# Conceptual Foundations of Design Problem Solving

Gerald F. Smith and Glenn J. Browne

*Abstract*—**Design problems, processes, and methods are topics of longstanding interest in fields such as architecture and engineering. Design tasks are also common in domains addressed by systems and management scientists. However, much scientific work in these fields indicates little awareness of design theories and research. This paper introduces systems and management scientists to the extensive design theory literature. More importantly, it consolidates and extends that literature by developing a deep conceptual analysis of design problems and problem solving. The analysis is built around five elements of design problems: goals, constraints, alternatives, representations, and solutions. These elements define the basic tasks or functional demands posed by design problem solving. The paper also identifies special difficulties faced by designers in systems and management science domains.**

## I. Introduction

DESIGN—the creation of a system or artifact—is ubiquitous in practical affairs, especially in domains addressed by systems and management science. Products are designed, as are information systems, production systems, business strategies, and organizations [59]. Several fields, notably architecture (cf., [12]) and engineering (cf., [9], [20], [63]) have extensive design literatures, which include theories/conceptualizations of design, empirical studies of the design process, and methods/aids for improving design activity. Though these fields deal with the design of physical things, theorists have broadened the definition of design to include the development of any complex system or course of action [7], [53], [71]. Design problems involve making something [67], where the making is not simply from an existing plan. Design research is concerned with human creations or ''artifacts.'' The terms ''artifact'' and ''system'' will be used interchangeably in this paper, though the latter is a broader construct that includes things not designed by humans (e.g., the solar system, living organisms).

Different kinds of problems pose different tasks or functional demands. Interventions are appropriate for a problem kind insofar as they respond to its characteristic functional demands [72]. Decision making, as traditionally conceived, hinges on the prediction of future states and the evaluation of related outcomes. Decision theorists have developed probability assessment techniques in response to the prediction task, and utility assessment methods to aid in the valuation of outcomes [75]. Performance problems—situations in which an existing system is performing unsatisfactorily—require one to determine the cause of the performance deficit. Quality management research has responded with such devices as cause-and-effect diagrams [34]. To understand and improve design problem solving, it is necessary to understand design problems and the functional demands they pose.

This paper presents a wide-ranging but coherent account of the design theory literature. Though intended for an audience of systems and management scientists, the account is quite general. It integrates research findings from traditional design fields as well as from such disciplines as artificial intelligence (AI) and software engineering [64]. Despite the paper's breadth, not all the vast design literature is covered. The paper focuses on design theory and the individual thinking or problem solving aspects of design; it omits such important practical matters as client relations and project implementation. The paper's level of analysis falls midway between abstract, conceptual accounts (e.g., [24]) and reports of specific design projects.

Though the paper draws extensively from the design literature, it does more than simply summarize previous research. It proposes a conceptual framework, claiming that design can be understood in terms of five concepts prominent in design thinking and practice: goals, constraints, alternatives, representations, and solutions. The natures of these concepts and their inter-relationships are clarified. For instance, the paper explains how the notion of ''function'' relates to goals and other motivators of design activity; it elucidates the relationship between goals and constraints; a typology of constraints is proposed; the constraint-generation process is detailed; the paper outlines three general methods for generating design alternatives; it specifies the purposes of graphic and other representations in design activity; it analyzes the nature of solutions to design problems; and it suggests the limitations of constraint-based, problem space, and parametric methods of design. Though these topics have been addressed by previous research, the present discussion advances our understanding of the issues and makes them accessible to a larger audience.

Thus, the goals of the paper are to develop a conceptual framework for design theory and to familiarize systems and management scientists with design research. It is hoped that the paper's conceptual account of design will

Manuscript received July 25, 1992; revised December 16, 1992 and February 11, 1993.

G. F. Smith is with the Information and Decision Sciences Department, Carlson School of Management, University of Minnesota, Minneapolis, MN 55455.

G. J. Browne is with the Department of Information Systems, University of Maryland, Baltimore, MD 21228.

IEEE Log Number 9209678.

0018-9472/93$03.00 

promote and support design research in domains—management, for instance—that encompass challenging design problems, but that historically have not been recognized as design fields.

## II. Elements of Design Problems

Design problems have been characterized as "wicked" [17], ill-defined [73], and as having many feasible solutions [40]. The design process has been described as constraint exploration [29], achieving fit between form and context [4], and as a function-to-structure transformation [25]. To understand how these diverse descriptions might each be applicable to design, it is necessary to develop a deeper understanding of design problems. Our analysis is constructed around five conceptual elements: goals, constraints, alternatives, representations, and solutions. Depicted in Fig. 1, these concepts are conspicuous in design research.

Their selection can also be justified on conceptual grounds. Adapting from Agre [1], a problem is an undesirable situation that is significant to and may be solvable by some agent, although probably with difficulty. As such, problems necessarily involve goals, the motivations that mark some situations as undesirable. The problem concept also entails the notion of solution. Lacking the possibility of improvement, there would be no point to problem solving activity. Alternatives are the precursors to solutions; they are mentally-envisaged possibilities that problem solvers identify and evaluate. Constraints are important in design because solutions must be created, and constraints define the space of feasible alternatives. Representations, pertinent to all problem solving, are especially salient in design because designed products are complex, because physical objects lend themselves to graphic depiction, and because the solution usually is a pictorial or other representation that informs construction of an artifact.

As Fig. 1 suggests, design problems involve real and knowledge-level elements. The designer forms mental or knowledge-level representations of aspects of reality in the course of producing a design for the construction of a real thing. Goals, the motivators of design activity, are real characteristics of agents that must be mentally recognized to influence design problem solving. Likewise for constraints: these denote characteristics of reality (e.g., tensile strengths of materials) that are hopefully included in the designer's knowledge. Alternatives are generated in light of this knowledge. Elaboration of an alternative is almost invariably supported by development of a visual representation, reconnecting design with the real world. Design solutions are fully elaborated representations of acceptable alternatives or the desired products themselves.

### A. Goals

Like all problem solving, "design begins with a need" [8, p. 59]. The notion of need is a motivation concept,

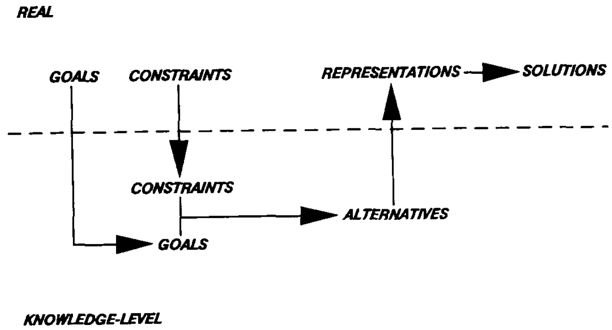



Fig. 1. Elements of design problems.

along with desire, want, preference, value, purpose, objective, and goal, the last of which is usually employed in the design literature. Unsatisfied goals or needs motivate and inform design activity, instigating design efforts and providing criteria for the evaluation of design products. Though much less apparent in other problem solving literatures, the concept of *function* also fills this motivational role in design research. Designed artifacts are "meant to achieve some functions" [16, p. 59], the functions of a design being "the goals that the design is intended to achieve when put in use" [80, p. 225].

Design goals are of various kinds. Most fundamental are needs deriving from human physiology [12]. Our bodies need food, warmth, and sleep, and we work most effectively in an environment that is physically comfortable. Architectural design and ergonomics have been attentive to such considerations, as well as to needs arising from social interaction (e.g., privacy, coordination of multiagent activities). Aesthetic goals—that something be visually appealing—seem equally innate to humans and are a core concern in architecture [30]. Furthermore, and especially in industrial societies, many artifacts are created to perform specific functions in systems that connect only remotely to basic human needs. A bushing in the engine mount of an airplane enables the craft to fly, to transport humans, and thereby to satisfy their social needs. But the bushing is designed in light of its local function, oblivious to the remote social purposes it serves.

Design goals invariably manifest a hierarchical structure in which subgoals are means to or specializations of their parents. Such structures are often depicted with tree or network diagrams. In addition to hierarchy, goal structures can exhibit conjunctive, disjunctive, and sequential relationships among elements [52]. Goals partially determine properties of the designed system, and this connection gives rise to other intergoal relationships. Two goals that imply the same property are cooperating (e.g., small size and low material cost); if the two have conflicting property implications, they are opposing [7]. Design goals vary in importance. Empirical research suggests that differences in designers' assessments of goal weights explain variation in design outputs [42]. Expert designers readily identify the most critical functions to be built into a system [68].

If unmet goals are to be addressed by designed systems, they must become known to the designer. Though goals can be stated in a brief that initiates the design project, goal identification is among the most difficult activities in design. Design is usually undertaken for a client or set of customers, so that the designer cannot introspectively determine all relevant needs. And, as systems analysts have discovered during the requirements determination phase of MIS design [19] [64], clients are rarely able to fully specify their needs. Jones [37] noted that existing needs may not reflect what people will want when new possibilities become available. In addition, systems can affect agents other than the intended customers, it being difficult to determine who these outlying stakeholders are and what relevant preferences they hold [61]. Finally, clients and designers often suboptimize by focusing on lower order goals, failing to consider the larger purposes served by the overall system. There is little to be gained from improved design of components of a fundamentally inadequate system. Nadler's ''planning and design approach'' [54] responds to this failing by requiring designers to assess the broader purposes involved, and the adequacy of existing systems to those purposes.

As a result of these difficulties, design work usually starts from an incomplete specification of goals, which is augmented as objectives are discovered during the design process [48]. But more than discovering goals, designing effects a clarification of goal descriptions and their translation into system specifications. Design goals cannot be directly mapped into solutions. There is a sizable gap between the desire for a more energy efficient automobile and an implementable design for such. These gaps are bridged by 1) means-ends analysis, the decomposition of general objectives into subgoals, means of achieving desired ends [48] and 2) conceptual translations, as from goals into functions, then into the artifact's causal structure, and finally into its form [26]. Definitions of design as a translation from one language to another suggest the importance of this process [10], [16], [25], [45], [68].

### B. Constraints

The notion of constraint is central to design. Indeed, design has been conceived as ''a process of expressing and exploring constraints'' [29, p. 134]. This centrality derives from the nature of design problems: something new must be created; human imagination is able to generate various possibilities; but this capacity and the alternatives it proposes must be managed by consideration of what is feasible. Constraints serve this purpose. They express relations among properties or variables of the proposed artifact and its environment or context [46]. ''Constraints are the rules, requirements, relations, conventions, and principles that define the context of designing'' [28, p. 56]. A characteristic of the environment (including the designer), or of the artifact as currently conceived, is constraining when it rules out or against potential settings of design variables.

Constraints are related to goals, Brown and Chandrasekaran [13] arguing that constraint is the more general concept, goals being one variety of such. Ullman, Dietterich, and Stauffer defined constraints as including ''specifications, requirements, needs, performance measures, and objectives'' [74, p. 196]. As mathematical programmers know, constraints and objectives can often be interchanged. Certainly goals are constraining in that options are restricted to those promising goal satisfaction. But a conceptual distinction can be made between what is wanted for its own sake (goals) and considerations that are not valued per se, but which rule out options nonetheless (constraints). Personal computers, designed for ease of use (goal), must be compatible with standard power supplies (constraint). This distinction blurs as artifacts become more remote from users, as functions replace needs (e.g., the airplane's engine mount bushing). An artifact's function is the purpose it serves in a larger system. Rather than being valued for its own sake, that function is determined or constrained by the overall system configuration.

Broadly understood, constraints are whatever is constraining. Admitting that goals are constraining, it is nonetheless useful to maintain a goal-constraint distinction. Whether as human needs or as functions to be performed by a system, goals motivate design [16]. Constraints, narrowly understood, direct designer attention towards what is doable. The goal-constraint distinction reflects the difference between desirability and feasibility. Other concepts are also employed in this regard. *Requirements* usually refer to goals and constraints expressed in formal problem statements [49]. *Specifications* can be precisely stated requirements or the actual settings of variables in the completed design. *Criteria* include goals and constraints considered during the evaluation of design alternatives.

Design theorists have differentiated constraints in many ways, the major distinctions being identified in Fig. 2. Constraints can apply to different entities. While the designed system is the usual target of restriction, constraints can also pertain to the design process and the process by which systems are produced from design descriptions [16]. Constraints can derive from different sources. Internal constraints reflect facts of nature, technological capabilities, and the evolving character of the artifact. External constraints are produced by particular agents and social realities. The fundamental requirement of any design is that it be physically realizable [9], [77], [82]. Realizability is an internal constraint that derives from technological capabilities and, ultimately, facts of nature. Internal constraints are also produced by the system's evolving form and the need for coherence among its components. Per Logan, ''later decisions are constrained by earlier decisions in that they are taken within the context of an existing partial solution, and each solution further limits the range of possible alternatives'' [44, p. 189].

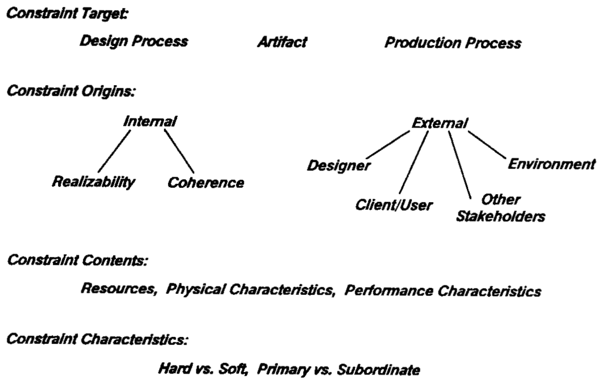



Fig. 2. Varieties of constraints.

Thus, design of modules of an information system is constrained by characteristics of databases with which they interface.

Other constraints originate externally, from design stakeholders or from the general environment. In addition to generating goals, the client/product user is a source of constraints (e.g., cookware must not weigh too much for the average cook). The designer also originates constraints, including time and knowledge limitations, as well as ''autonomous constraints'' [60]. These are discretionary commitments by which designers narrow the alternative space; a composer, for instance, chooses to write in a certain key. Because a system's effects often extend beyond its immediate users, other stakeholders are implicated in design. Notable among these are governmental bodies, whose regulations (e.g., building codes) restrict system specifications. Finally, because systems must fit a larger context of use, design is constrained by characteristics of the environment. By this we mean the existing and expected world of artifacts, institutions, and social relationships. As Jones [37] observed, design is complicated by the need to consider higher level systems and the larger community. It may be technically feasible to make an automobile that is 10 feet wide, but the dimensional constraints imposed by existing highways, garages, and other parts of our built world militate strongly against such designs.

Regardless of origin, each constraint has certain content; it relates to particular attributes of the artifact or the processes by which it is designed or manufactured. There are, for instance, resource constraints respecting the availability of time, money, materials, or expertise. Physical characteristics of the artifact—its size, weight, structure, appearance—can be restricted for various reasons. Statements of requirements often set performance criteria concerning such matters as operational efficiency and reliability. A constraint can be further described in terms of qualitative attributes: is it ''hard'' or ''soft'' [80], more or less firm? And is it primary or subordinate [77], a critical or less significant demand?

As the foregoing suggests, every design problem involves a huge set of potentially relevant constraints and considerations. This is the problem's **context**, defined as ''anything in the world that makes demands of the form'' [4, p. 19]. The context includes the designer's profession, participating organizations, culture, history, the physical environment, and any system impinging on the artifact [15]. The context cannot be fully described [4] or anticipated [58], since the artifact will alter its environment in unforeseeable ways [10]. Also, many constraints only come to bear if particular solution alternatives are pursued [33]. Consequently, while tasks are introduced to designers by means of briefs that specify requirements and constraints [27], such specifications are radically incomplete [15], [18], [30].

Given that constraints must be identified, how can this be done? Coherence constraints—those deriving from the problem's partial solution—are identified most readily, it being necessary for the designer to recognize commitments entailed by the emerging design. Reitman's [57] study of fugue composition demonstrates how a composer's design choices direct and constrain subsequent developments. AI researchers have formalized this process in systems that ''propagate commitments,'' generating the new constraints implied by each successive design choice [13], [74]. Other constraints are less easily discovered. Descriptive research points to the value of experience [21]. Or constraints can be identified through analysis of inadequacies in proposed solutions [41], [48]. This is consistent with the claim that design is driven by perceptions of misfit. [4], and with rapid prototyping strategies for the design of information systems.

But constraints are not just recognized and complied with; innovative design often involves modifying or even ignoring certain requirements. Designers must avoid the ''puzzle trap'' [40] of imposing unnecessary constraints (as in the classic nine-dot problem), and they must be alert to needless constraints set by others (e.g., clients). The definition of a design problem ''is a matter of deciding just how much of what already exists can be called into question'' [40, p. 42]. Problem space boundaries are negotiated [26], with certain constraints being relaxed to satisfy conflicts [30].

AI researchers [28], [29], [65] have attempted to treat design as a constraint satisfaction problem, solvable by search within a constraint space. These attempts have met with limited success and seem overmatched by real world design problems [13]. In practical design, the set of relevant considerations is huge and never maps neatly into well-defined conceptual spaces; it is impossible to generate all pertinent constraints; and innovative design does not just accept constraints as given, but revises and even creates such as a means of directing attention.

### C. Alternatives

Alternatives are a knowledge-level phenomenon, possibilities that can be mentally represented, as opposed to perceptible physical things. Along with goals and constraints, they are part of the design *problem space*. A fundamental category in information processing accounts of problem solving [70], problem spaces include the initial state, the goal state, all other states that can be considered, all possible state-to-state moves or operators, and all relevant knowledge [69]. The problem space concept, useful in understanding how humans address well-structured problems, has been employed in research on design [2], [26], [47], [52], [60]. But there are concerns about its applicability to design, the concept having been transformed in such applications. Brown and Chandrasekaran [13] argued that design does not have a unique problem space. Heath [30] proposed that design spaces are unstructured and person-dependent. Goel and Pirolli [26] presented an analysis of the design problem space that reads like a description of design problem solving. These conceptual revisions reflect the nature of design alternatives. Alternatives are not pre-existing, like moves in a chess game; they can emerge as full-fledged (but undetailed) product concepts, rather than being accumulated through a sequence of steps; the goals, constraints, and other knowledge pertinent to a problem are determined by the alternatives considered, rather than being part of a prespecifiable problem space [40]. Under such conditions, the notion of problem space reduces to that of mental representation.

Alternative generation is the most distinctive functional demand posed by design problems. Whereas alternatives in decision problems are known or can be discovered through search, design alternatives must be created. They can range from high-level design concepts for the overall system to proposals for minor components. Generation methods vary in the extent to which they employ reasoned analysis, experiential knowledge, and creative imagination.

Analytical approaches to alternative generation presume complete knowledge of requirements and often rely on decomposition. Archer's work [7], [8] is typical. Known goals and constraints allow alternatives to be conceived in terms of control variables that can be manipulated to achieve acceptable designs. Analytical strategies for generating alternatives have been pursued by some AI researchers. Transformation methods, often employing formal grammars, translate from one language to another, as from function to structure [47]. Constraint satisfaction techniques produce alternatives by exploring a constraint space [28]. Decomposition can reduce design to a search for primitive objects (e.g., the bolts, shafts, and gears constituting a machine) to be combined in constraint-satisfying configurations [13]. These methods treat alternative generation as a matter of reasoning from an appropriate conceptualization of the problem.

Analytical methods are vulnerable to charges that alternative spaces are too large and unstructured to be efficiently explored and that requirements cannot be prestated. Darke's [18] empirical research suggested that designers use a few objectives to reach an initial design concept that directs subsequent problem solving activity. Her notion of the "primary generator" motivated Hillier, Musgrove, and O'Sullivan's [31] claim that design is conjectural: rather than deeply analyzing a problem and then synthesizing a solution, designers quickly propose an alternative, using this conjecture to identify goals and constraints, thereby reducing the problem space. It is now accepted that "often *solutions* will be imagined before the corresponding *problem* is articulated" [15]. Early generation of alternatives promotes problem space development and exploration [17], [37].

But if conjecturalist accounts of design undermine analytical methods, they do not say how alternatives are generated. What has emerged in their support is a focus on the designer's memory or experiential knowledge as a source of design proposals. Echoing the cognitivist conclusion that problem solving expertise lies in domain knowledge rather than analytical methods [22], theorists argue that skilled designers have elaborate knowledge structures, also called schemas [35] or prototypes [25]. These include procedural knowledge (e.g., how to decompose a problem); declarative knowledge of the functions, structures, and behaviors of artifacts [25]; and experiential knowledge of previous problems, processes, and solutions. The latter support alternative generation by case-based reasoning [47]: recall relevant cases, select the most promising, and adapt its solution to fit the current problem. Criticizing traditional design methods as "misleading because they portray the designer as overly analytical," Klein [39, p. 176] argued that "much of the strength of an experienced designer comes from his ability to recognize the types of problem encountered, to recognize the typical ways of handling such issues, and to recognize the implications of contextual nuances."

Significantly, a designer's knowledge is not limited to the individual's direct experience. Design fields accumulate stores of collective knowledge that is made available through training and apprenticeship. Much of this knowledge exists as *design types* (or prototypes), exemplary solutions to certain kinds of problems [60]. Per Schon, "types should be seen as particulars that function in a general way, or as general categories that have the *fullness* of particulars" [67, p 183]. A field's stock of design types provides the designer with a repertoire of proven, ready-made, solution alternatives that can be adapted to the situation at hand. Empirical research demonstrates that this repertoire is frequently employed [3] [60]. Design types are apparent even in nascent fields like organization design, where functional and divisional forms have type status.

If alternatives cannot be derived analytically or recalled from memory, a final possibility is to generate them through a creative process. Though creativity is not well understood [55], it is assumed that creative products are more original than what can simply be recalled or inferred. Accounts of how alternatives can be creatively generated are varied and vague. Creativity techniques are commonly included in the arsenal of design methods [37]. Many focus on metaphor and analogy, the identification of relational parallels between the problem situation and a remote domain of experience [12], [60]. Others stress the importance of visualization, such as imagining scenarios involving interaction with the artifact [3]. It has been proposed that objects can be seen from various conceptual points-of-view, and that adopting different perspectives can be generative [14]. The notion of pattern has also been employed, researchers arguing that alternative generation involves making patterns or perceiving such in the problem space [37].

The relative prominence of memorial, as opposed to creative, processes in alternative generation drives many distinctions among design tasks [11], [13], [68]. For instance, Gero [25] distinguished between 1) routine design, in which all variables and their ranges can be determined from the designer's existing knowledge; 2) innovative design, in which there is a well-defined state space of potential designs, but the values of certain variables must be adjusted outside their normal ranges; and 3) creative design, which uses new variables to produce new artifact types, generating a new space of potential designs. A single problem could include routine, innovative, and creative subproblems.

In contrast to decision making, where alternative generation is usually completed before evaluation is begun [5], in design there is a close interaction between these activities. Design consists largely of nested cycles of alternative generation and evaluation, alternating phases of producing and reducing variety [58]. Few alternatives are generated during any cycle [78], but due to extensive problem decomposition, there are many cycles within even modest design projects. Alternative evaluation supports discovery of goals and constraints, knowledge of which informs subsequent generative activities [79]. An evaluated alternative might be rejected out of hand or, if it seems to have promise, the designer will decompose it, specifying subproblems to be resolved as part of the option's development. Consequently, the alternatives considered are often variations on an organizing principle [60], potential means of adapting an interesting higher order solution concept to the current situation. Alexander [4] characterized design as an error-correcting process in which inadequacies are identified and remedied as part of a gradual movement towards fit between form and context. Inadequacies are often discovered through actual use of the artifact, but evaluation efforts within the design process try to find such weaknesses before a physical artifact is constructed. Performance monitoring possibilities vary from one design domain to another. Proposal evaluation is more effective in fields like industrial design, where well-understood physical entities and processes are involved, than in a field like organizational design or strategic planning, where little causal knowledge is available.

### D. Representations

A representation is a depiction of something else. The notion encompasses pictures, verbal statements, and models, as well as the internal mental representations that drive human thinking. Per Simon, "a deeper understanding of how representations are created and how they contribute to the solution of problems will become an essential component in the future theory of design" [71, p. 153]. Representations have been integral to design practice since the leap forward from traditional trial-and-error methods of artifact creation. Design-by-drawing enabled the division of design labor and the communication of results required for the construction of complex artifacts [37]. Design has become the production of representations, rather than of systems per se, as a result of the split between design, production, and use in modern societies. Software engineering, for instance, involves an elaborate set of representations, ranging from flowcharts, through data flow and entity-relationship diagrams, to programming languages [64].

Representations are the language of design, a partially verbal, but primarily visual system of codes. The true product of design is a representation or plan for artifact construction that communicates specifications to those who actually make the product. But communication is not the primary purpose of representations; its major use is in support of the design process. Any system embodies a vast set of relationships among components and attributes. A representation models the most significant relationships, allowing designers to explore their implications. A diagram is effective to the extent that it includes physical or otherwise real implications for the system [4] or, equivalently, to the extent that it supports visual apprehension of the problem's relational structure [61]. By so doing, representations can support the translations, as from function to structure [25], that design effects; it can serve as a record of commitments, helping to maintain system coherence [26]; and it can help the designer discover unrecognized constraints [79].

Representations strongly influence design processes and products. Rowe [60] argued that the development of drawing techniques (e.g., the rendering of perspective) had a major influence on architectural design. Empirical research supports this claim. Eastman [21] found a correspondence between the representations used and the constraints designers are able to consider. In an experimental study, subjects who had access to suitable representations generated better designs faster [48]. Rusch [61] concluded that graphic activity can stimulate major reorganizations of design material and changes in the high-level concept being developed.

The variety of design representations reflects the available modeling languages, the various design contents that can be depicted, and the different levels of abstraction that can be adopted. Simon's [71] taxonomy of representations includes natural language, mathematical models, diagrams of physical objects and processes, and three-dimensional models. All are employed in design. Empirical studies have shown that individual designers use multiple representations during a project. Asked to create a system to allow consumers to pay bills by phone, a designer employed procedural representations of action scenarios, solid models, matrices of information, orthographic projections, notations, perspective drawings, and dimensions [10]. Indeed, representational systems are constantly being created, Eastman arguing that "most methodologies are in fact new representations that allow explicit comparison of information not previously relatable" [21, p. 30].

The selection of the representational mode is influenced by task and the designer's familiarity with available options. Ballay [10] found that representations are used episodically, with the designer attempting to do all that can be done in one mode at a time. He argued that a representation develops along three dimensions: inclusion or comprehensiveness; coherence, the degree of fit among components; and precision, the degree of dimensional specificity. Regarding inclusion, "the information in a sketch seems to get added in a consistent sequence that is partly idiosyncratic (the result of a designer's training and experience) and partly a response to the demands of the particular design problem" [10, p. 76]. Precision increases with the move from initial sketches to final products. But "imprecision seems to have a value of its own" [10, p. 77]: Low precision drawings indicate that content is tentative, marking issues that require further development. Representational activity is complete when the design description has sufficient information to enable system construction [25]. This is often indicated by representation in a conventionalized, domain-dependent, specification language (e.g., a musical score, programming language), although designers can add details beyond these requirements [26].

The power of representation contributes to what Lawson [40, p. 164] calls "the icon trap." One fails to recognize the limitations of models, mistaking the map for the territory. As a result, "problems which are not visually apparent tend not to come to the designer's attention" [40, p. 18]. Arguably, some problem contents are more amenable to representation than others [73]. A domain's whole notion of what is to be designed may be determined by what can be represented: organization design, for instance, may be too strongly anchored on the authority and inclusion relationships depicted in classic organization charts.

### E. Solutions

The solution to a design problem is a system or, more commonly, a description enabling system construction [25]. Solutions have several noteworthy characteristics. First, there is a great difference between problem description and solution description. Though many problems are stated in ways suggestive of possible solutions (e.g., the givens in a theorem proving task include elements that define solution states), in design, problem statements are often expressed in language remote from solution description. For instance, the need for warmth allows many responses—clothing, a heat source, shelter, relocation to a milder climate—each requiring specification in terms (e.g., blueprints for a dwelling) having little direct connection to the goal. The disparity between what is wanted and how that want can be satisfied motivates Rittel's [58] observation that solution space identification is a recurring difficulty in design. The disparity is also responsible for the pitfall of defining design problems in terms of solutions [32], [36], [40]: viable alternatives are removed from consideration by requirements statements calling for a certain kind of response.

Second, design solutions are complex [26]. Artifacts are complex systems of components having many attributes and interrelationships. Design descriptions must mirror this complexity if they are to support artifact construction. Third, as a consequence of complexity, it is difficult to validate design solutions. "There is no way of deciding beyond doubt when a design problem has been solved" [40, p. 40], since the design description may not be of an acceptable artifact. Validation difficulties underlie Rzevski's [62] claim that the best solutions are those which can be easily modified.

Fourth, most design problems have many acceptable solutions [28], [40]. In conjunction with solution complexity and validation difficulties, this is responsible for the satisficing nature of design problem solving [30]. Lawson states that "there are no optimal solutions to design problems" [40, p. 88]. More moderately, optimal design is a practical impossibility. Finally, while designers rarely generate more than one acceptable solution, there are usually enough degrees of freedom in the problem statement to allow subjective factors to influence process outcomes [40]. Several of these issues will be discussed in more detail.

Design problem complexity is widely acknowledged [2], [38] [71]. It stems from the fact that the intended artifact is a system of richly interactive components that must function in an unpredictable and even more complex environment. Humans generally respond to complexity by decomposing, reducing wholes into parts. Design is no exception. While nonreductionist solution strategies (e.g., case-based reasoning) can be used, decomposition is the prototypical means of addressing design problems [26], [35], [69]. The solution to a design problems "comes into being in stages . . . because the designer cannot tackle the whole problem in one fell swoop" [43, p. 118]. Design affords decomposition because systems can be understood as hierarchies of components at different levels [62], [71]. Levels are usually defined around the system's structural parts (e.g., the rooms of a dwelling) or constitutive functions (e.g., the fuel system of an automobile). While virtually all problems can be decomposed [72], design problems are especially recursive: they decompose into ever smaller design problems [16], [30]. Paraphrasing Wade [76], the design process is one of "make it or break it": Either design a solution for the current problem or decompose into subproblems and try again. When the "make it" option is selected, the functional demands of identifying goals and constraints, and generating and evaluating alternatives, become relevant. Thus, in the overall design process, these activities are performed many times for tasks at different hierarchical levels [45].

Decomposition is not a trivial task and there is evidence that much design skill lies in knowing when and how to decompose [35]. Since one usually decomposes into functional or structural parts, it is easiest when these dimensions are congruent, functions mapping neatly into structural elements [74]. Jones [37] distinguished "splittable" from "unsplittable" design problems on this basis, the latter being more difficult. For instance, while all the walls of a house serve as partitions, the load-bearing walls also provide structural support for the building. Redesign for the sake of repartitioning is complicated by the fact that such walls cannot be removed without providing another means of support. As Wise points out, "the best designs *are* the best designs because their partial physical and functional descriptions are richly interdependent" [81, p. 291]. Artifacts often serve multiple purposes, which the design must achieve in a structurally integrated whole. As a result, problems are, at best, "nearly decomposable" [71], with components being interactive or "leaky" modules [26]. Due to component interaction, subproblems cannot be solved in isolation. Since "the whole is at stake in every partial move" [66, p. 101], it is best to address the hardest subproblems first, to ensure that they can be solved, thereby avoiding the backtracking and wasted effort that results when irremediable tasks are uncovered late in the process [37]. The ability to identify critical (i.e., difficult) subproblems is another mark of designer skill [52].

Decomposition also creates the need for recomposition, integrating components into a whole. This can be viewed as a constraint satisfaction issue [47], each component creating coherence constraints to be satisfied by components with which it interacts. Theorists have proposed a "least commitment" strategy in which components are designed so as to minimally constrain other elements [26], [51]. The initial modules of an information system might be designed to be relatively standardized or self-contained, minimizing the restrictions they create for subsequent elements. All this highlights the fact that designed artifacts are systemic, explaining the prominence of systems concepts and methods in the design literature [23], [37], [71].

While designs are systems, they can also be seen as settings of values of design parameters—controllable variables—which result in desired values of dependent performance variables [43]. Termed "parametric design" [65], this approach was pursued during the design methods movement of the 1960s, when optimization and other OR/MS techniques attracted the interest of design theorists. Parametric design is thoroughly decompositional, reducing problems to sets of variables and relationships. There usually is no formal recomposition, it being assumed that part-whole relations can be captured in the low-level problem representation. While many theorists reject such an approach [4], [15], [17], [79], others are willing to think about design as a problem of selecting variables and setting their values [16], [25]. Rittel [58] proposed that the central difficulty of designing was constructing a system of functional relationships that connect design, context, and performance variables. Certainly the parametric approach is conceptually valid, in that design problems can be conceived in those terms.

However, its practical value is limited. Boundary searching [37] is a parametric method that explores the range of acceptable values for dimensions of an artifact. It presumes that relevant dimensions have been determined. But these cannot usually be identified until problems have been deeply decomposed. Consequently, parametric methods can only be used for routine problems or after the most difficult and creative parts of design work have been completed. Other difficulties reinforce these limitations: relations among variables are often not understood or explicitly specified. Designers rely on experience to identify alternatives that implicitly satisfy requirements. The number of potentially relevant variables, and the related size of the problem space, dwarfs the capacities of existing optimization techniques. And, of course, solving for each variable in isolation is pointless, due to the artifact's systemic nature.

Another objection to parametric design, and to any strictly "objective" design method, is the claim that solutions necessarily include subjective commitments by the designer, over and beyond the goals that motivate problem solving. Despite the many requirements that artifacts must satisfy, the set of acceptable designs is usually quite large, this underdetermination providing room for designer discretion [26]. Indeed, intervention is needed to structure the solution space and rank possibilities [30], [66]. Designers impose an "ordering principle" [43], often in the form of aesthetic criteria [30]. Consequently, design solutions cannot be identified or assessed in purely objective terms.

The latitude allowed subjective considerations drives distinctions among design activities in different fields. The key issue is the importance of functional requirements in the final product. There is agreement that architecture occupies a middle ground between engineering design and art [40], [80]. Engineering design is dominated by functionality, whereas art is exclusively concerned with form [9], [15], [30]. Architecture encompasses both considerations—a building has to satisfy user requirements and be attractive—although the profession's focus has evolved toward aesthetic concerns, with functional issues (e.g., heating, lighting) being passed off to engineers [6]. When functional demands are less constraining, there is more room for designers to include personal values and ordering principles in the design process.

## III. Conclusion

Design problems pose the following functional demands: the designer must identify relevant goals and constraints and must generate alternatives that reflect these requirements. Certain alternatives will be depicted in overt representations, enabling their elaboration and evaluation, and ultimately the selection of one as the preferred solution. As the preceding section suggests, these basic tasks sustain an effectively infinite set of variations, accounting for the difficulty of design problem solving and the variety of design aids.

If design is difficult per se, it is especially challenging in domains addressed by the systems and management sciences. This is due in large part to the lack of scientific understanding of domain phenomena. For instance, despite the important contributions of organizational theorists [50], we still do not adequately understand how organizations function. How then can we design them to function effectively? This lack of knowledge ramifies throughout the design process, making it difficult to identify goals and constraints, to determine viable decompositions, or to evaluate alternatives. Many of our knowledge deficiencies reflect the fact that humans are part of designed managerial systems, not just users of such systems. Viewed as system components, humans are incredibly complex and behave in ways that are difficult to predict or control.

Another difficulty derives from the nonphysical nature of many of these systems (e.g., organizations, business strategies, incentive systems). Being nonphysical, such entities do not lend themselves to graphic representation. While representational media have been developed, their ability to support the design process is questionable. Moreover, few of these design fields have anything like a specification language, a bottom-line representational system at which design activity can stop and from which the artifact can be constructed.

To illustrate these points, consider the field of software engineering. Information systems are of obvious importance in managerial affairs. Their design benefits from the fact that system functioning is driven by a fully specified, deterministic machine (the computer). As a result, the software engineering field has been able to develop a rich array of representational devices [64], including specification languages (programming languages) that precisely define system behavior. However, the major difficulties of software engineering occur at just those places where humans come fully on the scene: in the requirements determination phase and in the design of user interfaces. Design is difficult in managerial domains because humans are almost always "fully on the scene."

In *The Sciences of the Artificial* [71] first published in 1969, Simon proposed that most applied scientific disciplines were concerned with artificial or man-made, rather than natural, phenomena. They are, incipiently at least, design sciences. Design researchers [12], [17] have been unsure about the design-science relationship, there being controversy over whether design per se is scientific activity. Arguably, design is not itself a scientific process; designers do not act as scientists or produce scientific results. However, design can be the target of scientific investigation, design science being the intended product of such. Talk of science leads inexorably to demands for theory, explanations of relevant phenomena that are both deep and coherent. Critics of design research might charge that decades of activity have yet to produce a widely-accepted general theory of design. This criticism misunderstands the goal of science in such fields. Like management, design is too broad to be encompassed by unified general theories that are substantively deep. Rather, design science must focus on narrower targets, such as understanding the mental activities of designers, or developing design methods and aids. In doing so, design research properly appropriates relevant scientific theories from other fields, such as the information processing theory of human problem solving [56]. Such research must also be well-grounded conceptually, a need that this paper has attempted to satisfy.

## References


[1] G. P. Agre, "The concept of problem," *Educational Studies*, vol. 13, pp. 121-142, 1982.

[2] O. Akin, *Psychology of Architectural Design*. London: Pion Limited, 1986.

[3] —, "Expertise of the architect," in *Expert Systems for Engineering Design*, M. D. Rychener, Ed. New York: Academic Press, 1988.

[4] C. Alexander, *Notes on the Synthesis of Form*. Cambridge, MA: Harvard Univ. Press, 1964.

[5] E. R. Alexander, "The design of alternatives in organizational contexts: A pilot study," *Administrative Sci. Quarterly*, vol. 24, pp. 382-404, 1979.

[6] J. Archea, "Puzzle-making: What architects do when no one is looking," in *Computability of Design*, Y. E. Kalay, Ed. New York: John Wiley, 1987.

[7] L. B. Archer, "An overview of the structure of the design process," in *Emerging Methods in Environmental Design and Planning*, G. T. Moore, Ed. Cambridge, MA: MIT Press, 1970.

[8] —, "Systematic method for designers," in *Developments in Design Methodology*, N. Cross, Ed. New York: Wiley, 1984

[9] M. Asimow, *Introduction to Design*. Englewood Cliffs, NJ: Prentice-Hall, 1962.

[10] J. M. Ballay, "An experimental view of the design process," in *System Design*, W. B. Rouse and K. R. Boff, Eds. New York: North-Holland, 1987.

[11] K. R. Boff, "The Tower of Babel revisited: On cross-disciplinary choke-points in system design," in *System Design*, W. B. Rouse and K. R. Boff, Eds. New York: North-Holland, 1987.

[12] G. Broadbent, *Design in Architecture*. London: David Fulton Publishers, 1988.

[13] D. C. Brown and B. Chandrasekaran, *Design Problem Solving*. London: Pitman, 1989.

[14] L. L. Bucciarelli, "An ethnographic perspective on engineering design," *Design Studies*, vol. 9, pp. 159-168, 1988.

[15] L. L. Bucciarelli, G. Goldschmidt, and D. A. Schon, "Generic design process in architecture and engineering," in *Proceedings of the 1987 Conference on Planning and Design in Architecture*, J. P. Protzen, Ed. American Society of Mechanical Engineers, 1987.

[16] B. Chandrasekaran, "Design problem solving: A task analysis," *AI Magazine*, vol. 11, pp. 59-71, 1990.

[17] N. Cross, "Designerly ways of knowing," *Design Studies*, vol. 3, pp. 221-227, 1982.

[18] J. Darke, "The primary generator and the design process," *Design Studies*, vol. 1, pp. 36-44, 1979.

[19] L. J. Davies, "Designing from ill-defined problems," *Int. J. Informat. Manage.*, vol. 9, pp. 199-208, 1989.

[20] J. R. Dixon, *Design Engineering: Inventiveness, Analysis, and Decision Making*. New York: McGraw-Hill, 1966.

[21] C. M. Eastman, "On the analysis of intuitive design processes," in *Emerging Methods in Environmental Design and Planning*, G. T. Moore, Ed. Cambridge, MA: MIT Press, 1970.

[22] E. A. Feigenbaum, "The art of artificial intelligence: Themes and case studies of knowledge engineering," in *Proc. Int. Joint Conf. Artific. Intell.*, 1977.

[23] R. Foque, "Beyond design methods: Arguments for a practical design theory," in *Changing Design*, B. Evans, J. A. Powell, and R. Talbot, Eds. New York: John Wiley, 1982.

[24] W. W. Gasparski, "Praxiological-systemic approach to design studies," *Design Studies*, vol. 1, pp. 101-106, 1979.

[25] J. S. Gero, "Design prototypes: A knowledge representation schema for design," *AI Magazine*, vol. 11, pp. 26-36, 1990.

[26] V. Goel and P. Pirolli, "Motivating the notion of generic design within information-processing theory: The design problem space," *AI Magazine*, vol. 10, pp. 18-36, 1989.

[27] G. Goldschmidt, "Problem representation versus domain of solution in architectural design teaching," *J. Architect. Planning Res.*, vol. 6, pp. 204-215, 1989.

[28] M. Gross, S. Ervin, J. Anderson, and A. Fleisher, "Designing with constraints," in *Computability of Design*, Y. E. Kalay, Ed. New York: Wiley, 1987.

[29] —, 'Constraints: Knowledge representation in design," *Design Studies*, vol. 9, pp. 133-143, 1988.

[30] T. Heath, *Method in Architecture*. New York: Wiley, 1984.

[31] B. Hillier, J. Musgrove, and P. O'Sullivan, "Knowledge and design," in *Developments in Design Methodology*, N. Cross, Ed. New York: Wiley, 1984.

[32] A. Holgate, *The Art in Structural Design*. Oxford: Clarendon Press, 1986.

[33] R. M. Hunt, "The difficulties of design problem formulation," in *System Design*, W. B. Rouse and K. R. Boff, Eds. New York: North-Holland, 1987.

[34] K. Ishikawa, *Introduction to Quality Control*. London: Chapman and Hall, 1990.

[35] R. Jeffries, A. A. Turner, P. G. Polson, and M. E. Atwood, "The processes involved in designing software," in *Cognitive Skills and Their Acquisition*, J. R. Anderson, Ed. Hillsdale, NJ: Lawrence Erlbaum, 1981.

[36] J. C. Jones, "A method of systematic design," in *Conference on Design Methods*, J. C. Jones and D. Thornley, Eds. Oxford: Pergamon, 1963.

[37] —, *Design Methods*. New York: Wiley, 1980.

[38] S. Kendall, "On design reasoning: Lessons from teaching architectural technology," *Design Studies*, vol. 10, pp. 89-95, 1989.

[39] G. A. Klein, "Analytical versus recognitional approaches to design decision making," in *System Design*, W. B. Rouse and K. R. Boff, Eds. New York: North-Holland, 1987.

[40] B. Lawson, *How Designers Think*. London: The Architectural Press, 1980.

[41] S. G. Lera, "Empirical and theoretical studies of design judgment: A review," *Design Studies*, vol. 2, pp. 19-26, 1981.

[42] —, 'Synopses of some recent published studies of the design process and designer behaviour," *Design Studies*, vol. 4, pp. 133-140, 1983.

[43] P. H. Levin, "Decision-making in urban design," in *Developments in Design Methodology*, Nigel Cross, Ed. New York: Wiley, 1984.

[44] B. S. Logan, "Conceptualizing design knowledge," *Design Studies*, vol. 10, pp. 188-195, 1989.

[45] J. Luckman, "An approach to the management of design," *Operat. Res. Quarterly*, vol. 18, pp. 345-358, 1967.

[46] M. L. Maher, "Synthesis and evaluation of preliminary designs," in *Artificial Intelligence in Design*, J. S. Gero, Ed. New York: Springer-Verlag, 1989.

[47] —, "Process models for design synthesis," *AI Magazine*, vol. 11, pp. 49-58, 1990.

[48] A. Malhotra, J. C. Thomas, J. M. Carroll, and L. A. Miller, "Cognitive processes in design," *Int. J. Man-Machine Studies*, vol. 12, pp. 119-140, 1980.

[49] D. Meister, "A cognitive theory of design and requirements for a behavioral design aid," in *System Design*, W. B. Rouse and K. R. Boff, Eds. New York: North-Holland, 1987.

[50] H. Mintzberg, *The Structuring of Organizations*. Englewood Cliffs, NJ: Prentice-Hall, 1979.

[51] T. M. Mitchell, L. I. Steinberg, and J. S. Shulman, "A knowledge-based approach to design," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 7, pp. 502-510, 1985.

[52] J. Mostow, "Toward better models of the design process," *AI Magazine*, vol. 6, pp. 44-57, 1985.

[53] G. Nadler, "An investigation of design methodology," *Manage. Sci.*, vol. 13, pp. B642-655, 1967.

[54] —, *The Planning and Design Approach*. New York: Wiley, 1981.

[55] —, 'Systems methodology and design," *IEEE Trans. Systems, Man, Cyber.*, vol. 6, pp. 685-697, 1985.

[56] A. Newell and H. A. Simon, *Human Problem Solving*. Englewood Cliffs, NJ: Prentice-Hall, 1972.

[57] W. R. Reitman, "Heuristic decision procedures, open constraints, and the structure of ill-defined problems," in *Human Judgments and Optimality*, M. W. Shelly and G. L. Bryan, Eds. New York: Wiley, 1964.

[58] H. W. J. Rittel, "Some principles for the design of an educational system for design," *Design Methods and Theories*, vol. 20, pp. 359-375, 1986.

[59] W. B. Rouse, *Design for Success*. New York: Wiley, 1991.

[60] P. G. Rowe, *Design Thinking*. Cambridge, MA: MIT Press, 1987.

[61] C. W. Rusch, "The role of graphic activity in the design process," in *Emerging Methods in Environmental Design and Planning*, G. T. Moore, Ed. Cambridge, MA: MIT Press, 1970.

[62] G. Rzevski, "On the design of a design methodology," in *Design: Science: Method*, R. Jacques and J. A. Powell, Eds. Guildford, UK: Westbury House, 1981.

[63] A. P. Sage, *Systems Engineering*. New York: Wiley, 1992.

[64] A. P. Sage and J. D. Palmer, *Software Systems Engineering*. New York: Wiley, 1990.

[65] M. Sapossnek, "Research on constraint-based design systems," in *Artificial Intelligence in Design*, J. S. Gero, Ed. New York: Springer-Verlag, 1989.

[66] D. A. Schon, *The Reflective Practitioner*. New York: Basic Books, 1983.

[67] —. "Designing: Rules, types and worlds," *Design Studies*, vol. 9, pp. 181-190, 1988.

[68] J. J. Shah and P. R. Wilson, "Analysis of design abstraction, representation, and inferencing requirements for computer-aided design," *Design Studies*, vol. 10, pp. 169-178, 1989.

[69] H. A. Simon, "The structure of ill structured problems," *Artific. Intell.*, vol. 4, pp. 181-201, 1973.

[70] —, "Information-processing theory of human problem solving," in *Handbook of Learning and Cognitive Processes*, vol. 5, W.K. Estes, Ed. Hillsdale, NJ: Lawrence Erlbaum, 1978.

[71] —, *The Sciences of the Artificial*, 2nd ed. Cambridge, MA: MIT Press, 1981.

[72] G. F. Smith, "Towards a heuristic theory of problem structuring," *Manage. Sci.*, vol. 34, pp. 1489-1506, 1988.

[73] J. C. Thomas and J. M. Carroll, "The psychological study of design," *Design Studies*, vol. 1, pp. 5-11, 1979.

[74] D. G. Ullman, T. G. Dietterich, and L. A. Stauffer, "A model of the mechanical design process based on empirical data: A summary," in *Artificial Intelligence in Engineering: Design*, J. S. Gero, Ed. Amsterdam, The Netherlands: Elsevier, 1988.

[75] D. Von Winterfeldt and W. Edward, *Decision Analysis and Behavioral Research*. Cambridge, UK: Cambridge Univ. Press, 1986.

[76] J. Wade, *Architecture, Problems and Purposes*. New York: Wiley, 1977.

[77] M. B. Waldron, "Modeling of the design process," in *Intelligent CAD, I*, H. Yoshikawa and D. Gossard, Eds. Amsterdam, The Netherlands: North-Holland, 1989.

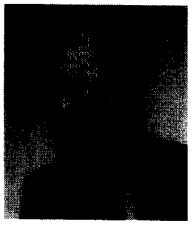
[78] A. Westerberg, I. Grossmann, S. Talukdar, F. Prinz, S. Fenves, and M. L. Maher, "Applications of AI in design research at Carnegie Mellon University's EDRC," in *Artificial Intelligence in Design*, J. S. Gero, Ed. New York: Springer-Verlag, 1989.

[79] A. Whitefield and C. Warren, "A blackboard framework for modeling designers' behaviour," *Design Studies*, vol. 10, pp. 179-187, 1989.

[80] R. A. Willem, "On knowing design," *Design Studies*, vol. 9, pp. 223-228, 1988.

[81] J. A. Wise, "Decisions in design: Analyzing and aiding the art of synthesis," in *Behavioral Decision Making*, G. Wright, Ed. New York: Plenum, 1985.

[82] H. Yoshikawa, "General design theory as a formal theory of design," in *Intelligent CAD, I*, H. Yoshikawa and D. Gossard, Eds. Amsterdam, The Netherlands: North-Holland, 1989.



**Gerald F. Smith** received the Ph.D. degree in decision sciences from the University of Pennsylvania.

He is an Assistant Professor in the Information and Decision Sciences Department, Carlson School of Management, University of Minnesota, Minneapolis. His primary area of research is managerial problem solving, with special interests in problem identification and definition, problem types, design, and quality problem solving.

**Glenn J. Browne** received the Ph.D. degree in information and decision sciences from the University of Minnesota.

He is an Assistant Professor with the Information Systems Department at the University of Maryland—Baltimore. His principal research interests are decision-making processes and decision support systems.