Expert Systems in Computer-Aided Design
John Gero (editor)
Elsevier Science Publishers B.V. (North-Holland)

INNOVATION AND CREATIVITY IN KNOWLEDGE-BASED CAD

Richard D. Coyne, Michael A. Rosenman, Antony D. Radford and John S. Gero

Architectural Computing Unit
Department of Architectural Science
The University of Sydney
Sydney, 2006, Australia

We examine the creativity of knowledge-based design systems from a narrow information processing perspective. As a property of the design process innovation and creativity can be identified by observing both the quality of the product, and also the characteristics of the process itself. The key theme running through our discussion is the acquisition of knowledge as the key to understanding creativity. This involves not only the ability of a system to acquire knowledge, but also its ability to control its own processes and change its own structure. In order to discuss this view a model of design systems is put forward in which we distinguish between interpretative and syntactic subsystems. Examples of automated design systems serve as vehicles for exploring the potential of knowledge-based systems for innovation and creativity.

1. INTRODUCTION

Knowledge-based CAD systems exhibit the potential to accomplish a wide variety of tasks in the service of the design professions. The question is often raised: do CAD systems ever contribute substantially to the creative process? Indeed, can CAD systems ever *be* innovative or creative? It is our intention within this paper to explore the issues of innovation and creativity with reference to recent work carried out in the Architectural Computing Unit. We intend to discuss innovation and creativity in information processing terms, to demonstrate what is currently achievable, and to suggest the processes which a system needs to be able to perform to earn the designations *innovative* or *creative.*

Various attempts have been made to characterize innovative and creative thinking. A common view is that creative people are those who are able to overcome the 'psychological inertia' which encumbers the thinking of less creative people (Altshuller, 1984). Creative thinkers are able to break out of 'conventional' modes of thinking. This view has been popularized by deBono (1967) in his use of the term 'lateral thinking'. Various views on the subject of creativity are provided by Vernon (1970), Storr (1972) and Koestler (1976). Although these writings are based on theories of psychology, they do not generally provide models which are sufficiently formal to be directly applicable to CAD (Lansdown, 1985).

There are several ways in which commonly understood ideas about creativity and innovation can be incorporated into models of the design process. Two views are presented in this paper. The first is that innovation and creativity can be discussed qualitatively, pertaining to the product and the process of design. The second is a knowledge-based view in which innovation and creativity are seen as different aspects of the design process. It will be argued that this latter model fits comfortably within a knowledge-based model of design.

We wish to adopt a distinction between innovation and creativity that is becoming common currency within design and linguistic theory. Design is concerned with search within spaces of possible designs. We adopt the view that *innovation* can be characterized as a process of exploration within a defined space. *Creativity*, on the other hand, is concerned with exploration within a space that is only partially defined. In the case of innovation the space being explored can be infinite in extent, such as that defined by a linguistic grammar. The generation of a sentence in a language can be regarded as *innovative*. In the case of creativity we are concerned with the search for the 'paraphernalia' that defines that space. In the case of a natural language (considering only its syntax) we could say that the 'paraphernalia' defining the space of all possible discourses is its grammar. To produce a discourse which also involved the production of a new grammar would be regarded as *creative*. This distinction may not appear immediately relevant to natural language, but it is an important concern in design.

We generalize the idea of creativity in design by contending that the 'paraphernalia' which defines search spaces in design is *knowledge*. As well as the search for designs, creativity therefore involves the search for the knowledge defining the space within which that search should take place. This view appears to capture something of the richness, the complexity and even the intractibility of design as a process.

In exploring innovation and creativity in design we pursue a linguistic model of design concerned with spaces of designs delineated by semantic and syntactic subsystems. This enables us to identify various reasoning processes applicable to design systems. We discuss, with reference to examples, how each of these processes can be said to contribute to the innovative and creative character of a design system.

The view of innovation and creativity we are adopting is only one of several. Prior to a discussion demonstrating its utility we discuss an alternative view which contributes to an understanding of innovation and creativity in design.

2. A QUALITATIVE MODEL OF CREATIVITY

In this discussion we will concentrate on the view that creativity is a qualitative term. It subsumes the idea of innovation. In this and subsequent discussion we will restrict ourselves to the context of the *design* process: that is, the production of descriptions of artifacts – artifacts which serve to modify our environment in some way, are manifested in space, and are to be manufactured, or constructed.

The term 'creativity' may be discussed in relation to both *product* and *process*. In the former case we mean: the artifact is the *product* of a process the creativity of which is assessed by the quality of the product. In the latter case we mean that a process is designated as creative irrespective of the quality of the product. Both of these approaches will be considered here.

2.1 Product

There are three ways in which an artifact can be said to be the *product* of a creative process: in relation to the degree to which it is innovative, in relation to its value, and according to the richness of interpretations that can be placed upon it.

2.1.1 Innovation

We may say that an artifact is the product of a designer (or a design system) who is particularly creative if the artifact is different in some way to the products of other designers. We may qualify this by saying that the designer is creative if something is produced that never existed before. If we have a way of measuring the originality of artifacts then we are able to compare one with another, and determine which designer is more creative than the rest. In this view creativity is dependent on *temporal context*. To invent the steam engine in 1987 would not be regarded as very creative, nor for that matter, would the re-discovery of the general theory of relativity – even if it were derived by someone whose mathematical training did not extend beyond high school level.

A novel artifact may also serve as a paradigm – effectively serving to define a class of things. Hence, certain buildings throughout the history of architecture are regarded as products of highly creative minds in that they embody the essence of a particular school or style of design, or a building type.

2.1.2 Value

Another view of creativity is that it is related to the *value* of an artifact in some way. We may describe a designer as creative if something is produced which is generally regarded as of high value – by whatever criteria we may wish to employ: for example, in terms of beauty, simplicity, sophistication, suitability for purpose or marketability.

2.1.3 Richness of Interpretation

As well as exhibiting qualities of high value an artifact may also be *multi-valued*. Certain artistic endeavours thrive on ambiguity (Lansdown, 1985; Cohen et al, 1985). An artifact may be regarded as the product of a creative process if it can be interpreted in many different ways: 'there is always something new to discover!' Within the realm of architectural design this may be manifested as the ability of a building to assume many roles such that it can serve several purposes, including those for which it was not originally intended. Richness can also be evident in terms of the social and other messages that it may convey – in the sense in which a building fits within a semiotic system (Broadbent, 1977).

2.2 Process

A further explanation of creativity is to see it as an attribute of an information process. This aspect of creativity will be considered under the headings of entropy, efficiency and richness.

2.2.1 Entropy

A design system may be regarded as creative if it produces a complex artifact description from information which contains the 'seeds' of the design to only a very small degree at the outset, and if the initial state exhibits a low level of information content. Hence, the 're-invention' of the steam engine by someone with no knowledge of mechanisms and of physics may be regarded as more creative than a similar process carried out by an engineer. The process by which a schoolboy comes up with the general theory of relativity might rank highly in terms of creativity because the information which he would have had to process would be at a fairly rudimentary state. The fact that the discovery had already been made by someone else would not significantly affect our view of his creative ability. (We may think that he had wasted his time, however.)

A computer system which selected from a database of building designs might therefore be regarded as less creative than a system which generates descriptions by combining building elements, and then selecting from among these. In the former, the designs are already in existence. In the latter case the designs have to be generated, as well as evaluated. There is also a link between the idea of creativity and that of generality. The more general the descriptions available to a system the greater the capacity of the system to be creative. This view of creativity is closely connected to Shannon's definition of *entropy* (Shannon and Weaver, 1949). Entropy can be measured in terms of the amount of organization, or structure, exhibited between the components of a system. A high entropy implies little organization. The higher the entropy of a design system, the greater its creative potential.

The following aspects of process serve as qualifications to the issue of entropy as a criterion of creativity.

2.2.2 Efficiency

Efficiency has a bearing on the degree to which we may describe a system as creative. If a rule-based generative system exhibits a high entropy due to the unstructured nature of its processes it may be regarded as more creative than a highly structured system. However, this is conditional on the fact that designs are realized in a finite time.

2.2.3 Richness

If a design system is capable of producing only one design for a given context where other designs could serve equally well, then the system may be regarded as less creative than one which is able to generate a range of designs. Richness in terms of process may also be exhibited if a system is able to accommodate a wide range of task environments. This also reflects the *adaptability* of a system.

This view of creativity as concerned with the quality of a design measurable in terms of both its product and its process is important because of its mapping with the way we are often accustomed to talking about creativity in design. It presents difficulties, particularly in terms of how we define entropy in knowledge-based systems. A further model will be presented here and explored in greater detail.

3. A KNOWLEDGE-BASED MODEL OF INNOVATION AND CREATIVITY

In general, a computer-aided design system can be described in terms of objects, operations on objects, and some controller which decides which operation to bring into action and when. Objects and groups of objects undergo changes in states under the transformations brought about by various operators under a control strategy. Sets of operators constitute the *knowledge* of the system.

The argument is presented here that knowledge serves to define spaces of designs. The exploration of design spaces in order to produce a design exhibiting certain properties is an *innovative* act. If this exploration also involves the search for knowledge then the process can be described as *creative*. We explore this view by developing a model of the design process.

In this discussion the linguistic model of computation will be explored, namely that computational processes in CAD can be described in linguistic terms. (The application of language theory to design has been explored, and its utility demonstrated, by Stiny [1975, 1980], Stiny and March [1981] and Mitchell [1983].) In this model we consider two types of computational systems, those concerned with interpretation and those concerned with the definition of languages. An interpretative system provides a mapping between some statements about the world, such as a collection of 'facts' in a computer 'database', and the *meaning* of those statements. Taken together, the statements in a database constitute a 'sentence' in some language. The second type of system is concerned with the rules which define the syntax of such a language. Mappings between sentences and meanings is a concern of semantics. On the other hand, the definition of what constitutes a legal sentence in the language is an issue of syntax.

These two types of systems can serve to define spaces of designs and spaces of interpretations. An interpretative system infers the meaning of a design. It maps an individual design onto a set of interpretations within a space of interpretations. It can also provide a mapping from particular meanings to a space of designs which can be interpreted as embodying those meanings.

A syntactic system also serves to define a space of designs – these are the designs in the language made possible by its grammar. We may therefore characterize a design system as being concerned with the production of designs which belong to the space defined by the language, but which also belong to the space defined by the interpretative system, that is, accord with some specified set of meanings.

In such a system the meanings of a design are simply the attributes of the artifact description which are not explicit in the description but must be inferred in some way – they are *implicit* attributes: that rooms bear particular relationships to each other; that the building costs a certain amount; or that it exhibits a certain performance in terms of energy efficiency. All possible designs which map onto a particular set of such meanings constitutes a space of designs. In all likelihood such a space will be extremely large. Design spaces can also be defined in terms of languages. These may embody particular conventions or styles (March and Stiny, 1985). We would expect a design to constitute a member of a set formed by the intersection between the space defined by the interpretative system and that defined by the syntactic system.

These two types of systems, which might reasonably be thought to constitute subsystems in a design system, will be discussed in turn. That which defines design spaces can be termed *design knowledge.*

These two types of systems will be characterized in terms of an understanding of logic and reasoning, prior to a discussion of how this view of computation may be said to account for innovation and creativity.

3.1 Interpretative Systems

The mode of reasoning we are generally able to discuss with any assurity is *logical deduction*. In making deductions we employ rules to chain through related propositions to establish the consistency of new statements with given statements. This type of reasoning lends itself to verification, and we recognize it in good argument. It can also be readily formalized.

We may well ask what relevance this type of system has to the derivation of meaning. The interpretation of a set of logical statements is simply that which can be logically inferred from them. In logic, the meaning of a set of statements is that which is not stated explicitly, but which is *implicit* in those statements. Logical deduction therefore provides a mechanism for interpretation.

Deduction is the basic building block of formal reasoning systems. It is generally recognized, however, that there are two other modes of reasoning to which humans have recourse: namely *induction* and *abduction*. Induction can be seen as the process by which logical rules are derived. Recent developments in machine learning suggest that it is possible to model inductive reasoning. The principal advocate of this view has been Simon (1977), and inductive systems which demonstrate the utility of this view have been developed by Winston (1975) and

Michalski (1983).

A third mode of reasoning has been proposed, principally by Peirce (1839-1914) (Feibleman, 1970). This type of reasoning is sometimes called *abduction*. This is the derivation of statements about the world given logical rules and some logical consequences. Peirce argued that abduction is a valid mode of human reasoning, Eco (1984) sees it operating in Science, Charniak and McDermott (1985) use it to explain the process of medical diagnosis, and March (1976) sees it as the key reasoning mode operative in the design process. Again, it is a type of reasoning which has been successfully modelled in a formal way.

A deductive reasoning system can be called an 'axiomatic system'. It consists of axioms and premises and is driven by deduction. When a proposition is proved true it remains true independent of any other deductions. The set of such logic statements is said to be *monotonic*. The utility of axiomatic systems as a means of modelling human reasoning has been called into question, and this has given rise to the investigation of other reasoning models, notably the idea of reasoning by *circumscription*, or *non-monotonic* reasoning.

The driving force in logic systems is the quest for consistency. Theorems are postulated and if they prove to be inconsistent with the axiomatic system then they are effectively rejected. There is no room for inconsistent statements in formal logic. Certain controls can be imposed on a theorem prover, however, which facilitate responses other than failure in the event of inconsistency.

If there is an inconsistency then perhaps the axioms should be rejected or modified in some way. Another strategy might be to adjust the premises rather than the theorems or axioms. The notion that it may be the 'givens' that are responsible for failure in a logical system, and the issue of how to control their modification in the light of inconsistencies is more fully addressed by the subject of non-monotonic reasoning (McDermott, 1980; Doyle, 1981; Moore, 1985).

3.2 Syntactic Systems

Can the processes within syntactic systems be described in a similar manner to those of interpretative systems? A simple manifestation of non-monotonicity occurs where the axioms of the system serve to change the state of the premises. This can be seen in the case where the axioms serve as re-write, or transformation rules, as in a linguistic grammar. That a statement may be explicit in one state but not in another need cause no problem as long as we remember that the truth or falsity of certain statements is relative to the states to which they belong. The various states can be defined by the axioms that bring them into being. Such a system can be readily modelled by ensuring that statements are tagged with state identifications in some way. In representing the system we generally adopt the shorthand convention of dropping the reference to states. A transformation system can be regarded as a particular type of axiomatic system and it can be considered to define a language.

A system defines a language if it consists of three components: a vocabulary of elements; a set of rules for transforming 'strings' of vocabulary elements; and a start state. A natural language system, formulated along these lines, is concerned with deducing whether a sentence is a 'legal' sentence in the language. The axioms of the system are the grammar rules, or rules of syntax. The rules determine the process of re-writing by which words in the sentence are replaced by certain non-terminal vocabulary elements (syntactic categories). If the sentence conforms to the syntax of the system then the process of substitution eventually produces the start state. This process is called *parsing*, and is a manifestation of the deductive process described in the interpretative system above.

How can induction and abduction be characterized in such a system? Induction is the process by which a set of transformation rules is derived. The process of abduction results in the definition of a space of designs which conform to the language. In an interpretative system abduction generally results in a space defined by bounds and constraints, but in a syntactic system the rules employed in parsing actually facilitate the generation of designs and partial designs. The transformation rules can be employed to bring about changes to the start state, and subsequent states, such that a space of states is generated. The transformations produce descriptions of designs conforming to the language.

The fundamental role of creativity in design is therefore clear. Creativity is operative where design spaces are undergoing formulation and redefinition. Creativity is strongly linked to *induction* and concerns the acquisition of design knowledge. This accords with Eco's (1984) view of *creative abduction* in which reasoning from an interpretation to a design is taking place, but where the rule (knowledge) is uncertain or unknown.

Design systems are therefore characterized by certain processes based on models of reasoning. In the next section an attempt is made to describe these processes more formally.

4. A STRUCTURE FOR CATEGORIZING DESIGN PROCESSES

We may expect a creative system to be one which engages in all of these modes of reasoning. Creativity is concerned with deduction and with the processes by which spaces of designs are defined, but also with induction – the process by which new knowledge is acquired. From the above analysis we can characterize the various components of a design system in terms of the following sets:

designs	D
vocabulary of elements	V
knowledge	K
interpretations	I
contextual information	C

The types of knowledge (K) with which a design system is concerned may be characterized as follows:

$$K = \{ K_i, K_g, K_c, K_f \}$$

where:

K_i = interpretative knowledge;
K_g = generative knowledge;
K_c = control knowledge; and
K_f = formulative knowledge.

We need not follow all the processes described in the preceding section. For example, it is difficult to formulate a meaningful mapping between parsing and design. The following mappings characterize various reasoning processes applicable to design. Those that have been considered in the above discussion include the semantic mappings:

$I = \tau(K_i, D_j)$	interpretation	(1)
$K_i = \tau(I, D_1, ..., D_n)$	induction of interpretative knowledge	(2)
$D = \tau(K_i, I)$	abduction	(3)

where $D_1, ..., D_n$ denote specific instances of designs and τ represents some kind of transformation function, and syntactic mappings:

$V_0 = \tau(V, K_g, D)$	induction of start state	(4)
$V = \tau(K_g, D)$	vocabulary induction	(5)
$K_g = \tau(V, D_1, ..., D_n)$	generative knowledge induction	(6)
$D = \tau(V, K_g)$	generation	(7)

where V_o is a particular vocabulary element designated as a start state,

$$V_o \in V.$$

Another important mapping is the production of interpretations where no design exists. This characterizes the search, not for the meaning of an artifact, but for the meaning we would want an artifact to exhibit:

$I = \tau(C, K_i, V, K_g)$	goal induction	(8)

These symbolic models are not intended to be exhaustive, but they form the basis of a taxonomy of design systems. There may be other mappings. What these models say about control processes will be addressed in Sections 6.

5. REASONING MODELS AND DESIGN SYSTEMS

The intention here is to demonstrate how these various processes – described by equations (1) to (8) above – can be realized in a design system, and to explore how each can be said to form the basis of an innovative or creative system. In this discussion both the qualitative and the knowledge-based models of innovation and creativity will be considered.

5.1 Deduction

In a design context deduction provides a mechanism for interpretation as described above by the symbolic model:

$$I = \tau(K_i, D_j).$$

Given a description of a design (D_j) and some interpretative knowledge (K_i) a set of interpretations (I) can be derived. These interpretations constitute the performances of the artifact.

In the system TOPOLOGY1 (Akiner, 1985), low level descriptions are interpreted to derive higher level descriptions. For example, the descriptions 'space *a* is next to space *b*' or 'space *a* is above space *c*' may be derived from low level descriptions in terms of the coordinates of the spaces. A similar process can also be carried out in determining building regulations compliance. A building description is interpreted to produce performances which are then evaluated. Figure 1 shows an example of an expert system dealing with building regulations compliance (Gero et al, 1986).

This example is concerned with a two-bedroom flat designed using a commercial CAD system, called EAGLE (Carbs Ltd, 1985). The flat is labelled, *flat1*, and its description is interpreted and evaluated by a model-based expert system. This system makes use of a knowledge base containing rules about minimum room sizes. The statement:

diagnostic on area is_ *area of the room is ok,*

is a derived or high-level fact that means that the room *bathroom1* complies with the floor area requirements of the regulations. It is derived from explicit or low level information about the dimensions of lines bounding the rooms and the objects contained within the flat and the rooms.

As shown in the example above, deduction as a mechanism for interpretation produces new descriptions – that is, new information. Is this process innovative or creative? It can be argued that because the information so produced is implicit in the system nothing new is created. Such a system, it can be argued, can never create anything which is not already fully described within it. We might, however, tend to designate a system as creative if the information produced is not readily obvious.

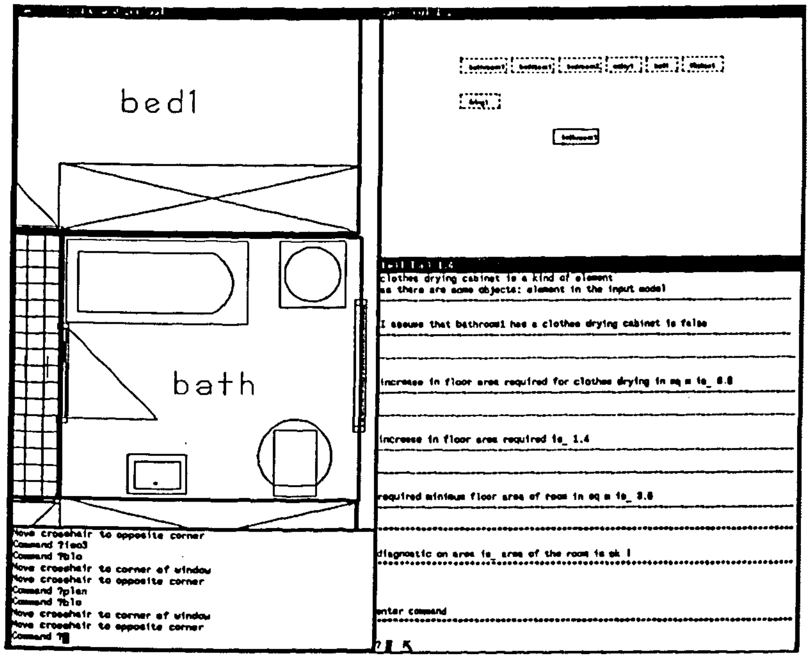


Figure 1. Screen display from an expert system containing a knowledge base related to building regulations interrogating the database of a commercial CAD system to check whether the two bedroom flat shown conforms to the regulatory requirements.

It has been argued that whether or not deduction *appears* to create anything new is a psychological issue and depends on our familiarity with the information contained in the system (Hospers, 1956). It can be argued, therefore, that deduction *is* an innovative process. According to a qualitative definition of creativity the degree to which we say that a deductive system is creative depends on the degree that the conclusion is apparent from the original premises.

However, according to the knowledge-based model this system does not appear to constitute creativity. A deductive system cannot create or hypothesize notions outside its strictly laid down axioms. The utility of deduction as a model of creativity is therefore limited. Deduction may, however, constitute a necessary component of a creative system. Any creative system will need to interpret information given to it, and it will also need to evaluate information that it generates itself. Deduction also serves as a tool by which we can model other reasoning processes (such as induction and abduction) which may also constitute creative systems.

5.2 Abduction

Abduction was introduced in Section 3. Abduction does not necessarily produce a bounded space – this can only be assured if the logical system represents a closed world.

In a closed world the assumption is made that the axioms of the system provide a complete mapping from a set of possible interpretations to the premises constituting an artifact description. The distinction between open and closed worlds can be summarized with reference to the following simple axiom:

if B **or** C **then** A

If we know B is true then, irrespective of whether the system describes a closed or open world, A can be inferred. If this is a closed world system we can also ask: if we know A is true what can be inferred? The answer is: B or C. In an open world the assumption is that the truth of A could also be conditional on other statements not expressed within the system. In this case, knowing A does not definitely conclude B or C. That an interpretative system (as a logic system) defines a bounded space of designs is conditional on the fact that the closed world assumption applies.

Another way of addressing this issue is to consider whether a set of interpretative rules are descriptive or prescriptive in character. The axioms within a system of building regulations (assuming they are consistent) are generally prescriptive. They say what must be true about a building in order for it to constitute a legal design. They can be said to effectively define a space of legal designs. If an attempt is made to run an interpretative system 'backwards' then we would expect some kind of disjunctive statement about various possible design descriptors – statements which effectively bound the space of possible designs.

In a design context abduction leads to the definition of a space of designs as suggested by the symbolic model:

$$D = \tau(K_i, I).$$

Given a specification (I) and some interpretative knowledge (K_i), a set of designs (D) is derived.

Is this process abduction or simply a special case of deduction? This question is pertinent when we consider that the derivation of artifact descriptions can also be inferred by deductive statements such as:

if waterproof roof is required **then** roofing is sheet metal (9)

This is a legal deductive statement. Given that a waterproof roof is required we can deduce that the roofing must be sheet metal. Statement (9) seems to model simple design decision-making – we interpret a design goal to arrive at an artifact description. Statement (9) suggests that deduction can produce design descriptions as well as serving as a mechanism for interpretation. A comparable interpretative statements is:

if roofing is sheet metal **then** roof is waterproof (10)

This apparent paradox can be resolved if we consider that the goals of the design should not be confused with the goals of the reasoning process. Although statement (9) is modelled deductively it uses abductive reasoning. This is explained in greater detail below.

We can consider the goals of a design system as the performances of the artifact. In this example the goal is the performance (the roof is waterproof). It is necessary for the design system to arrive at a description of an artifact which satisfies this goal. Statement (9) takes the design goal as a given premise and the 'solution' to this goal as the goal of the reasoning system. The roles have been interchanged. To further highlight the process we consider what would occur if, together with statement (9), the following statement also exists:

if waterproof roof is required **then** roofing is terracotta tiles (11)

Statements (9) and (11) declare that, in order for the roof to be waterproof, there is a possiblity that the roofing be either sheet metal or terracotta tiles. Obviously, at any one time the roofing cannot be both. (The MYCIN system [Shortliffe and Buchanan, 1975] incorporates similar reasoning processes, making use of probability measures – or degrees of certainty – to reason from evidence to hypotheses.)

Abduction can be demonstrated with the following statements:

if A **then** B (12)
if A **then** C (13)
if B **then** not(C) (14)
if C **then** not(B) (15)

Given A we should be able to deduce

B **and** not(C) **or**
C **and** not(B)

Statements (12) and (13) can be rewritten as

if A **then** B **or** C (16)

and further, statements (14), (15) and (16) can be rewritten as:

if A **then** p1.B **and** p2.C (17)

where the period (.) indicates product, p1 is the plausability of B and p2 is the plausability of C:

$p1 \in \{1,0\}$,
$p2 \in \{1,0\}$
$p1 + p2 = 1.$

We can now take the statement:

If B **or** C **then** A (18)

together with statements (14) and (15). Using abduction, and the assumption that the only relevant premises existing are B and C (the closed world assumption), we can derive that either B or C must be true. Therefore, we can see that statement (18) is equivalent to statement (12) and (13), and their corresponding representations in statements (16) and (17). The latter statements are thus shown to be cases of abduction modelled with deductive statements.

This can be demonstrated further with reference to an expert system called RETWALL (Hutchinson, 1985; Rosenman et al, 1986) which is an expert system for the selection and design of earth retaining structures. The goal of RETWALL is to produce the design for an earth retaining structure. The procedure it follows is to first select a set of possible solutions, then select a final choice based on an implicit cost criterion, and finally refine the properties of the structure by defining its parameters (geometry, dimensions and reinforcement). The rules in RETWALL are of the abductive reasoning type modelled deductively as described above. A typical rule is shown below:

```
r413(if
  'height of earth retaining structure (in mm)' is_greater_than 1800 and
  'height of earth retaining structure (in mm)' is_less_than 10000 and
  'Reinforced concrete wall is aesthetically acceptable' and
  'Labour and materials are available for  reinforced concrete'
  then
  possible('type of earth retaining structure' is_ 'concrete cantilever') and
  'Reinforced concrete wall is suitable for this application').
```

RETWALL then employs a deductive reasoning mechanism (either goal-driven or data-driven) to arrive at a suitable description of a structure which meets the given requirements. The system effectively interprets the specifications of the problem in order to arrive at a design.

An alternative approach to that taken by RETWALL is to represent the knowledge as rules for interpreting designs – to derive their performances – and to employ *abductive* reasoning to derive a set of possible solutions. This procedure can be demonstrated by considering the knowledge contained within a set of building regulations, as demonstrated above. Instead of asking the system to derive the floor area requirements for a given room we can ask it to furnish us with all the rooms, and their features, which meet a given minimum floor area requirement. For example, we can ask the system to find those solutions which satisfy the goal: that the minimum floor area is 11.0 square metres. The results are shown in Figure 2. In the current implementation this procedure is paralleled by asking the system why a certain conclusion was *not* reached. This is part of the explanation facililty of the BUILD expert system shell (Rosenman, 1986). The procedures are equivalent, and the result is that the system produces the various options in order for the requirements to be met.

Abduction results in a space of solutions rather than a single solution (except for the special case where only a single solution exists). By specifying the performance to be achieved we obtain a set of design descriptions. By specifying a different set of performance requirements a different set of design descriptions is obtained. By specifying several performance requirements we can narrow the solution space to that of the the intersection of several individual ones. This is illustrated in Figure 3.

Artifact descriptions can be produced by means of statements such as:

if shelter is required **and** view is to be maintained
then provide a glass partition.

According to a qualitative understanding of creativity we could say that a greater degree of creativity can be exhibited by abstracting the statements and employing more general levels of descriptions. Such a system would then be capable of employing general principles to arrive at a variety of solutions which are not

---

```
explain why_not 'required minimum floor area of room in sq m' is_ 11.0.
-----------------------------------------------------------------------

building is_a dwelling-house or flat and
room type is_ habitable room and
not(room is_a kitchen) and
number of habitable rooms contained is_greater_than 1 and
there exists at least one room with floor area of at least 14 sq m and
not(there exists another room with floor area of at least 14 sq m or
    increase in floor area of habitable room requirement)
    needed to prove
required minimum floor area of room in sqm is_ 11.0

room is_a bedroom or living room or lounge room or music room or kitchen or dining
room or sewing room or study or playroom or sunroom
    needed to prove
room type is_ habitable room
```

Figure 2. Output from the use of 'abductive reasoning' in the context of building regulations.

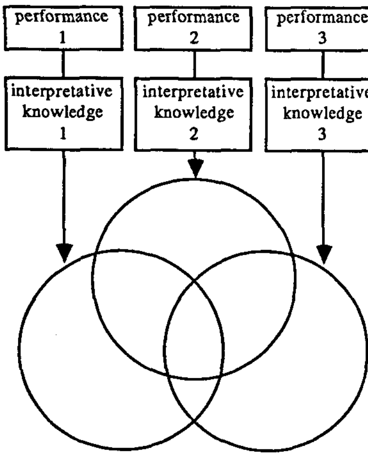


Figure 3. A solution space as the intersection of spaces created by abduction from several interpretation systems.

explicit in the original statements. We can illustrate this by considering the following set of logical rules:

if shelter is required **and** view is to be maintained
then weatherproofing is required **and** transparency is required

if weatherproofing is required
then prevention of weather penetration is required

if prevention of weather penetration is required
then a waterproof barrier is required

and a set of facts such as:

partitions are barriers
partitions are made of solid materials
glass is a material
glass is solid
glass is waterproof
glass is transparent
a stream of pressurised air prevents weather penetration
air is transparent

The above set of statements illustrate how solutions to design problems can be derived – solutions which are not explicitly stated. In this case an air curtain is suggested as a possible solution as well as a glass partition. Of course, the set of statements shown above does not represent the entire set of knowledge that a design system would require. (A further example is EDISON which is a system for the design of mechanical devices from knowledge of basic physics [Dyer et al, 1986].)

5.3 Generation

A generative design system is concerned with the production of a set of designs from a vocabulary of design elements and a body of generative knowledge, according to the syntactic model:

$$D = \tau(V, K_g).$$

In order to demonstrate the role of generation in creative systems the EAVES detailing system (Radford and Gero, 1985; Mitchell and Radford, 1986) will be considered. In the EAVES system there is a vocabulary of building elements, which are linked according to a body of generative knowledge (in this case described as a grammar of rules) to generate members of a language of designs for eaves details. The system operates from one of three initial states for the design (single brick external wall, double brick external wall or timber framed external wall), and the rules of Figure 4. link a condition in the state of the design to a set of consequents. The consequents of the rules constitute modifications or additions to the state of the design.

The vocabulary and grammar of the system are inferred from examples of Australian domestic architecture of the Federation period, around 1915. Designs are generated by successively matching the rules of the grammar and firing those that are applicable, Figure 5. The design is complete when there are no more rules left to fire.

To what extent may the processes of this system be regarded as innovative or creative? Adopting the qualitative view of creativity and considering the criteria outlined in Section 2, we can look first at the system's *products* . The products are cetainly valuable; because of the derivation of the rules they are designs which have been proved to function well and look good. They are not, though, necessarily *original*; because of the derivation of the rules all the designs fit within a class of established 'solutions', although some may be unexpected. This is because they are the result of combinations of rules which have not been commonly used by human designers. From this viewpoint the system is being less creative when it generates the design in Figure 5, which is a common form of eaves detail in Sydney houses of the period, than when it generates a design with a sloping fascia and no gutter, which is much less common. We can look at the *richness* of the system; it will generate a very wide variety of designs, which is wider, in fact, than those generated by the architects of the period. Does this, then, make it more creative than the architects of the period?

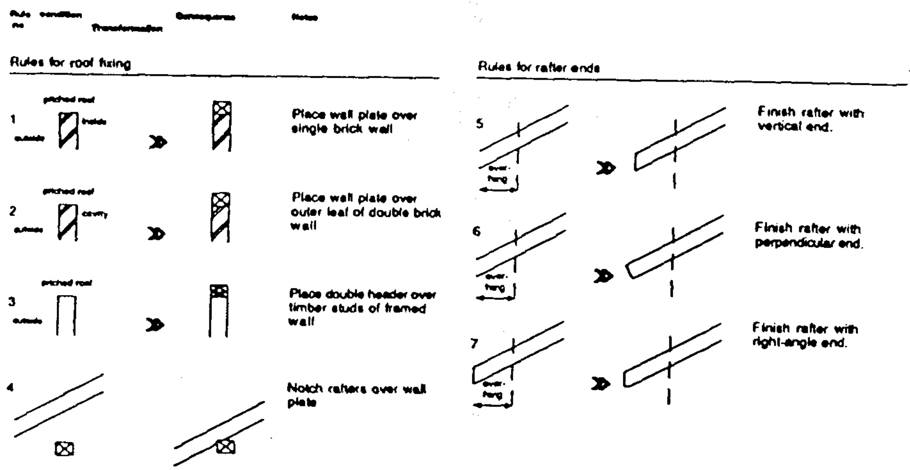


Figure 4. The first few rules from a rule set for generating designs for eaves details.

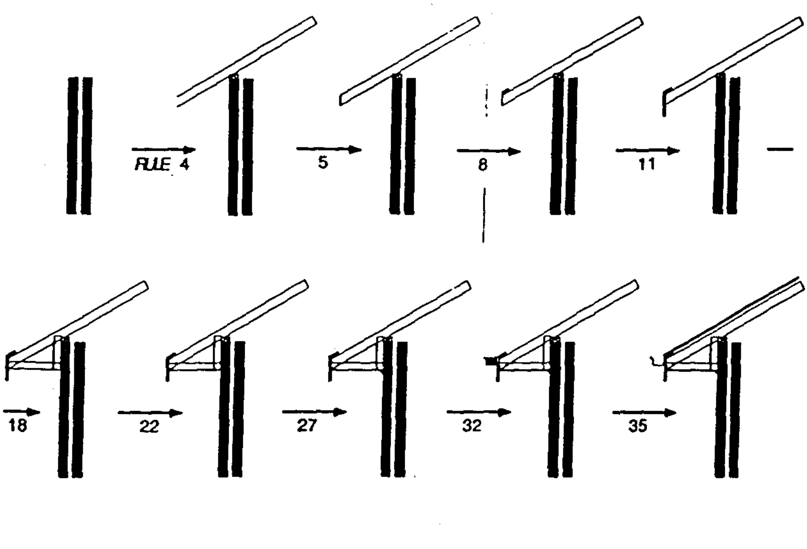


Figure 5. An example of the state of an eaves design as it develops through the sequential application of design rules.

The system as described is essentially syntactic in nature. As such it contains no knowledge about intent or meaning. The knowledge that makes possible the mapping between intentions and designs is supplied by the human operator of the system who either makes selections between competing rules or evaluates the designs so produced. In Section 6 the view will be presented that the knowlege by which grammar rules are selected can be made explicit. This provides a key to the integration of semantic and syntactic systems in a design domain.

A shortcoming in the *creative* ability of this type of system is that it is incapable of producing a configuration of elements which is not anticipated in the formulation of the generative knowledge. This issue is demonstrated with the following example.

An example of the implications of viewing creativity from the point of view of the product is provided by an experiment in the development of a grammar for generating a class of designs by the Australian award-winning architect Glenn Murcutt (Hanson and Radford, 1986). (This grammar has not yet been implemented as a computer system). The architect is best known for a series of domestic scale buildings in which he has developed, and continues to develop a strong and practical architectural form for the Australian landscape and climate. The basic method was to infer from writings on his work, and from a corpus of examples of his pavilion-like country houses, an ordered set of condition and consequent rules that appear to apply, Figure 6.

As a case study, the grammar was tested by applying the rules to develop a design for a house. This was carried out in parallel with the architect who started with the same initial design state (site and brief). The interest here lies not in comparing the grammar and the architect's design for this house (these are compared in detail by Hanson and Radford, 1986), but in comparing the grammar design with a then unpublished architect's design for another site, that of the Ball-Eastaway house at Glenorie, Figures 7 and 8. To some extent the similarities are fortuitous; the grammar can and has been used to generate other designs which are not so close. The degree of similarity results from the choice of which rules to apply, in this case a choice made by the human driver of the grammar. The issue we wish to consider, though, concerns the value of the result. The Glenorie house won an award for architecture, and the jury "considered this an intensely intellectual work". Does this mean that the plan generated by the grammar is "an intensely intellectual work"? (To be accurate, the jury were assessing much more than the house plan, but for the sake of this discussion let us assume that the designs are in all respects identical). The answer has to be 'yes': the design is the same whatever its origin. The 'intellectual work', though, and the creativity can, in both cases, be traced back to the human designer. In the grammar it is encoded in the vocabulary and rules.

The system as described here does not demonstrate the creative component of design as outlined in the knowledge-based model. Glenn Murcutt said of the grammar: "But this won't show me the future, it is only interested in the past". He said that if he used the system his designs would never improve, never develop. He could only produce variations of his past designs. The grammar is a time and

Rules to establish the preliminary form.

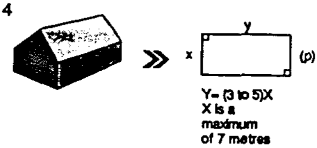
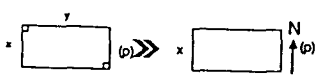
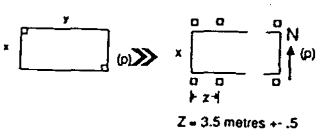
Rule no.	condition	Transformation	Consequence	Notes
1	Site in Urban Area.	»		(urban form) Open sites on the outskirts of cities are still urban.
2	Site in country or site is semi-rural	»		(pavilion form) Semi-rural = for example, a country town. This rule just directs the grammar to the appropriate set of prelim-inary rules. Later rules differentiate Between rural and semi-rural houses.
3				(one pavilion) No rules are apparent when more than one pavilion is to be used. * Murcutt says there are very clear rules: if the building is too long in one direction, then go back, and because he wants sunlight in back areas the pavilion is repeated using the roofplane to get sunlight.
4				This rule says pavilions are rectanges in plan. (p) means the rule is in plan
5				This orientation rule applied to all five single pavilion houses in the sample but can be changed by view, wind, slope, etc.
6				The rules are parametric, so do not have to give exact sizes for variables. *This rule needs more definition. Murcutt tells us it is the materials that decide the grid. "Timber will span 3m very beautifully, over 3.2m move into steel" up to 4.8 to 5m. Lightweight steel also spans about 3m. But 3m is too small for a bedroom, so in his recent houses they may "push out" through the grid. Walls almost always line up with the grid, but there are exceptions.

Figure 6. The first few rules from a rule set for generating designs for pavilion houses.

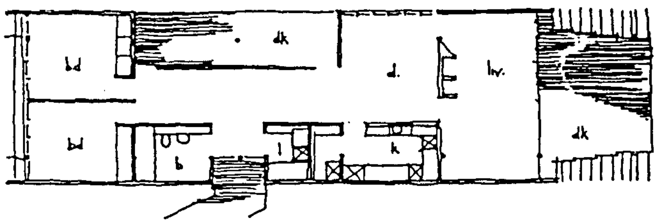


Figure 7. The plan of the Ball-Eastaway house at Glenorie, designed by Glenn Murcutt.

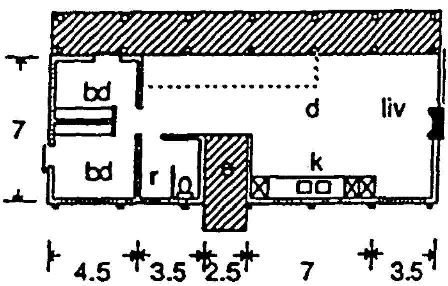


Figure 8. A house plan generated by executing the pavilion grammar.

domain frozen representation of his work, rich within very fixed boundaries which can only be changed by explicitly adding to the vocabulary and rules. The human is always refining his ideas, building on what has been done in the past. New vocabulary and rules are added, but they are added by the 'human system' itself, not inserted by an outside agency as is necessary in the grammar system. The human exhibits a richness in the *process* not mirrored in the grammar. Clearly, the acquisition of design knowledge is a key process within a creative system. This will be addressed in greater detail in the following section.

5.4 Induction

In this discussion induction is concerned with the derivation of interpretative knowledge from a set of designs and their interpretations, according to the syntactic model:

$$K_i = \tau(I, D_1, ..., D_n),$$

and also with the derivation of a syntactic grammar from a set of designs and a vocabulary:

$$K_g = \tau(V, D_1, ..., D_n).$$

We will consider the induction of interpretative knowledge first. Induction is a process by which general rules are derived from individual cases. A kind of naïve induction can be characterized as follows: knowing a proposition that A is true and that B can be deduced from this fact we might infer the statement:

if A **then** B.

In reality the process of induction is much more complex than this. There may be other factors impinging on the fact that B follows from A. A deductive inference rule is a generalization which accounts for different cases of A and that which can be inferred from A. If the rules contain only explicit mappings between cases and interpretations then no generalization has taken place. An inductive system must be able to derive rules which account for cases which have not been previously encountered.

A simple algorithm for the derivation of generalizations has been devised by Hunt et al. (1966) under the title of CLS (Concept Learning System). Further automated learning systems have arisen following a similar approach – notably ID3 (Quinlan, 1979), and the commercially available programs, Expert-Ease and Rulemaster (Michie et al, 1984). These programs are intended specifically for the derivation of expert knowledge from large numbers of examples. Little work has proceeded in the area of machine induction and design systems. However, we demonstrate a simple application of a CLS-type learning system in the derivation of a few rules – in this case rules about the set-backs of buildings from boundaries in residential design.

Figure 9 shows a learning set which consists of positive and negative examples of setback conditions. Those diagrams which are drawn with bold lines are those which comply with the standard. The other examples do not comply. The task of the learning system is to discover what are the essential features of the designs which contribute to their compliance or non-compliance, and to construct general rules to account for the factors that are important. The system is told about the five attributes and the various values the attributes may take (Figure 10). Each example is also described in terms of the attributes and their values. There are two interpretations (generally called classes) possible: compliance (yes) and non-compliance (no). The system is also told which examples comply.

The CLS algorithm does not directly produce a deductive inference rule, but rather a procedural *decision tree*. The decision tree appropriate to the learning set is illustrated in Figure 11. The branch ends of the tree represent the classes (yes and no) and the other nodes are various attributes which have a bearing on the classes. The arcs of the tree are the values of the attributes. From the tree of Figure 11 we therefore find that: if the setback is greater than 3.0 then the design complies with

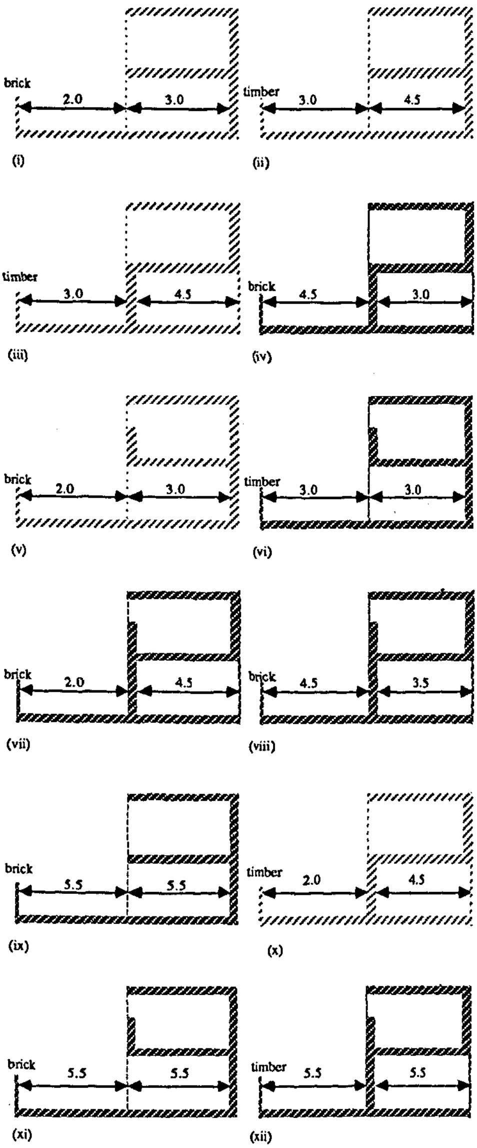


Figure 9. A learning set pertaining to site boundary conditions. The diagrams are of a cross section of a building near the property line and show the material of the boundary wall, distance from boundary, room depth and window conditions for two storey structures. Buildings drawn with heavy lines comply with the standards.

the standard. If it is less than or equal to 3.0 and the upstairs window is primary then it does not comply, and so on. This type of representation can be readily converted to the inference network of Figure 12. The nodes n1, n2 and n3 serve no purpose other than to make the tree more readable – they do not have any meaning. The rules can be further represented as logic statements as follows:

if (setback > 3.0) **or** n1 **then** complies
if (upstairs window = secondary) **and** n2 **then** n1
if (downstairs window = none) **or** n3 **then** n2
if (downstairs window = primary) **and** (setback > 2.0) **then** n3

It is noteworthy that the system has been able to discern that the wall material and the room depth do not have any bearing on whether the design complies with the standard. In general the CLS algorithm is intended for larger learning sets than that demonstrated here. Other approaches have been devised, notably by Winston (1970) and Michalski (1983). Clearly, it would be desirable if it were not necessary to spell out the characteristics of the design in detail, but if the system could infer significant descriptive features from a geometrical, or other low-level, description of the design. The STAR knowledge acquisition system is an attempt to address these issues (Michalski, 1983).

Understanding the knowledge by which a syntactic grammar can be derived is a more difficult problem than the induction of interpretative rules. The acquisition of grammar rules has been addressed in natural language by Anderson (1977) and Berwick (1980). We may consider the derivation of the grammar for the house design systems described above in Section 5.3. A grammar constitutes a hypothesis, the testing of which is accomplished by attempts to employ the grammar to re-generate known designs. The default mechanism of learning is therefore that of generate and test, which is enhanced by the imposition of various control heuristics. Formulating hypothetical grammars requires domain dependent knowledge which embodies assumptions about what to look for in example designs: such as spatial organization, perimeter configurations, common adjacencies, common connections, fenestrations and room proportions. The derivation of grammars appears to be an expert task the knowledge for which is difficult to externalize.

We may also suppose that a system should learn, not only from other designs, but also from its own activities. Having generated certain designs these can constitute the learning set from which new rules are induced. Such a system improves its own performance in time. This suggests that a highly creative system must be dynamic.

Induction does not produce artifact descriptions, but rather knowledge (as rules in this example) which can facilitate interpretative and syntactic processes in design

Attribute	**Values**
upstairs_window	primary, secondary
downstairs_window	primary, none
set_back	continuous
room_depth	continuous
fence_material	brick, timber

Figure 10. Table of attributes and corresponding values.

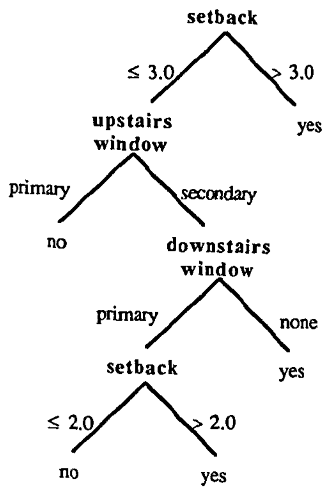


Figure 11. Decision tree resulting from analysis of the learning set of Figure 9.

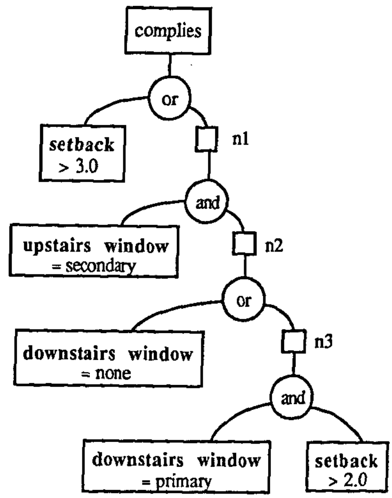


Figure 12. Inference network derived from decision tree of Figure 11.

systems. Induction is a process by which disparate information in the form of discrete statements about individual cases is distilled into rules of interpretation and generation. To this extent induction contributes substantially to the creative character of a design system.

6. CONTROL OF DESIGN SYSTEMS

Here we consider the role of innovation and creativity in the control of design systems. In the discussion so far it is assumed that the 'vocabulary' elements of a design system are statements about physical form. The database on which a design system operates may consist of statements about lines and points, or higher level descriptions of objects as in the previous example. These descriptions can be interpreted by an interpretative system as objects in the manner outlined above. Descriptions which conform to the syntax of some language can also be generated as outlined. But there is no stipulation in our model that the objects of the system have to relate to form. There are other types of entities on which a system could operate, such as those which have no mapping onto physical objects (they may remain in the realm of 'ideas'). Can we also regard the components of the system itself as objects upon which the system may operate?

One manifestation of this is to consider the transformation rules which operate in a grammar. Because these rules bring about changes in states they can be regarded as actions. It is possible to devise a design system in which the objects are actions, and vocabulary elements are statements about the relationships between actions. There is therefore knowledge by which we can interpret sequences of actions, and there is syntactic knowledge which defines a language of such actions. The sequence of actions generated by such a system would constitute a plan of actions. This system effectively serves as a controller for a generative system. The operators (that is, the axioms) of a design system serve as objects in a control system. This will be demonstrated with reference to a system for spatial layout design (Coyne, 1986; Coyne and Gero,1986).

Designers appear to possess a ready vocabulary of permissible actions: for example, to put objects down, to move objects, erase them, and make then larger or smaller. In fact the full range of geometrical transformations is available. But actions can also be described in relational terms: for example, to place an object next to another object, or to locate an object to the north of another object. There are, of course, further transformations by which objects change their meanings, such that lines become walls. Taxonomies of actions can be devised which indicate the relationships between different actions in much the same way that it is possible to draw up taxonomies of relationships between objects (Akiner, 1985). These taxonomies constitute semantic mappings between actions.

A major concern in the selection of appropriate actions is the resolution of conflicts between 'competing' actions. For example, in the realm of spatial synthesis the placement of one room may preclude that location as a site for another room. The action which locates one room is incompatible with the action which locates another. Designers seem to adopt strategies which enable them to actively avoid situations where conflicts occur, or to detect conflicts at an early stage when they are relatively easy to rectify. One strategy is to proceed from highly abstract levels of operations to less abstract levels. A common type of strategy is to consider that designers frequently regard spatial layouts at three levels of abstraction at least. One is the level of *topology* – of dimensionless 'bubble' diagrams or, more formally, as nodes on an adjacency network. Plans are then considered as *geometrical* entities which are only loosely formed. A final level of abstraction may be where dimensional constraints are taken into acccount. This is at a *metrical* level.

The transformation rules of the meta-grammar are therefore devised to detect patterns within partially developed plans, and to transform plans from highly abstract sequences into more specific sequences of actions. Partial plans therefore proceed from actions which state that rooms are to be *put* in place to actions about adjacencies, then to placements in terms of orientation. The final task of assigning dimensions to spaces is not considered here. What is the nature of the rules which operate on plans?

An example of a rule is:

if the plan contains a set of actions concerned with locating spaces
and there is no commitment to order as yet
then order the actions according to the importance of the spaces on which they operate.

This is a simple heuristic of the type that a designer might employ in beginning a layout task. There are eleven such rules in the system.

The layout task can be visualized as at least three distinct tasks: of organizing nodes on an adjacency network; formulating sequences of actions about the placement of spaces; and as the configuration of geometrical entities. Each of these abstractions has its own pertinent body of knowledge. Figure 13 is a screen display of the system showing the various stages in the synthesis process. The top left window shows the tasks being processed, the bottom left window shows a network representation of the spatial layout, the bottom right window shows the plan of actions and the top right window depicts the resulting layout after the execution of the plan of actions.

Can we also make explicit the knowledge by which the generative rules of a control system are selected and ordered? This knowledge would constitute *meta-control* knowledge. It may also be possible to represent the knowledge which controls that type of system. The hierarchical representation of control knowledge provides a system for organizing design knowledge such that certain strategies can be made explicit at various levels of abstraction. As we move into more abstract levels of control we move into strategies which become less dependent on the context of the task at hand. For example, rules for manipulating groups of subtasks could be based on general strategies such as: consolidation, diversification, convergence and divergence. These are often regarded as general-purpose creative problem-solving strategies. This view has been explored in greater detail by Coyne and Gero (1986).

In the system illustrated in Figure 13 the product of the system is effectively a procedure which, when implemented, produces a design. It is therefore a system which contains explicit knowledge about control. The processes which constitute this control system are essentially the same as those that were discussed in Section 4. We can refer to the interpretation of actions, the abduction of 'plan spaces', the generation of plans of actions and the acquisition of control knowledge. The system described above is primarily concerned with the generation of plans of actions. Induction would be operative in such a system if it was concerned with modelling the acquisition of design strategies, and with processes of exploration and discovery. The point to be made here is that creativity within the control process is an indispensable element of design.

7. DISCUSSION

Innovation and creativity have been discussed as properties of the design process. Two models have been considered. Firstly, innovation and creativity can be identified by observing the quality of the product, and also the characteristics of the process itself. In the second model we considered *innovation* as the production of a design from within a space of designs defined by a body of design knowledge. A *creative* design is one that is the product of a process in which the knowledge defining the space of designs also has to be discovered. This capacity for creativity

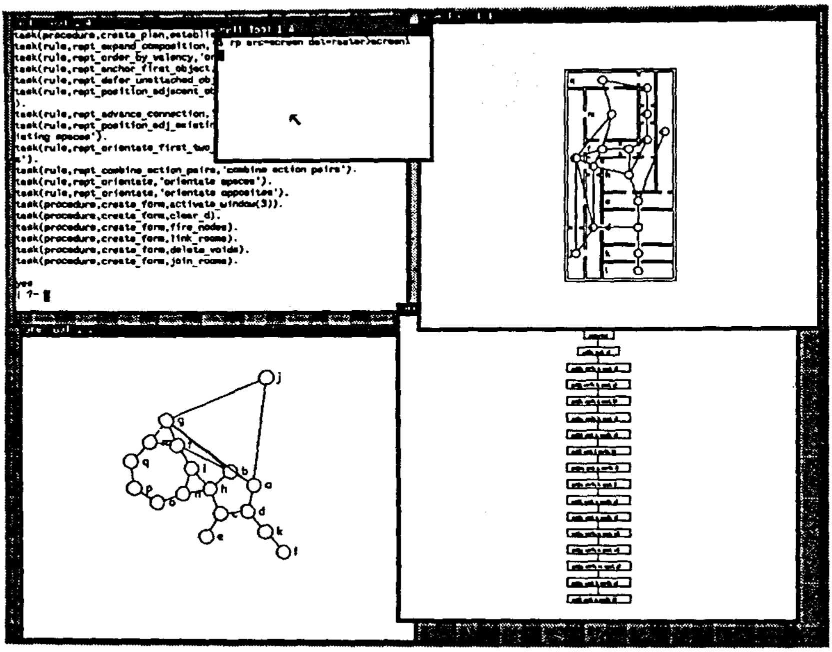


Figure 13. Screen display of the various stages in an expert planning system for the design of room layouts.

is exhibited in the ability of a system to acquire knowledge, but also in its ability to control its own processes and change its own structure.

In order to discuss this ability a model of design systems has been put forward in which we distinguish between interpretative and syntactic subsystems. These two types of system serve to set bounds on spaces of designs. The processes of reasoning applicable to these types of systems have been described with reference to terminology appropriate to logic: deduction, induction and abduction. In interpretative systems the first two processes can be translated as interpretation and knowledge acquisition. In syntactic systems the three processes are: parsing, knowledge acquisition and generation. A structure has been proposed by which the processes applicable to design can be characterized. Examples from automated design systems serve to demonstrate the creative potential of systems based on these processes in terms of both an 'object level' of representation and also in terms of control knowledge representation. A highly creative system is one which exploits all of the modes of reasoning mentioned, and does so at various levels of control abstraction.

We have also considered the qualitative aspect of creativity. Is it possible to measure the innovative and creative ability of a design system? Since there are no accepted standards for measuring the creative ability of designers, perhaps it is too ambitious to attempt to measure that of systems. However, we can suggest some ways in which different systems may be compared.

(i) Systems may be compared if there is some uniformity in the way they are described – for example, as rule-based systems. In this case we can look at the knowledge units (rules) and measure both the number of homogeneous rules and the change each rule produces. A system consisting of a large number of simple rules might therefore be regarded as more innovative or creative than a system consisting of a small number of rules which constitute large 'jumps' in reasoning.

(ii) A system may be regarded as more or less creative depending on the efficiency of the processing required. A generate and test system may consist of a small number of very simple operators, but the design may not be realized in a finite time.

(iii) Innovation and creativity may be measured according to the primary reasoning mechanism operating. In the semantic realm, logical deduction might be considered less creative than induction or abduction. In the syntactic realm, parsing is relatively uncreative, whereas the induction of knowledge and generation employing this knowledge are relatively more so.

(iv) Because we are interested in the production of artifact descriptions, the production of a set of rules may not be regarded as particularly creative. However, if the rules are subsequently employed in a generative system, then the system is more creative than if the rules were already supplied. We may therefore consider systems creative according to the *combinations* of reasoning mechanisms taking place.

In each of the above we can utilize entropy as a measure of the creative potential of a knowledge-based CAD system. However, it is necessary to measure not just the new entropy of a system but also the entropy of the control process as this affects the speed with which the entropy reduces or increases. If a system undergoes a process which increases its entropy then it is likely to be potentially more creative than one that does not.

There are, no doubt, other criteria measuring creative potential than those outlined above. These are suggested merely as an introduction to the topic, the development of which is beyond the scope of this paper.

ACKNOWLEDGEMENTS

Research support is from the Australian Research Grants Scheme, the Australian Telecommunications and Electronics Research Board and the University of Sydney Postgraduate Research Studentship Scheme.

REFERENCES

Akiner, V.T. (1985). *TOPOLOGY1 A System that Reasons About Spaces*, PhD Thesis, University of Sydney, Sydney.

Altshuller, G.S. (1984). *Creativity as an Exact Science*, Gordon and Breach, New York.

Anderson, J.R. (1977). Induction of augmented transition networks, *Cognitive Science*, vol.1, no.2, pp.125-157.

Berwick, R.C. (1980). Computational analogues of constraints on grammars: a model of syntactic acquisition, *Proc. 18th Conference of the Association for Computational Linguistics*, Association for Computational Linguistics, pp.49-53.

Broadbent, G. (1977). A plain man's guide to the theory of signs in architecture, *Architectural Design*, July/August, pp.474-482.

Carbs Ltd, (1985). EAGLE 3D Modelling System Command Reference Manual, Cadcom Pty. Ltd. 130 Pacific Highway, Greenwich, New South Wales.

Charniak, E. and McDermott, D. (1985). *Introduction to Artificial Intelligence*, Addison-Wesley, Reading, Massachusetts.

Cohen, H., Cohen, B. and Nii, P. (1985). *The First Artificial Intelligence Coloring Book*, Kaufmann, Oxford.

Coyne, R.D. (1986). *A Logic Model of Design Synthesis*, PhD Thesis, University of Sydney, Sydney.

Coyne, R.D. and Gero, J.S. (1986). Semantics and the organization of knowledge in design, *Design Computing*, vo.1, no.1, pp.68-89.

deBono, E. (1967). *The Use of Lateral Thinking*, Penguin, Harmondsworth, Middlestex.

Doyle, J. (1981). A truth maintenance system, *Readings in Artificial Intelligence*, Webber, B.L. and Nilsson, N.J. (eds.), Tioga, Palo Alto, California, pp.496-516.

Dyer, M.G., Flowers, M. and Hodges, J. (1986). EDISON: an engineering design invention system operating naively, *Proc. 1st International Conference on Applications of Artificial Intelligence in Engineering Problems*, Springer-Verlag, Berlin, pp.327-342.

Eco, U. (1984). *Semiotics and the Philosophy of Language*, MacMillan, London.

Feibleman, J. (1970). *An Introduction to the Philosophy of Charles Sanders Peirce*, MIT Press, Cambridge, Massachusetts.

Gero, J.S., Rosenman, M.A. and Manago, C. (1986). A model-based expert system shell, *First Australian Artificial Intelligence Congress*, Melbourne (to appear).

Hanson, N.R.L. and Radford, A.D. (1986). On modelling the work of the architect Glenn Murcutt, *Design Computing* (to appear).

Hospers, J. (1956). *An Introduction to Philosophical Analysis*, Routledge and Kegan Paul, London.

Hunt, E.B., Marin, J. and Stone, P.T. (1966). *Experiments in Induction*, Academic Press, New York.

Hutchinson, P.J. (1985). *An Expert System for the Selection of Earth Retaining Structures*, MBdgSc Thesis, Department of Architectural Science, University of Sydney, Sydney.

Koestler, A. (1976). *The Act of Creation*, Hutchinson, London.

Lansdown, J. (1985). Computing in the creative professions, *Research Paper*, Department of Design Research, Imperial College, London.

March, L. (1976). The logic of design and the question of value, in *The Architecture of Form*, March, L. (ed.), Cambridge University Press, Cambridge.

March, L. and Stiny, G. (1985). Spatial systems in architecture and design: some history and logic, *Environment and Planning B*, Vol.12, pp.31-53.

McDermott, D. (1980). Non-monotonic logic I, *Artificial Intelligence*, vol.13, pp.41-72.

Michalski, R.S. (1983). A theory and methodology of inductive learning, *Artificial Intelligence*, v.20, pp.111-161.

Michie, D., Muggleton, S., Riese, C. and Zubrick, S. (1984). RuleMaster: a second generation knowledge engineering facility, *Radian Technical Report MI-R-623*, Radian Corporation, 8501 Mo-Pac Blvd., Austin, Texas.

Mitchell, W.J. (1983). *The Logic of Architecture*, unpublished manuscript.

Mitchell, J. and Radford, A,D. (1986). Adding knowledge to computer-aided detailing, AUSGRAPH '86, Australasian Computer Graphics Association, Sydney, pp.31-35.

Moore, R.C. (1985). Semantic considerations on nonmonotonic logic, *Artificial Intelligence*, vol.25, pp.75-94.

Quinlan, J.R. (1979). Discovering rules by induction from large collections of examples, in *Expert Systems in the Micro-Electronic Age*, D. Michie (ed.), Edinburgh University Press, Edinburgh.

Radford, A.D. and Gero, J.S. (1985). Towards generative expert systems for architectural detailing, *Computer Aided Design*, vol.17, no.9, pp.428-435.

Rosenman, M.A. (1986). *BUILD Expert System Shell Users Manual*, Architectural Computing Unit, Department of Architectural Science, University of Sydney, Sydney.

Rosenman, M.A., Gero, J.S., Hutchinson, P.J. and Oxman, R. (1986). Expert systems applications in computer-aided design, *CAD86*, Butterworths, Guildford (to appear).

Shannon, C. and Weaver, W. (1949). *The Mathematical Theory of Communication*, University of Illinois Press, Urban, Illinois.

Shortliffe, E.H. and Buchanan, B.G. (1975). A model of inexact reasoning in medicine, *Mathematical Biosciences*, vol.23, pp.351-379.

Simon, H.A. (1977). Does scientific discovery have a logic? in *Models of Discovery*, H.A. Simon, Reidel, Dordrecht, Holland.

Stiny, G. (1975). *Pictorial and Formal Aspects of Shape and Shape Grammars*, Birkhauser Verlag, Basel.

Stiny, G. (1980). Introduction to shape and shape grammars, *Environment and Planning B*, Vol.7, pp.343-351.

Stiny, G. and March, L. (1981). Design machines, *Environment and Planning B*, Vol.8, pp.245-255.

Storr, A. (1972). *The Dynamics of Creation*, Secker and Warburg, London.

Vernon, P.E. (1970). *Creativity*, Penguin, Harmondsworth, Middlesex.

Winston, P.H. (1970). Learning structural descriptions from examples, in *The Psychology of Computer Vision*, P.H. Winston (ed.), McGraw-Hill, New York.