amsterdam, The Netherlands: North-Holland, 1985.
[36] G. L. Glegg, *The Science of Design.* Cambridge, U.K.: Cambridge Univ. Press, 1973.
[37] V. Goel and P. Pirolli, "Motivating the notion of generic design with information-processing theory: The design problem space," *AI Mag.*, vol. 10, no. 1, pp. 18–38, 1989.
[38] S. J. Gould, *Ontogeny and Phylogeny.* Cambridge, MA: Belknap, 1977.
[39] S. A. Gregory, "The boundaries and internals of expert systems in engineering design," in *Proc. 2nd IFIP Workshop Intelligent CAD*, 1988, pp. 7–9.

[40] N. R. Hanson, "An anatomy of discovery," *J. Philosophy*, vol. 64, pp. 321–352, 1967.
[41] ——, *Patterns of Discovery.* Cambridge, U.K.: Cambridge Univ. Press, 1958.
[42] L. Harrisberger, *Engineersmanship.* Belmont, CA: Brooks/Cole, 1966.
[43] P. Henderson, *Functional Programming: Application and Implementation.* Englewood Cliffs, NJ: Prentice-Hall, 1980.
[44] G. Holton, *Introduction to Concepts and Theories in Physical Science.* Reading. MA: Addison-Wesley, 1952.
[45] V. Hubka, *Principles of Engineering Design.* London, U.K.: Butterworth, 1982.
[46] V. Hubka and W. E. Eder, *Theory of Technical Systems: A Total Concept Theory For Engineering Design.* Berlin, Germany: Springer-Verlag, 1988.
[47] M. H. Huhns, and R. D. Acosta, "Argo: An analogical reasoning system for solving design problems," Tech. Rep. AI/CAD-092-87, Microelectronic and Computer Technology Corp., Mar. 1987.
[48] T. Ishida, H. Minowa, and N. Nakajima, "Detection of unanticipated functions of machines," in *Proc. Int. Symp. Design and Synthesis*, Tokyo, Japan, 1987, pp. 21–26.
[49] *Design: Science: Method*, R. Jaques and J. A. Powell, Eds. Guildford, U.K.: Westbury, 1980.
[50] P. N. Johnson-Laird, *The Computer and the Mind.* Cambridge, MA: Harvard Univ. Press, 1988.
[51] J. C. Jones, "A method of systematic design," in *Conf. Design Methods*, J. C. Jones and D. Thornley, Eds. Oxford, U.K.: Pergamon, 1963, pp. 10–31. Reprinted in *Development in Design Methodology*, N. Cross, Ed. New York: Wiley, 1984.
[52] ——, *Design Methods: Seeds of Human Futures*, 2nd ed. New York: Wiley, 1980.
[53] E. Kant and A. Newell, "Problem solving techniques for the design of algorithms," *Inf. Process. Manage.*, vol. 20, no. 1–2, pp. 97–118, 1984.
[54] J. L. Kolodner, R. L. Simpson, and K. Sycara, "A process model of case-based reasoning in problem solving," *Proc. IJCAI-85*, Los Angeles, CA, 1985, pp. 284–290.
[55] A. W. Kornfeld and E. C. Hewitt, "The scientific community metaphor," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-11, no. 1, Jan./Feb. 1981.
[56] C. S. Krishnamoorthy, H. Shivakumar, S. Rajeev, and S. Suresh, "A knowledge-based systems with generic tools for structural engineering," *Structural Eng. Rev.*, vol. 5, no. 1, 1993.
[57] T. S. Kuhn, *The Structure of Scientific Revolutions.* Chicago, IL: Univ. of Chicago Press, 1962.
[58] ——, "Postscript—1969," in *The Structure of Scientific Revolutions*, enlarged 2nd ed. Chicago, IL: Univ. of Chicago Press., 1970, pp. 174–210.
[59] ——, "Reflections on my critics," in *Criticism and the Growth of Knowledge*, I. Lakatos and A. Musgrave, Eds. Cambridge, U.K: Cambridge Univ. Press, 1970, pp. 231–278.
[60] ——, "Second thoughts on paradigms," in *The Essential Tension*, T. S. Kuhn, Ed. Chicago, IL: Univ. of Chicago Press, 1977.
[61] I. Lakatos, "Falsification and the methodology of scientific research programmes," in *Criticism and the Growth of Knowledge*, I. Lakatos and A. Musgrave, Eds. Cambridge, U.K: Cambridge Univ. Press, 1970, pp. 91–196.
[62] ——, *Proofs and Refutations.* Cambridge, U.K: Cambridge Univ. Press, 1976.
[63] *Criticism and the Growth of Knowledge*, I. Lakatos and A. Musgrave, Eds. Cambridge, U.K: Cambridge Univ. Press, 1970.
[64] P. Langley, H. A. Simon, G. L. Bradshaw, and J. M. Zytkow, *Scientific Discovery: Computational Explorations of the Creative Process.* Cambridge, MA: MIT Press, 1987.
[65] J. Lansdown, *Design Stud.*, vol. 8, no. 2, pp. 76–81, 1987.
[66] L. Laudan, *Progress and its Problems.* Los Angeles, CA: Univ. of California Press, 1977.
[67] B. Lawson, *How Designers Think: The Design Process Demystified.* London, U.K.: Architectural, 1980.
[68] K. Lewin, T. Dembo, L. Festinger, and P. S. Sears, "Levels of aspiration," in *Personality and the Behavior Disorder*, J. M. Hunt, Ed. New York: Ronald, 1944.
[69] C. E. Lindblom, "The science of muddling through," *Public Admin. Rev.*, vol. 9, pp. 79–88, 1959.
[70] J. Luckman, *Oper. Res. Quart.*, vol. 18, no. 4, pp. 345–358, 1967.
[71] M. L. Maher, "A knowledge-based approach to preliminary design synthesis," Rep. EDRC-12-14-87, Carnegie Mellon Univ. Eng. Design Res. Center, Pittsburgh, PA, 1987.
[72] O. Maimon and D. Braha, "A mathematical theory of design: Representation of design knowledge (pt. I)," *Int. J. General Syst.*, vol. 25, no. 2, 1997.
[73] ——, "On the complexity of the design synthesis problem," *IEEE Trans. Syst., Man, Cybern. A*, vol. 26, pp. 142–151, Jan. 1996.
[74] S. M. Manton, "Engineering for quality," in *Taguchi Methods*, Bendell, Disney, and Pridmore, Eds. Boston, MA: IFS, 1989.
[75] M. Masterman, "The nature of a paradigm," in *Criticism and the Growth of Knowledge*, I. Lakatos and A. Musgrave, Eds. Cambridge, U.K: Cambridge Univ. Press, 1970, pp. 59–90.
[76] J. Mostow, "Toward better models of the design process," *AI Mag.*, pp. 44–57, Spring 1985.
[77] R. A. Mueller, and J. Varghese, "Knowledge-based code selection in retargetable microcode synthesis," *IEEE Design and Test*, vol. 2, no. 3, pp. 44–55, 1985.
[78] S. S. Murthy, and A. Addanki, "PROMPT: An innovative design tool," in *Proc. 6th Nat. Conf. Artificial Intelligence*, Seattle, WA, 1987.
[79] A. Newell, and H. A. Simon, *Human Problem Solving.* Englewood Cliffs, NJ: Prentice-Hall, 1972.
[80] ——, "Computer science as empirical enquiry: Symbols and search," (ACM Turing Award Lecture), *Commun. ACM*, vol. 19, pp. 113–126, Mar. 1976.
[81] J. N. Nilsson, "Logic and artificial intelligence," *Artif. Intell.*, vol. 47, pp. 31–56, 1991.
[82] A. F. Osborn, *Applied Imagination.* New York: Scribner, 1963.
[83] H. M. Paynter, *Analysis and Design of Engineering Systems*. Cambridge, MA: MIT Press, 1961.
[84] J. S. Penberthy, *Incremental Analysis and the Graph of Models: A First Step Toward Analysis in the Plumber's World*, M.S. thesis, Dept. Elect. Eng. Comp. Sci., Mass. Inst. Technol., Cambridge, 1987.
[85] G. Pahl and W. Beitz, *Engineering Design.* Berlin, Germany: Springer-Verlag, 1988.
[86] K. R. Popper, *The Logic of Discovery.* London, U.K.: Hutchinson, 1959 (originally published 1935).
[87] S. Pugh, *Total Design.* New York: Addison-Wesley, 1990.
[88] S. Pugh and D. G. Smith, "CAD in the context of engineering design: The designers viewpoint," in *Proc. CAD*, London, U.K., 1976, pp. 193–198.
[89] S. Pugh and I. E. Morley, *Toward a Theory of Total Design*, Univ. Strathclyde, Design Division, 1988.
[90] J. Rehg, S. Elfes, S. Talukdar, R. Woodbury, M. Eisenberger, and R. Edahl, "CASE: Computer-aided simultaneous engineering," *AI in Engineering Design*, J. S. Gero, Ed. Berlin, Germany: Springer-Verlag, 1990, pp. 339–360.
[91] A. L. Ressler, "A circuit grammar for operational amplifier design," Tech. Rep. 807, Artificial Intelligence Laboratory, Mass. Inst. Technol., Cambridge, 1984.
[92] C. Rieger and M. Grinberg, "The declarative representation and procedural simulation of causality in physical mechanisms," in *Proc. 5th Int. Joint Conf. Artificial Intelligence*, 1977, pp. 250.
[93] J. R. Rinderle, *Measures of Functional Coupling in Design*, Ph.D dissertation, Mass. Inst. Technol., Cambridge, 1982.
[94] ——, "Function and form relationships: A basis for preliminary design," Rep. EDRC-24-05-87, Carnegie Mellon Univ. Engineering Design Research Center, Pittsburgh, PA, 1987.
[95] H. W. Rittel and M. M. Webber, "Planning problems are wicked problems," *Policy Sci.*, vol. 4, pp. 155–169.
[96] G. Roylance, "A simple model of circuit design," Tech. Rep. 703, Mass. Inst. Technol., Artificial Intelligence Laboratory, Cambridge, 1983.
[97] *Expert Systems for Engineering Design*, M. Rychener, Ed. New York: Academic, 1988.
[98] D. A. Schon, *The Reflective Practitioner.* New York: Basic Books, 1983.
[99] L. D. Serbanati, *IEEE 9th Int. Conf. Software Engineering*, 1987, pp. 190–197.
[100] G. S. Shina, *Concurrent Engineering and Design for Manufacture of Electronics Products.* New York: Van Nostrand Reinhold, 1991.
[101] J. N. Siddall, *Optimal Engineering Design: Principles and Applications.* New York: Marcel Dekker, 1982.
[102] H. A. Simon, "The structure of ill structured problems," *Artif. Intell.*, vol. 4, pp. 181–200.
[103] ——, "Style in design," in *Spatial Synthesis in Computer Aided Building Design*, C. M. Eastman, Ed. New York: Wiley, 1975.
[104] ——, *Administrative Behavior*, 3rd ed. New York: Free Press, 1976.
[105] ——,*The Science of the Artificial.* Cambridge, MA: MIT Press, 1981.
[106] ——, *Models of Bounded Rationality*, vol. 2. Cambridge, MA: MIT Press, 1982.
[107] E. Simoudis and J. S. Miller, "The application of CBR to help desk applications," *Proc. 1991 Case-Based Reasoning Workshop*, Washington, DC, DARPA.
[108] *Basic Questions of Design Theory*, W. R. Spillers, Ed. Amsterdam, The Netherlands: North-Holland, 1972.

[109] D. Sriram and K. Cheong, "Engineering design cycle: A case study and implications for CAE," in *Knowledge Aided Design*. New York: Academic, 1990.
[110] S. Sriram *et al*., "Knowledge-based system applications in engineering design: Research at MIT," *AI Mag.*, vol. 10, no. 3, pp. 79–96, 1989.
[111] L. I. Steinberg, "Design as refinement plus constraint propagation: The VEXED experience," in *Proc. 6th Nat. Conf. Artificial Intelligence*, 1987, pp. 830–835.
[112] N. P. Suh, *The Principles of Design.* New York: Oxford Univ. Press, 1990.
[113] N. P. Suh, A. C. Bell, and D. C. Gossard, "On an axiomatic approach to manufacturing and manufacturing systems," *J. Eng. Industry*, vol. 100, no. 5, pp. 127–130, 1978.
[114] W. Swartout and R. Balzer, "On the inevitable intertwining of specification and implementation," *Commun. ACM*, vol. 25, pp. 438–440, July 1982.
[115] K. Sycara and D. Navinchandra, "Integrating case-based reasoning and qualitative reasoning in design," in *AI in Design*, J. Gero, Ed. Ashurst, MI: Computational Mechanics, 1989.
[116] C. Tong, "Knowledge-based circuit design," Ph.D. dissertation, Stanford Univ., Stanford, CA, 1986.
[117] C. Tong and D. Sriram, Eds., *Artificial Intelligence Approaches to Engineering Design*. Boston, MA: Academic, 1991.
[118] S. Toulmin, *Human Understanding.* Princeton, NJ: Princeton Univ. Press, 1972.
[119] D. Ullman, L. Stauffer, and T. Dietterich, "Preliminary results of experimental study of the mechanical design process," Tech. Rep. 86-30-9, Dept. Comp. Sci., Oregon State Univ., Corvallisa, 1986.
[120] K. T. Ulrich, "Computation and pre-parametric design," Tech. Rep. 1043, Mass. Inst. Technol., Artificial Intelligence Laboratory, Cambridge, 1988.
[121] G. T. Vollbracht, "The time for CAEDM is now," *Comput. Aided Eng.*, CAD/CAM section, vol. 7, pp. 28, Oct. 1988.
[122] P. H. Winston, *et al.*, "Learning physical descriptions from functional definitions, examples and precedents," Memo 679, Artificial Intelligence Laboratory, Mass. Inst. Technol., Cambridge, 1983.
[123] K. L. Wood and E. K. Antonsson, "Engineering design-computational tools in the SYNTHESIS domain," *The Study of the Design Process*, workshop, Ohio State Univ., Columbus, Feb. 1987, pp. 8–10.
[124] H. Yoshikawa, "General design theory and a CAD system," in *Proc. IFIP W.G 5.2 Man-Machine Communications CAD/CAM*, Tokyo, Japan. Amsterdam, The Netherlands: North-Holland, 1982, pp. 35–38.
[125] D. Braha and O. Maimon, "Computational analysis of the evolutionary metaphor of design processes," Tech. Rep. 7-95, ADMS Lab., Boston Univ., Boston, MA, 1995.

Dan Braha received the Ph.D. degree in industrial engineering from Tel-Aviv University, Israel, in 1994. He served as Post-Doctoral Fellow, Department of Manufacturing Engineering, Boston University, Boston, MA, until September 1995.

He is currently an Assistant Professor, Department of Industrial Engineering, Ben-Gurion University, Israel. His research within engineering design focuses on developing methods to help the designer move from the conceptual phase to the realization of the physical device. To achieve this objective, he has developed the formal general design theory (FGDT), which is a mathematical theory of design. He has developed methods to solve routine design with group technology, genetic algorithms, and simulated annealing, to define computerized architecture, structure, and databases for the conceptual design process, and to develop a framework of logic decomposition and case-based reasoning methodology for mechanical design. He is also co-authoring, with O. Maimon, a book on the foundations of engineering design. Additional interests include real-time scheduling of flexible manufacturing systems, inventory management, risk and decision analyses, and group technology. His papers have been published in several journals, including the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, *International Journal of Production Research*, *Journal of the Operational Research Society*, and *International Journal of General Systems*.

Oded Maimon is an Associate Professor, Department of Industrial Engineering, Tel-Aviv University, Israel. He is the former chairman of the department, and is now on sabbatical at Boston University, Boston, MA. Before joining Tel-Aviv University, he was with the Massachusetts Institute of Technology and Digital Equipment Corporation.

His research interests are in design, scheduling, and automation. He established research and educational labs for design and automation at Tel-Aviv University and Boston University. He has published extensively in leading academic journals.