New Knowledge-Based CAD Models of Design

J S GERO*

ABSTRACT

Knowledge-based systems utilise concepts from artificial intelligence. They are the bases of new models of design which have the potential to extend the utility of computers in design. This paper briefly reviews current research supporting new knowledge-based CAD models of design before describing and elaborating two such models. One is case-based design and the other is creative design.

Background

The early work on computer-aided design fell into two distinct and disparate groupings. The first was concerned with analysis methods embodied in computer programs. This has resulted in today's finite element method techniques and programs. The finite element method is, nowadays, a mature technology. The second was concerned with graphics, commencing with Ivan Sutherland's SKETCHPAD. For a long time this form of graphics led researchers to concentrate on data structures to support graphical image making. Later, the emphasis shifted away from graphical image making to the representation of models of objects leading researchers to concentrate on models and data structures for geometric modelling.

This work was and continues to be based on particular paradigms of the roles of the computer in the design process. In the work characterised by the finite element method the paradigm assumes that a sufficient representation can be encoded to allow analysis to be automated. In the work characterised by geometric modelling the paradigm assumes that the representation is the problem since the analysis is left to the designer. Neither of these is based on a paradigm which gives the computer a more active role in the entire process of designing.

From the beginning of the 1980s there has been a burgeoning interest in understanding and using knowledge-based approaches drawn from artificial intelligence. These approaches are often couched under such labels as information technology, knowledge-based systems, expert systems and so on. What they all have in common is the move from using the computer with algebraic models and numerical values for the variables in those models to symbolic models and symbolic values for the variables in those models. Along

* *Professor of Design Science, Key Centre of Design Computing, University of Sydney, NSW 2006 Australia.*

with the move to the use of knowledge-based systems has come an increasing interest in expanding the role of computers and redefining computer-aided design in the service of design.

Research Supporting New Models of Design

New knowledge-based models of design based on the artificial intelligence paradigm make use of the fundamental concepts of:

- symbolic variables
- separation of knowledge from control
- symbolic reasoning.

Ten of the most significant research areas supporting these new design models will be briefly described before elaborating two models in more detail.

Representation in Design

A fundamental problem for artificial intelligence and design remains the one of representation. What is it that a designer knows and how does it get represented in a computer? There are two disparate kinds of knowledge of interest here: that concerned with design processes and that concerned with the artefact as it is being designed. Even if there is no concern with what a human designer knows there is still the question of what knowledge a computational model of design needs and how to represent it.

Knowledge-based design has moved from being treated as a knowledge-lean problem to being treated as a knowledge-rich problem. Thus, increasing amounts of knowledge need to be formalised, structured and represented. Three kinds of knowledge with their associated processes need to be represented:

- case knowledge (episodes or precedents)
- generalised or compiled knowledge (derived from cases)
- first principles knowledge.

In traditional CAD systems only the artefact as it is being designed or after it has been designed is represented. Only the physical description is usually represented in these systems. In knowledge-based CAD systems much more needs to be represented to increase the utility of the system.

Design Semantics

Two issues are mentioned here. The first issue is the coding-decoding problem. Coding is the process of converting the human's understanding of an artefact into a representation in a computer. The traditional way is to draw defining images of the artefact in order to construct a model of it. Decoding is the process of converting a representation into human interpretable images. How does a system decode a representation that has been altered after it has been coded or if the representation is being decoded in a different context.

One important aspect of design is the shifting context it creates for its own activities. Such changes in context offer the opportunity for *emergence*—where an interpretation of the semantics of a representation is made which is different to that explicitly made in the representation.

The second issue concerns how do you represent in an explicit and manipulable form the intentions, purposes or functions of the intended artefact in such a manner that they can be used. This has important implications for data exchange between designers and for data exchange standards.

Reasoning in Design

Much of the reasoning machinery brought across from artificial intelligence has been concerned with monotonic logics, with consistency maintenance and with resolving conflicting constraints. These reasoning processes have been developed for a static world. The design world by its very nature is not static and the appropriate reasoning mode is abductive (i.e. what could be) rather than deductive (i.e. what must be). Expert systems generally use deductive reasoning so their use in design synthesis presents special difficulties not encountered when they are used in design analysis. It is common in design to maintain inconsistent beliefs for a time and to resolve conflicting constraints by designing them away.

Combinatorial Explosion in Design

Abductive reasoning brings with it the very real likelihood of combinatorial explosion of potential inferences. As soon as a system deals with what could be rather than what must be it could go on indefinitely. Constraint propagation, planning and heuristics are common ways of addressing combinatorial explosion. However, alternate approaches based on evaluating the satisfaction of solutions or solution directions are likely to be more useful in design.

Indexing in Design

Design occurs in a knowledge-rich and knowledge-intensive environment. However, the more knowledge that is coded into the system the harder it is to find what is useful. The introduction of multimedia with its emphasis on visual images in pixel only form brings with it unique indexing problems. Free text indexing is readily available today and to a lesser extent semantic text-based indexing is available. No such free visual image indexing systems yet exist. Much design knowledge can be placed into one of the three categories of: cases (episodes or precedences), generalised knowledge based on cases and first principles knowledge. What, when and how to index: these still remain difficult questions to answer.

Dynamic Modification—Learning in Design

In design synthesis, unlike in fields which rely exclusively on deductive processes, obtaining the same solution each time for the same problem is considered a failure of design. Designers learn from doing design and learn from their own and other's designs. This learning results in a dynamic modification of both the knowledge and knowledge structures used to represent the knowledge. Learning in knowledge-based systems is a well-developed area but it is only recently that learning is being included in knowledge-based design systems and further research is needed.

Situation Recognition in Design

An important research area for artificial intelligence in design is how to produce systems capable of recognising situations at a semantic (strategic) level rather than simply at the syntactical (tactical) level. Much of the interest in non-routine design lies in the emergence of newly recognised situations, situations which were not produced intentionally but by extension. Situation recognition is particularly important in design by analogy as occurs in case-based design.

Collaborative Design

Designers rarely work alone, design has become so complex an activity that many specialist designers are involved. Synchronous collaboration across networks is the goal. Designers working in disparate locations sharing the same 'whiteboard' should be able to work on the same drawing at the same time. How to provide real-time computational support to improve collaboration between individuals in a design team has become a critical issue. Ideas from distributed artificial intelligence, networked whiteboards, and emergence, provide useful starting points.

Non-Routine or Creative Design

Design and creativity are often treated synonymously by many people. Clear definitional distinctions have been drawn between routine and non-routine design with the acceptance that not all design is creative. Basic questions remain: are there principles of creativity; are there creative processes; what kind of computational support can be provided in a non-routine design context? Considerable research is currently underway aimed at addressing these questions with the goal of ultimately providing additional computer support at the very early stages of design; the stages currently not supported by CAD.

Evaluation in Design

The evaluation processes in design include not only the evaluation of the a priori defined technical performance of the designed artefact but an assessment of emerging performance as well as the assessment of its socio-ethical value. This latter aspect currently eludes any formal description. However, these issues need to be addressed.

New Knowledge-Based CAD Models of Design

A number of new knowledge-based CAD models of design are under investigation and development building upon some of the outcomes of some of the issues raised in the previous section. Two of these new knowledge-based CAD models will be described here.

Case-Based Design

Case-based reasoning is a well-defined paradigm in artificial intelligence. It is based on the premise that humans also reason from specific experiences rather than by only following a set of general guidelines. For example, reasoning from precedents is one of the basic methodologies in law. Case-based reasoning relates a current situation to the closest most specific experience in memory and uses that experience to solve the problem at hand. It is thus a memory-based approach rather than a computation approach, whereby solutions to problems already solved need only be retrieved rather than computed again. The key factors in case-based reasoning are the storage of cases as complete patterns of experiences including the reasoning process, the ability to be reminded of the most appropriate case and the application of that case to the current situation. Application of the case may either be a direct application if the current situation and that of the case match exactly, or there may be a need for some modification of the case. This modification may be of various degrees of severity. Case-based reasoning uses the strategies of modification and repair to effect such modifications. New cases are produced either as variations on the previous case or, in extreme situations, as new cases if considerable modification took place. Case-based reasoning thus incorporates a learning capacity in the form of new cases being incorporated into a dynamic case base.

Searching for a case is based on indexing cases with regards to various factors, e.g. goals and attributes. The more efficient the indexing, the more efficient the search. Retrieval is a matter of pattern matching, i.e. matching a pattern of requirements to an existing set. This match may be exact or partial. In the case of partial matches, some criteria are required to determine the 'best' partial match. Matches may be made to parts of several cases and a new case results from combining elements from these cases, if consistency is satisfied.

The processes involved in case-based design are search, match, retrieve, select, modify, repair and store.

Search—given a problem description of requirements including functions to be achieved, required behaviour performances, the design environment and even constraints on values of structure variables, the case base must be searched to find an appropriate design case. The utility of case-based designing is strongly dependent on the efficiency of the search procedure. Searching could be sequential, parallel or direct using an indexing mechanism. Indexing must be done on the function, behaviour, structure and context features.

Match—an appropriate case for consideration is found with regards to the matching of above mentioned features. Perfect matching, i.e. where the required features are found exactly in a case, is unusual. Partial matching occurs when some of the features are matched or the features are matched to some degree.

Retrieve—a case which matches to some defined degree needs to be retrieved for consideration. This may or may not involve display of these cases to the users for perusal and consideration.

Select—a selection of a single case as the basis for determining the design solution has to be made. Alternatively, if only part of a design case is required, then several design cases may require to be selected, and the necessary parts of each extracted. In either situation, the 'best' matching design case should be selected. Selection of the 'best' design case can be on the basis of the most similar or the most useful match. Selection can be carried out by the system or by users after consideration of an appropriate set of candidates retrieved by the system. Selection by the system based on partial matching entails such factors as the importance of the features matched as well as how close they are matched.

Modify —where a design case is selected which does not match the design requirements sufficiently, some modifications will be necessary. This may involve the replacement of variables with other variables or simply the alteration of some values of variables.

Repair—in many situations, a modification to an existing design case based on substitution of variables or modification of values will cause some performance failure in some other behaviour or function. For example, decreasing the cross-sectional area of a column to satisfy some new spatial requirement may cause buckling. Other modifications may be considered but none may be satisfactory. One of two directions now needs be taken. Either an alternative design case is selected based on the new information known regarding the necessity for modifications and the effects of modifications or the current selected design case is modified in such a way as to make it acceptable. This latter process is known as the process of *repair* in case-based reasoning.

Store—after a design case has been modified or repaired, a new design case has been generated. If this new design case is considered to be sufficiently important as a design experience different to existing design cases, then it must be stored in the case base with appropriate indexing. Where the failure of solutions is seen as an important piece of information to the anticipation of future problems, this must be noted in the design case.

Case-based computer-aided design introduces a new paradigm into CAD and moves the potential utility of CAD from the drafting and documentation phase closer to the conceptual design phase.

Non-Routine or Creative Design

Non-routine design or *creative design* can be defined as that class of design activity when all the variables which define the structure and behaviour are not known a priori nor necessarily are all the processes needed to produce them. The implication of this conceptualisation of non-routine design is that the focus is on processes for the introduction of new variables into the design and their integration into the existing variable structure. It is suggested that this is one basis for the production of potentially creative designs.

For a given set of variables and processes operating within a bounded context any model will construct a bounded state space. Creative design can be represented in such a state space by a change in the state space. Routine design does not change the state space, it simply searches within it. There are two classes of change to the state space possible: addition and substitution. The *additive class of state space change* is represented in Figure 1 where the new state space, S_n, totally contains the original state space, S_o. The implication of the additive class of state space change is that new variables are added to the existing stock of variables.

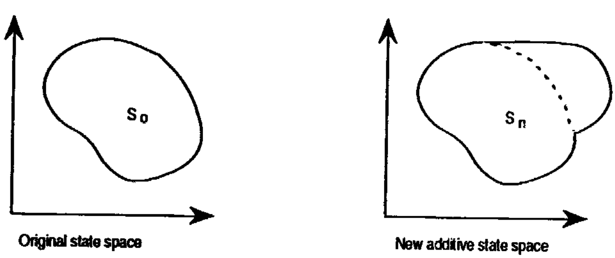

Figure 1. The Change in State Space Due to the Addition of New Variables

The *substitutive class of state space change* is represented in Figure 2 where the new state space, S_n, does not cover the original state space, S_o. The implication of the substitutive class of state space change is that some (or in the extreme case all) of the existing variables are deleted and new ones are added to the remaining stock of variables.

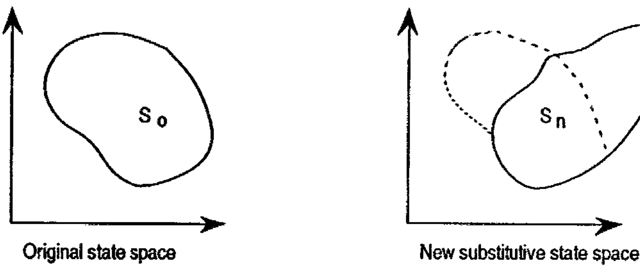

Figure 2. The Change in the State Space Due to the Substitution of New Variables

Whilst the additive and substitutive classes of state space change have been presented as if the variables being used are structure or behaviour variables only, this need not be the case. Modifications to the knowledge structures and to the contents of knowledge structures fall into these classes also and have the potential to be part of the creative process. For example, take a rule-based system for the production of a design. A design is produced by a defined sequence of executions of the rules, i.e. the plan or control, for a given set of rules. Concern with determining which is the best plan for the given rules places that endeavour in the realm of routine design. However, if there is a process for modifying the rules themselves within the planning process then it is possible to produce designs with behaviours or structures outside the original state spaces. Computational systems which exploit this concept are now being developed.

Creative computer-aided design systems make use of a variety of such processes, chief amongst them are the following:

- combination
- mutation
- analogy
- first principles
- emergence.

Combination—as a creative design process combination involves the addition of components from two separate designs. This combination is expressed in terms of the addition of variables. One common computational model for carrying out this combination is based on modelling the design process as a genetic algorithm. Here the genetic process of cross-over is the analog of combination. Novel designs can be produced this way.

Mutation—as a creative design process mutation involves a modification to an existing design variable to produce a new design variable. Typical mutation operators include the algebraic and set theoretic operators. Thus, division, for example, divides a single variable into two like variables. Such an operation can affect the resultant topology of the artefact. Mutation is also a process in genetic algorithms.

Analogy—analogy is defined as the product of a process in which specific coherent aspects of the conceptual structure of one design are matched with and transferred to another design. Based on the nature of the knowledge transferred to the new design, analogical reasoning processes can be placed into one of the two classes of transformational analogy or derivational analogy. Transformational analogy adapts the structure of a previous design to be useful in the present design. Derivational analogy applies the design process used in a previous design to the production of the current design. The effect of transformational analogy is the introduction of new variables into the current design.

First principles—first principles relies on causal, qualitative or computational knowledge used abductively to relate intentions (functions) to behaviour and behaviour to structure without the use of compiled knowledge. Design using first principles is the least developed of the processes described so far.

Emergence—emergence is the process whereby extensional properties of a design are recognised beyond its intentional ones, i.e. properties which were not intentionally explicit are recognised and made explicit. Computational models of emergence are only now being developed concentrating on shape emergence.

Non-routine or creative computer-aided design is at the cutting-edge of research and development in computer-aided design. Its goal is to improve the effectiveness and quality of designs by providing computational support to designers during the conceptual phase of design.

These two models of design—case-based computer-aided design and creative computer-aided design—are developments founded on concepts from knowledge-based systems which have allowed an expansion of the possible roles of computers in the design process. This is the beginning of a redefinition of computer-aided design.

References

Brown, D and Chandrasekaran, B (1989), *Design Problem Solving: Knowledge Structures and Control Strategies*, Morgan Kaufmann, San Mateo, CA.

Brown, D, Waldron, M and Yoshikawa, H (eds) (1992), *Intelligent Computer-Aided Design*, North-Holland, Amsterdam.

Coyne, R D, Rosenman, M A, Radford, A D, Balachandran, M and Gero, J S (1990), *Knowledge-Based Design Systems*, Addison-Wesley, Reading.

Gero, J S (ed.) (1991), *Artificial Intelligence in Design '91*, Butterworth-Heinemann, Oxford.

Gero, J S (ed.) (1992), *Artificial Intelligence in Design '92*, Kluwer, Dordrecht.

Gero, J S and Maher, M L (eds) (1993), *Modeling Creativity and Knowledge-Based Creative Design*, Lawrence Erlbaum, Hillsdale, NJ.

Green, M (ed.) (1992), *Knowledge Aided Design*, Academic Press, London.

Tong, C and Sriram, D (eds) (1992), *Artificial Intelligence in Engineering Design, Vols I and II*, Academic Press, Boston.

Topping, B (ed.) (1992), *Optimization and Artificial Intelligence in Civil and Structural Engineering—Vol. II*, Kluwer, Dordrecht.

Rychener, M (ed.) (1988), *Expert Systems for Engineering Design*, Academic Press, San Diego.

c. 1993, Management of Information Technology for Construction, K. Mathur et al (Eds), World Scientific Publishing Co., Singapore.