RESEARCH IN DESIGN COMPUTING: AN ARTIFICIAL INTELLIGENCE FRAMEWORK

John S Gero
Key Centre of Design Computing
Department of Architectural and Design Science
University of Sydney NSW 2006
Australia
email: john@arch.usyd.edu.au

1. Introduction

Design computing has often been considered a subset of computer applications that assist the designer in documenting and analysing complex designs. As one of many areas in which computer applications have been developed, design computing has relied on software developers and vendors to implement and market software with the relevant features and utilities to support some aspects of design activity. The development sin artificial intelligence and cognitive science over the last decades have provided new intellectual platforms to support research into design. In this paper we consider design computing as a research area which utilises these intellectual platforms, one in which the results of the research lead to more than additional computer programs and in fact lead to a better understanding of designing and computer support for designing.

Considering design computing as a research area, we identify three sets of goals:

(i) to develop theories, models and methods of designing as a process;
(ii) to use these theories, models and methods as the basis for the development of tools;
(iii) to use these theories, models and methods as the basis for teaching.

The first set of goals utilises both artificial intelligence and cognitive science as the providers of research methods to support design computing research. In order to achieve the first set of goals, it is sometimes useful to consider computational models of design as a way of simulating design processes. However, human designers can also provide the basis for developing theories, models, and methods of designing. The second set of goals looks at the implications of particular theories, models, and methods of designing when considering computer support or automation of specific design tasks. This set of goals has a more direct correlation with the majority of design computing research currently taking place at universities. The third set of goals brings this understanding of design processes to bear on how we teach design. Here again, the focus is not entirely on computer applications for design, but on the use of computational models and/or cognitive models of design to inform design teaching.

The Key Centre of Design Computing at the University of Sydney carries out teaching and research in the area of design computing. There are approximately 300 undergraduate architecture students, 60 graduate design computing students, 15-20 doctoral students, and 10 academic and research staff at the Key Centre. The artificial intelligence framework for design computing research presented here is based on research that has taken place at the Key Centre over the last 25 years.

Design computing research can be pursued using a variety of scientific methods, an artificial intelligence framework that we find to be both useful and distinctive is based on the following three research methodologies:

(i) empirically-based research (cognitive models);
(ii) axiom-based research (computational models); and
(ii) conjecture-based research (computational models).

Empirically-based research involves the development of experimental studies of designers that result in cognitive models of designing, which then form the basis of artificial intelligence models. Axiom-based research involves the identification of a set of axioms and their consequences to derive a logic-based computational model of designing. Conjecture-based research involves an analogy between a cognitive or computational process that leads to a computational model specific to designing. This paper briefly describes the characteristics of each of the three paradigms and gives examples of research projects at the Key Centre that illustrate the approach and preliminary results obtained through the different paradigms.

2. Empirically-Based Design Computing Research

Empirically-based research uses the experimental paradigm in which experiments are set up and then data is collected and analysed to produce a set of results. These results are then used as the basis of either the development of a hypothesis or the confirmation of a hypothesis about designing. These hypotheses then form the basis of an artificial intelligence model of designing. Typical approaches to empirically-based design computing research are: direct observation of the results of designing; surveys of designers' perceptions; and protocol studies of individual and collaborating designers designing. New protocol analysis methods have been developed and are being applied to produce novel results concerning the behaviour of designers as they are designing which has significance for the development of computational tools for designers.

Protocol analysis of designers

Protocol studies are a means of obtaining data from verbal utterances. Designers are asked to "think aloud" while they are designing. While they are designing they are video- and audio-taped. The designer's verbal utterances are transcribed. The transcription is then used, along with design theory, to develop a coding scheme. The transcription in then coded and finally analysed. An increasing number of possible analyses. In addition to “think aloud” protocols use is made of “retrospective” protocols where the designer does not talk during the design session but is videotaped. The designer is shown the videotape immediately after the session finishes and is asked to think aloud about what he or she was thinking during the designing process while the tape in running. This is then videotaped and used as the basis for the transcription, etc. The steps are listed below:

- taping
- transcription
- code development
- coding
- analysis

The results of such studies provides grounded insight into the behaviour of designers as they are designing. These insights form the basis of the development of computational support tools for designers.

An experimental study of designers

Designers were asked to carry out a specified design task and the "talk aloud" method was employed. Each designer was videotaped and a rich coding scheme was developed based on both design theory and the need to accommodate the data in the transcription. The development of the coding scheme is a crucial aspect of the protocol analysis method. The coding scheme developed here used five generic categories. The advantage of the use of categories is that they allow for an additional confirmation phase in the analysis since they exhibit an interdependence. The five categories developed were (Gero and McNeill, 1998):

- problem domain: abstraction level
- function-behaviour-structure
- analysis and evaluation
- synthesis micro-strategies
- design macro-strategies

Protocol analysis results

At a gross level a designer's time can be spent either on postulating solutions, called structure, or in reasoning about the function and behaviour of possible or postulated designs (Gero, 1990). Figure 1 shows a typical distribution of the time spent between these two large classes of activities by a designer. It is interesting to note that it is almost twenty minutes into the session, for this design, before any structure is proposed. As the design session proceeds the designer moves from spending time on function and behaviour to increasingly spending time on structure.

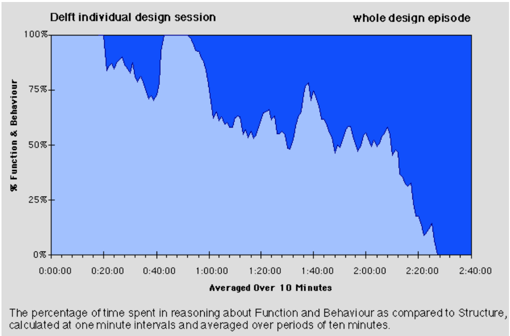

Figure 1. Typical plot of distribution of time spent on function and behaviour (light), as against structure (dark), for an experienced designer (Gero and McNeill, 1998).

Considerable detail about various aspects of designers' behaviour can be determined using the protocol analysis method. Figure 2 shows the spectrum of design event lengths across a typical design session. What is surprising is the very short duration of each design event. Without experiments with human designers such information would not become available.

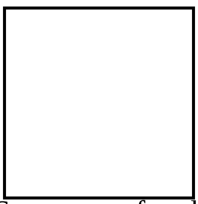

Figure 2. Spectrum of coding design event lengths (Gero and McNeill, 1998).

3. Axiom-Based Design Computing Research

Axiom-based research produces computational models of design through the identification of a set of axioms and the logical consequences of the axioms. This approach to design computing research involves:

(i) specifying relevant axioms
(ii) deriving logical consequences of the axioms
(iii) mapping the axioms and their consequences onto a particular domain to derive new results.

For example, an axiomatic logic-based shape representation allows for the uniform representation of shapes with or without curved boundaries, the consequences of which are representations of complex shapes that can be manipulated with logical implications (Damski and Gero, 1996). Consider the universe of discourse as the space defined in Figure 3. The axiom is that the space can be divided into two complementary spaces.

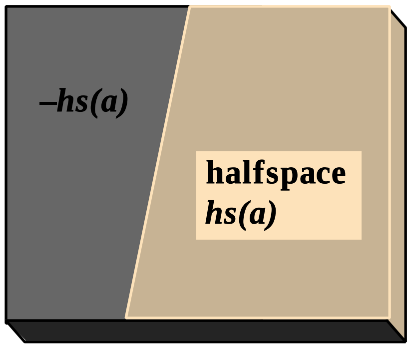

Figure 3. A space divided into two halfspaces, labelled hs(a) and -hs(a).

The following can be defined or inferred from the axiom:

a predicate hs(a) is defined for the halfspace a and -hs(a) for the halfspace a'

hs(a) is defined as True and -hs(a) as False

a volume V is the logical difference of hs(a1), hs(a2),..... hs(an)

a shape S is the logical addition of V1, V2, V3,..... , Vm.

Consider the painting in Figure 4 which shows a girl with a hat, along with a set of labelled halfplanes. The representation of such near arbitrary shapes is computationally extremely difficult if the designer wishes to reason further about them. The axiomatic approach described here can handle these shapes.

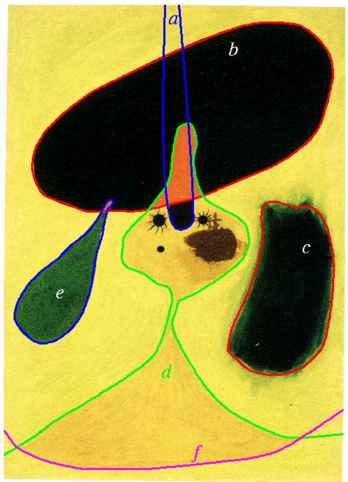

Figure 4. Miro's Girl in a Hat.

The girl's hat is defined by:

The girl's head and body is defined by:

$$\neg hp(c) \quad hp(d) \quad \neg hp(e) \quad hp(f)$$

From such representations we can carry out a variety of shape and topological computations even though the original shapes are difficult to represent numerically.

4. Conjecture-Based Design Computing Research

4.1 Conjectures Based on Analogies with Human Design Processes

The development of theories, models and methods of designing often relies on identifying an analogy with other processes. This research paradigm starts with a relevant computational process or cognitive model of design and develops a specific computational model of design. Some examples of computational models based on an analogy with cognitive models of design include: case-based design (design based on precedents; representation of cases including multimedia representations); design prototypes (knowledge chunking); graphical emergence (emergence of shapes, objects, semantics and style from drawings); design by analogy (between domain analogies in particular); and qualitative reasoning in design (qualitative representation and reasoning about shapes and spaces). The development computational models of designing need not rely entirely on cognitive models of designers, there is the potential to identify an analogy with other computational processes and apply them to a design domain. This type of research borrows heavily from computing fields such as artificial intelligence to produce specific computational models of design; for example: evolutionary systems (genetic engineering and co-evolution); and neural networks (emergence models).

Shape emergence

Emergence is the process of making properties, which were previously only implicit in a representation, explicit. In the visual domain it is a common human process (Gottschaldt, 1926; Granovskaya et al, 1987). Figure 5 clearly demonstrates the phenomenon. If the right-hand figure is drawn using a CAD system, its representation will be that of six objects located in geometrical space. However, for humans the dominant features are the central star and triangles. None of the features seen by the human observer can be "seen", ie, are represented by the CAD system.

Figure 5. A single object and a composite object, made of 12 copies of the single object, which exhibits strongly emergent shapes.

From the work of the Gestalt psychologists and more recently that of the cognitive psychologists, it is possible to construct computational models of shape emergence based on concepts drawn from their research.

Humans appear to distinguish foreground from background in their reading of shapes. In order to emerge shapes which were not previously represented a process which manipulates the foreground and background can be constructed. What is done is to take the primary or originally represented shape and "unstructure" it so that it now becomes part of the background, producing an image composed of unstructured shapes only. A structuring process is then passed over this background to emerge foregrounds which may include both the primary shape and newly represented shapes. Gero and Yan (1993) have developed such a process based on a new representation, infinite maximal lines, along with a structuring process.

The concepts behind shape emergence can be extended to emerge shape semantics, where the shape semantics are derived from visual patterns of shapes. Since these patterns were not originally represented they are emergent when there is a computational process which can find and represent them. From seeing drawings, various visual patterns are perceived by the human viewers. Designers can find different visual patterns from what was intended to be drawn. The newly discovered visual patterns may play a crucial role in developing further ideas in the same design if the designer is willing to adapt the visual pattern which was not there at the moment of drawing. Regardless of adaptability, visual patterns from shapes are defined as shape semantics when the patterns match the criteria for predefined labels, such as visual symmetry, visual rhythm, visual movement and visual balance. Figure 6 shows the Temple of Thebes which exhibits emergent visual movement.

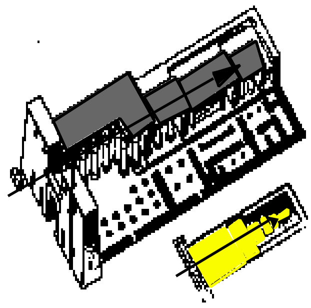

Figure 6. The Temple of Thebes which exhibits emergent visual movement.

Gero and Jun (1998) have developed a computational model of shape semantic emergence which is based on three computational processes:

object correspondence
grouping
shape semantics emergence.

Shape semantics play an important role in organising decisions, providing order, and generating final form in visually-oriented design. They appear to have a special role in architectural design in particular. Architecture reflects its main design concept through visual organization of structures. Visual organization of structures is shown as visual semantics of the design and is perceivable to designers. However, current computer-aided drawing, computer-aided drafting and computer-aided design systems prevent the discovery of visual shape semantics. Inadvertently such systems have enforced fixation so that it is not surprising that they are not used in the early stages of architectural design.

4.2 Conjectures Based on Analogies with Other Processes

There are many computational processes in artificial intelligence which do not derive their ideas from what humans do when they design. These include neural networks, evolutionary systems, and various object representation schema.

Genetic engineering

Genetic engineering can be used to form design concepts computationally. The practice of genetic engineering in natural organisms involves locating genetic structures which are the likely cause of specified behaviours in the organism. This provides a direct analog with concept formation. The behaviour of the organism is an observable regularity which maps onto a concept and the structure of the genetic material which causes that behaviour is a representation of that concept, albeit a representation which has to be expressed for the concept to appear. The practice of genetic engineering is akin to reverse engineering.

Consider Figure 7 where the population of designs is divided into two groups (it could be more). One group exhibits a specific regularity whilst the other does not. The goal is to locate a common structure in the genotypes of those designs which exhibit this regularity. Genetic engineering at this symbolic level uses pattern matching and sequence analysis techniques to locate these genetic structures. Of particular interest in this form of concept formation is the separation of position-dependent structures from position-independent structures. The implication of the former is that the concept depends on either other concepts or a "situation" for it to apply, whilst in the latter case the concept is independent of any situation.

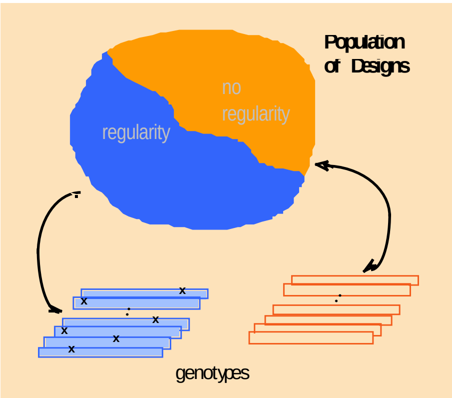

Figure 7. Genetic engineering is concerned with locating groups of genes' regularity, marked as X in the genotypes of those design which exhibit a specific behavioural regularity.

Take as an example the 8 genes shown in Figure 8 represented in the form of state transition rules. These genes are used to form the genotypes of designs within which a regularity is sought.

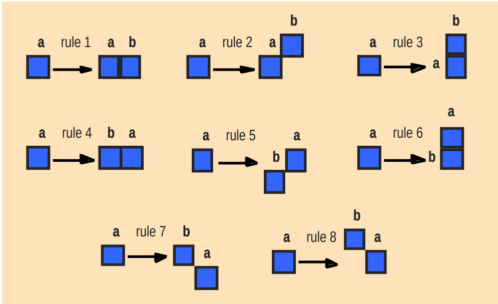

Figure 8. A set of 8 genes in the form of shape transition rules [Gero and Kazakov, 1996].

Figure 9 shows 10 designs produced from those genes. Each design is searched to determine some common regularity.

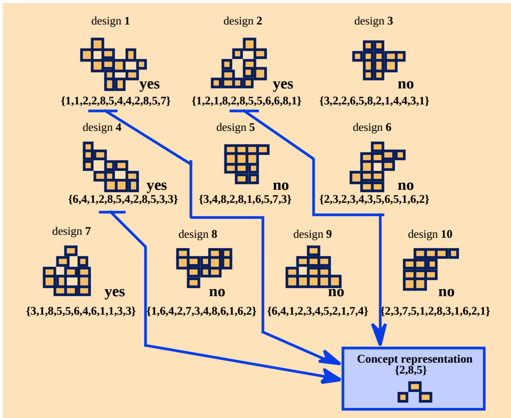

Figure 9. A set of 10 designs produced with the genes in Figure 3 and evaluated according to their regularity. Genetic engineering techniques emerge the gene group {2, 8, 5} as being the likely cause of that regularity, after [Gero and Kazakov, 1996].

From Figure 9 can be seen that a concept has been found. There is no semantic label for that concept since such labels need to be grounded in human experience, but there is a symbolic representation and its graphical interpretation, which is appropriate for this context.

5. Conclusions on AI-Based Design Computing Research

This paper has described an artificial intelligence framework within which design computing research is carried out. The three paradigms which have proven to be most useful are:

(i) empirically-based research (cognitive models);
(ii) axiom-based research (computational models); and
(iii) conjecture-based research (computational models).

A number of research projects from the Key Centre of Design Computing, University of Sydney, have been outlined as vehicles for each of these paradigms. Each of the projects uses one of the paradigms listed. The conduct of research for each of the projects is different and in some cases quite different. Empirically-based design computing research looks like experimental cognitive science research. Axiom-based design computing research looks like mathematical/logic research. Conjecture-based design computing research looks like theoretical engineering research. Thus, design computing research spans a range of research paradigms. What both the projects and the framework of paradigms imply is that design computing research has now reached a level of maturity that allows it to operate as a sub-discipline of design science rather than as simply a means of producing software packages. In this it contributes directly to the three goals enunciated in the Introduction. It is one of the primary means of developing theories, models and methods of designing as a process. It uses these as a basis for the development of design tools, and is beginning to use the theories, models and methods as a basis for teaching (although this has not been presented in this paper).

What directions are open for design computing research using artificial intelliegnce and cognitive science approaches? Not so much what projects should be pursued rather what strategic directions may yield results which inform us about designing and produce processes of value. As empirically-based research produces more results, we should have a greater understanding or how human designers design. Such knowledge will have implications for both how information technology can be interfaced with human designers and, perhaps more importantly, provide new conjectures for design computing research to explore in order to provide the foundation for more useful tools for designers. Similarly, as the other approaches yield insights into designing they may provide the foundation for novel tools.

Acknowledgments

The projects described here have been supported by grants from the Australian Research Council. Computational support has been provided by the Key Centre of Design Computing.

References

Damski, J. C. and Gero, J. S. (1996) A logic-based framework for shape representation, *Computer-Aided Design* **28**(3):169-181.

Gero, J. S., (1990). Design prototypes: A knowledge representation schema for design, *AI Magazine* **11**(4): 26-36.

Gero, J. S. and Jun, H. (1998) Emergence of shape semantics of architectural shapes, *Environment and Planning B: Planning and Design* (to appear).

Gero, J. S. and Kazakov, V. (1996). Evolving building blocks for design using genetic engineering: a formal approach, *in* J. S. Gero (ed.), *Advances in Formal Design Methods for CAD*, Chapman and Hall, London, pp.31-50.

Gero, J. S. and McNeill, T. (1998). An approach to the analysis of design protocols, *Design Studies* **19**(1): 21-61.

Gero, J. S. and Yan, M. (1993). Discovering emergent shapes using a data-driven symbolic model, *in* U. Flemming and S. Van Wyk (eds), *CAAD Futures'93*, North-Holland, Amsterdam, pp. 3-17.

Gottschaldt, K. (1926). Uber den Einfluss der Erfahrung auf die Wahrnehmung von Figuren. *Psychologische Forschung* **8**: 261-317.

Granovskaya, R. M., Bereznaya, I. Y. and Grigorieva, A. N. (1987). *Perception of Forms and Forms of Perception*, Lawrence Erlbaum, Hillsdale, New Jersey.

This is a copy of the paper: Gero, J. S. (1998) Research in design computing: an artificial intelligence framework, , *in* X. Huang, S. Yang and H. Wu (eds), *International Conference on Artificial Intelligence for Engineering,* HUST Press, Wuhan, China, pp.5-12.