LBL-31742
BS-306

Presented at the Focus Symposium on "Computer-Assisted Building Design Systems" of the 4th International Symposium on System Research, Informatics and Cybernetics, Baden-Baden, Germany, August 3-4, 1993.

The Limits of Intelligence in Design

Konstantinos Papamichael

Building Technologies Program
Energy and Environment Division
Lawrence Berkeley Laboratory
University of California
1 Cyclotron Road
Berkeley, California 94720

and

Jean-Pierre Protzen
Department of Architecture
University of California
232 Wurster Hall
Berkeley, CA 94720

May 1993

This work was partially supported by the Assistant Secretary for Conservation and Renewable Energy, Office of Building Technologies, Building Systems and Materials Division of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098.

The Limits of Intelligence in Design

K. M. Papamichael
Building Technologies Program
Lawrence Berkeley Laboratory
University of California, Berkeley
Berkeley, CA 94720, U.S.A.

J. P. Protzen
Design Theories and Methods
Department of Architecture
University of California, Berkeley
Berkeley, CA 94720, U.S.A.

Abstract

A new, comprehensive design theory is presented, applicable to all design domains such as engineering and industrial design, architecture, city and regional planning, and, in general, any goal-oriented activity that involves decision making. The design process is analyzed into fundamental activities that are characterized with respect to the nature of knowledge requirements and the degree to which they can be specified and delegated to others, in general, and to computers in particular.

Throughout the history of research in design theories and methods, from the 1940s with operations research and optimization, through the 1960s with the characterization of design problems as "wicked," or "ill-defined," design has been understood as a rational activity, that is "thinking before acting." The new theory presented in this paper suggests that design is "thinking and feeling while acting," supporting the position that design is only partially rational. Intelligence, "natural" or "artificial," is only one of two requirements for design, the other being emotions. Design decisions are only partially inferred, that is, they are not entirely the product of reasoning. Rather, design decisions are based on judgment that requires the notion of "good" and "bad," which is attributed to feelings, rather than thoughts.

The presentation of the design theory extends to the implications associated with the limits of intelligence in design, which, in-turn, become constraints on the potential role of computers in design. Many of the current development efforts in computer-aided design violate these constraints, especially in the implementation of expert systems and multi-criterion evaluation models. These violations are identified and discussed in detail. Finally, specific areas for further research and development in computer-aided design are presented and discussed.

Keywords

computer; decision; design; intelligence; judgement; knowledge; plan; quality; theory; tool.

Introduction

A major effort is under way during the recent years to utilize the power of computers to assist designers directly in the design process, possibly at the "early stages of design", where most of the important design decisions are made. The general approach relies on the use of Artificial Intelligence techniques, mainly in the form of the so-called Expert or Knowledge-Based Systems (Rich 83). The main idea in these efforts is the identification of "rules" in the form of *conditional* *statements* that are then used in *automated logical inferences*. Example applications within the building design domain include *diagnosis* of problems with various types of building equipment (Haberl et al. 1989; Ruberg and Cornick 1988), *selection* of various building components and systems (Degelman and Kim 1988; Tuluca et al. 1989), and various "intelligent" design systems (Pohl and Chapman 1988; Cameho and Hittle 1989; Pohl et al. 1990; Case et al. 1990; Tham et al. 1990; Jafar at al. 1991; Mayer et al. 1991).

These initial attempts have led to the identification of problems in *knowledge acquisition* and *representation*, as well as *integration* with existing software (Hall and Deringer 1989). The reason for such problems is the lack of a comprehensive theory about design, which would serve as the foundation for the development computer-based design tools. Most of the efforts to-date concentrate on computer modeling issues, assuming design theories that do not represent the design process accurately or explicitly enough at the level required for computer-based applications. The need for a comprehensive design theory is now more apparent than ever before.

Background

The attempts to understand and handle design problems in a systematic way have been classified into two generations (Rittel 1972). The first generation was initiated during the Second World War and was based on the concept of *systems analysis* dictating the handling of design and planning problems in a rational, straightforward, systematic way, following specific steps, or *phases*. There are several variations on what these *phases* are, however very similar: *A system's analyst understands the problem, gathers information, analyzes it, generates solution(s), implements them, tests them, and, if necessary, modifies them.* In a particular branch of the first generation of systems approach, *operations research*, the system's analyst finds the *best* solution by defining the *solution space*, the *constraints* and the *measure of effectiveness*, the latter of which is then *optimized*.

The second generation of systems approach was initiated in the late 1960s and was based on the concept of *wicked* problems, dictating the handling of design and planning problems in an *argumentative* way. Design problems were characterized as *ill-defined*, or *wicked* problems, in contrast to the *tame* problems of science and engineering. *Wicked* problems were characterized by a number of *properties* which contradict the first generation approach and limits the application of systems analysis and operations research to *tame* problems: *Wicked problems have no definite formulation. Every formulation corresponds to a statement of the solution and vice versa. There is no stopping rule for wicked problems and the terms "correct" and "wrong" are not applicable to them, while there are no exhaustive, enumerable list of permissible operations. Wicked problems are seen as discrepancies between a situation as is and a situation as it ought to be, and can be considered as symptoms of other wicked problems of a higher order. In addition, there is neither an immediate nor an ultimate test to check the appropriateness of solutions. Wicked problems are essentially unique and their treatment is the equivalent of a one-shot operation, for which the wicked problem solver, in contrast to the scientist, has no right to be wrong* (Rittel 1972).

Both generations of systems approach accept design as a *rational* activity, that is, *thinking before acting*. The second generation of systems approach offers a further elaboration based on *the four paradoxes of rationality* to conclude that *design is not possible* (Rittel 1972; Papamichael 1991).

However, this major theoretical conclusion has been accepted as a paradox rather than as an indication of a theoretical shortcoming. Moreover, although the second generation of systems approach identifies the theoretical shortcomings of the first generation, it still accepts and follows its methods and procedures, especially with respect to multi-criterion evaluation and optimization (Musso and Rittel 1967; Rittel 1985). A more careful consideration is then necessary to resolve these issues and establish a comprehensive design theory that is consistent and compatible with design practice.

The theoretical model of the design process presented in the next section is applicable to all design domains. The design process is analyzed into fundamental activities, which are then further and continuously analyzed hierarchically until the very basic, elementary design activities are identified and explicitly specified. Moreover, the related design knowledge is characterized with respect to origin, specifiability, reapplicability and openness to argumentation.

Design Theory

Design is defined as *an activity aimed at producing a plan which is expected to lead to a situation with specific intended properties and without undesired side- or after-effects* (Rittel 1985; Papamichael 1991). Based on the above definition, design presupposes a discrepancy between a situation *as is* and a situation *as it ought to be*, involving at least four distinct activities:

- the formulation of the specifications of the *as-is*, or *current* situation,
- the formulation of the specifications of the *ought-to-be*, or *desired* situation,
- the generation of a *plan* to lead from the *current* situation to the *desired* one, and,
- the checking for undesired side- and after-effects.

However, the specifications of the *desired* situation are formulated in terms of *performance* characteristics, while a plan is formulated in terms of *descriptive* characteristics. As a result, designers need to somehow *translate* descriptive characteristics into performance ones, an activity referred to as *simulation*. Moreover, consideration of identified undesired side- or after-effects is the equivalent of *updating* the performance specifications of the desired situation. Also, when the simulated performance characteristics do not match the desired ones, then the descriptive characteristics are *modified* in hope for a *solution*, or the desired performance characteristics are *degraded*, or the designer *gives up*, which is the equivalent of accepting the *current situation*. Even if a solution without side- or after-effects has been found, the performance specifications of the desired situation may be *improved* in hope for a *better solution*.

Based on the above considerations, the activity of design can be further decomposed into seven main activities, three initial ones and five that are performed iteratively according to five main decisions (Figure 1). All activities contribute towards the development of three sets of specifications:

- the specifications of the *desired performance,*
- the specifications of the potential *future description*, and,
- the specifications of the potential *future performance*.

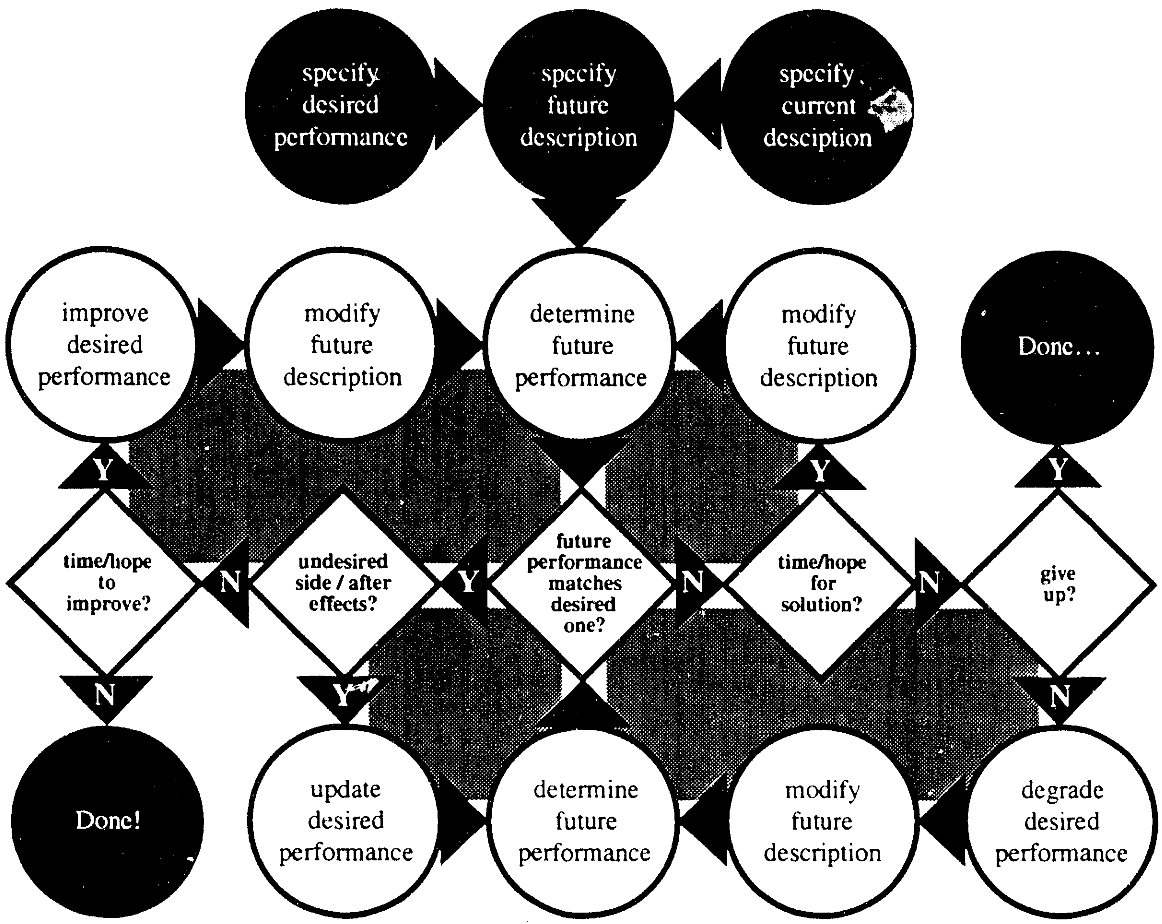

Figure 1. Schematic diagram of the design process.

Formulation of the *desired performance*

Performance is specified through the use of *performance variables*, that is variables based on the values of which designers judge the *appropriateness* of design solutions. The *desired performance* is specified as a set of *performance criteria* (or *design criteria*), which are *conditions* on the values of performance variables. The formulation, then, of the specifications of the *desired performance* requires two activities:

- The determination of the *performance variables* to be considered, and
- The formulation of the *performance criteria*.

Performance variables are determined initially based on the consideration of the design program objectives, and then during the design process through consideration of undesired side- and after-effects of potential design solutions. A performance variable can be considered either *directly*, or *indirectly* through *deliberation* towards the identification of a *set* of new performance variables whose consideration is the *equivalent* to that of the deliberated performance variable (Musso and Rittel 1967; Rittel 1985; Papamichael 1991). Since each new performance variable can also be considered through deliberation, performance variables can be realized as a hierarchical, tree-like structure, where the root represents the *overall performance* and the branches represent the *terminal performance variables*. Continuation of deliberation on the terminal performance variables results on the formulation of performance criteria, that is *conditions* on the values of performance variables that specify *good* and *bad* performance. However, "good" and "bad" are *felt*. The desired performance is then more accurately defined as a prescription of what *feels good* and what *feels bad*.

Based on the above considerations and accepting rationality as "thinking before acting," design is only *partially rational*, since it involves feeling as well as thinking. In fact, design is part of the linear process of life, which is a continuous design process of its own (Figure 2). Our *actions*, as

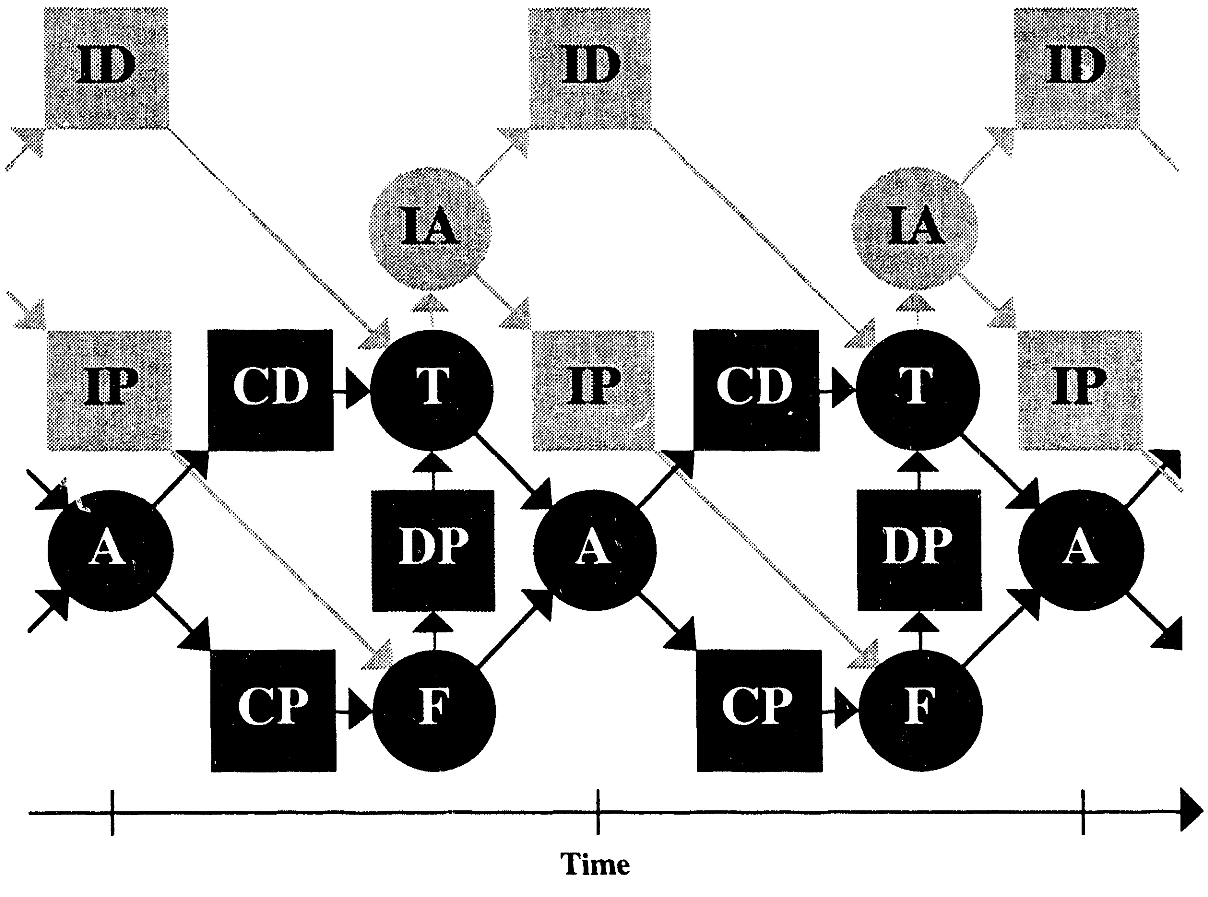

Figure 2. Design is the equivalent of living in one's imagination, however continuously affected by real life itself, which is a continuous design process of its own.

well as the actions of others, (A), affect the continuously changing current situation. At any given time we *think* (T) the *current description* (CD) and we *feel* (F) the *current performance* (CP), formulating a *desired per, rmance* (DP) and proceeding with *actions* (A). Design is the equivalent of considering *imaginary actions* (IA) that produce *imaginary descriptions* (ID), along with their associated *imaginary performances* (IP), which, however, are *thought* (T) and *felt* (F) in real life. This is how we carry design over time, interrupting and/or continuing/terminating it, according to how we feel the priorities of our life. All design decisions of the iterative model of the design process (Figure 1) are affected by real-life considerations that are not necessarily related to the specific design problem at hand. As a conclusion, design is not "thinking before acting," nor "feeling and thinking before acting." Rather, design is *"feeling and thinking while acting."*

Performance criteria may or may not be specified for all performance variables. Design is a *compromise* between what is *desirable* and what is *possible*. The design process is the equivalent of *exploring* what is *possible* under the specific design context and *adjusting* performance criteria accordingly, since what is *desirable* may not be *possible* (e.g., zero cost). The performance criteria are formulated *throughout* the design process. The desired performance is either *updated*, *improved*, or *degraded*. It is updated through the identification of side- and after-effects to include new performance variables, possibly along with the associated performance criteria. It is improved when a solution has been found but there is still enough time and hope for a better solution. It is degraded when the potential future performance does not match the desired one and there is no time or hope for a solution. This updating, improving, and degrading of the desired performance continues *throughout* the design process. *The final version of the desired performance is that of the final design solution.*

Performance variables can be measured either on nominal or continuous scales. Performance criteria that are formulated as conditions on performance variables that operate on continuous scales (e.g., cost) are called herein *quantitative*, and can be specified with respect to both, performance *acceptability* as well as performance *improvement* and *degradation*. Performance criteria that are formulated as conditions on performance variables that operate on nominal scales (e.g., esthetics) are called herein *qualitative*, and can be specified only with respect to performance *acceptability*. Quantitative criteria are specified as acceptable *value ranges*, while qualitative criteria *may be* specified as acceptable *value sets*. However, qualitative criteria are usually *not specified at all*, in which case the delegation of judgement is impossible.

The judgement on a qualitative criterion is actually a deliberated judgement, in which the branching criteria *are not independent*, that is *the appropriateness of the value of each of the branching performance variables depends on the values of the rest of the branching performance variables*. However, the branching criteria of a deliberated judgement *can never be independent*. They are always linked through their *relative importance*.

A deliberated design criterion represents the desired performance, which, however, may or may not be possible. If it is not possible, that is if a solution that satisfies all branching criteria cannot be found, then the branching criteria have to be *modified*. If it is possible, and there are two or more solutions that satisfy all branching criteria, selecting one is the equivalent of further specifying the desired performance, that is *modifying* it. Both cases, then, result in *modification* of the branching criteria. Here is where the relative importance of performance criteria is considered to select which performance criterion to modify and how. In fact, since the relative importance of performance criteria is required for their modifications, *its determination is part of the formulation of the performance criteria, rather than an independent activity*. The relative importance is what makes all branching criteria interdependent, resulting in all deliberated criteria being qualitative ones.

There are two main approaches to consider *quantitative* branching criteria of a deliberated criterion. One is through the consideration of a multidimensional space defined through acceptable value ranges for each quantitative criterion, and the other is through the consideration of an *index*, that is a function of all quantitative branching criteria (Papamichael 1991). An equivalent to this latter approach is the one most commonly used in multi-criterion evaluation models. The most sophisticated and elaborate versions require the specification of *transformation functions*, which transform the values of all terminal performance variables into *values of appropriateness*, or *goodness*, measured on a common scale for all performance variables. Moreover, such models require the specification of the relative importance of all branching criteria of each deliberated judgement through the assignment of *weighting factors*, and the specification of an *aggregation function* through which the appropriateness of the deliberated performance variable is then computed (Musso and Rittel 1967; Rittel 1985). As explained, however, the relative importance is not independent from the acceptable value ranges, or the transformation functions, in this case. This approach can actually be used to demonstrate this interdependency, since the *weighting factors* and the *appropriateness values* for each of the branching performance variables appear always together, as a product, and cannot be separated to allow for independent measurement (Rittel 1990). Weighting factors are inappropriate, because they represent a *reconsideration* of the relative importance of performance criteria, in addition to the one that led to their formulation.

Based on the above considerations, all deliberated judgements represent qualitative criteria and the aggregation of the appropriateness of the values of the terminal performance variables is not delegable and cannot be specified. As a result, designers consider only the values of the terminal performance variables to judge the overall appropriateness of design solutions, essentially ignoring the intermediate branching. The overall evaluation of potential design solutions can be simply expressed as a *how-much-of-this-for-how-much-of-that* consideration.

The desired performance is, then, only *partially specifiable*, thus being only *partially delegable*. In practice, it is never explicitly specified or considered in any orderly fashion. The formulation of performance criteria is concurrent with the generation of alternative design solutions and their evaluation. The final version of the desired performance is the performance of the final solution, that is the plan that the designer decides to commit to.

Development of the potential *future description*

Deliberation on a performance variable can be seen as the identification of the variables that affect it. A deliberated performance variable can then be seen as a *function* of the branching performance variables. In the same way, the terminal performance variables are functions of variables that describe potential solutions. The names of these variables depend on design domain. In architectural design they are variables that describe the building and its context. In automotive design they are variables that describe the automobile and its context.* These

* In fact, design domains are usually specified through reference to a concept such as building, automobile, or city.

variables are called *control* variables, because they *control* the values of performance variables (Papamichael 1991).

Control variables are either *design*, or *context* variables (Rittel 1973; Papamichael 1991). Design variables are the control variables whose values are controlled by the designer. Context variables are the control variables whose values are not controlled by the designer. Since the terminal performance variables are functions of design and context variables, design can be seen as the direct control of the values of the design variables in order to indirectly control the values of performance variables. This indirect control is also partial, since the values of performance variables are affected by the values of the context variables as well.

Modifications of potential descriptions are made when one or more performance criteria are not met, based on the relationship of design and performance variables for the specific set of values of context variables. When a performance criterion is not met, each design variable that affects it represents an option for modifying potential solution. However, modifying the value of a design variable does not guarantee that the required performance will be met. Determination of the value of the specific performance variable is required.

Design variables usually affect the values of more than one performance variable. Trying to improve performance with respect to one or more performance variable may, then, result in degrading performance with respect to one or more of the rest of the performance variables. Such trade-offs among performance variables due to interdependencies is what makes performance criteria difficult to meet.

The development of the potential future descriptions is then specifiable and, thus, delegable within a given design domain. The related knowledge consists of the values for context variables, the available alternative values for design variables, either objects or attributes, and the relations among context, design and performance variables (Papamichael 1991).

Determination of the potential *future performance*

Since design is aimed at producing a potential description of a situation, rather than actually creating it, the only means to determine the potential performance is through *simulation* Simulation of performance can be achieved in various ways: through drawings, scale models, hand calculations, nomograms, charts, or computer-based algorithms.

Simulation procedures to determine the values of performance variables are used at various degrees of detail. They range from simple, relatively effortless ones for quick estimates of the order of magnitude (e.g., sketches, crude scale models, simplified calculations), to complicated, relatively elaborate ones that are highly detailed and accurate (e.g., working drawings, detailed scale models, sophisticated calculations). Quick estimates of the order of magnitude are usually used during the initial, "schematic" phases of design, while complicated, accurate simulations are mostly used after most of the required design details have been addressed.

Determination of the potential performance is specifiable within a specific design domain, thus delegable. In fact this is what most of the design education covers, since this is what designers mostly do during the time allocated to a design project. In many cases, however, the determination of the potential performance is almost concurrent with the determination of the potential description and appears to be the latter, as is the case with sketching, drafting and drawing.

Conclusions & Recommendations

The shortcoming of both generations of systems approach is the consideration of design as a rational activity. The first generation assumed it without even recognizing it. The second generation recognized it, identified the four paradoxes of rationality, however, still assuming that design is a rational activity.

The new theory presented in this paper suggests that design is "thinking and feeling while acting," supporting the position that design is only partially rational. Intelligence, "natural" or "artificial," is only one of two requirements for design, the other being emotions. Design decisions are only partially inferred, that is, they are not entirely the product of reasoning. Rather, design decisions are based on judgment that requires the notion of "good" and "bad," which is attributed to feelings, rather than thoughts. Design is the equivalent of imaginary life, however affected by real life itself, which in turn can be seen as a continuous design process. When more than one performance criteria are considered for the evaluation of alternative design solutions, the delegation of judgement is not possible, since the relative importance of performance criteria is not specifiable. Designers do not "know" the relative importance of design criteria. They feel it continuously throughout the design process, reformulating it as they compromise between what is desired and what is possible.

This new theory suggests very well-defined limits of the role of intelligence in design, which become *constraints* on the potential use of computers and suggest reconsideration of the research efforts towards the development of computer-based design tools. Most of the current efforts incorporate design process models that violate these constraints through use of *multi-criterion evaluation* techniques, *conflict resolution* methods and *optimization* algorithms. Such models are inappropriate for design, because they force designers to make *premature judgements*, by requiring an explicit, a-priori knowledge of the desired performance. Moreover, most existing design tools that incorporate "expert" or "knowledge-based" systems, may mislead designers due to "hidden" subjective preferences, that are "acquired" from "experts," or taken from standards and codes, and are used as "rules" for automated decision-making.

All design activities that involve thinking can be delegated to others, e.g., computers. The ones that require feeling, however, cannot be delegated and require the attention of the designer her/himself. This new design theory suggests that research and development efforts should concentrate on computer-based *simulation of performance*, *factual databases* and, most important, appropriate *user interface*, so that computers' memory and computational speed can be used effectively to assist designers in making better judgements.

Acknowledgments

This work was partially supported by the Assistant Secretary for Conservation and Renewable Energy, Office of Building Technologies, Building Systems and Materials Division of the U.S. Department of Energy under contract No. DE-AC03-76SF00098.

References

Camejo, P.J. and Hittle, D.C. (1989); An expert system for the design of heating, ventilating, and air-conditioning systems; *ASHRAE Transactions*, Vol. 95, Part 1.

Case, M. P., McConkey, I., McGraw, K. and Lu, S. C-Y. (1990); Multiple cooperating knowledge sources for the design of building energy systems; ASHRAE Transactions, Vol. 96, Part 2.

Degelman, L. O. and Kim, Y. S. (1988); A knowledge-based system for energy efficient building design; Proceedings of the Third National Conference in Microcomputer Applications in Energy, Tucson, Arizona (pp. 83-89).

Haberl, J. S., Norford, L. K. and Sparado, J. V. (1989); Intelligent systems for diagnosing operational problems in HVAC systems; *ASHRAE Journal*, Vol. 31, No. 6 (pp. 20-30).

Hall, J. D. and Deringer, J. J. (1989); Computer software invades the HVAC market: Overview of knowledge-based systems research and development activities in the HVAC industry; *ASHRAE Journal* (pp. 32-44).

Jafar, M. J., Bahill, A. T., Osborn, D. E. (1991); A knowledge-based system for residential HVAC Applications; *ASHRAE Journal*, January (pp. 20-26).

Mayer, R., Degelman, L. O., Su, C. J., Keen, A. Griffith, P., Huang, J., Brown, D. and Kim, Y. S. (1991); A knowledge-aided design system for energy-efficient buildings; *ASHRAE Transactions*, Vol.97, Part 2.

Musso, A. and Rittel, H. (1967); Measuring the performance of buildings: Report about a pilot study; Washington University, St. Louis, Missouri.

Papamichael, K. M. (1991); Design process and knowledge: Possibilities and limitations of computer-aided design; Ph.D. Dissertation, University of California, Berkeley.

Pohl, J. and Chapman, A. (1988); An expert design generator; *Architectural Science Review*, Vol. 31 (pp. 75-86).

Pohl, J., Meyers, L. and Chapman, A. (1990); ICADS: An intelligent computer-aided design environment; *ASHRAE Transactions*, Vol. 96, Part 2.

Rich, E. (1983); Artificial intelligence; New York, NY: McGraw-Hill.

Ruberg, K. and Cornick, S. M. (1988); Diagnosing window problems: Building an expert system; Proceedings of the Fourth Conference on Building Science and Technology, Toronto, Ontario (pp. 101-120).

Rittel, H. W. J. (1972); On the planning crisis: systems analysis of the "first and second generation"; *BERDIFTSOKONOMEN*, No. 8 (pp. 390-396).

Rittel, H. W. J. (1985); Unpublished lectures in Design Theories and Methods, Department of Architecture, University of California, Berkeley, California.

Rittel, H. W. J. (1973); Some principles for the design of an educational system for design; *DMG-DRS Journal: Design Research and Methods*, Vol. 7, No. 2.

Rittel, H. W. J. (1990); Personal discussion.

Tham, K. W., Lee, H. S. and Gero, J. S. (1990); Building envelope design using design prototypes; *ASHRAE Transactions*, Vol. 96, Part 2.

Tuluca, A. N., Turner, G. E. and Krarti, M. (1989); Expert system for selection of standardized research greenhouses; *ASHRAE Transactions*, Vol. 95, No. 2.

END

DATE
FILMED
3 / 30 / 94