3
A Locus for Knowledge-Based Systems in CAAD Education

John S. Gero

Department of Architectural and Design Science
University of Sydney

This paper outlines a possible locus for knowledge-based systems in computer-aided architectural design education. It commences with a review of computer-aided architectural design and knowledge-based systems. It then proposes their use at various stages in CAAD education.

Introduction

There are a number of paradigms to consider in computer-aided architectural design education. We could think of CAAD education as a means to equip student architects with the ideas and skills needed to use CAAD systems in their work after they graduate. We could think of CAAD education as a means of specialising architecture students to control the development of CAAD systems and even to contribute to the development of such systems themselves. We could think of CAAD education as a means of providing architecture students with commercial-quality facilities to expedite the production aspects of their design processes. We could think of CAAD education as a means of providing a laboratory for architecture students to explore a new design medium. We could think of other paradigms but these four cover the major issues. In this paper, we briefly examine the potential role of knowledge-based systems in each of these paradigms.

The next section provides an overview of computer-aided architectural design. The following reviews knowledge-based systems. The final section examines the locus of knowledge-based systems in CAAD education.

Computer-Aided Architectural Design

Computer-aided architectural design has passed through a number of distinct phases. It commenced in the 1960s with a concern for graphical representation of objects being designed. This graphical genesis still manifests itself in today's computer-aided architectural design systems. In the 1970s there was a concern for object modeling to support graphical representation. The aspect being modeled was geometry and topology. It is often simply called geometric modeling. There was the recognition that aspects other than geometric were also needed, so many systems allowed the inclusion of non-geometric attributes by attaching them to geometric entities.

By the end of the 1970s and the early 1980s geometric modeling had reached sophisticated levels. At the same time analysis tools were finding their way into computer-aided architectural design systems. However, with some exceptions, computer-aided architectural design systems were not concerned with providing direct assistance to designers in their design decisionmaking processes. The exceptions derived their impetus from operations research techniques but did not find widespread acceptance. Recently, there has been renewed interest in using computer-based methods as direct aids to design decision making.

Designing, whether with the aid of computers or not, involves transforming a description expressed in function terms to a fixed description expressed in structure terms. *Functions* are the requirements, specifications or goals. Part of designing involves determining the functions. *Structure* is the set of elements and their relationships that go to make up an artifact. When looking at the description of structure there is no explicit function evident. Similarly, function contains no structure. Since these two classes have no descriptors in common how can one be transformed into the other.

A designer's experience allows him to map function onto structure. This is how the abductive rules in expert systems encode this knowledge. However, such a direct mapping does not allow for any reasoning about the transformation process since it is a direct mapping. How does a designer incorporate new structures? It is suggested that both function and structure are translated into a homogeneous concept, namely, behaviour. Function is decomposed into expected *behaviour*. If this behaviour is exhibited by the structure then the function is produced. From the structure the actual behaviour can be deduced. In engineering the deductive process of producing the actual behaviour is called "analysis". Further it is suggested that function, structure, and behaviour are bound together into a single conceptual schema through experience (Gero, 1989).

What are Knowledge-Based Systems?

Knowledge Engineering

Knowledge-based systems are computer programs in which the knowledge is explicitly coded rather than implicitly encoded. They make use of knowledge engineering. *Knowledge engineering* is a subfield of artificial intelligence. It is concerned with the acquisition, representation and manipulation of human knowledge in symbolic form. Human knowledge is thought of as being reasoning (rather than the simple ability to acquire facts as you might find in an encyclopedia). Just as the industrial revolution can be considered to have automated mechanical power, and the computer revolution to have automated calculation, so knowledge engineering automates reasoning.

Feigenbaum (1977) defines the activity of knowledge engineering as follows:

> *The knowledge engineer practices the art of bringing the principles and tools of artificial intelligence research to bear on difficult application problems requiring experts' knowledge for their solution. The technical issues of acquiring this knowledge, representing it, and using it appropriately to construct and explain lines of reasoning are important in the design of knowledge-based systems... The art of constructing intelligent agents is both part of and an extension of the programming art. It is the art of building complex computer programs that represent and reason with knowledge of the world.*

The fundamental structure used to represent reasoning and, hence, knowledge, is symbolic inference. Inference is based on well established logic principles and has been extended to operate on symbols. The obvious advantage of inferencing is that it does not require an a priori mathematical theory such as is found in, say, hydraulics or structures. It can be used to manipulate concepts. Barr and Feigenbaum (1981), talking about the applicability of knowledge engineering in conceptual areas, state:

> *Since there are no mathematical cores to structure the calculational use of the computer, such areas will inevitably be served by symbolic models and symbolic inference techniques.*

Expert Systems

Expert systems have been defined as knowledge-based computer programs which use symbolic inference procedures to deal with problems that are difficult enough to require significant human expertise for their solution. Human experts can be compared with conventional computer programs in the following respects (Lansdown, 1982):

• Human expertise arises from the possession of structured experience and knowledge in a specific subject area. These skills grow as more and more experience is gained.
• Human experts can explain and, if necessary, defend the advice they give and are aware of its wider implication.
• Human experts determine which knowledge is applicable rather than proceeding algorithmically.
• Human experts can, and frequently have to, act with partial information. In order to supplement this, they ask only sufficient and pertinent questions to allow them to arrive at a conclusion.

Conventional computer programs differ markedly from programs which act as experts.

• They are usually complex and difficult for anyone other than their designers to understand.
• They embody their knowledge of the subject area in terms designed for computational efficiency such that this knowledge is intertwined with the control parts of the program. Thus, the knowledge is implicit in the program in such a way which makes it difficult to alter or change.
• They cannot suggest to their users why they need a particular fact nor justify their results.

Thus, expert systems aim to capture the ability of human experts to ask pertinent questions, to explain why they are asking them, and to defend their conclusions. These aspects are unrelated to a specific domain of knowledge and apply to all experts. Expert systems are computer programs which attempt to behave in a manner similar to rational human experts. They all share a common fundamental architecture even if the knowledge encoding mechanisms differ. An expert system will have the following components:

• an inference engine - this carries out the reasoning tasks and makes the system act like an expert
• a knowledge base - this contains the expert's domain specific knowledge and is quite separate from the inference engine
• an explanation facility - this interacts with both the knowledge base and the inference engine to explain why an answer is needed at a particular point or how a question can be answered; further it is used to explain how a conclusion was reached or to explain why a specific conclusion could not be reached
• a state description or working memory - this contains the facts which have been inferred to be true and those which have been found to be false during a particular session, as well as the facts provided by the user or another system.

• a knowledge acquisition facility -this allows the knowledge base to be modified and extended.
• a natural language interface - few expert systems have this yet.

Knowledge-Based Design Systems

Expert systems were originally developed to carry out diagnosis using classification concepts. They readily lend themselves to engineering analysis and evaluations, i.e. design analysis. Design analysis may be considered as the interpretation of a design description. The facts which describe an object and the knowledge by which properties of the object can be derived can be modelled as formal axiomatic systems. The advantage is that knowledge becomes amenable to formal proof procedures and the mechanism of logical inference (Kowalski, 1983).

Workable systems can be devised which operate on the basis of formal reasoning. This is particularly so in the case of interpreting the properties and performances of buildings where the theory by which interpretations can be made is well understood. This is generally the case, for example, when evaluating the performance of buildings for compliance with the requirements of building codes.

Expert systems of this type are also applicable to the synthesis of designs, particularly for those classes of design problem which can be subdivided into independent subproblems. But expert systems which are applicable to the more general class of design problem can also be devised.

Expert systems for design analysis are well-described in the literature (Sriram and Adey, 1986a; Sriram and Adey, 1986b; Sriram and Adey, 1987a; Sriram and Adey, 1987b; Sriram and Adey, 1987c; Gero, 1988a; Gero, 1988b; Gero, 1988c; Dym, 1985; Maher, 1987; Pham, 1988). Expert systems for design synthesis can also be found in the above references as well as in Rychener (1988), Gero (1985), and Gero (1987). The foundations of the use of expert systems and knowledge-based systems for design analysis and design synthesis are presented in Coyne et al. (1989).

Locus of Knowledge-Based Systems in CAAD Education

Learning How to Use CAAD Systems

Current CAAD systems involve constructing geometric and topological models of buildings. Often the manuals for these systems are many hundreds of pages long. Some are even over a thousand pages long. The typical student learning curve to develop facility with such a system looks like that shown in figure 1.

The complexity of these systems, their lack of standardisation and their lack of an intuitive approach and interface means that the development of skills to use them in a productive fashion is enormously time-consuming. Further, it takes considerable teaching staff time to provide the necessary tutorial effort to support this skill development.

Much of the knowledge in the manuals could be translated into the knowledge base of an expert system. This expert system could sit alongside the CAAD system as a desktop accessory and provide both information and guidance to the CAAD system user. It could be used by both novices and experienced users, each of whom would utilise different knowledge. Bennett et al. (1978) developed an expert system to guide users in the proper use of a large computer-aided analysis package. The approach is presented graphically in figure 2. The expert system is considerably more than an on-line manual. It can be interrogated by the user and extensive dialogue between the user and the expert system is likely to occur. A further sophistication would be to have a generic expert system for this task which semi-automatically acquired the knowledge about the CAAD system directly from the users manual. Such an expert system would have a model of CAAD systems as its foundation.

Learning About Developing CAAD Systems

CAD systems used to be considered general purpose, i.e. the same system was used by engineers and by architects. It was soon realised that specialisation was needed for each professional group. Thus, we see CAAD systems being developed which share elements with systems designed for other professionals but having approaches built into them that make them unique to architecture. We are beginning to see *expectation-driven systems* that contain domainspecific knowledge being developed. Such CAAD systems embody considerably more architectural ideas than before. A small percentage of architecture students will end up developing the next generation of CAAD systems and it is to these students that such an approach is addressed.

An expert system that contained knowledge about the design ideas behind current CAAD systems and about CAAD system principles could be used to assist students to learn about developing CAAD systems. This approach is presented graphically in figure 3. The expert system would sit between the user and a variety of CAAD systems. it would draw specific instance examples from the CAAD systems to demonstrate system design principles.

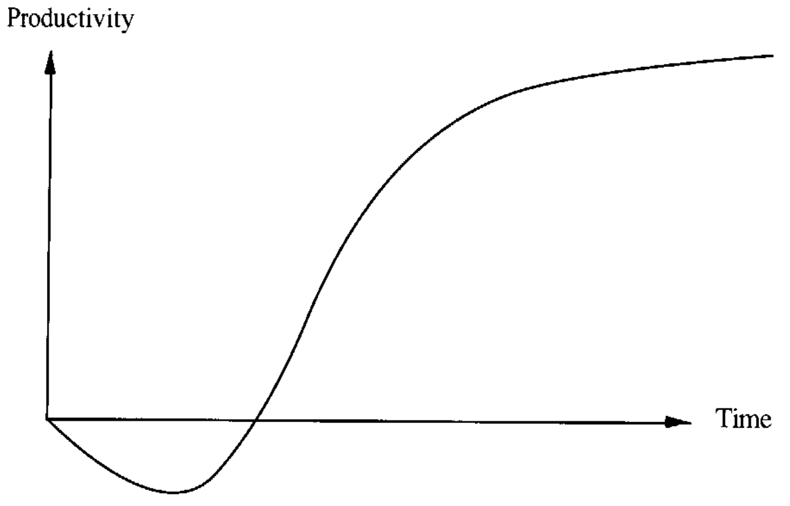

Figure 1 Typical time-productivity learning curve

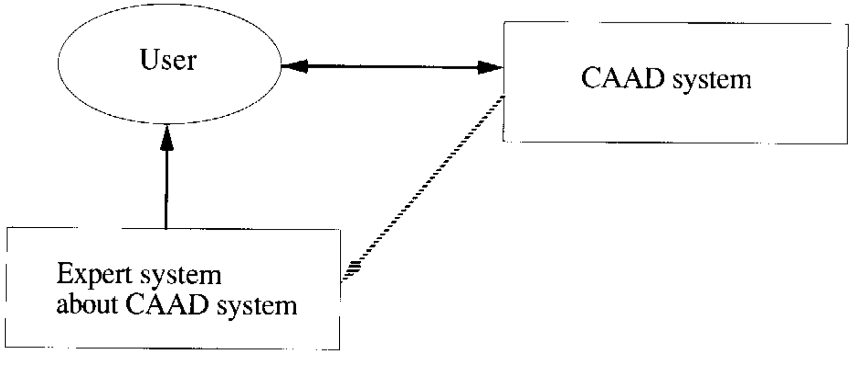

Figure 2 Using an expert system to inform and guide users of the CAAD system.

Learning About Production

It takes experience to develop an intuitive understanding of what needs to be presented at the various stages in developing a design. The information presented to the client at the concept stage is quite different to that needed by the city planning department. This in turn is quite different to that needed by the contractor. This professional practice knowledge could be encapsulated in the knowledge base of an expert system. This expert system would interact with the user to determine the stage of the project and then control the CAAD system to produce the appropriate documentation. This is shown graphically in figure 4.

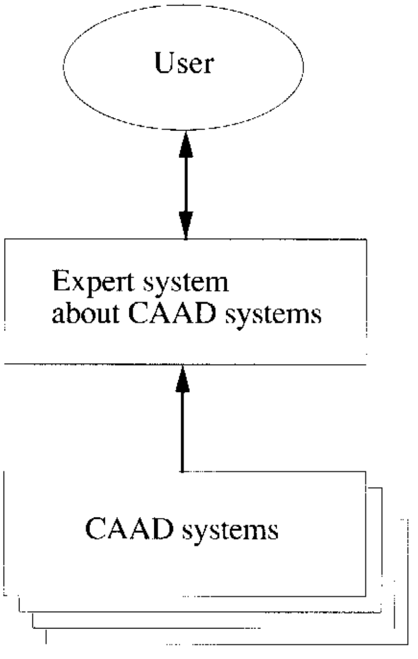

Figure 3 Using an expert system to teach about CAAD systems design and development

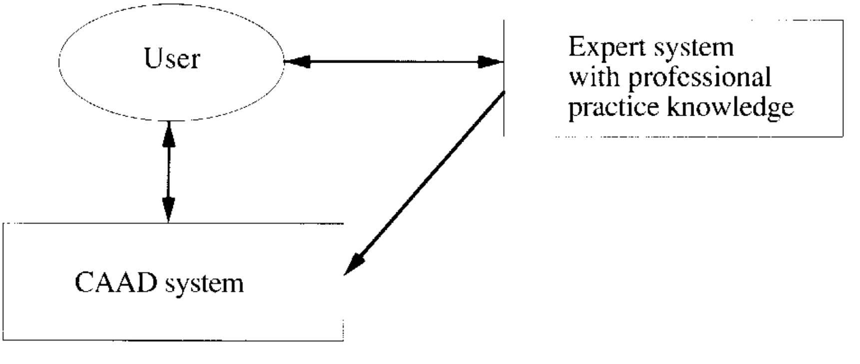

Figure 4 Using an expert system to teach about and to control the documentation produced at various stages of a project.

Since there are no legal consequences if the documentation is incomplete, it would be sufficient if the expert system controlled the CAAD system to produce indicative documentation in some areas.

Learning About Design Using a Wove! Medium

Probably the most interesting approach is to consider the use of artificial intelligence in computer-aided architectural design systems as presenting a novel design medium It is only now that *research into* knowledge-based design is beginning to show results. The inclusion of symbolic reasoning, one of the bases of artificial *intelligence, is* changing our expectations of CAAD systems. Further, it is re-awakening the *interest in* design theory and methods as we develop more powerful *representation tools* (Coyne et al., 1989; Gero and Maher, 1988).

We are beginning to *see the* emergence of knowledge-based CAAD systems (figure 5). It is still very early but the promise is *there. New* ideas about how design experience can be acquired, *represented, and* manipulated offer the potential that we can produce systems closer to our *expectations than* existing systems (Gero and Rosenman, 1989).

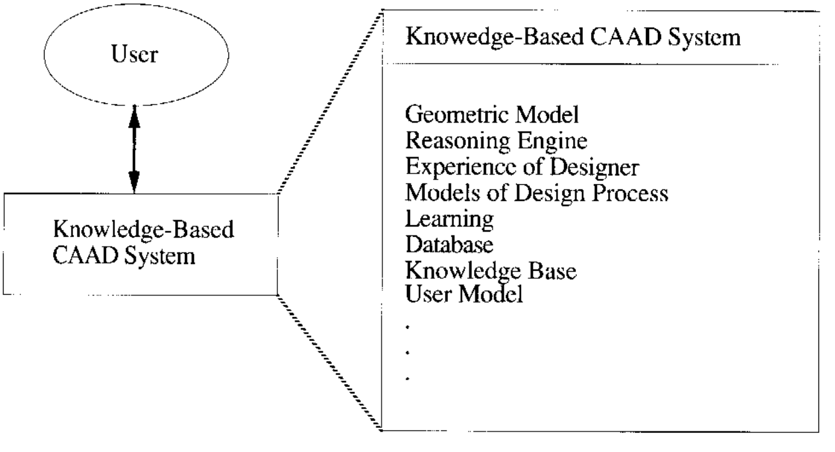

Figure 5 Knowledge-based CAAD systems

Conclusion

The first three paradigms of the use of knowledge-based systems are founded on the body of research into tutoring using artificial intelligence (Woolf, 1988). Intelligent tutoring systems are concerned with how to guide a student to learn new concepts- Intelligent tutoring systems have been successfully constructed. The fourth paradigm draws ideas but not systems from other domains. It is here that we, as educators and researchers, will have to carry out both the research and development.

References

Barr, A. and Feigenbaum, F. (eds). 1981. *Handbook of* Artificial Intelligence, Vol.1. Los Altos: William Kaufmann.

Bennett,J. S., Creary, L. A., Engelmore, R. M., and Melosh, R. F. 1978. SACON: A *knowledge-based consultant in* structural *analysis,* heuristic Programming Project Rep. No. HPP-78-23. Stanford: Computer *Science Department, Stanford* University.

Coyne, R. C., Rosenman, M. A., Radford, A. D., Balachandran, M. and Gero, 1.5. 1989. Knowledge-Based Design Systems. Reading, MA: Addison-Wesley.

Dym, C. L. (ed.). 1985. Applications of *Knowledge-Based Systems* to Engineering Analysis and Design. New York: American Society of Mechanical Engineers.

Feigenbaum, F. A. 1977. "The art of artificial intelligence: *themes and* case studies in knowledgeengineering". pp.1014-1029. IJCAI-77. Los Altos: William Kaufmann.

Gero, J. S. (ed.). 1985a. *Knowledge Engineering in Computer-Aided Design*. Amsterdam: North-Holland.

Gero, J. S. (ed.). 1987a. Expert Systems in *Computer-Aided Design. Amsterdam:* North-Holland.

Gero, J. S. (ed). 1988a. Artificial *Intelligence in* Engineering: Design. Amsterdam: Elsevier / CMP.

Gero, J. S. (ed). 1988b. Artificial *Intelligence in* Engineering: Diagnosis and Learning.

Amsterdam: Elsevier / CMP.

Gero, J. S. (ed.). 1988c . Artificial *Intelligence in* Engineering: Robotics and Processes. Amsterdam: Elsevier / CMP.

Gero, J. S. 1989. 'Prototypes: a basis for knowledge-based design". pp.11-17. In 1. 5. Gem and T. Oksala (eds), *Knowledge-Based Systems* in Architecture. Helsinki: Acta Polytechnica Scandinavica.

Gero, J. S. and Maher, M. L. 1988. "A future role of knowledge-based systems in the design process". pp.81-90. In T. Mover and H. Wagter (eds). CAAD Futures '87. Amsterdam: *Elsevier*.

Gero, J. S. and Rosenman, M. A. (1989). "A conceptual framework for knowledge-based design *research at* Sydney University's Design Computing Unit", in J. S. Gero (ed.), Artificial *Intelligence in* Engineering: Design *11, CMP,* Southampton, pp.36l-380.

Kowalski, R. 1983 . Logic for Problem *Solving.* Amsterdam: Elsevier-North Holland.

Lansdown, J. 1982. Expert systems: their impact *on* the *construction industry.* London: RIBA Conference Fund.

Maher, M. L. (ed.) 1987. Expert Systems for Civil *Engineers: Technology* and Application. New York: American Soc. Civ. Eng.

Pham, D. T. (ed.) 1988. Expert Systems in *Engineering. Berlin:* ITS Publications / Springer-Verlag.

Rychener, M. *(ed.) 1988. Expert Systems for Engineering Design. New York: Academic Press.*

Sriram, D., *and Adey, It (eds.) 1986a. Applications of Artificial Intelligence in Engineering Problems-Volume 1. Berlin:* Springer-Verlag.

Sriram, D., and Adey, It (eds.) 1986b. Applications of Artificial Intelligence in Engineering Problems-Volume II, Berlin: Springer-Verlag.

Sriram, D., *and Adey,* R. *(eds.) 1987a. Artificial Intelligence in Engineering: Tools and Technique. Southampton: Computational* Mechanics Publications.

Sriram, D., *and Adey, R. (eds.) (1987b). Knowledge-Based Expert Systems in Engineering: Planning and Design. Southampton: Computational* Mechanics *Publications.*

Sriram, D., *and Adey,* R. *(eds.) (l987c). Knowledge-Based Expert Systems in Engineering: Classification, Education and Control. Southampton: Computational Mechanics Publications.*

Woolf, B. (1981). 'Intelligent tutoring *systems: a survey". pp.1-43. In H. E. Shrobe and AAAI (eds), Exploring Artificial Intelligence. San Mateo: Morgan* Kaufmann.