Artificial Intelligence for Engineering Design, Analysis and Manufacturing (1998), *12,* 117–121. Printed in the USA.

Dimensions of machine learning in design

DAN L. GRECU AND DAVID C. BROWN

AI in Design Group, Department of Computer Science, Worcester Polytechnic Institute, Worcester, MA 01609, U.S.A.

(RECEIVED June 27, 1997; REVISED October 17, 1997; ACCEPTED November 10, 1997)

Abstract

Many of the design systems developed in recent years incorporate some machine learning. The number of such systems already available, and the multitude of design learning opportunities that are slowly being revealed, suggest that the time is ripe to attempt to put these developments into a systematic framework. Consequently, in this paper we present a set of dimensions for machine learning in design research. We hope that it can be used as a guide for comparing existing work, and that it may suggest new directions for future exploration in this area.

Keywords: Design; Dimensions; Machine Learning; Taxonomy

1. INTRODUCTION

Design represents one of the most complex problem-solving domains addressed by artificial intelligence (AI). Despite the progress made in the last decade to advance the use of AI techniques in design, existing systems have difficulty coping with the diversity and quantity of knowledge required, as well as with the variety of reasoning involved.

In general:

- A design problem requires knowledge from various domains and uses a broad range of representations.
- Design problem solving is based on the ability to carry out many specialized tasks, such as analysis, abstraction, evaluation, and explanation, each involving different reasoning abilities.

Portions of some design domains have been analyzed and formalized, providing solid support for the search for solutions. However, much of designing still relies on good knowledge and heuristics. Maintaining the quality of designs and the efficiency with which they are produced requires continual evaluation and improvement of design knowledge and methods, including heuristics.

For designers, such improvements have been based on recording and learning from notable events and attributes that have occurred during the development of designs. Learning from designs, and learning during designing, is as old as design activity itself.

Adding adaptation to a design system is clearly desirable. Even though learning does not always reach the optimal solution, experience should eventually bring noticeable and worthwhile improvements over initial designs and design processes. These are measured in terms of higher quality, shorter design times, and lower costs.

There has been increasing acknowledgment that computational design systems can and should include the ability to learn, and there is an increasing amount of research on machine learning in design (as demonstrated by the papers in this special issue).

Knowledge acquisition and machine learning are the main tools that support the process of change in a design system. Knowledge acquisition emphasizes the transfer of knowledge from the outside world into the system and relies less on transformations inside the system. The primary goal of knowledge acquisition is to extend the system's operation, by the addition of new knowledge.

Learning, while being based on the perception of events and feedback, focuses on transformations that affect performance. The meaning of "performance" includes both the quality of the solution offered by the system and the efficiency of the processes which generate that design (or designs).

Many of the design systems developed in recent years do incorporate some learning. They illustrate the different ways

Reprint requests to: Dan L. Grecu, AI in Design Group, Department of Computer Science, Worcester Polytechnic Institute, Worcester, MA 01609, U.S.A. Tel: (508) 831-5006; E-mail: dgrecu@cs.wpi.edu

https://doi.org/10.1017/S0890060498122047 Published online by Cambridge University Press

in which design is open to adaptive techniques. With learning, design systems can try to cope with increasingly more complex problems. The number of examples of adaptive design systems already available, and the multitude of design learning opportunities that are slowly being revealed, suggest that the time is ripe to attempt to put these developments into a systematic framework.

2. NEED FOR DIMENSIONS

It is not this paper's intent to review the use of machine learning in design systems (for a recent review, see Duffy, 1997). However, we believe that the field is now important enough and active enough that it is useful to try to characterize it and to attempt to provide a framework for future work.

Consequently, in this paper we present a set of dimensions for machine learning in design research. We hope that it can be used as a guide for comparing existing work, and that it may suggest new directions for future exploration in this area.

The set of dimensions chosen is inspired mainly by the existing attempts to apply machine learning to design. It is by no means complete, and probably not the only possible analysis of the research literature. It represents a hypothesis for discussion. We expect that future authors [including those responding in this special issue to our draft of these ideas (Grecu & Brown, 1997)] will have different opinions, especially as new developments in the field occur.

We do not claim that there is no overlap between the coordinates within each dimension, or that they exhaust the possibilities within that dimension. The purpose of defining the coordinates is to identify the main points of focus within each category. Some of the coordinates "flow" into each other, and it would be difficult to define a clear line of demarcation between them. Their representative character *and* their distinctiveness have been simultaneously considered in singling them out for inclusion.

As a final observation, we would like to note the potentially distributed nature of design systems. The description of the dimensions has been kept as general as possible, to encompass both paradigms—distributed and nondistributed. Except for cases where there is an explicit reference to either the distributed or the nondistributed type of a design system, the statements made are assumed to be valid in both cases.

We now present the proposed dimensions of machine learning in design:

- The triggers of learning.
- The elements supporting learning.
- What gets learned.
- Availability of knowledge for learning.
- Methods of learning.
- Local versus global learning.
- Consequences of learning.

3. DIMENSIONS

The following dimensions have been produced from an analysis of the research literature. No significance should be attached to the order.

3.1. Triggers of learning

The situations that trigger learning in a particular design environment have an impact on the choice of learning techniques that apply and, at run-time, the frequency of occurrence of learning situations.

- *Failure* presents a system with the challenge of identifying the context and the reasons for its occurrence. Failure can occur in a partial or temporary form (as in backtracking in a subproblem), or it can be equivalent to the failure of finding a design solution—where a solution may or may not be known to exist.
- *Success* cannot always be taken for granted at the end of a design process. For many design problems it is hard to know how a good solution can be reached, despite having a significant amount of resources. Being able to identify factors that facilitate the achievement of good performance is as challenging as finding causes for failure—i.e., both the credit and blame assignment problems are hard.
- *Differences between expected and real values* indicate a learning situation, and can include failure and, perhaps, success situations. One advantage of design systems is that they allow monitoring of the evolution of virtually any design parameter or other factor. The design process may generate parameter values that fall outside the range considered to be normal. While often being too local to be immediately evaluated as good or bad, parameter fluctuations can be captured and used to do prediction as well as evaluations of the design.
- *The need to improve abilities* may be a built-in long-term goal, or it may occur as a requested extension to the set of objectives of the design system. An explicit change of goals, for example, calls for adjustments to satisfy these new requirements.

3.2. Elements supporting learning

Learning in design depends on three supporting elements: *a representational structure* that can be evaluated and updated; *the support knowledge*, such as rules, plans, or actions, used to generate new entities; and *the feedback* that is used to decide how to modify the representational structure.

For example, the representational structure can be the heuristics in a design planning module, the support knowledge

https://doi.org/10.1017/S0890060498122047 Published online by Cambridge University Press

for learning can be a set of previous design plans, and the feedback can consist of evaluations of how these design plans performed. The learning component can update the planning module with new planning heuristics extracted from the set of plans based on the available feedback.

The representations used for design problem-solving are chosen according to the design task and the domain to which they will be applied. It is not always the case that system developers consider the possibility of modification when making their representation decisions. Feedback and support knowledge usually can be more easily customized to facilitate learning.

Examples of feedback and support knowledge used in design are as follows:

- *Critique* and *praise* reflect utility factors in design. They often represent a point of view and can provide opposing feedback elements. *Estimates* may be preliminary sources of rating information. *Evaluations* provide assessments of decisions or values with respect to a goal or a set of goals. All of these elements can be provided from external sources, by system users, or by sources internal to the design system, incorporated in separate system components. Most often, critiques, praise, estimates, and evaluations are used at run-time.
- *Feedback after completing the design task* usually comes from outside the design system and reflects evaluations provided by humans to the solution(s), and possibly the ingredient decisions, of the design system. This feedback can refer either to the current design solution or, on a comparative basis, to an entire set of past design solutions. It may be about all or part of the solution(s). Feedback may be directed to the whole design system or to some particular reasoning component.
- *Analyses of failures and conflicting elements* (e.g., goals or decisions) are concerned with factors that introduce "stress" in the design process. Especially in the case where this analysis uses design rationale provided by the user, this feedback can extend the range of adaptation of the design system beyond what can be achieved automatically.
- *Sequences of design decisions* support the generation of improvements from the perspective of design as a decision-making process. Either utilities or newly compiled decisions (in the "knowledge compilation" sense) may be generated by examining the decision sequencing process.
- *Design histories* (e.g., traces of information flow, knowledge exchange, and negotiation between design systems) provide a basis for insight into entire parts of the design process. Certain aspects that characterize global design performance are obtainable only by looking at the information recorded throughout entire design sessions. Patterns of design activity also can be determined only by viewing collections of traces reflecting specific aspects of designing (e.g., communication, information retrieval, negotiation, etc.).

3.3. What gets learned

The object of the learning process can be any aspect deemed to be critical for design performance, and therefore the choice always will be relative to the goals of the design system.

It is not necessary that learning focuses on only *one* target. Any element that is essential for design quality or process performance can be included in a separate learning process, or sets of elements can be combined and modified as a whole within a learning process.

The following examples illustrate some of the elements that are preferentially targeted in the learning process:

- *Constraints relating parameters or other elements of the design* are at the core of every design problem. Some constraints might not be visible or known because they occur dynamically, only at run-time. Other constraints remain hidden because they are distributed over several design system components. Learning can make them explicit.
- *Dependencies between design parameters* are critical for the coordination phases of the design process. Violation of such dependencies usually leads to insufficiently informed decisions, making backtracking and redesign more likely. Hence, increased knowledge of dependencies is worth having.
- *Support in favor of or against a decision*, whether expressed as utilities or as rationale, is the basis for guiding the design process. Any change in support is likely to influence the final design outcome.
- *Design rules, methods, and plans* are among the ingredients of almost any design reasoning system. More generically, any extension of the basic problem-solving knowledge of a design system may be the main target for learning.
- *Analogical associations* indicate similar patterns of reasoning and/or behavior in different situations. The discovery of analogies represents the starting point for abstracting reasoning mechanisms and transferring them to other parts of the design system or into new design contexts.
- *Preferences* provide a ranking to be used in selection processes. Preferences might be only partially explicit, due to a large range of values, and might be computed only as needed. Alternatively, when they are used to characterize the selection process of another component of the design system, they initially might be unknown. In either case, preferences that become explicit or hinted at during the design process can be learned as a means to reduce future uncertainty. This should reduce failure and conflict.
- *Preconditions and postconditions for rules, actions, and tasks* are essential for limiting the reasoning search

https://doi.org/10.1017/S0890060498122047 Published online by Cambridge University Press

space and for avoiding failure. Both types of conditions can be refined or learned as a result of experience.

- *Consequences of design decisions* help establish the utility of design actions. Learning about these end results of design decision-making facilitates the assessments and predictions that guide design development.
- *Failures and conflicts* can be classified into types, which then allow recognizing typical situations that are likely to create them.
- *Heuristics for failure recovery and conflict resolution* often result by looking at "recordings" of situations that have ended with a solution. Failure recovery and conflict resolution are frequently carried out with only a limited look-ahead, and information synthesized in hindsight can prove extremely useful for future situations (Cross, 1978).
- *Successful designs and design processes* can simply be learned as cases, or can be used as knowledge to support the generation of new structures (e.g., via generalization) that are very likely to lead to successful outcomes when reused.

3.4. Availability of knowledge for learning

The learning mechanism depends on the way in which the supporting knowledge is made available. This has an impact on the *quantity* of available knowledge, the *frequency* with which it is provided (occasionally, periodically, or permanently), its *scope* (local or global), and the limitation of its *validity* in time (applies to a limited set of situations or always). In some circumstances it may also affect is *quality*.

- *Direct communication* represents a stream of messages either between the design system and the user or between design system modules. The persistence of information is short, and it either has to be stored or immediately used for learning. The contents can refer to virtually any aspect of the design or design process.
- *Indirect communication* (e.g., between design systems via a blackboard) usually conveys information that is less time-critical, and its frequency tends to be lower than for direct communication. The knowledge quantity, its scope, or its validity in time are as unrestricted as in the case of direct communication.
- *Records of the state of the design* provide an image of the design evolution and of the context and impact of design actions. The design state directly reflects the quality of the design and indirectly reflects the design process. It covers design aspects ranging from the local to the global level. The persistence of this support for learning is low; therefore, the adaptive component has to process the available information immediately or it has to store relevant aspects of states for later analysis.
- *Repositories of designs and interaction histories* provide unrestricted opportunities for generating information for learning from the point of view of time restrictions. The information stored usually is limited to aspects considered relevant at the time of recording. This knowledge source is usually the most useful setting for nonincremental learning techniques.

3.5. Methods of learning

Virtually any learning technique can be applied to design. The fact that not all of them have been used to the same extent so far is due mainly to the relatively short time that the field of learning in design has been given attention by researchers. The following list presents some of the many possible examples.

- *Explanation-based learning* has a potential in design problems where a logic-based representation of design states and actions has proven to be effective (Mitchell et al., 1986).
- *Induction* (Fisher, 1987) has been one of the most widely used techniques, ranging from the development of new design concepts, predicting unknown design parameters, to the modeling of the behavior of design agents (Grecu & Brown, 1996).
- *Knowledge compilation* (Brown, 1991) can be used to generate macro-operators in planning or to recombine knowledge for design decomposition or configuration.
- *Case-based and analogical learning* is one of the basic techniques for knowledge reuse in design (Maher & Pu, 1997).
- *Reinforcement learning* supports action selection during design in systems where the emphasis lies on the design generation process and where the right sort of feedback is available (Whitehead, 1991; Tan, 1993).
- *Genetic algorithms* implement design adaptation by recombining parts of an initial set of completely described designs (Bentley & Wakefield, 1995). They also might be applied to design generation knowledge, such as grammars (Brown, 1997).
- *Neural networks* are a relevant approach whenever learning design configuration patterns or behavior patterns can improve design performance (Ivezic & Garrett, 1994).

3.6. Local *versus* global learning

The overwhelming majority of learning applications in design describe learning in an individual design system or module. Distributed design systems and agent-oriented design systems have recently attracted increasing interest, and part of the effort to implement learning has been transferred into this new area. Virtually all of the learning paradigms implemented in an individual design system keep their relevance when mapped onto a single design agent. We call this type of learning *local learning*.

Multiagent design systems involve a set of processes that involve at least two agents. Some of these processes imply the active participation of several agents, while some of them imply the use by one agent of knowledge about another agent. In either case, the learning related to such processes is a *distributed learning process*.

At the "extreme" end of distributed learning lie global effects, which result as a consequence of local changes at the level of individual agents (Hutchins, 1991; Shoham & Tennenholtz, 1994; Weiss, 1993). These effects can be regarded as *global learning* caused by partial views and feedback at the local level, which nevertheless result in a new consistent behavior of the system in its entirety.

3.7. Consequences of learning

The performance of the design system, and the success of the learning mechanism, can be measured on two different scales:

- Design improvements mean a higher quality design solution. Design quality measures can provide feedback about the impact of learning on individual aspects of the design. Even though a single global design quality indicator is rarely available, the variations of individual parameters describing the design produced provide a multifaceted view of the learning achieved.
- *The improvement of the design process* through learning implies improved design system efficiency. In some design systems, this can be a way to avoid resource limitations. Redirecting these "saved" resources toward achieving improved design quality becomes increasingly important as more difficult design problems are attempted.

4. CONCLUSIONS

This paper presents a set of dimensions for machine learning in design research for use as a guide for comparing existing work and to suggest new directions for future exploration in this area. The set of dimensions may not be complete, and there may well be other possible analyses, but we consider those presented here to be a useful contribution.

The number of design systems that include learning, and the many possible uses for learning in design systems, suggest that this is an appropriate time to try to analyze these developments and opportunities in a systematic manner.

REFERENCES

Bentley, P.J., & Wakefield, J.P. (1995). The table: An illustration of evolutionary design using genetic algorithms. *Proc. Conf. Genetic Algorithms in Engineering Systems: Innovations and Applications*, IEE Conference Publication No. 414.

Brown, D.C. (1991). Compilation: The hidden dimension of design systems. In *Intelligent CAD, III* (Yoshikawa, H., & Arbab, F., Eds.) (pp. 99–108). North-Holland, Amsterdam.

Brown, K. (1997). Grammatical design. *IEEE Expert 12(2)* 27–33.

Cross, J. (1978). Negotiation as a learning process. In *The Negotiation Process—Theories and Applications* (Zartman, W., Ed.), 29–54. Sage Publications.

Duffy, A.H.B. (1997). The "what" and "how" of learning in design. *IEEE Expert 12(3)*, 71–77.

Fisher, D. (1987). Knowledge acquisition via incremental conceptual clustering. *Machine Learning 2*, 139–172.

Gordon, D., & Subramanian, D. (1993). A multistrategy learning scheme for agent knowledge acquisition. *Informatica 17*, 331–344.

Grecu, D.L., & Brown, D.C. (1996). Learning by single function agents during spring design. In *Artificial Intelligence in Design '96* (Gero, J.S., and Sudweeks, F., Eds.), 409–428. Kluwer Academic Publishers.

Grecu, D.L., & Brown, D.C. (1997). Dimensions of machine learning in design. Web page, *http://cs.wpi.edu/~dcb/AID/taxonomy.html*. Computer Science Department, WPI, Worcester, MA.

Hutchins, E. (1991). Organizing work by adaptation. *Organization Sci. 2(1)*, 14–39.

Ivezic, N., & Garrett, J.H. (1994). A neural network-based machine learning approach for supporting synthesis. *Artif. Intell. Engrg. Design, Anal. Manuf. 8(2)*, 143–162.

Maher, M.L., & Pu, P. (1997). *Issues and Applications of Case-Based Reasoning in Design.* Lawrence Erlbaum Associates, Mahwah, NJ.

Mitchell, T., Keller, R., & Kedar-Cabelli, S. (1986). Explanation-based generalization: A unifying view. *Machine Learning 1(1)*, 47–80.

Shoham, Y., & Tennenholtz, M. (1994). *Co-Learning and the evolution of social activity*. Report No. CS-TR-94-1511. Stanford University, Stanford, CA.

Sycara, K., & Lewis, M. (1990). Cooperative negotiation in concurrent engineering design. In *Cooperative Engineering Design*, 269–297. Springer-Verlag, New York.

Tan, M. (1993). Multi-agent reinforcement learning: Independent vs. cooperative agents. *Workshop on Reinforcement Learning, Proc. 10th Int. Conf. on Machine Learning*, 330–337. Morgan Kaufmann, San Mateo, CA.

Weiss, G. (1993). Learning to coordinate actions in multi-agent systems. *IJCAI-93*, Vol 1, 311–316.

Whitehead, S. (1991). A complexity analysis of cooperative mechanisms in reinforcement learning. *AAAI-91*, 607–613.

Dan L. Grecu is a Ph.D. candidate at Worcester Polytechnic Institute, where he is a member of the AI in Design Group. He holds B.S. and M.S. degrees in Computer Science. He is a member of the ACM, the AAAI, and Sigma Xi. His research interests lie in the areas of artificial intelligence in design, machine learning, multiagent systems, and cognitive modeling. His current work is focused on models of adaptive group design. His recent publications are on conflict resolution in multiagent design and multiagent learning in design.

David C. Brown is a Professor of Computer Science at Worcester Polytechnic Institute, and has B.Sc., M.Sc., M.S., and Ph.D. degrees in Computer Science. He is a member of the ACM, IEEE Computer Society, and the AAAI. His Editorial Board memberships include the journals *Artificial Intelligence for Engineering, Design, Analysis and Manufacturing* and *Concurrent Engineering: Research and Application*. He is on the committees of the AI in Engineering and the AI in Design Conferences, as well as the IFIP WG 5.2 workshops and conferences. He organized the IFIP IntCAD91 conference. His research interests include computational models of engineering design, and the applications of artificial intelligence to engineering and manufacturing. He is the author, with B. Chandrasekaran, of the book *Design Problem Solving: Knowledge Structures and Control Strategies*, Pitman Publishing, Ltd., and a co-editor of *Intelligent Computer Aided Design*, Elsevier Science Publishers B.V. (North-Holland).