AI Magazine Volume 11 Number 4 (1990) (© AAAI)

Creating a Scientific Community at the Interface Between Engineering Design and AI

David Steier

■ *On January 13-14, 1990, a workshop organized by EDRC was held to discuss the topic of creating a scientific community at the interface between engineering design and AI, in order to identify problems and methods in the area that would facilitate the transfer and reuse of results. This report summarizes the workshop and follow-up sessions and identifies major trends in the field.*

Motivation

It has been over thirty years since a group of engineers at Westinghouse created a program to design generators and motors using heuristic method[1]. The program produced commercial-quality designs, and may have been the world's first expert system. Since then, the subfield of artificial intelligence devoted to engineering design applications has blossomed, especially in the last decade. In order to form a genuine scientific community out of the group of researchers active in this area, it is reasonable to ask a hard question: What do we know?

On January 13-14, 1990, twenty-three researchers in the field participated in a workshop designed to produce at least the beginnings of an answer. The workshop, held in Pittsburgh, Pennsylvania, was organized by the Engineering Design Research Center (EDRC) of Carnegie Mellon University (CMU) and sponsored by the National Science Foundation. Based on the observation that the research at the interface between engineering design and artificial intelligence was not as cumulative as it could be, the goal of the workshop was to identify problems and methods in the area that would facilitate the transfer and reuse of results. The workshop itself was by invitation only to permit focussed interaction, but was followed by a public session on the afternoon of January 24, at which the discussions of the workshop were reported and five additional CMU EDRC members made presentations and coordinated discussions. This paper reports on both workshop and follow-up sessions at several levels of abstraction: first, short summaries of each session are given; secondly, major trends, as evidenced by the presentations and discussions, are identified; and, finally, some assessment of how the workshop as a whole fulfilled its original goals.

The workshop organizers began by making their own assumptions explicit in the invitation to the participants:

1. It is both feasible and desirable to study engineering design scientifically. While gaps in current theories of engineering design exist, no portion of the design process is assumed a priori to be beyond analysis.
2. The design and implementation of computer programs that automate or support engineering design is a fruitful activity for researchers in the field.
3. Artificial intelligence concepts and techniques, including explicit representation of and reasoning about design goals, search of problem spaces, and methods for learning from experience, aid in the construction of engineering design systems.
4. It will be easier to make progress in this field if the research goals are structured so as to facilitate description, dissemination, and building on previous work.

We recognized that these assumptions would apply to only a subset (albeit a large one), of researchers interested in design yet the belief was that some limiting set of assumptions had to be made to enable fruitful discussions. Part of the evidence that this tactic was successful was that relatively little time at the workshop and follow-up session was spent defining the term design itself.

With these assumptions made explicit, the organizers then took the unusual step of asking the prospective participants not to give detailed presentations of their own research. Participants were requested instead to prepare short descriptions of two or three key results or techniques that they felt should be part of the core of a scientific discipline of engineering design and artificial intelligence. These descriptions were then copied and distributed to the participants prior to the workshop, and the workshop schedule was organized using clusters of related contributions as session themes (with some participants speaking multiple times). We should acknowledge that some of the invited researchers protested that the field was in fact not yet at a stage to identify such key results or techniques or that those results could not yet be called scientific. On the other hand, such objections did not deter the researchers from submitting a contribution or attending the workshop.

. . . no portion of the design process is assumed a priori to be beyond analysis.

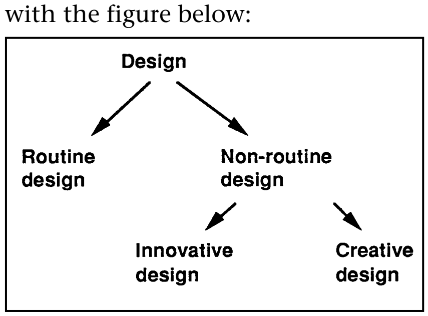
Workshop and Follow-up Sessions

By the time of the workshop, over two dozen contributions were submitted by the participants, and four session themes were chosen: models of design processes, coping with design complexity, representations for design, and goals and methods of the field. Each session consisted of five to ten individual presentations, interleaved with time for discussion. Each individual presentation was limited to eight minutes, followed by two minutes for clarification questions; time limits were strictly enforced with the aid of the VOID (Verbal Overrun Inhibitory Device), a darkroom timer hooked up to shut off the overhead projector at the end of a speaker's allotted time. Workshop participants soon demonstrated that they were capable of adapting to real-time constraints, and the VOID proved effective at keeping the workshop on schedule.

The session on models of design processes was divided into three topic areas, the first of which was using design process models to understand design. I began the session by presenting (on behalf of Allen Newell of Carnegie Mellon University) three lessons from human design: first, that human design appeared to be a combination of rapid kernel idea selection followed by a longer phase of elaboration and refinement of the kernel idea; second, that human design proceeds by alternately stipulating and retrieving partial functions and structures; and third, that specific structure plays an important role in the discovery of additional constraints for the design, in addition to the more commonly perceived role of approximation of the final design by a succession of partial artifacts.

The next speaker, Mary Lou Maher (also of CMU), argued that three distinct models of design have been developed, and ought to be generally applicable to a variety of design applications, while being specific enough to be useful: decomposition of complex problems into simpler ones; case-based reasoning to draw on design experience in the form of specific episodes; and systematic transformations for design as in grammar-based approaches.

Tim Smithers of the University of Edinburgh noted that design ought to be considered as exploration rather than search of a single space with well-defined boundaries. The very ill-structured nature of most design means that the designer must spend a substantial portion of the design time refining the formulation of the problem. This claim was given additional support by John Gero of the University of Sydney, who gave further structure to the ill-structured problem of defining creative design with the figure below:



Routine design, Gero claimed, occurs when all functions and structures are known before design begins (thus parametric design is a special case of routine design). When the possible design structures are not known, design becomes innovative. When neither structures nor the allowable design functions are known, then design becomes creative. Creative design extends the space of possible artifacts that the designer can produce.

The second topic area of the session was architectural support for design models. Barbara Hayes-Roth of Stanford University began by presenting a taxonomy of design problems, methods, and domains and gave instances of research to fit in the categories. Different design methods and heuristics are effective in different domains, depending on knowledge availability. She conjectured that a general and flexible computational architecture is best equipped to adapt to each situation, supporting the integration and acquisition of knowledge from a variety of sources. This conjecture is sup-

As our models represent the actual design process with increasing fidelity, they will become increasingly complex, and strategies will be needed to cope with that complexity.

ported by the work of Hayes-Roth and her colleagues on the BB1 blackboard system. B. Chandrasekaran of Ohio State University presented a slightly different approach towards the use of architectures to support design, focusing on task structures as a technique for specializing an architecture to the demands of a specific design method, such as propose-critique-modify. Hayes-Roth then presented a specific technique, based on a model of cross-domain analogy and Anderson's spreading activation memory in ACT*, for retrieving information within a successor to BB1. Chandrasekaran concluded the session by discussing the use of a functional representation of devices for retrieval of relevant design cases.

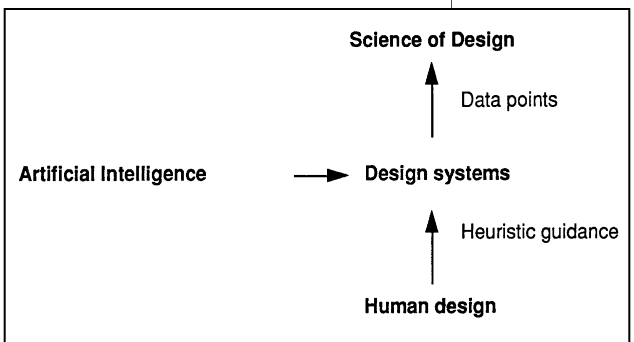
The third topic in the session discussed models of team design. Art Westerberg of CMU observed that most failures in design occur for organizational rather than technical reasons and introduced the idea that a design environment should provide support of the integration of many partial models of the artifact being designed. Sarosh Talukdar, also of CMU, argued that we need to devote more effort to the prediction and resolution of design conflicts, offering genetic algorithms as a possibility for capturing the evolutionary nature of the design process. John Goldak of Carleton suggested in his talk that one way to achieve success in supporting large-scale designs was to study carefully the way that VLSI design software has structured domain knowledge, and to transfer these methods to other applications. This suggestion was reinforced by John Hopcroft of Cornell, who moderated the discussion that followed.

The topic of the next session, coping with design complexity, was in one sense an extension of the session on design models. As our models represent the actual design process with increasing fidelity, they will become increasingly complex, and strategies will be needed to cope with that complexity. I began the session on the topic of progressive deepening. This pattern of repeated exploration of design paths in order to acquire new information on each pass is observed in human performance on many design, and design-related, tasks. Chris Tong of Rutgers University discussed how progressive deepening has been used in several automatic design systems. These use exploratory design to add structure to a problem, rendering it amenable to methods of routine design. Tong also began the next topic: using knowledge from previous experience to reduce search. His SCALE algorithm improves the decomposition of a problem by reducing subproblem interactions. Jack Mostow, also of Rutgers, presented several lessons he and his colleagues have learned about the applications of derivational analogy to design; for example, design plans can be acquired by first recording user-selected transformation steps in an interactive design system, and then generalizing those steps using explanation-based learning techniques. The third topic of the session was the use of abstraction. Ken MacCullum of the University of Strathclyde argued that the acquisition and use of abstractions at multiple levels is crucial for good designs. Mostow then discussed automated techniques for discovering useful abstractions, illustrating them with systems, ABSOLVER and POLLYANNA, built by his students. The session was concluded by several presentations by Mostow, Tong, and Tom Dietterich (of Oregon State University) on the use of knowledge compilation. This term describes a class of techniques for producing efficient search algorithms for specific classes of problems, generated by integrating algorithm design and domain-specific knowledge.

The third session concentrated on representations for design. The better adapted a representation is to a particular design problem, the easier it will be for a designer to search the space of designs for a high-quality solution. Chuck Eastman of UCLA discussed two representational concepts based on database techniques: first, data modelling methods for engineering product models, and second, design transactions, in which each transaction satisfies certain integrity constraints. He also discussed shape grammars, a formalism for representing geometric knowledge pioneered by George Stiny. Gero then summarized the motivations behind the concept of design prototypes, which independently represent the function, structure and behavior of engineered artifacts. Talukdar discussed the representation of designs as n-tuples of partial models. Tetsuo Tomiyama of Tokyo had planned on discussing the role of logical formalisms and qualitative physics for representing engineering knowledge, but was not able to attend the workshop. Sanjay Mittal of Xerox Palo Alto Research Center discussed dynamic constraint satisfaction for engineering design. Panos Papalambros of the University of Michigan presented an example of how structural design problems could be solved using mathematical optimization, and argued that methods for combining AI and optimization techniques ought to be given more attention in research on engineering design.

The final session took a broader perspective and directly addressed the nature of a scientific community in this area. Herb Simon of CMU initiated the topic on theories and data in design science by presenting some of his ideas on problem formulation, satisficing, the order-dependence of design, and the need to combine techniques from whatever disciplines are appropriate. This talk was followed by Steve Fenves' (also of CMU) talk on how design theories should serve to explain design processes. John Gero emphasized how design science must work more strongly to collect data on design. In the second topic of this session, the topic suggested by Gero was explored in detail, in terms of what are the sources of data on design. Dietterich discussed the use of protocol analysis, based on his studies of mechanical design, while I discussed the need to perform comparative design studies using common examples, drawing on my work on algorithm design studies. The session and the workshop was concluded by a discussion on infrastructure for the field.

The follow-up session, held two weeks later, began with a brief review of the workshop presentations and discussions, followed by an extension of the format to allow additional researchers to present their "golden nuggets" for design science, some of which were not explored in depth during the workshop itself. All the speakers were affiliated with CMU. Newell presented his conception of how the field should progress in terms of the diagram below:



AI researchers produce systems that design, each of which serves as a data point from which to induce a science of design, since each system is a detailed design model. Studies of human design are used to guide the creation of design systems, since humans are currently the only intelligent agents possessing the robustness and flexibility that we would like our design systems to have.

Mark Fox listed several examples of design process, such as indexing, filtering, analogy, composition, and constraint satisfaction, and discussed the role of opportunism in design. Ulrich Flemming argued that we should focus on formal models of design to produce designers without human weaknesses, dividing the labor so that human capabilities would be relied upon only for guidance as needed. Rob Woodbury extended this further with specific examples of shape grammars, structure and solid grammars, and non-manifold representations. Such representations, he noted, had desirable properties in terms of expressiveness and simplicity, and enabled efficient search in geometric-based domains such as building and truss design. Susan Finger argued that we should be focusing more on representation of the designed artifacts themselves than on the design process, claiming that process-oriented research hasn't made as much progress as artifact-oriented research, such as the development of formal geometric models. Dan Siewiorek presented several nuggets from his work with his student on building knowledge-based design systems, including the observation that engineers developing heuristics for different synthesis problems are already actually practicing AI without knowing it. In his experience it was more productive (in terms of knowledge integrated into a system per unit time) to train domain experts in AI techniques than vice versa. Ultimately, automated, domain specific knowledge acquisition systems will be required to achieve truly expert performance.

Major trends

I noted five trends that underlay much of the discussion at the workshop and follow-up session:

1. A near-consensus on the complementary roles of function and structure in design: Terms such as function and structure were often used in different senses by the workshop participants, and some attention was devoted to clarification of these. Some used the term function in the sense of the purpose of some structure relative to the design goals, where others used function to describe the behavior exhibited by some structure. Newell's abstract for the workshop noted that descriptions of structures are generally much less problematic than description of functions (in the first sense), which are conceptual entities used to decompose the design process. In either case it does not seem that design is a straightforward mapping from functional specifications to structural descriptions. Both structure and function seem to play important roles in retrieving design knowledge (as noted by Newell, Chandrasekaran, and others) and both are elaborated during design. Yet we are far from a complete theory of functions, or of a theory to link function and structure. One concept, which did not come up in the abstracts[2] or discussions, but which does seem to provide a function-structure link, is that of design feature. Features are aspects of the structure of a design, but their importance for any given process depends on the functions being fulfilled.
2. The beginnings of a common vocabulary for modelling design: As noted by Hayes-Roth, a theory of design must describe problem types, methods for problem solutions, and problem domains. She proposed arrangement as a candidate problem type, which she found occurring in domains as diverse as biochemistry, construction management, and architectural layout. Examples of methods are propose-critique-modify (Chandrasekaran), decomposition (Maher), and progressive deepening (Steier, Tong). A variety of design domains were discussed at the workshop. Among them were aircraft design (Goldak), digital logic (Mostow), and structural design (Dietterich, Papalambros). Terms to describe approaches for acquiring design data, such as protocol analysis (Dietterich), and for implementing design models, such as design architectures (Hayes-Roth, Chandrasekaran) were also used at the workshop. As Fenves pointed out, the establishment of such common vocabulary is one of the prerequisites for the establishment of a scientific community in this area. Yet, the field still seems to have some distance to go before terms are defined with precision, and commonly used in comparison and evaluation of work.
3. An increasing recognition of the importance of studying design-in-the-large: Several speakers (Westerberg, Goldak, Hopcroft, Talukdar) noted that issues arise in team design that are not often encountered in research on individual designers. Among these are the knowledge representation and integration process for multiple partial artifact models, and the prediction and resolution of design conflicts. Tools, such as n-Dim (Westerberg), that recognize the organizational context in which most real design takes place are just beginning to be built. There was a clear sense in the workshop that the issues ought to be studied further and that there is a need to devote more research effort to observing and analyzing large-scale design.

4. A growing collection of methods for acquiring and applying abstract or approximate design knowledge: Design prototypes (Gero) are one method for capturing general design knowledge. Abstract design algorithms (Mostow, Tong, Dietterich), and design task structures (Chandrasekaran) are others. A notable development is the use of knowledge compilation to specialize abstract algorithm schemes into efficient algorithms in several design domains. MacCullum observed that an important issue in developing systems that use such abstract knowledge is how the abstractions are acquired. Mittal observed from his studies of designers in the PRIDE and COSSACK projects that representations used to articulate knowledge may not be the most appropriate for use in automated design systems. A wide range of methods for producing abstractions were discussed. Some of the methods assigned a central role to knowledge engineering, suggesting systematic, formal analysis of the artifacts (Flemming, Woodbury, Finger), comparative study of common examples (Steier), or protocol analysis (Dietterich). Others have stressed automated techniques for acquiring such knowledge from experience (Mostow, Tong). Mostow in particular has recently focused on methods for acquiring useful approximations. It is likely that robust design systems will result from combinations of these methods (Simon).
5. The continuing proliferation of radically different knowledge representations for design: Several participants (Mittal, Newell, Simon, Smithers) noted that both constraint formulation and satisfaction are important in design. Mittal's research has focused on representing such constraints and their dynamic nature explicitly. Other representations mentioned at the workshop and follow-up sessions included shape grammars (Eastman, Flemming, Woodbury), engineering data models (Eastman) or n-tuples of such models (Talukdar), logical expressions (Tomiyama), systems of equations (Papalambros), and bond graphs (Finger). All the participants seemed to agree that a representation should be adapted to the application for a special-purpose system. The development of a theory that would guide the selection of representations for a particular application, especially a complex one, remains a research question. One way to acquire the data on which to build such a theory would be to encode the same engineering design knowledge in different representations for the purposes of evaluating the efficacy of different representations in a particular context.

Going for the Gold

When we began to plan for the workshop, we intended this report to include a list of two dozen or so golden nuggets that would form a core of a scientific discipline of engineering design and AI. While the workshop was successful in terms of generating stimulating presentations and discussion, there was not very much progress towards the production of the list, nor is there a clear method—at least not to this participant—for producing such a list by post-workshop armchair analysis. The workshop did produce some candidates for the list, although not the two dozen we had hoped for. These candidates would include techniques such as the use of functions as memory retrieval cues, and terms such as design task structures or derivational analogy. In retrospect it is clear that any field which claims to have golden nuggets must also propose the assay for those nuggets. In fact methods for evaluating contributions were absent from most of the workshop presentations.

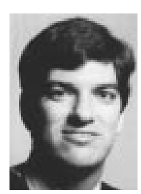
I am still convinced that a list of golden nuggets for this field (eventually leading to a handbook of techniques) would be both very useful to have and feasible to produce, but perhaps only with additional research and several more iterations on the workshop format. We probably need to understand better what is meant by a theory of design, a process begun over twenty years ago by Herb Simon's book, *The Science of the Artificial*. We need to concentrate on the role of data in such theories, perhaps establishing repositories for design case histories as was suggested at the workshop. Both design processes and designed artifacts need to be studied. And we would need further convergence on assumptions, vocabulary and certainly, criteria for evaluating contributions. I believe the workshop provided a useful forum to discuss these issues, essential for further progress of the budding scientific community in engineering design and AI.

Acknowledgements

The workshop was sponsored by NSF grant ECD-8943164. I am grateful to Professors Allen Newell , Art Westerberg, and Mary Lou Maher for assistance in organizing the workshop, and to the head of the EDRC, Fritz Prinz, and EDRC staff Nancy Monda, Mary L. Ray, and Sylvia Walters for their administrative support. Newell, Westerberg, and Peter Patel-Schneider also provided useful comments on an earlier draft of this report.

Notes

1. Godwin, G. L (1958). Digital Computers Tap Out Designs for Large Motors . . . Fast. *Power*, April.
2. As pointed out by Caroline Hayes, who has built a feature-based machining planner.



David Steier is a research scientist at the Engineering Design Research Center at Carnegie Mellon University. His research interests center on the application of AI techniques, especially integrated problem-solving and learning architectures, to design problems. He received his PhD in computer science from Carnegie Mellon in 1989, and is co-author (with P. Anderson) of the book, Algorithm Synthesis: A Comparative Study.