The ‘what’ and ‘how’ of Learning in Design

Alex H B Duffy
CAD Centre
University of Strathclyde
75 Montrose Street
Glasgow G1 1XJ
United Kingdom
Tel: +44 (0)141 552 4400
E-mail: alex@cad.strath.ac.uk

Abstract
Previous experiences hold a wealth of knowledge which we often take for granted and use unknowingly through our every day working lives. In design, those experiences can play a crucial role in the success or failure of a design project, having a great deal of influence on the quality, cost and development time of a product. But how can we empower computer based design systems to acquire this knowledge? How would we use such systems to support design? This paper outlines some of the work which has been carried out in applying and developing Machine Learning techniques to support the design activity; particularly in utilising previous designs and learning the design process.

Key words: Design experience, machine learning, past design utilisation, design reuse, design process learning, agent based learning.

Final submission to: IEEE Expert special issue on AI in Design, 25 February 1997

1. Learning and Design

The importance of experience in design is that of providing a wealth of knowledge of the past which can be brought to bear on the present. Experience presents one of the most powerful resources possessed by a designer. Experiential knowledge involves the knowledge of previous designs, processes/events and the external factors involved e.g. the “actors”, agents, environmental concerns, etc. Learning is a process which helps to maintain (i.e. update and evolve) experiential knowledge. It also helps to promote the flexibility of experiences by removing highly specific details and generating more generally applicable knowledge.

Learning is a fundamental human process. While it is a commonly occurring phenomenon it is one which is poorly understood and difficult to formalise. Learning alters a human's state of knowledge and hence has a direct influence on the ability of humans to solve problems. Learning is concerned with events or entities of the current or past whereas design is directed at creating the new and defining the future.

Design is a creative problem solving activity. It often involves the exploration of alternative, and new concepts, to meet a foreseen demand or social development. The task of design is to create a specification of a concept, given a set of requirements to be achieved within a given environment. In many areas of engineering design, complexity prevents the designer producing the design in a single step. Instead the design is separated into a number of phases, each corresponding to an increasing level of detail.

We are only starting to learn what or when designers learn, and how to complement that learning process within a computer based system in order to enhance a designer's "creative" activity [1-3].

2. Machine Learning in Design

The dynamic nature of knowledge is becoming an increasingly important issue in the development of computer based design systems. Without the capacity to learn, design systems present designers with a near static knowledge source that is incapable of maintaining its knowledge and therefore will eventually become obsolete unless continually updated by knowledge engineers or system developers. Mainstream research into Computer Aided Design (CAD) has concentrated on the formulation of solutions and has neglected the generation and modification of the knowledge used to create these solutions. To a lesser extent the same criticism can also be directed at research into the development of Artificial Intelligence in Design (Intelligent CAD) systems which although stressing the explicit use of knowledge still have a considerable way to go to adequately address the important issue of learning in design.

Given that it is difficult to impart design experiences into computer based systems, automatic machine learning provides an effective mechanism for improving or increasing a system's knowledge. That is, by developing and applying Machine Learning (ML) techniques appropriate to design it is now possible, to a certain degree, to capture relevant design expertise and enhance Intelligent CAD systems. Thus, such systems are not only able to carry out automated tasks, give guidance, advice and assistance to designers but also to learn from various knowledge sources and design activities.

The development and application of Machine Learning in Design (MLinD) received little attention until the late 1980's when it started to receive more community wide consideration [4]. Fortunately work on the application of Machine Learning techniques to design has continued to emerge as researchers and the Intelligent CAD community have come to realise the potential usefulness of computational learning to aid design. The result is a considerable amount of work in the MLinD field and the reader is pointed elsewhere for further reading [5-10].

While research in MLinD is gaining momentum Reich et al [in 6] criticise it for not being scaleable to real design problems and offers an alternative approach to using machine learning in design practice more effectively. They describe and comment on six steps in the development and use of ML programs and discuss the relative pros and cons of inductive learning. They describe learning activities in design as a base to guide the development of ML programs for design practice. In particular, an information modelling and memory sharing system and its utility to support their interpretation of machine learning in design is presented. Prevalent throughout their paper is the message that the knowledge learnt has to be responsive to the multiple perspectives of the design participants.

Reich [in 6] also considers research itself as a design activity and focuses on bringing design rationale and machine learning to bear in the design of research. He does this by discussing the development of a specific research project directed at developing a machine learning design system and argues for improving the methodology of doing research on machine learning applications for design in general.

3. Machine Learning

Machine Learning is a branch of Artificial Intelligence (AI) concerned with the study and computer modelling of learning processes. It offers the potential not only to alleviate the problem of knowledge acquisition but also to enhance the system's performance in solving problems. Work on ML can be traced to the late fifties with early systems being based on neural models of the brain. Since then, considerable research has been undertaken and a number of working ML systems have been developed. The early ML systems have been classified by Carbonnel [11] on the basis of their learning strategy (i.e. rote, instruction, analogy, examples and observation), the type of knowledge acquired, and the domain of application. Kocabas [12] on the other hand classified machine learning techniques according to the "level" at which knowledge representations (e.g. rules, frames, predicate logic, semantic networks, classifiers, conceptual clustering, genetic algorithms) can be expressed, i.e. knowledge, symbol or device level. For details as to ML techniques in general the reader is also referred to Michalski et al's work [13].

The main ML techniques applied in design can be considered as:

- *Agent-based learning* is an approach which focuses upon utilising inductive techniques to gain knowledge of the interaction between different agents. Agents can be individual systems, such as Knowledge Sources in blackboard architectures, or any self-sufficient unit which performs a single or set of functions (see Lander in this special issue). They differ from other software in that they are generally developed to interact with other agents in some way. Agent-based learning focuses upon acquiring knowledge of the nature of interaction between agents and can use any ML technique to do so.
- *Analogical reasoning* is an approach directed at finding solutions to problems based upon retrieving knowledge from similar previous experiences (see Goel in this special issue). The previous solution is adapted to suit the new problem. Case Based Reasoning (CBR) (See M L Maher and de Silva Garza's review in this special issue) is a particular example of analogical reasoning. CBR involves approaches to representing, indexing, and organising past cases and processes for retrieving and modifying selected instances. The emphasis of ML based analogical reasoning is on

the generalisation of abstract properties, trends or concepts and the retrieval of a "best" match to help previously unsolved problems.

- *Induction* methods focus upon the generation of explicit symbolic knowledge from previous data, experience or examples. There are various induction methods the outcome of which can be a set of rules, patterns, logical relationships or concepts that reflect a generalisation of the input. The gained knowledge is generated through inference rules and statistical, and probabilistic, analysis. Clustering is an inductive approach that involves comparing concepts, determining a measure of similarity and grouping those concepts into a classification.
- *Genetic Algorithms* (GAs) are based on an approach analogous to biological genetics. Concepts are represented by a specific sequence of identifiers, i.e. a code. New concepts are generated using similar approaches to natural evolution, such as mutation and crossover, resulting, for example, in some improved designs. In crossover a *child* concept is generated from two *parents* through the inheritance of different elements of the code from both parents. Parts of the code can also be mutated in order to generate a new child. Newly created concepts can then be evaluated to determine their performance with selected concepts proceeding to the next step.
- *Knowledge Compilation* can use a variety of ML techniques, but has a particular focus. The motivation behind Knowledge Compilation is to simplify deeper, more fundamental, knowledge to make it more reusable and efficient. The possible outcomes have been presented by Brown [in 4] as increased efficiency, a change in the representation level, reduced amount of reasoning, facts turned into procedures ("proceduralisation"), deep knowledge transformed to surface knowledge, and surface knowledge transformed into more efficient surface knowledge. See also Goel [14] for further elaboration.
- *Neural Network* systems are based upon what is believed to be the brain's learning mechanism. The network is represented by interconnected independent processing units, i.e. nodes. Each interconnection has a weighting associated with it. A weighting specifies the degree of interconnection strength between nodes. A network consists of an input and an output "layer" of nodes, with interconnecting layers in between. The input layer represents the incoming data that is propagated through the interconnecting network to the output layer. The output layer represents the target information being learned. The network is then "trained" to associate the most likely outputs given particular inputs. Training a network involves various methods to change the weightings to reflect the associations between particular sets of inputs to identified outcomes. Once trained, new input scenarios can be fed into the system and based upon it's learned associations it then predicts the most likely outputs.

It would seem that most effort has tended to focus upon learning from previous design knowledge and the design process itself.

4. Learning from previous designs

A feature of design is the effective utilisation of previous design solutions. These solutions hold a wealth of explicit and implicit knowledge and can be interpreted differently depending upon the needs of the designer(s). With the emergence of Intelligent CAD systems, mechanisms have been developed to facilitate the effective retrieval of appropriate previous design cases and to aid in the utilisation of knowledge inherent in previous designs. One of the most notable of these is Case Based Reasoning. CBR systems are directed at the selection and modification of appropriate instances. Designers on the other hand not only use individual previous design cases but, also by learning and understanding salient features of specific designs, they can abstract or generalise their knowledge. This generalised knowledge can then be used to aid in future design scenarios.

Experienced designers can evolve relevant heuristics for a particular problem from previous designs. They can develop relationships between design solutions for alternative and changing uses or viewpoints. In addition to gaining new and general experiential knowledge, designers also learn what knowledge is suitable for their particular problem and why. This process of learning about a domain, *Domain Exploration* [15], can be time consuming and is recognised as being a very important aspect of any designer, or design's, development.

What is needed is a design tool which can effectively utilise previous design knowledge, which learns from its past, accommodates new knowledge and the needs of the designer, presents the designer with a dynamic knowledge source capable of abstracting knowledge, generalising knowledge, evolving heuristics and generating multiple views of knowledge, all to meet the needs of the designer(s). In other words, a design tool which supports a dynamic memory of past design knowledge, i.e. a dynamic design tool.

The maturity of ML learning techniques now provides the field of Intelligent CAD with an opportunity to work towards a more dynamic design tool and to re-examine computer-based utilisation of previous designs.

4.1 Analogical Reasoning

The comparison and use of generic abstractions, rather than specific cases, is a key process in analogical reasoning. The generation of those abstractions is the focus of the machine learning element of analogical reasoning in design. Bhatta and Goel [in 7] use machine learning techniques to generalise *Behaviour-Function* (BF) knowledge (see Tomiyama and Umeda in this special issue) independent of the *Structure* of the design. The generalised BF knowledge can then be used to support cross domain analogical reasoning. Thus, their system, IDEAL, supports the sharing of knowledge across different domains. Having found analogical concepts, knowledge can be shared between the domains at an abstract level, thus allowing abstracted processes and "principles" to be utilised in very different design problem solving. For example, from their experiences, a designer of a Coffee Maker may learn basic thermodynamic processes and principles and apply that learned knowledge to help them design their home heating system.

4.2 Induction based

The work of induction based approaches can be considered within the three main areas of generating multiple viewpoints to suit a designer's need, supporting the exploration of domain knowledge, and aiding the synthesis of the design solution.

<u>Multiple Viewpoints</u>

The utility of experiential knowledge and learning in design is reminiscent of Schank's understanding of human memory, i.e. *Dynamic Memory* [16], who believed that knowledge structures in memory change when new experiences are encountered or when the present structure is no longer applicable. This train of thought led to the development of Kerr and Duffy's *Customised Viewpoint* (CV) [in 6] approach which explicitly models a designer's need for knowledge and, using clustering and generalisation mechanisms, creates relevant abstraction hierarchies from previous design cases. Thus the approach recognises that the knowledge to be learned by a system should match that required by a designer. It is based upon the view that designers require various viewpoints of previous designs at different times for different reasons. For example, they may need to view aspects such as geometric, spatial, or numerical knowledge; the breakdown of structures such as compositional or taxonomic; and different perspectives determined by a designer's focus of attention. This concept has led to the development of Duffy and Duffy's PERSPECT system [15] which supports the generation of knowledge according to designers' knowledge requirements.

<u>Domain Exploration</u>

The functionality of the CV approach led to Duffy and Duffy's concept of *Shared Learning* [15] which advocates that to obtain maximum benefit from computational learning, its introduction into Intelligent CAD should not concentrate solely on empowering computers with automated learning capabilities but instead it should be directed towards a means of 'sharing' the learning activity between designers and computers and ensuring that the knowledge represented in the design systems reflects that which is of relevance, useful and understandable to designers. Thus, Shared Learning is directed at supporting Domain Exploration.

A text based analysis approach for constructing design representations from domain documents is presented by Dong and Agogino [in 17]. They present a technique for inducing a representation of the design based upon the syntactic patterns contained in textual documents describing that design. Firstly a "dictionary" of noun-phrases is created. These phrases are then clustered to discover inter-term dependencies. A Bayesian belief network also describes a conceptual hierarchy of those phrases, for the particular document domain. They integrate the document learning system with an agent-based collaborative design system to facilitate information sharing between designers.

Synthesis

Synthesising and configuring a design artefact are complex design activities involving "conceptualising" ideas, selecting appropriate components to satisfy functional design requirements and connecting those ideas and components in such a way as to determine the best overall design solution. Consequently, numerous design solutions are often generated and evaluated in order to ascertain the best solution, given the particular stage of design.

Reich's BRIDGER system is directed at the synthesis of cable-stayed bridges [18]. The system uses a clustering approach to generate two hierarchical classification structures from previous bridge designs: a *synthesis* and a *default hierarchy*. The synthesis hierarchy incorporates "existing" attributes (i.e. those used to describe the previous design). The default hierarchy is based upon "derived" attributes (i.e. those generated by the designer to describe the proportionality of previous designs, such as length to width ration). The contents of these hierarchies are generalisations of previous designs, which the system helps designers use to synthesise a new bridge design. It not only supports the retrieval of previous design cases but also extends this concept by supporting the automatic generation and selection of abstract design classes from those cases. Thus, the system provides a means to integrate the advantages of case based with previous design abstractions. Maher and Li [in 7] extends this concept by applying statistical analysis to generalise numerical empirical knowledge for the associated abstractions. While Henderson and Bailin [in 7] apply concept formation to dynamically generate hierarchical repositories of software artefacts for design reuse.

Persidis and Duffy's NODES system [19] provides knowledge modelling and design analysis support during the synthesis and modification of a design solution. Knowledge of previous design concepts are stored in Concept Libraries which provide a framework for representing, utilising and generalising knowledge of instances and classes of previous designs. Previous designs and their abstractions are organised into taxonomic hierarchies of concepts which are commonly used in a domain. The knowledge in the concept libraries is dynamically modified by induction to augment and update the knowledge of the design domain. That is, NODES induces generalised knowledge from newly defined design solutions, i.e. value ranges, nominal features and compositional (*part-of)* relations. Thus, NODES's "experience" base is automatically updated and new knowledge generalised up through the concept hierarchies to reflect newly created, and acquired, design solutions. This knowledge can then be used not only as guidance to a designer but also in the synthesis and configuration of new concepts. For example, the generalised knowledge of a concept's composition can be used to assist in automated decomposition.

4.3 Knowledge Compilation

Analysis

Cerbone [in 20] presents a three step compilation approach to augment numerical optimisation design. In the first step learning methods are used to partition the overall optimisation task into sub-problems. These sub-problems are then further simplified by using an inductive learning approach to reduce the number of independent variables. Finally, an "inductive learning algorithm is used to derive selection rules that associate problem instances to sets of candidate solutions" [(20, Page 699).] Using this approach within two design domains the approach is shown to produce a 95% speedup over traditional optimisation methods.

Schwabacher et al [in 17] are also motivated towards improving the numerical optimisation process through the reformulation of analysis "rules". They present an inductive learning approach for automatically generating rules that associates the design goal with past experiences. That is, for each design goal, constraints are generated from a set of past designs. These constraints are then used to optimise a design to meet those newly defined goals. Schwabacher et al test their approach in the domain of racing-yacht-hull design and conclude that both speed and reliability of the optimisation is improved, approaching the best performance possible.

4.4 Neural Networks

Optimising the configuration of a design solution, predicting attributes, and synthesising the design solution can be considered as the main areas of neural network support.

Configuration Design

Each design configuration (see Schreiber and Wielinga in this special issue) will have particular strengths and weaknesses in respect to its desired functionality, quality, cost and any other evaluation criteria. Optimising the configuration of the design artefact is the focus of Murdoch and Ball's [in 9] work. They use a "Technical Merit" measure to rank previous design configurations in order to identify prevalent characteristics that contributed to their success (high technical merit). Component parameters of previous design solutions are clustered into archetypes using a neural network approach. These archetypes are then compared using "duty, reliability and cost" as criteria to identify the main characteristics that contribute to either a high or low technical merit. Thus, the approach provides a basis upon which to support a designer in the creation of new design configurations that will have potentially high technical merit, and provides guidance in component and feature selection.

Prediction

Ivezic and Garrett [in 7] use the neural network approach to help predict function, form and behaviour attributes given new design requirements and partial attributes. In effect this approach is attempting to capture the relationships between the design variables (i.e. the requirements and solution attributes). On the other hand Liu and Gan's system, SPRED-1 [21], as with Rogers and Lamarsh's system, NETS/PROSS [in 20], uses previously analysed structures to train the neural network and consequently predict, for a new structural design, a set of preliminary design parameters (e.g. maximum internal forces, deflections etc.) from a set of conceptual design parameters. Thus, their approach attempts to emulate behavioural relationships. A similar approach is used by Varma et al [in 17] to facilitate the retrieval of past design cases to suit new requirements.

Synthesis

Neural networks have also been used to generate room designs. Coyne and Newton's [22] work trains the neural network to learn of room contents e.g. a particular type of kitchen design could have a double sink, a fridge-freezer etc. Given a partial definition of a room design the network predicts the remaining contents. This illustrates how a neural network can learn general descriptions of room designs and consequently use this to complete (i.e. synthesise) partial room designs.

5. Learning the design process

The design process that produced a design can be very complex. The capture of the decisions and their effect on the evolution of the design solution is often referred to as a *design plan*, rationale or sometimes history. Thus, the design rationale represents, to some degree, the rules used to carry out the design process and to evolve the design artefact to produce an acceptable solution. Design rationales can be saved for particular designs as part of the design procedure and then replayed under the right circumstances to help solve all or part of a new design problem. That is, the "rules" in the design rationale are replayed and used to help construct a new, or partial, design solution to meet new requirements e.g. BOGART [in 23]. The capture of design rationales is a basic form of learning and the reader is referred elsewhere for further information (see J Lee in this special issue). Having said this, the work which focuses upon the use of ML techniques to generalise or abstract additional knowledge from design rationales is presented in this article.

5.1 Agent based learning

Decisions made during design interact in a variety of ways and a decision by one agent can adversely constrain or indeed free the decision of others. Constraint based reasoning can alleviate, to a degree, the control and conflict resolution aspects of conflict resolution (see Bowen in this special issue). However, with the advent of Computer Supported Co-operative Working and Concurrent Engineering environments the computational issues inherent in multi-agent problem solving are becoming ever more evident and consequently problematic. Any means to aid our understanding of the learning issues in multi-agent design can only be welcomed. While relatively new, even within the still embryonic field of MLinD, some work is being directed at supporting the learning aspects of multi-agents in design.

Grecu and Brown have been addressing learning issues in agent-based design [2] and have carried out some initial experiments on learning in multi-agent based design systems [in 17]. They have studied the learning characteristics of *Single Function Agents* (SiFAs)[*] in order to investigate "how difficult it is to learn about other agents, how good the prediction of behaviour of the other agents will be, and how much learning contributes to reducing the interaction overhead". They used a concept formation approach to carry out their investigations and illustrated that what is learned about other agents depends upon the type of agent being considered, that further study is required to fully investigate the prediction behaviour of other agents, and that within different negotiation strategies the learning system outperformed the non learning system by up to 40%.

Tang combines agent based and design process based learning within his blackboard based design system [24]. The system records design histories between different agents (knowledge sources) within the blackboard architecture and uses a truth maintenance approach to maintain those histories for use in future design explorations. During new design problem solving activities these histories are used to aid in selecting suitable candidates for particular design requirements and are evolved to reflect new design problem solving experience. Tang augments this approach by learning from (clustering and generalising) and reusing previous design cases in small-molecule drug design [in 6].

5.2 Analogical reasoning

ARGO [in 23] uses abstract design plans and analogical reasoning to aid in new design problem solving. The abstract plans represent previous experiences in the form of design plans with their associated pre and post conditions. Similar problem types can then be matched against different abstract plans and the most appropriate used to help solve the new design problem. ML techniques are not used to aid in the generation or generalisation of those plans. The work by Wang & Howard [in 7] illustrate how this can be accommodated, though they too have yet to implement appropriate abstraction and generalisation techniques to make their systems more effective.

A limitation of most design process learning systems is that they require the design plan(s) to have been saved during a previous design session(s) or knowledge acquisition phase(s). But what if this was not the case? Britt and Glagowski's work is concerned with inferring the design process from a previous design case/solution [in 9]. They present a new approach, termed Reconstructive Derivational Analogy, that creates a design plan from an existing previous design solution that had no previously saved plan. That is, an existing design solution is used to automatically construct a design plan (history) of a possible decision route that may have led to the creation of that solution. The "reconstructed" plan can then be replayed for new requirements and used to help create a new design solution.

[*] A SiFA is a single function knowledge based agent which has a single "target" and point of view, e.g., select a material from the point of view of cost.

5.3 Genetic Algorithms

Gero et al [in 7] and Gage [in 17] present approaches to learning shape grammars based on GAs. Shape grammars are formal methods (rules) of generating various spatial layouts (See K Brown in this special issue). They can be used recursively and change one shape into another shape. They model generalised design knowledge and can be used to generate new design solutions to meet particular requirements. Gero et al apply the GA approach to explore alternative combinations of grammar rules, while Gage builds upon their work and applies a GA approach to multi-criteria optimisation. Their objective is to find the optimal sequence of application of the rules to best meet the defined requirements. The optimised sequence can then be reused as a starting point for new design scenarios. Thus, not only can the newly generated solutions form the basis of learned knowledge (see Schnier and Gero [in 17]) but also new grammars are learned by monitoring the success of those solutions.

5.4 Knowledge Compilation

Chabot and Brown [in 7] present a Knowledge Compilation approach using constraint inheritance. Routine design knowledge in the form of constraints is represented in the *Design Specialists and Plans Language* (DSPL). DSPL models steps in a design process. Design is carried out by executing the DSPL code. They focus upon automatically constructing new DSPL constraints to aid in new design problem solving. The system modifies and augments its constraints knowledge so as to better support a later routine design problem.

Liu and Brown [in 25] present a knowledge compilation approach to partitioning (grouping design decisions) and ordering (specifying the problem solving sequence) design problem solving knowledge in routine design. The partitioning and ordering knowledge is referred to as *decompositional knowledge* and is represented as design plans. They present a mechanism for compiling this knowledge and evaluate its effectiveness within two design domains. Their empirical results show that the compiled decompositional knowledge aids problem solving efficiency.

6. Conclusion

The systems and approaches discussed above are not exhaustively reviewed. Rather, the overview has intended to be indicative of just some of the main focus areas in the Machine Learning in Design (MLinD) field, touching on how they are assisting in the utilisation of experiential design knowledge. MLinD is still in the embryonic stages of its development but offers tremendous potential to actively support designers in all of their problem solving and knowledge requirement activities. No doubt some day non-learning systems will be relegated to the deep recesses of our own previous experiences.

So “who’s learning what in AI in Design”? Hopefully you will have learned a little from reading this article.

Acknowledgement

In writing such an article it is fitting to distinguish those that one has learned from. Of course it would take up too much space to include such a list. Suffice it say that I am indebted to Dr S M Duffy, Mr Sim Siang Kok and Mr I M McKay for their research efforts and their critical views on the work in the MLinD field. I would also like to thank the reviewers of the paper for their valuable recommendations and constructive comments. I am particularly grateful to Professors D C Brown and W Birmingham for their support and understanding in light of unforeseen circumstances.

Biography

Alex Duffy is currently a Senior Lecturer and Director of the CAD Centre at the University of Strathclyde. His main research interests have been the application of knowledge based techniques in early stage design, product and product knowledge modelling, machine learning techniques and past design utilisation, and design co-ordination. He has been active in the application of Machine Learning techniques in design since the mid 80's and has been instrumental in organising a number of workshops and special issues on this topic. After completing a Shipwright designer/draughtsman apprenticeship and a further two years in the shipbuilding industry he went to the University of Strathclyde to obtain his degree in Naval Architecture and a Ph.D. in knowledge based computer support for conceptual engineering design. He is a member of the British Computer Society (MBCS), a Fellow of the Institute of Engineering Designers (FIED), and the leader of a European (EU) Basic Research thematic network sub-group working in Design Co-ordination. He is on the advisory board of a number of international conferences and editorial boards, chairs the International Engineering Design Debate (EDD) and is a member of the IFIP Technical Committee Working Group 5.8 on Product Specification and Documentation.

References

1. Persidis, A. and Duffy, A., *Learning in engineering design,*, in Selected and Reviewed Papers and Reports from the IFIP TC/WG5.2 Third Intelligent Workshop on Computer Aided Design, Osaka, Japan, September 1989, H. Yoshikawa, F. Arbab, and T. Tomiyama, Editors. 1991: Elsevier Science Publishers B.V. (North-Holland) Amsterdam The Netherlands. p. 251-272.,
2. Grecu, D.L. and Brown, D.C., *Dimensions of Learning in Agent-based Design*, in Machine Learning in Design Workshop, 4th International Conference on Artificial Intelligent in Design (AID'96), A.H.B. Duffy, D.C. Brown, and A. Goel, Editors. 1996: Stanford University, CA, USA.,
3. Arciszewski, T. and Michalski, R.S., *Inferential design theory: A conceptual outline*, in Proceedings of the Third International Conference on Artificial Intelligence in Design (AID'94), J.S. Gero and F. Sudweeks, Editors. 1994, Kluwer Academic Publishers, The Netherlands: Lausanne, Switzerland.,
4. Yoshikawa, H., Arbab, F., and Tomiyama, T. *Intelligent CAD III, IFIP TC/WG5.2 Workshop, Osaka, Japan, September 1989.* 1991. Amsterdam, The Netherlands: Elsevier Science Publishers B.V. (North-Holland).,
5. Maher, M.L., Brown, D.C., and Duffy, A.H.B., eds. *Machine Learning in Design Workshop*. 2nd Int. Conference on Artificial Intelligence in Design (AID'92), Pittsburgh, U.S.A. 1992.,
6. Duffy, A.H.B., *Special Issue on Machine Learning in Design.* Artificial Intelligence in Engineering, 1993. **8**(3).,
7. Maher, M.L., Brown, D.C., and Duffy, A.H.B., *Special Issue: Machine Learning in Design.* Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM), 1994. **8**(2).,
8. Duffy, A., Brown, D.C., and Maher, M.L., eds. *Machine Learning in Design Workshop*. 3rd Int. Conference on Artificial Intelligence in Design (AID'94), Lausanne, Switzerland. 1994.,
9. Duffy, A.H.B., Brown, D.C., and Maher, M.L., *Special Issue: Machine Learning in Design.* Artificial Intelligence for Engineering Design Analysis and Manufacturing (AIEDAM), 1996. **10**(2).,
10. Duffy, A., Brown, D.C., and Goel, A., eds. *Machine Learning in Design Workshop*. 4th Int. Conference on Artificial Intelligence in Design (AID'96), Stanford, U.S.A. 1996.,

11. Carbonnel, J.G., Michalski, R.S., and Mitchell, T.M., *Machine learning: a historical and methodological analysis,.* AI Magazine, 1983. **4**(3): p. 69-79.,
12. Kocabas, S., *A review of learning,.* The Knowledge Engineering Review, 1991. **6**(3): p. 195-222.,
13. Michalski, R.S., Carbonell, J.C., and Mitchell, T.M., eds. *Machine Learning: An Artificial Intelligence Approach*. 1983, Morgan Kaufman.,
14. Goel, A.K., *Special Issue on Knowledge Compilation.* IEEE Expert, 1991. **6**(2).,
15. Duffy, S.M. and Duffy, A.H.B., *Sharing the learning activity using Intelligent CAD.* Artificial Intelligence for Engineering Design Analysis and Manufacturing (AIEDAM), 1996. **10**(2): p. 83-100.,
16. Schank, R., *Dynamic Memory: A theory of remindings and learning in computers and people*. 1982, Cambridge UK: Cambridge University Press.,
17. Gero, J.S. and Sudweeks, F., eds. *Artifical Intelligence in Design (AID'96)*. Proceedings of the Fourth International Conference on Artificial Intelligence in Design, Stanford, U.S.A. 1996, Kluwer Academic Publishers, The Netherlands.,
18. Reich, Y. and Fenves, S.J. *The formation and use of abstract concepts in design,.* in Concept Formation: Knowledge and experience in unsupervised learning. 1991: Morgan Kauffman.,
19. Duffy, A.H.B., Persidis, A., and MacCallum, K.J., *NODES: a Numerical and Object based modelling system for conceptual engineering DESign.* Knowledge-Based Systems, 1996. **9**: p. 183-206.,
20. Gero, J.S., ed. *Artifical Intelligence in Design (AID'92)*. Proceedings of the Second International Conference on Artificial Intelligence in Design, Pittsburgh, U.S.A. 1992, Kluwer Academic Publishers, The Netherlands.,
21. Liu, X. and Gan, M. *A preliminary structural design expert system (SPRED-1) based on neural networks,.* in Proceedings of the First International Conference in Artificial Intelligence in Design. 1991. Butterworth-Heinemann Oxford UK.,
22. Coyne, R.D., Newton, S., and Sudweeks, F., *A connectionist view of creative design reasoning,*, in Modelling Creativity and Knowledge-Based Creative Design, J.S. Gero and M.L. Maher, Editors. 1993, Lawrence Erlbaum Associates, Hillside, New Jersey, USA. p. 177-209.,
23. Tong, C. and Sriram, D., eds. *Artificial Intelligence in Engineering Design Volume II*. 1992, Academic Press, Boston.,
24. Tang, M.X., *Knowledge-based design support and inductive learning,* Ph.D. Department of Artificial Intelligence. University of Edinburgh. 1994. Edinburgh,
25. Gero, J.S. and Sudweeks, F., eds. *Artifical Intelligence in Design (AID'94)*. Proceedings of the Third International Conference on Artificial Intelligence in Design, Lausanne, Switzerland. 1994, Kluwer Academic Publishers, The Netherlands.,