

COMPUTATIONAL MODELS OF CREATIVE DESIGN PROCESSES*

JOHN S. GERO

University of Sydney

1. Introduction

Artificial intelligence has provided a rich set of concepts for design researchers to work with. It has extended and considerably elaborated ideas from the design methods movement and from systems theory. There are three views that can be taken about artificial intelligence in design. The first is that it provides a framework in which to explore ideas about design; the second is that it provides a schema to model human designing; and the third is that it has the means to allow the development of tools for human designers. An extension of the third view is that these tools point to ways to automate certain processes in design.

One of the fundamental concepts of artificial intelligence which finds favour with design researchers is the externalisation of knowledge. This has led to the two major research concerns:

representation of design knowledge; and
processes for designing

2. Representation of design knowledge

The three main symbolic knowledge representation approaches of rules, semantic networks and frames were soon augmented by a sub-symbolic approach based on neural networks. These four still remain the base knowledge representation approaches. However, meta-symbolic or conceptual approaches which could be implemented in these base approaches soon began to be developed. These approaches aimed to represent important conceptual relationships which made up design knowledge (Coyne et al., 1990).

Orthogonal to these issues is that of compiled or generalised design knowledge versus case or episodic design knowledge. In compiled knowledge a set of design cases is generalised so that an abstraction of their characteristics is produced. The generalised knowledge can be instantiated for use in producing designs which are similar to the original set of designs. Episodic design knowledge keeps individual design cases separately without generalising them.

* This chapter is based in part on a presentation made to EuropIA'91.

3. Processes for designing

Artificial intelligence brought with it a large number of processes applicable to problem solving tasks. If design could be cast as some sort of problem solving task then it could use the array of available processes. Most prominent amongst these processes are rule chaining, constraint-based processing and planning. Each has been used by design researchers to model design processes (Coyne et al., 1990).

Rule chaining forms the basis of many expert systems. The expert systems technology readily lends itself to a variety of design related tasks and continues to be explored. However, there is a recognition that the synthesis activity in design is more akin to abduction in logic than it is to deduction whereas rule chaining is more akin to deduction. Backward chaining, it is suggested, matches abduction. This is likely to be so only if the knowledge encoded in the rules is represented abductively.

Constraint-based processes lend themselves readily as design processes since designing involves formulating requirements. These requirements may be treated as either objectives or goals which point to the directions of improving performance or behaviour or as constraints which restrict the values of the variables. However, the underlying assumptions behind most of the processes are not necessarily in accord with those in design. For example, constraint-based systems assume a fixed world in which to operate whereas part of designing involves changing the world.

Planning is an attractive concept drawn from artificial intelligence and is related to constraint propagation. Its success relies on the existence of fixed but divisible goals. Although each of these processes may be used within designing, none of them characterises design itself as a process. Design is seen as being more complex than any or all of these.

4. Routine and non-routine design

It is convenient to characterise design as routine or non-routine. *Routine design*, in computational terms, can be defined as that design activity which occurs when all the necessary knowledge is available. It may be expressed as being that design activity which occurs when all the knowledge about the variables, objectives expressed in terms of those variables, constraints expressed in terms of those variables and the processes needed to find values for those variables, are all known *a priori*. In addition, routine design operates within a context which constrains the available ranges of the values for the variables through good design practice. None of this is to imply that routine design is not complex or is even easy.

Non-routine design can be subdivided into two further groups: innovative design and creative design. *Innovative design*, in computational terms, can be defined as that design activity which occurs when the context which constrains the available ranges of the values for the variables is jettisoned so that unexpected values become possible. This produces two effects, one for the design process and the other for the product or artefact. In terms of the design process, variable values outside the usual

ranges have the potential to introduce unexpected as well as unintended behaviours which can only be brought into formal existence if additional knowledge capable of describing them can be introduced. For example, in designing a structural beam to carry a load across a gap there are standard depth-to-span ratios for different materials. If the depth of the beam is made much larger than these then there is the likelihood that the beam will buckle. However, if no buckling knowledge is applied to its design (and buckling is not normally considered in the design of such beams) then no buckling will be found. In terms of the artefact, innovative design processes produce designs which recognisably belong to the same class as their routine progenitors but are also ‘new’.

Creative design, in computational terms, can be defined as that design activity which occurs when a new variable is introduced into the design. Processes which carry out this introduction are called ‘creative design processes’. Such processes do not guarantee that the artefact is judged to be creative, rather these processes have the potential to aid in the design of creative artefacts.

The remainder of this chapter is concerned with creative design processes. Section 5 addresses a schema for the representation of design knowledge which provides a framework for what follows. Section 6 describes four creative design processes. Section 7 concludes the chapter with a brief discussion on the implications of these processes.

5. Design prototypes: A schema for design knowledge representation

5.1. OUTLINE

A *design prototype* (Gero, 1990) is a conceptual schema for representing a class of a generalized grouping of elements, derived from like design cases, which provides the basis for the commencement and continuation of a design. Design prototypes do this by bringing together in one schema all the requisite knowledge appropriate to that design situation.

A designed artefact may be broadly interpreted in terms of the three variable groups of function, structure and behaviour. The level of specificity in each of these depends on the granularity and level of abstraction being represented. Thus, at an early stage of designing an appropriate design prototype may contain primarily function and behaviour with little information on structure. Whilst at a later time an appropriate design prototype will contain considerable detail in the structure group. A design prototype brings together these three groups and the relations between them which includes processes for selecting and obtaining values for variables. Design prototypes draw from such sources as prototype theory (Osherson and Smith, 1981) and scripts (Schank and Abelson, 1975). Prototype theory construes membership of a concept to be determined by its similarity to that concept’s best exemplar. Design prototypes use the notions of generalization to produce the prototype. Although closely related to scripts, design prototypes include semantics and are not time sequence

bound.

Although it is well recognized that there is no function in structure and vice-versa that there is no structure in function, human design experience produces a connection between function and structure. Once that connection is learned it is very difficult to unlearn. Once the connection between behaviour and structure is made and the connection between behaviour and function is made it forms the basis of much of a designer's knowledge. It is function, structure, behaviour and the relationships between them which form the foundation of the knowledge which must be represented in order for specific design processes to be able to operate on them. In natural discourse the distinction between function and structure sometimes becomes blurred to the extent that the label of the structure takes on the meaning of the function. For example, the label of a particular copier 'Xerox' is slowly taking on the meaning of its function, i.e. to copy. However, if reasoning is to occur in transforming function to structure then a clear separation must be made between them and between function, structure and behaviour.

5.2. STRUCTURE OF DESIGN PROTOTYPES

A design prototype separates function (F), structure (S), expected behaviour (B_e) and the structure's actual behaviour (B_s). It also stores relational knowledge between them (K_r) as well as qualitative knowledge (K_q), computational knowledge (K_c) and context knowledge (K_{ct}).

Relational knowledge provides and makes explicit the dependencies between the variables in the function, structure, behaviour categories and can take the form of a dependency network. Relational knowledge identifies the relevant variables in going from function to behaviour, from behaviour to structure and in the inverse direction. Relational knowledge allows for the specialization of the information in a design prototype to a specific design situation.

Qualitative knowledge (a subset of qualitative reasoning) is an adjunct to relational knowledge and provides information on the effects of modifying values of structure variables on behaviour and function. Included here are the normal ranges of values of variables found in the generalization. Qualitative knowledge can be used to guide any decision making process.

Computational knowledge is the quantitative counterpart of qualitative knowledge and specifies symbolic or mathematical relationships amongst the variables. Computational knowledge is used to determine values of variables.

Constraints appear in both the qualitative knowledge and computational knowledge. Constraints on function appear as expected behaviours, constraints on structure reduce the range of possibilities.

Context knowledge identifies the exogenous variables for a design situation and specifies that values for these variables must come from outside the design prototype, i.e. from the context (C).

In addition there is knowledge concerning the design prototype itself (K_p). This

comprises the *typology* (t) of the design prototype which identifies the broad class of which the design prototype is a member, while *partitions* (p) represent the subdivisions of the concept represented by the prototype. Partitioning a design prototype supports viewing it from many perspectives. Once the partition or combination of partitions is selected only information pertaining to these partitions will be made available. In this sense, partitioning of design prototypes ultimately reduces the space of potential designs. Finally, the structure has to be represented in a canonical form standard for each domain. This representation is the *design description* (D). Often the representation is in the form of drawings.

A design prototype, P , may be represented symbolically as

$$P = (F, B, S, D, K, C)$$

where	B	=	behaviour (B_s, B_e)
	B_e	=	expected behaviour
	B_s	=	structure's actual behaviour
	C	=	context
	D	=	design description
	F	=	function
	K	=	knowledge ($K_r, K_q, K_c, K_{ct}, K_p$)
	K_c	=	computational knowledge
	K_{ct}	=	context knowledge
	K_p	=	prototype knowledge (t, p)
	K_q	=	qualitative knowledge
	K_r	=	relational knowledge
	p	=	partition
	S	=	structure
	t	=	typology

In summary, a design prototype brings together all the requisite knowledge appropriate to a specific design situation. Although the contents of a design prototype are developed by individual designers, like-minded designers will tend to agree on its general contents. Thus, a design prototype concerned with initial design of a house is likely to include such notions as style, location on site, orientation, existence of spaces based on their functional activities, building planning codes, and so on. A designer will draw on many design prototypes during the course of developing any design.

Whether a design prototype is used as a formal schema for compiled or generalised knowledge or whether it is used as a conceptual schema, its utility lies in the framework it provides. This framework is equally useful to structure design cases or episodes.

6. Creative design processes

Five creative design processes with their computational analogs will be discussed. These five are:

1. combination
2. mutation
3. analogy
4. first principles
5. emergence

6.1. COMBINATION

Combination as a creative design process involves the addition of two design prototypes or some subset of them (Figure 1). It can occur at the function, behaviour or structure level, i.e.

$$\begin{aligned} F_{new} &= F_{existing1} \cap F_{existing2} \\ B_{new} &= B_{existing1} \cap B_{existing2} \\ S_{new} &= S_{existing1} \cap S_{existing2} \end{aligned}$$

Fig. 1. A graphical example of design using combination (after Rosenman and Gero, 1992)

However, the implications of each of these is different. Combining functions alone does not necessarily imply that new behaviours have to be produced since the new set of functions might be achieved with the existing behaviours. Although, in general, new behaviours will be expected to produce the new functions. Thus,

$$F_{new} \rightarrow \begin{cases} B_{existing} \\ B_{new} \end{cases}$$

where \rightarrow = implies. Similarly, combining behaviours alone also does not necessarily imply either new structures or new functions although generally it will. Thus,

$$B_{new} \rightarrow \begin{cases} S_{existing} \\ S_{new} \end{cases}$$

$$B_{new} \rightarrow \begin{cases} F_{existing} \\ F_{new} \end{cases}$$

Similarly with structure. Thus,

$$S_{new} \rightarrow \begin{cases} B_{existing} \\ B_{new} \end{cases}$$

Normally, structures are combined to produce a new structure since the structure is the basis of the artefact. More precisely, the structure is described by a set of structure variables, SV , which describe the elements of the structure and their relationships. Thus, if S_1 is represented by a set of structure variables $\{SV_{11}, SV_{12}, \dots, SV_{1n}\}$ and S_2 is represented by $\{SV_{21}, SV_{22}, \dots, SV_{2m}\}$, combination of S_1 and S_2 could occur when either some of SV_{1i} are substituted by SV_{2j} or are augmented by them.

For example

$$SV_{new} = \{SV_{11}, SV_{12}, \dots, SV_{2j}, SV_{2k}, \dots, SV_{1n}\}$$

If there is only structure variable substitution then this matches the notions of crossover in genetic algorithms (Goldberg, 1989). Using the genetic metaphor from natural biology, structure is divided into two levels: genomes or genotypes which are composed of genes, and phenotypes which are the expressions of those genes as an artefact. The genomes constitute a recipe for the production of the artefact. It is possible to combine the genes of two different genotypes to produce new phenotypes which potentially improve the behaviours of the phenotypes represented by the combined genotype. The representation of the ‘design genes’ is a research topic. Current work at the University of Sydney makes use of shape transformation rules and shape grammars (Stiny, 1980). Resultant forms (phenotypes) are produced by an ordered execution of the shape rules. One approach is to code the potential order of the rules as the genes and to evolve an order which, when applied over the rules, improves the resulting behaviours of the phenotype.

The effect of combination is the introduction of a new variable into the original structure and hence combination meets the formal definition of being a creative design process.

6.2. MUTATION

Mutation is the alteration of a structure variable by an external agent (Figure 2). It draws on the genetic metaphor also in that it is the genes that are mutated not the phenotype. In design the genes that produce the structure as artefact are the most interesting to mutate.

Mutation can be modelled as

$$\begin{aligned} S_{new} &= \varphi_m(S_{existing}) \\ \text{or } SV_{new} &= \varphi_m(SV_{existing}) \end{aligned}$$

Fig. 2. A graphical example of design using mutation (after Rosenman and Gero, 1992)

where φ_m = a mutation operation.

Of interest in creative design is the use of mutation to produce new variables (Jo and Gero, 1991). Typical mutation operations include the algebraic and set theoretic operators. Thus, division, for example, divides a single variable into two. Such operations can affect the resultant topology of the artefact. Mutation operators fall into two classes: homogeneous and heterogeneous. Homogeneous operators are those that produce new variables of the same class as the variable being mutated. For example, a length is mutated into two lengths. Heterogeneous operators are those that produce new variables of a different class to the variable being mutated. For example, a length is mutated into a length and an angle. Heterogeneous mutations require additional knowledge to incorporate them into the existing design prototype (Gero and Maher, 1991). Since mutation produces new structure variables it meets the formal definition of being a creative design process. Mutation need not be limited to structure variables, however. It is possible to conceive of mutation being applied to behaviour variables also to produce new behaviours which can be interpreted as new functions. This also meets the definition of a creative design process. In natural evolution the introduction of a new behaviour is a significant event.

6.3. ANALOGY

Analogy is defined as the product of processes in which specific coherent aspects of the conceptual structure of one problem or domain are matched with and transferred to another problem or domain (Figure 3). Based on the nature of the knowledge transferred to the new problem, analogical reasoning processes can be placed into one of two classes: transformational analogy and derivational analogy (Carbonell, 1983, 1986).

Transformational analogy adapts the structure of a past solution to the new problem. Derivational analogy applies the successful problem solving process to the pro-

Fig. 3. A graphical example of design using analogy (after Rosenman and Gero, 1992)

cess of producing a solution of the new problem.

Analogies can operate on the function, behaviour or structure of knowledge. Analogy requires a target and a source. Most analogies are drawn between situations in the same domain although interesting analogies can be drawn between situations in different domains (Qian and Gero, 1992).

Thus,

$$\begin{aligned} B_{target} &= \tau_a(B_{source}) \\ F_{target} &= \tau_a(F_{source}) \\ K_{target} &= \tau_a(K_{source}) \\ S_{target} &= \tau_a(S_{source}) \end{aligned}$$

where τ_a = an analogical operation.

As with combination as a creative design process the implications of each of these is different. The effects of a new F_{target} , B_{target} or S_{target} are the same as for combination. The effect of a new K_{target} can be described as

$$K_{target} \rightarrow \begin{cases} F_{existing} \\ F_{new} \end{cases}$$

$$K_{new} \rightarrow \begin{cases} B_{existing} \\ B_{new} \end{cases}$$

$$K_{new} \rightarrow \begin{cases} S_{existing} \\ S_{new} \end{cases}$$

Thus, new target variables do not guarantee novel target results although it is most likely. The effect of analogy on structure is the introduction of a new variable into the original structure. The effect of analogy on function, behaviour and knowledge

may be the introduction of a new variable into the original structure. Hence analogy meets the formal definition of being a creative design process. Computational models of analogy are well developed, particularly for transformational analogy and these have been used in design (Qian and Gero, 1992).

6.4. FIRST PRINCIPLES

First principles relies on causal, qualitative or computational knowledge used abductively to relate function to behaviour and behaviour to structure without the use of compiled knowledge (Figure 4) (Cagan and Agogino, 1987).

Fig. 4. A graphical example of design using first principles (after Rosenman and Gero, 1992)

Thus, first principles can be modelled as

$$S = \tau_k(B)$$

where τ_k = abductive knowledge-based transformation.

Design using first principles is the computational process that is the least developed because of the difficulty in relating behaviour to structure without the use of compiled knowledge. However, the division of the problem into basic independent behaviours is the crux of the idea so that the compiled knowledge can be utilised at the level of indivisible behaviours. Since the use of first principles introduces new variables it meets the formal definition of a creative design process. Computational models of first principles processes generally rely on the use of qualitative physics and have been used in design (Williams, 1991).

6.5. EMERGENCE

Emergence is the process whereby extensional properties of a structure are recognised beyond its intentional ones (Mitchell, 1992), (Figures 5 and 6).

Emergence of structure can be modelled as

$$S_e = \tau_e(S)$$

where S = intentional structure
 S_e = emergent structure
 τ_e = emergence transformation by substitution

Fig. 5. Triangles 1, 2 and 3 are drawn (intentional shapes)

Fig. 6. Some emergent shapes (extensional shapes) inferred from the original intentional shapes in Figure 5.

Design using emergence is computationally still being researched. Since emergence is often observed in the behaviour of designers it plays a potentially important role in design. It can be characterised as replacing the representation of a structure by another representation—that is, deleting one or many structure variables and replacing them with others. Since the use of emergence introduces new variables it meets the formal definition of a creative design process.

The computational models of emergence are only now being developed and rely on separating the representation of geometry and topology from a symbolic representation from which geometry can be inferred. The effect of this is to break the nexus between intentional shapes and a fixed representation from which only those shapes can be found.

7. Discussion

Artificial intelligence, which is a particular paradigmatic way of examining and articulating knowledge, provides a rich context in which to explore design as a set of processes with their computational analogs. The view taken is that design needs to be understood prior to the development of design support tools. Whilst the artificial intelligence paradigm cannot claim to provide an ‘answer’ to what design is, it does provide one framework for expeditious exploration particularly of creative design processes.

Most of the effort to date has been concerned with routine design since that is both better understood than non-routine and covers the majority of design in practice. However, concepts from artificial intelligence can be used to articulate non-routine design and in particular creative design and some of its related processes. This is still very much a research area so no claims can be made about progress at the tool making level. However, the understanding of creative design characterised in the manner described in this chapter is advanced through the framework outlined.

References

- Cagan, J. and Agogino, A. M.: 1987, Innovative design of mechanical structures from first principles, *AI EDAM: Artificial Intelligence in Engineering Design, Analysis and Manufacturing*, **1** (3): 169–189.
- Carbonell, J. G.: 1983, Learning by analogy: formulating and generalising plans from past experience, in Michalski, R. S., Carbonell, J. G. and Mitchell, T. M. (eds), *Machine Learning: An Artificial Intelligence Approach*, Tioga, Palo Alto, California, pp. 137–161.
- Carbonell, J. G.: 1986, Derivational analogy: a theory of reconstructive problem solving and expertise acquisition, in Michalski, R. S., Carbonell, J. G. and Mitchell, T. M. (eds), *Machine Learning II: An Artificial Intelligence Approach*, Morgan Kaufmann, Los Altos, California, pp. 371–392.
- 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.: 1990, Design prototypes: a knowledge representation schema for design, *AI Magazine*, **11**(4): 26–36.
- Gero, J. S. and Maher, M. L.: 1991, Mutation and analogy to support creativity in computer-aided design, *Proceedings CAAD Futures '91*, ETH, Zurich, pp. 241–249.
- Goldberg, D. E.: 1989, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading.
- Jo, J. and Gero, J. S.: 1991, Design mutation as a computational process, in Woodbury, G. (ed.), *The Technology of Design*, ANZAScA, University of Adelaide, Adelaide, pp. 135–143.
- Mitchell, W. J.: 1992, A computational view of design creativity, in Gero, J. S. and Maher, M. L. (eds), *Modeling Creativity and Knowledge-Based Creative Design*, Lawrence Erlbaum, Hillsdale, New Jersey, pp. 25–44.
- Osherson, D. N. and Smith, E. E.: 1981, On the adequacy of prototype theory as a theory of concepts, *Cognition*, **9**(1): 35–58.
- Qian, L. and Gero, J. S.: 1992, A design support system using analogy, in Gero, J. S. (ed.), *Artificial Intelligence in Design '92*, Kluwer, Dordrecht, pp. 795–813.
- Rosenman, M. A. and Gero, J. S.: 1992, Creativity in design using a prototype approach, in Gero, J. S. and Maher, M. L. (eds), *Modeling Creativity and Knowledge-Based Creative Design*, Lawrence Erlbaum, Hillsdale, New Jersey (to appear).

- Schank, R. C. and Abelson, R.: 1975, Scripts, plans and knowledge, *Proceedings of the Fourth International Joint Conference on Artificial Intelligence*, American Association for Artificial Intelligence, Menlo Park, California, pp. 151–157.
- Stiny, G.: 1980, Introduction to shape and shape grammars, *Environment and Planning B*, **9**: 359-367.
- Williams, B. C.: 1991, Interaction-based design: constructing novel devices from first principles, in Brown, D. C., Waldron, M. and Yoshikawa, H. (eds), *IntCAD'91 Preprints*, Ohio State University, Columbus, pp. 247–268.