Grammatical Design and Bounded Creativity

KN. Brown and J. Cagan

EDRC 24-124-96

GRAMMATICAL DESIGN AND BOUNDED CREATIVITY

K.N. BROWN AND J.CAGAN
Computational Design Laboratory
Department of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract. In this paper, grammatical design and creative design arc shown to be compatible by introducing the notion of bounded creativity, a recognition that grammars limit but do not determine die design process.

Introduction

Grammatical design methods provide a computational basis for both design support systems and automated design. They do so by specifying legal primitives and manipulations of those primitives, like desi|H space (i.e., the set of all possible stales in the design process phis the legal transitions from state to stale) is thus completely, but implicitly, specified before design begins. This notion of a completely specified space leads to the argument that creativity is not possible in grammatical design. That is, the developer of the grammar has described all possible design paihs, and a designer is simply selecting one of these pre-specified solutions. In this paper, we argue that this view is mistaken.

We suggest that this argument involves an overly restrictive View of creativity, and would deny creativity to many human designers. Grammars do, we concede, limit the design options. However, grammatical design spaces can be vast and effectively unsearchable, and thus the limitation may not be significant. It is our contention that the defining characteristic of creativity lies in the control of the design process as much as in the space in which the process operates. When discussing creativity, we differentiate between the final artifact and the process used to reach thai artifact; we will consider only the process here. We define creativity (a creative process) as follows:

> **Creativity is a property of an agent that behaves in a manner that is beyond its standard practice for the current goal.**

Thus, for a given design problem, there might exist a body of standard techniques, taking a designer through a series of actions, specifying the order in which sub-solutions should be specified, and prescribing methods for achieving certain objectives. Creativity, in our definition, occurs when a designer diverges from these standard procedures in the process of solving the problem. We contend that grammatical design is compatible with this definition, and in particular, with what we will call *bounded creativity.*

First, we will review the basic work in grammatical design. We then consider the relationship between design spaces and control of the design process within those spaces, and introduce the notion of bounded creativity. We then present a more concrete example, illustrating our ideas. Finally, we consider other definitions of design creativity in the light of our definition.

Design Grammars

A grammar is a formal specification of a possibly infinite set, consisting of a set of primitives and a finite set of production rules which specify transformations of these primitives. By recursive application of the rules to the primitives, a grammar may be used to generate members of the set (called the *language* of the grammar); alternatively, by applying the rules in reverse, the grammar may be used to recognize (or *parse*) members of that set. Formal grammars were first proposed by Chomsky (1956) to specify the syntactic structure of natural language. The use of grammars for the design of complex structures was popularized by the development of the shape grammar formalism (Stiny, 1980). In this formalism, rewrite-rules are recursively applied to two-dimensional shapes to produce languages of two-dimensional shape designs. Parametric shape grammars can also be defined, in which rule schemas with variable parameters are specified, and rule application proceeds with an instantiated version of one of the rule schemas. Note that a grammar specifies the legal transformations, but, in general, dotes not specify which legal transformation should be applied in any given situation.

Viewed in the abstract, the whole design process can be considered as a generative process. Stiny and March (1981) proposed *design machines,* while Fitzhorn (1989) has proposed a formal computational theory of design. In both of these papers, the design process is modeled as the interaction of constraints and the design context with the grammar rules used

to generate the designs. Research within this framework has taken two complementary paths. Firstly, there has been some effort in building grammatically-based design tools, allowing designers to explore design spaces interactively (Carlson, 1994a; Heisserman & Woodbury, 1994). Secondly, because a grammar specifies all possible design steps, they provide a means to formalize the interpretation of those design steps as designs are in progress (see, for example, Stiny (1981) or Brown *et al.* (1992)), and hence offer a method of feeding back the results of those interpretations into the design process, directing design towards better solutions (Cagan & Mitchell, 1993).

A number of specific shape grammars have been presented in the literature. For example, Stiny and Mitchell (1978) present a grammar of Palladian villas, in which, by designing within the constraints of the production rules, it is possible to generate villa floor plans in the style of Palladio. In similar style, Hemming (1987) presents ? grammar of Queen Anne, houses. Moving away from the issues of style, Brown *et al.* (1993) present a grammar which specifies the language of all axi-symmetric objects manufacturable on a given lathe. Finally, Fitzhorn (1990) presents grammars specifying the languages of constructive solid geometry and boundary representations - that is, languages of realizable solids.

There would appear to be two main benefits of grammatical systems. The first is that they offer a means for automating aspects of the design process, from simply making available the individual transformations in a computer support tool, through interpreting design decisions, to full automation. The second is that specific design grammars focus the generative process on designs with certain characteristics. Thus the first two grammars above constrain designers to producing designs which correspond to a certain style. Brown *et* a/.'s machining grammar, on the other hand, offers no guidance on the style of the finished design, but guarantees an object that can be manufactured. Finally, Fitzhorn's grammar offers no help with either style or manufacturability, but simply generates feasible solids. The use of such grammars clearly involves a trade-off, balancing the effectiveness of the design process for achieving specific goals with the obvious limitations placed on a designer. The notion, however, of specifying all legal steps leads to the argument that there is no scope for creativity when using a design grammar. This last question is addressed in this paper.

Grammars, Control and BdUnded Creativity

In this section we consider the relationship between grammatical design and creativity, and attempt to demonstrate that, far fkom removing creativity from the design process, grammatical design provides the framework for what we **will call *bounded creativity.***

As described above, a grammar is a specification of a set of available primitives and the means to manipulate and combine those primitives during the design process. The grammar is a (finite, implicit) formal specification of a set of designs. Thus it may appear that once a grammar has been selected, the creativity in the design process has been pre-empted, as the designer is now effectively reduced to selecting individual designs from the set specified by the developer of the grammar. However, this is an unnecessarily restrictive view of grammatical design. In any substantive grammar it is unlikely (and perhaps impossible) that the developer has been conscious of all the possible designs in the space it defines. Thus even if the term "creative" was restricted to end products of the design process that are previously unseen in the current context, it would still be arguable that grammatical design can produce creative designs. More significantly, though, the use of a grammar does not entail the use of any particular control strategy, and thus if "creative" is to be applied to the design *process* rather than the artifacts, creativity and grammatical design are still compatible.

In this view, the standard practice of a designer refers to the methods by which he or she carries out the design process, or, in terms of grammatical design, the methods by which the designer searches through the space. Standard practice can be considered to be the knowledge and experience built up over time about particular design problems. Creativity, in this model, happens when a designer transcends the limitations of the known procedures, and attempts, for example, different strategies, different orderings of sub-design instantiation and refinement, the use of different primitives, the use of different combinations of the familiar primitives, and so on.

As an illustration, consider the space, O, of all conceivable objects. The design task is to generate an object in this space that satisfies certain criteria (Y, say). Creativity happens when a designer moves beyond standard methods. In this situation, the only limit to creativity is the designer.

Consider now a grammar which generates a subspace X of O. X may still define an infinite space, and generating a design within X that satisfies Y may still be a difficult task. See, for example, Carlson (1994b) and Cagan (1994). Simply using a grammar has not solved the problem, although it has made it easier by providing a computable method of generating candidate designs. Suppose, however, we have a standard method of controlling search within the design space for X, which, given initial conditions in the form of specific constraints and objectives within Y, produces designs in X that satisfy those objectives. Call that known method K. The design problem is now largely solved, in that following the **dictates of method** K will give us satisfactory designs. Figure 1 illustrates this schematically, showing a single design path using method K. However, there may be many other designs in X which would satisfy our objectives but which cannot be generated by following method K. Similarly, there may be many different ways of generating the same designs as K, but without following that method. Following the method gives a routine solution to the given design problem. A creative solution, on the other hand, would be to depart from K, and generate designs by some new path - a much more difficult proposition. Thus although a grammar is being used, creative design is still possible. Figure 2 shows the same design space as before, but with two creative design paths, one resulting in a creative artifact and one resulting in a known artifact but through a creative process.

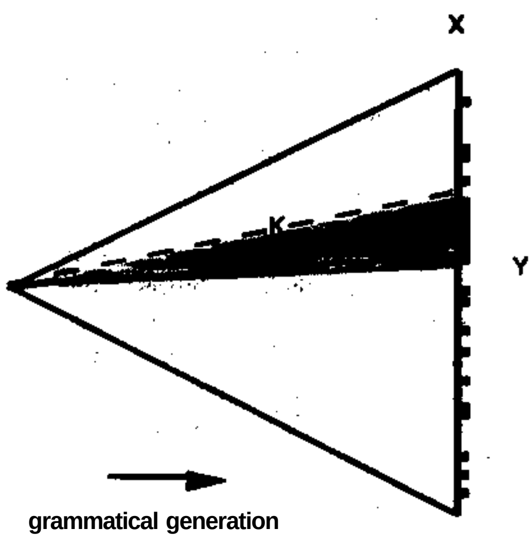

Figure 1. A single known design path

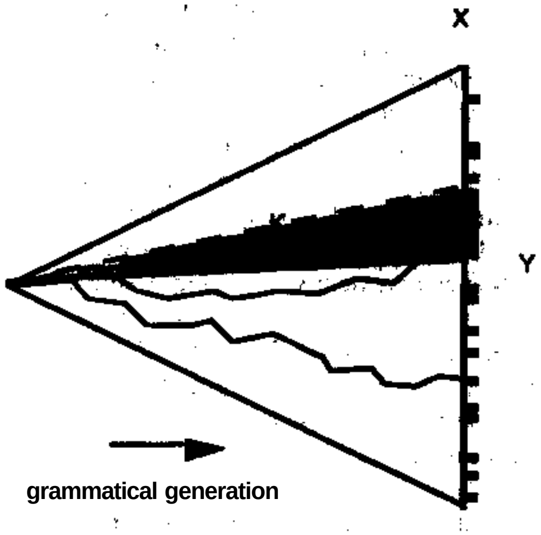

Figure 2. Two creative design paths

Consider now a more restrictive grammar that generates all feasible solutions to the constraints and objectives in Y, and only those solutions. The method K might still be applicable, or, more likely, a variant of K can be applied to this new grammar. Following method K is still not guaranteed to produce all solutions to Y, and thus a divergence from the method still gives a creative solution. However, the design problem is how much easier, because any divergence from K is guaranteed to produce a feasible solution. Figure 3 shows one routine and one creative design path for this design space.

Finally, we might consider a grammar based directly on the method K. This grammar would then generate all solutions and only those solutions that would have been generated by method K in the first grammar. In this case, by our definition, creativity within the grammar is not possible, as the standard control procedure is the only way to generate designs, and obviously then it can generate all designs in the space. In this situation, the use of a grammar and a particular control strategy would have removed creativity from design. Figure 4 shows a design path in this scenario.

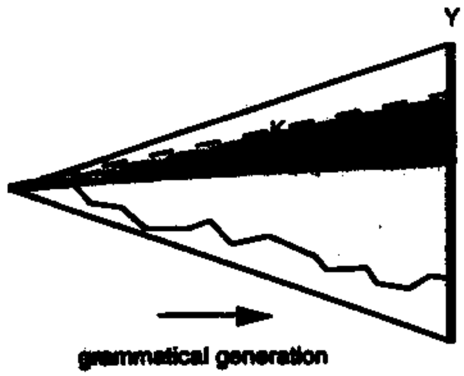

Figure 3* **Space of a grammar that only generates designs of criteria Y.**

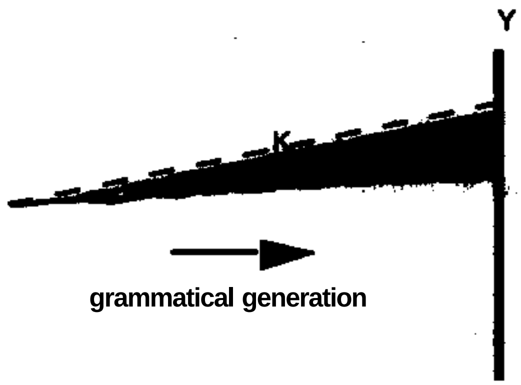

Figure 4. Space of a grammar that only generates designs by known method.

Thus it is our contention that the scope for creativity is not necessarily a function of the formalism used in the design process, but rather a function of the formalism and the control mechanism used on that formalism. We might express it using the following notation. Let C be a control mechanism, and let G be a grammar. Let r(C,G) be the set of all paths obtainable by applying C to G. Let s(G) be the set of all paths through the space defined by G. If s(G) = r(C,G), then there is no scope for creativity in the system (C,G). If s(G) 3 r(C,G), then there is scope for creativity. This leads us to the notion of *bounded creativity*[1], where the control mechanism corresponds to

[1] by analogy to Simon's (1957) definition of bounded rationality.

standard procedures, and the formalism places limits on the creativity possible within the system. Thus:

> <u>Bounded creativity</u> is a property of an agent that behaves in a manner that is within certain limits beyond its standard practice for the current goal:
> if N is a path to a design such that K £ r(C,G), then the agent was creative;
> if N is a path to a design such that N *e* s(G)\r(C,G), then the agent was boundedly creative.

We will regard a grammar and a control mechanism together as forming a design system. Generally, as we broaden the range of the known procedures or we narrow the scope of the grammar (i.e., as r(C,G) and s(G) converge) the scope for creativity decreases.

Machining Example

To illustrate these arguments, consider the design of an aircraft hinge component as laid out in Ullman (1992). The spatial requirements for this component and its interface points are shown in Figure 5. Ullman describes four possible solutions to this problem as shown in Figure 6. Consider the design of this hinge described by a solids grammar:

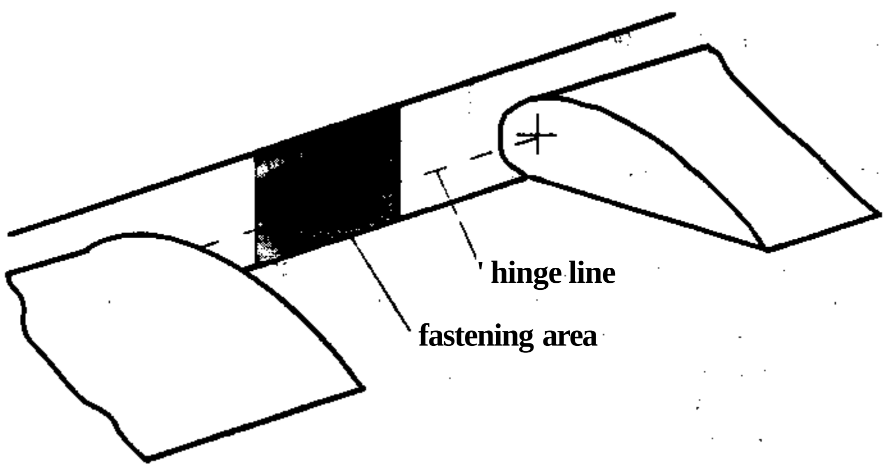

Figure 5. Spatial requirements

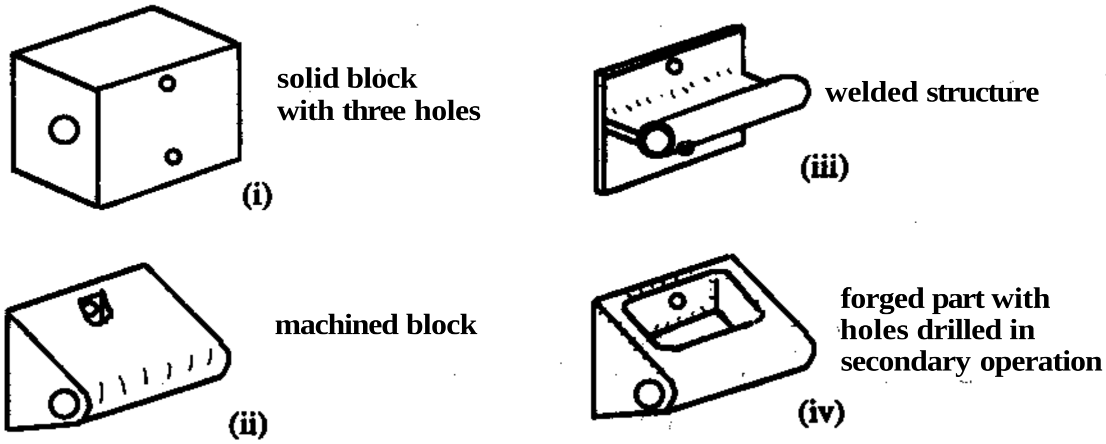

Figure 6. Possible solutions

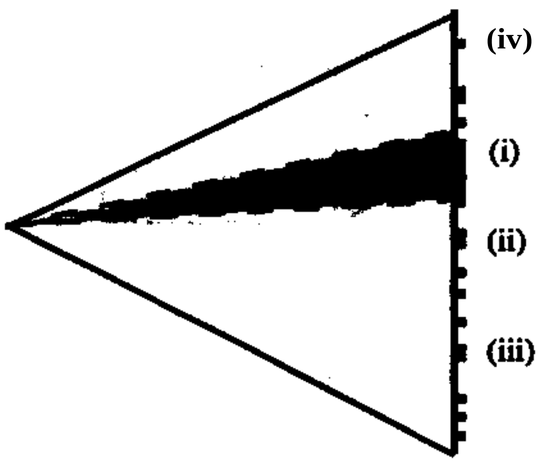

Figure 7. Space of all design solutions for this problem.

Consider a control procedure that constructs a hinge as follows with the solids grammar:

> **Given fastening area, minimum thickness, locations for 2 bolts, bolt sizes;**
> **1) Construct solid Mock (B) within size constraints (Figure 8a).**
> **2) Construct solid cylinders of bolt size (Cl and C2) (Figure 8a).**
> **3) Construct Solid cylinder of hmge pin size C3 (Figure 8a)**
> **4) Position B with cottier datum at coordinate origin (Figure 8b).**
> **5) Position centers of Cl ami C2 relative to datum ongin (Figure 8b).**

6) Position center of C3 offset from datum to hinge line location (Figure 8b)
7) Perform Boolean subtraction of Cl, C2 and C3 from fi (Figure 8c).

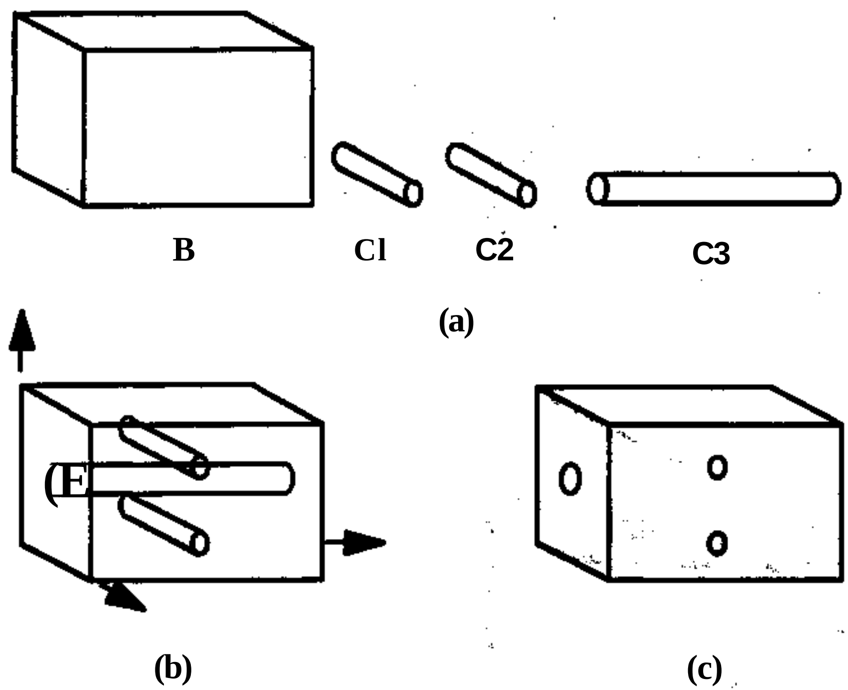

Figure 8. Known solution to **problem.**

This procedure generates blocks with holes like design (i), and generates a variety of designs with this characteristic. This is indicated by the shaded triangle in Figure 7. One could imagine other creative processes to generate the same configurations. For example, consider the solid elements in Figure 9. Boolean operations on these elements could also generate design (i), although not very efficiently. One could also imagine other creative procedures to generate designs (ii) - (iv) or other functional hinges. Some of these designs could be realizable solids yet not manufacturabie with traditional methods (e.g., a light weight design of a block with bolt holes and a sphere removed from the inside; see Figure 10)[2]. Some of these designs are manufacturabie, but not only from machining operations, as with designs

[2]Actually, with layered manufacturing techniques this is manufacturabie; however, not with traditional methods.

(iii) and (iv). As seen from Figure 7, all functional hinges are indicated by the black boxes, although some are not machinable. In addition there are some designs generated by a solids grammar that are machinable, but not functional (such as a solid block with no holes)* Time are not valid designs (ami are not elements within the black boxes) but are valid paths in the space of the solids grammar.

Figure 9. Solid elements that can create known solution.

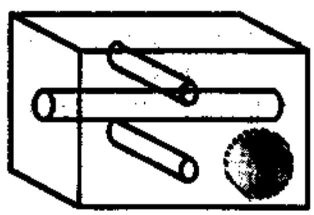

Figure 10. Solution generated from a solids grammar, yet non-manufacturable (with traditional methods).

Next consider a general three-axis machining grammar analogous to Brown *et aL's* (1993) axi-symmetric grammar. This grammar generates all and only those designs manufacturable by machining operations as shown in Figure 11. We can apply a similar control procedure to the one laid out above to design hinges like type (i) to this grammar generating the designs of the shaded triangle. A creative designer is still able to go outside of the standard procedure to generate hinge (ii) or other creative solutions. We can still generate machinable but not functional designs such as a block with no holes. However, only machinable designs can be generated (the block with the sphere removed is no longer valid). Thus the functional, yet not machinable, solutions (iii) and (iv) are no longer valid. The size of the design space is reduced, and a limit placed on creativity, yet we can still generate creative designs within the language of the grammar.

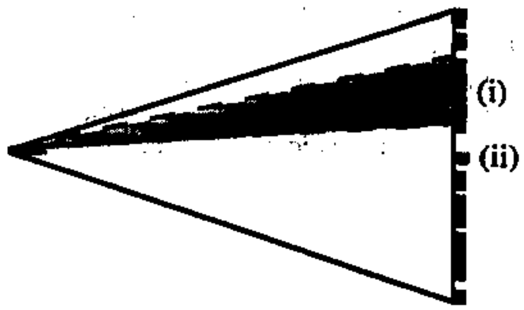

Figure 11. Space generated from general three-axis machining grammar.

Now consider a grammar that specifies only machinable hinges, Again the space of designs is limited (the solid block is no longer valid since it is not functional). Any design generated from the grammar is acceptable. We can still apply the same control procedure to produce solid blocks with 3 holes that are functional. However we can also still generate an infinite number of designs that fall outside of that procedure and can still generate creative designs (Figure 12). Note that we could have constrained the solids grammar to be functional and then machinable, resulting in the same space.

Figure 12. Space generated from grammar for machinable hinges.

Figure 13. Space of a grammar that only generates hinges by known method.

Finally, consider a grammar that only generates solid blocks with 3 holes that function as a hinge, effectively encapsulating our original control procedure. There is now no way to go outside of the procedure and remain valid within the grammar, and thus there is no scope for creativity within the system (Figure 13). It is only in this situation (hat the grammar removes the possibility of being creative in the design of hinges. Of course the evaluation of die hinges could change ami these blocks could find a different creative functional purpose (such as to be door stops); however that is outside the scope of our arguments.

Relation to Other Models of Creativity

We have argued that the use of grammars does not remove creativity, but rather supports a bounded creative process. There are a variety of definitions of creativity in design, most of which are consistent with our definition of bounded creativity and grammars. We consider some of these definitions below.

Fischer (1993)[3] uses Hayes[1] (1978) definition of creativity as consisting of acts which are "novel or surprising". This is consistent with our definition of diverging from the standard procedure. He uses a system for the design of pinball machines as an example of a restricted set of objects whose combination (and there are many) can be creative. He suggests that the use of the restricted set (thereby putting bounds on the set of moves) enhances the effectiveness of the design process. He also discusses the problem of these constructions not being limited to interesting or useful artifacts; we state that a more restrictive grammar could provide such a filter, and still permit creative solutions.

Coyne and Subrahmanian (1993) state that creative solutions occur when "a solution is found within a given formulation in a region of the space of design solutions never examined before". This again corresponds to a designer diverging from the standard procedure.

Cagan and Agogino (1993) focus on the designed artifact; however, in their introduction they state that creative design processes can produce innovative or routine designs. The model presented in this paper illustrates how this might happen; a design path may be non-standard but may produce a routine design (see Figure 2). Navinchandra (1992a;b) views innovation as being divergence from the common practices of the "design culture". Many of these earlier definitions of design classifications differentiated between routine, innovative, or creative designs (Brown and Chandrasekaran, 1989; Rosenman and Gero; 1993; Prabhakar and Goel, 1992); in our model we

[3]Many of the references in this section were originally presented in 1989 at the *1st International Round Table Conference on Computational Models of Creative Design,* but were published in (Gero and Maher, 1993).

differentiate only between routine and non-routine or creative. Rosenman and Gero describe results from a grammar as being innovative and not creative; however, because these Innovative* designs fall outside of routine practice, we define them to be creative.

Meyer *et al.* (1992), while addressing the use of grammars in design, state that "creative design involves the generation of novel objects[11]; and that "there is knowledge that cannot be formalized as a grammar, such as new organizations of elements that may lead to a creative design solution"; they use analogy and mutation to change the design space. We argue that the scope for creativity lies in the process as well as just the space. Woodbury (1993) also discusses the use of mutation in grammatical design for creativity, proposing methods of mutating the rules of the grammar itself to achieve new spaces. Mitchell (1993) presents the notion of creativity stemming from a continually changing set of grammatical rules. The idea of creativity arising from a change in the grammar rule set is not in conflict with our definition of creativity as these changes are one way to go outside the standard practice; however, a change in the grammar rule set is not required.

Finally, Faltings (1992) gives a definition of creativity as "the act of creating a problem solution which is not computable from the agent's design or domain knowledge". We find this definition too restrictive because it would rule out all creativity from grammatical systems. We have suggested that within a solid modeling grammar system people can be and are creative despite the restricted system.

Conclusions

If we are ever to achieve effective automated design tools and design support systems, computational design tools will have to be formal and concise. Grammatical representations of the design process are, by definition, formal, concise and computable. It has been argued thai such representations must rule out creativity. We have shown, contrary to this argument, that grammars support a form of creativity, which we have called bounded creativity. Bounded creativity captures many instances of human design processes that are conventionally called creative.

Acknowledgments

The authors would like to acknowledge the National Science Foundation, under grants DDM-9300196 and DDM-9258090, and the Engineering

Design Research Center, an NSF sponsored center at Carnegie Mellon University, for supporting this work.

References

Brown, K. N., Sims Williams, J. H. and McMahon, C. A. (1992) "Grammars of Features in Design", ***Artificial Intelligence in Design V2*** **(J. S. Gero,** ***e*L*** **Pittsburgh, Kluwer Academic Publishers, The Netherlands, pp. 287 - 306.**

Brown, K. N., McMahon, C. A. and Sims Williams, J. H. (1993) "A Formal Language for the Design of Maauftcturable Objects", in ***Formal Design Methods for CAD (B-18),*** **(J. S. Gcro & B. Tyugu, eds.), North Holland, pp. 135-155.**

Brown, D. C. and Chandrasckaran, B. (1989) ***Design Problem Solving: Knowledge Structures and Control Strategies,*** **Pitman, London.**

Cagan, J. and Agogino, A. M. (1993) 'Inducing OptimaUy Directed Non-Routine Designs", in ***Modeling Creativity and Knowledge-Based Creative Design*** **(J. S. Gero, and M. L. Maher, eds.), Lawrence Eribaum Associates, Hillsdale, NJ., pp. 273-293.**

Cagan, J. and Mitchell, W. J. (1993) "Optimally Directed Shape Generation by Shape Annealing", ***Environment and Planning*** **fi, Vol. 20, pp. 5-12.**

Cagan, J. (1994) "Research Issues in the Application of Design Grammars", in ***Formal Design Methods for CAD (B-18K*** **(J. S. Gero & E. Tyugu, eds.), North Holland, pp. 191-198.**

Carlson, C. (1994a) "A Tutorial Introduction to Grammatical Programming", in ***Formal Design Methods for CAD (B-18),*** **(J. S. Goo & E. Tyugu, eds.), North Holland* pp. 73-84.**

Carlson, C. (1994b) "Design Space Description Formalisms", in ***Formal Design Methods for CAD (B-18),*** **(J. S. Gero & E. Tyugu, eds.), North Holland, pp. 121-131.**

Chomsky, N. (19S6) "Three Models For The Description Of Language", ***IRE Transactions on Information Theory,*** **IT-2, No. 3, pp. 113-124 [Reprinted in** ***Readings in Mathematical Psychology*** **Volume 2, (R. D. Luce, R. R. Bush & E. Galanter, eds.) pp. 105-124, John Wiley and Sons, New York, 1965].**

Coyne, R. F. and Subrahmanian, E. (1993) "Computer Supported Creative Design: A Pragmatic Approach", in ***Modeling Creativity and Knowledge-Based Creative Design*** **(J. S. Gero and M. L. Maher, eds.), Lawrence Eribaum Associates, Hillsdale, N.J., pp. 295-328.**

Faltings, B. (1992) "Supporting Creativity in Symbolic Computation", ***Preprints 2nd International Round-Table Conference, Computational Models of Creative Design, 6-10 December, Heron Island, Queensland,*** **pp. 191-205.**

Fitzhorn, P. A. (1989) "A Computational Theory of Design", ***Preprints NSF Engineering Design Research Conference,*** **College of Engineering, University of Massachusetts at Amhcrst.**

Fitzhorn, P. A. (1990) "Formal Graph Languages of Shape", ***(AI EDAM) Artificial Intelligence for Engineering Design, Analysis and Manufacturing,*** **Vol. 4, No. 3, pp. 151-164.**

Flemming, U. (1987) "Mem Than The Sum Of Their Parts: The Grammar Of Queen Anne Houses", ***Environment and Planning B,*** **Vol. 14, pp. 323-350.**

Fischer, G. (1993) "Creativity Enhancing Design Environments", in *Modeling Creativity and Knowledge-Based Creative Design* (J. S. Gero and M, L, Maher, eds.), Lawrence Erlbaum Associates, Hillsdale, N. J., pp. 235-258.

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

Hayes, J. R. (1978) *Cognitive Psychology - Thinking and Creative*, Dorsey Press, Homewood, Illinois.

Heisserman, J. and Woodbury, R. (1993) "Geometric Design with Boundary Solid Grammars", in *Formal Design Methods for CAD (B-18),(J.S.* Gero & E. Tyugu, eds.), North Holland, pp, 85-105.

Meyer, S., Zhao, F. and Fenves, S. J. (1992) "A Representation for Integrating Different Models of Creative Design", *Preprints 2nd International Round-Table Conference, Computational Models of Creative Design, 6-10 December, Hewn Island, Queensland*, pp. 207-241.

Mitchell, W. J. (1993) "A Computational View of Design Creativity", in *Modeling Creativity and Knowledge-Based Creative Design* (J. S. Gero and M. L. Maher, eds.), Lawrence Erlbaum Associates, Hillsdale, NJ., pp. 25-42.

Rosenman, M. A. and Gero, J. S. (1993) "Creativity in Design Using a Prototype Approach", in *Modeling Creativity and Knowledge-Based Creative Design* (J. S. Gero and M. L. Maher, eds.), Lawrence Erlbaum Associates, Hillsdale, N.J., pp. 111-138.

Navinchandra, D. (1992a) "Innovative design systems, where are we and where do we go from here? I. Design by Association", *Knowledge Engineering Review,* vol. 7, no. 3, September, pp. 183-213.

Navinchandra, D. (1992b) "Innovative design systems, where are we and where do we go from here? II. Design by Exploration", *Knowledge Engineering Review,* vol. 7, no. 4, December, pp. 345-62.

Prabhakar, S. and Goel, A. (1992) "Performance-Driven Creativity in Design: Constraint Discovery, Model Revision, and Case Composition", *Preprints 2nd International Round-Table Conference, Computational Models of Creative Design, 6-10 December, Heron Island, Queensland*, pp. 101-127.

Simon, H. A. (1957) *Models of Man* , Wiley, New York.

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

Stiny, G. (1981) "A Note on the Description of Designs", *Environment and Planning B,* Vol. 8, pp. 257-267.

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

Stiny, G. and Mitchell, W. J. (1978) "The Palladian Grammar", *Environment and Planning B,* Vol. 5, pp. 5-18.

Ullman, D. G. (1992) *The Mechanical Design Process,* McGraw Hill, New York, pp. 203-205.

Woodbury, R. F. (1993) "Design Genes", in *Modeling Creativity and Knowledge-Based Creative Design* (J. S. Gero and M. L. Maher, eds.), Lawrence Erlbaum Associates, Hillsdale, N.J., pp. 211-232.