Routine Creativity?

David C. Brown
Computer Science Department, Worcester Polytechnic Institute, USA

1. Introduction

It is common wisdom that people should be given tasks that computers can't do well, and computers should be given tasks that people can't do well. So in design computing why are we attempting to study computational design creativity?

The main answer is that the field (like many others) progresses by tackling simpler problems first and moving towards harder ones. Routine parametric design and design checking were starting points, moving gradually to Configuration and most recently to harder problems such as distributed/collaborative design and to creative design: moving from routine to non-routine [Brown 1996]. One goal has always been to build working systems, while another is to learn more about the knowledge and reasoning used for each type of design activity studied.

Computational design creativity is hard to study, and until fairly recently it has received very little attention, even though it is widely held to be very important both from intellectual and economic points of view. It has mostly been studied by looking at analogical reasoning and genetic algorithms: i.e., the focus has been on extreme non-routine cases. There are hard sub-problems and other ways of moving towards creative systems that are worth considering. This paper suggests an alternative approach to computational design creativity research that might be labeled "routine creativity".

2. Current approaches

The author believes that most current computational creativity research tends to be based on *the goal of transformational creativity* [Boden 1994]. A lot of it appears to be based on or influenced by larger scale, general reasoning, such as Analogy (for example see [Yaner & Goel 2008]), Genetic/Evolutionary Algorithms (for example see [Koza 2008]), and Conceptual Blending [Turner & Fauconnier 1995] (for example see [Nagai et al. 2009]).

The *consequences* of this goal are that:

1) researchers tackle a very hard problem "head on", making slow progress;
2) these powerful methods don't give a clear idea of what their limits are—knowing what a method *can't* do is as important as knowing what it can do;
3) the computational methods used don't always match what people can do, and therefore don't provide very good hypotheses about human creativity; and
4) detailed psychologically based hypotheses about what people might be doing tend to be ignored.

3. Research Alternatives

One approach is to take a well understood but not intentionally creative approach and see how it might be modified in order to produce results that people would be willing to say are creative, due to their novelty and other characteristics [Besemer 2006]. A secondary goal would be to determine whether the post-modification mechanisms could meet the criteria for transformational creativity.

This alternative addresses the four "consequences" given above. Although this too may produce slow progress, #1 is addressed by working on several smaller problems to examine the impact of their solutions. By using a routine design (RD) problem-solving method (PSM), #2 is addressed as we know the limits quite well. By picking a RD PSM which is already based on expert behavior we stand a better chance of addressing #3, and #4 can be addressed by focusing on using modifications based on hypotheses about the ingredients of creative reasoning that can be found in the psychological literature (see [Brown 2008]).

Of course, from this author's point of view, an obvious candidate for this alternative is to look at what changes can be made to Design Specialists and Plans Language (DSPL) based routine design systems [Brown 1996b] [Brown & Chandrasekaran 1989] in order to produce more creative outputs. However, this isn't the only candidate.

This approach focuses on working "upwards" towards creativity, by examining smaller, ingredient decisions that make a difference to the result. It should be possible to investigate the degree of impact produced by changing the internal reasoning mechanisms in a DSPL system. This will contribute to our understanding of which less extreme reasoning mechanisms impact judgments of increased creativity. This is the foundation on which more extreme methods rest, as many authors agree that creativity is "…an outcome of subsets of … processes acting in concert…" and not just a single reasoning mechanism [Ward et al. 1999].

3.1. Using Cognitive Science and Psychology

A second alternative is to look more carefully at what cognitive science and psychology tells us about creativity. Everyone agrees that "novelty" is a key ingredient of the production and evaluation of creativity in a designed product, while some others add "surprise". Novelty appears to be the principle component of *all* models of creativity, and all creativity metrics. Judging both originality and surprise appears to be quite difficult, and needs much more attention. Srinivasan & Chakrabarti [2010], as well as others, have already made useful contributions to this problem.

Suggestions about the many ingredients of creative reasoning and its evaluation from the literature include:

A. *Novelty:* surprising and original; recognizing, evaluating and seeking it.

B. *Domain Knowledge:* having lots of it; being able to search it; finding relevant knowledge; rich interconnections; different representations; knowledge of its potential; similarities and differences; not just hierarchical representations.

C. *Heuristic knowledge:* having lots of it; for selecting ways to think (such as planning, simplification, analogy, etc.)

D. *Constraints:* being able to drop, weaken or invert them; having meta-knowledge about them to enable their modification.

E. *Combinations:* novel combinations of old ideas; combination of apparently unrelated ideas.

F. *Associative reasoning:* a quality of over inclusiveness; ability to associate the apparently unrelated.

G. *Suppressing inhibitions:* allows less relevant ideas/methods to “intrude” into the problem solving process.

H. *Abstract and imprecise descriptions:* such as for intermediate solutions and goals.

I. *Alternative methods:* for making decisions; for making goals more concrete.

J. *Critical assessment:* as an antidote to inclusiveness; identify misfits; heuristically eliminating very weak ideas and potential mistakes; resist pruning too strongly to just the routine ideas; resist too much novelty.

K. *Problem recognition:* error detection; recognition of product inadequacies; recognition leads to formulation of new goals.

L. *Concept expansion:* constructing, stretching, extending, modifying and refining concepts.

M. *Analogical reasoning:* far (cross domain) and near (same domain); depends on intentions and goals.

N. *Visualization:* mental simulation to examine existing things in new situations.

O. *Meta-reasoning:* breaking away from functional fixedness; abandoning old, unsuccessful problem-solving strategies; using meta-knowledge.

P. *Least commitment:* keeping options open as long as possible; suspending judgment; producing multiple partial solutions.

Q. *Forgetting:* productive forgetting; good mental management.

4. Ingredients of Routine Design Reasoning

We will concentrate on taking a well understood but not intentionally creative approach to see how it might be modified in order to produce results that people would be willing to say are creative.

Routine design means that everything about the design process is known in advance, including the knowledge needed. However, neither the resulting design nor the trace of use of the knowledge is known in advance. Typically, routine design knowledge is highly compiled: in the “knowledge compilation” sense of the term [Goel et al. 1991]. The DSPL language allows such routine design knowledge to be written down.

As previously presented [Brown 1992], the ingredient types of reasoning supported by DSPL are: Basic Synthesis; Criticism; Decomposition; Evaluation; Execution; Ordering; Patching; Planning; Recomposition; Retraction; Selection; Situation Recognition; Suggestion Making.

Note that they are not independent, as some of these items involve other items, and are therefore at a different level of abstraction. The connection between this list and the mechanisms of DSPL are summarized in Table 1 below. We use the "ingredients" list terms from above, but acknowledge that some have meanings that vary in the literature: e.g., "Synthesis" can also mean combining or generating, instead of calculating or selecting, hence we use the modifier "Basic".

In DSPL, each Specialist contains Plans and plan selection knowledge. They each represent a subproblem, solving it by plan selection and execution. Plans are precompiled, ordered sequences of actions intended to provide the design for a subproblem. Each Plan provides a decomposition as well as sub-solution recomposition. Sponsors evaluate the suitability of a Specialist's plans for use in a particular situation, while a Selector picks the most suitable Plan.

Steps are the building blocks of the design process, providing a value for an attribute of the design by calculation, or by selection using pattern matching. Tasks group Steps, and therefore define additional problem decomposition. Constraints test values and, on failure, make suggestions about patches. Redesigners attempt to patch the design, guided by suggestions, in order to correct a constraint failure. Failure Handlers (FHs) recognize failing situations that might be patchable, or can trigger suggestion-guided backtracking.

Table 1: The Ingredients of Routine Design Reasoning

Type	*DSPL*	*Action*
Basic Synthesis	Step	Calculate, or select.
Criticism	Constraint	Values are tested/compared.
Decomposition	Plan Task Step	All three have sequences of actions.
Evaluation	Sponsor	Determine the quality of a plan.
Execution	Plan execution	Carry out the actions in a plan.
Ordering	Plan Task Step	All three have ordered actions.
Patching	Redesigner	Can change an attribute's existing value.
Planning	Plan Plan Sponsor Plan Selector	Hierarchically arranged collections of plans with plan selection produce a dynamically constructed design plan.
Recomposition	Plan	Each plan action adds its subproblem's solution to the overall design.
Retraction	Backtracking	One or more recent design decisions can be retracted and a re-design phase entered.
Selection	Plan Selector Step	The selector selects from amongst suitable plans, while a step selects from amongst suitable values for an attribute.
Situation Recognition	Plan Sponsor Step FHs	All three can make context sensitive decisions, based on recognizing patterns of previous actions or design decisions.
Suggestion Making	Suggestion	If any "agent" (e.g., a Constraint, or a Step) used by another fails, it passes suggestions (about how the failure might be fixed) back to the agent that called it from 'above'.

5. Modifications to Routine Design Reasoning

We restrict possibilities by assuming that modifications are made without creating new agents (i.e., no additional reasoners are added), but that new mechanisms are allowed to be ‘called’ or added for exploiting meta-knowledge or meta-reasoning.

We assume that modifications are based on an RD knowledge-base constructed from DSPL, or something similar. We assume that the base system is doing configuration by selection between alternative pre-determined configurations.

As such an RD system is probably highly compiled, values will be constrained early to avoid failure later in the design process. The RD system could be considered to be very *tight* or *loose*, depending on how much earlier constraints restrict later decisions. One would expect tighter systems to be harder to modify to produce more creative results. This will require further study.

5.1. Matching Creative and Routine Reasoning

In Table 2, the rows show the suggestions (A-Q) about creative reasoning from the literature, while the columns show the 13 ingredients of routine design reasoning. The table entries indicate places where relevant modifications might occur: others might be possible.

Table 2: Some possibilities for modifications

	Synth	**Crit**	Decomp	Eval	Exec	Order	Patch	Plan	Recomp	Retr	Sel	Recog	Sugg
Novelty	**y**	**y**		y			y				y	y	y
Domain	**y**	**y**	y	y			y	y			y	y	y
Heuristic		**y**	y	y			y	y			y	y	y
Constr.		**y**		y									
Combin.							y	y	y		y	y	
Assoc.		**y**						y	y		y	y	
Suppress		**y**	y	y				y	y	y	y	y	y
Abstract	**y**	**y**						y			y	y	y
Alt.	**y**		y			y	y	y			y	y	y
Assess		**y**		y				y			y	y	
Recog.		**y**					y				y	y	y
Expand									y				
Analogy	**y**		y			y	y	y					
Visualiz.		**y**											
Meta.	**y**	**y**	y			y	y	y			y	y	y
Least C.	**y**		y		y								
Forget		**y**											

Note that the first two columns (marked in bold) were considered in more detail. Investigating the other 187 possibilities is more challenging and would require significantly more study. However, it is striking to see how many there are. The entries made in Table 2 were first done by considering each ingredient of routine reasoning in turn against all 17 of the creative reasoning suggestions, and then by considering it again in the opposite direction (i.e., for each of the suggestions, against all 13 of the ingredients).

6. Summary & Conclusions

The fine-grained analysis proposed by this paper is almost the antithesis of normal computational creativity research, in which the “blue skies” methodology is adopted: that is, to put it crudely, it isn’t any good unless it appears impossible. That approach tends to move researchers past more “mundane” problems, leaving them to be tackled later, with less prestige, or even left undone.

Creative systems may well be fuelled by important, large scale reasoning methods, such as analogy, that try to address the goal of transformational creativity, but they will be supported and enhanced by smaller scale reasoning such as has been presented here. It is important to note how many opportunities there are for potentially interesting research into creative design systems given this ‘humble’ Routine Design basis.

Note: *This workshop paper is an abbreviated version of the paper accepted for the main DCC’10 conference.*

References

Besemer S.P. (2006) *Creating Products in the Age of Design*, New Forums Press, Inc.

Boden M.A. (1994) What is Creativity?, in M.A. Boden (ed.), *Dimensions of Creativity*, The MIT Press, pp. 75-117.

Brown D.C. (1992) The Reusability of DSPL Systems, Workshop on Reusable Design Systems. *Second International Conference on AI in Design*, Carnegie Mellon University, Pittsburgh, PA.

Brown D.C. (1996) Routineness Revisited, in M. Waldron and K. Waldron (eds.), *Mechanical Design: Theory and Methodology*, Springer-Verlag, pp. 195-208.

Brown D.C. (1996b) DSPL: Design Specialists and Plans Language, 1983-1996. *web.cs.wpi.edu/Research/aidg/DSPL.html*

Brown D.C. (2008) Guiding Computational Design Creativity Research, to appear in *Studying Design Creativity*, J.S. Gero (ed.), Springer. (Web page available at *web.cs.wpi.edu/~dcb/Papers/sdc08-paper-Brown-25-Feb.pdf*)

Brown D.C. & Chandrasekaran B. (1989) *Design Problem Solving: Knowledge Structures and Control Strategies*. Research Notes in Artificial Intelligence Series, Pitman Publishing, Ltd., London, England.

Goel A.K., Bylander T., Chandrasekaran B., Dietterich T.G., Keller R.M., Tong C. (1991) Knowledge Compilation: A Symposium, *IEEE Expert*, 6(2), pp. 71-93.

Koza J.R. (2008) Human-Competitive Machine Invention by Means of Genetic Programming, in special issue on Genetic Programming for Human-Competitive Designs, L. Spector (ed.), *AI in Engineering, Design, Analysis and Manufacturing*, Cambridge University Press, 22(3), pp. 185-193.

Nagai Y, Taura T. & Mukai F. (2009) Concept blending and dissimilarity: factors for creative concept generation process, *Design Studies*, 30(6), pp. 848-675.

Srinivasan V. & Chakrabarti A. (2010) Investigating Novelty-Outcome Relationship in Engineering Design, in special issue on Creativity: Simulation, Stimulation, and Studies, M.L. Maher, N. Bonnardel & Y.-S. Kim (eds.) *AI in Engineering, Design, Analysis and Manufacturing*, Cambridge University Press, 24(2).

Turner M. & Fauconnier G. (1995) Conceptual integration and formal expression, *Metaphor and Symbolic Activity*, 10(3), pp. 183-204.

Ward, T.B., Smith, S.M., and Finke, R.A. (1999) Creative Cognition, in R.J. Sternberg (ed.), *Handbook of Creativity*, Cambridge University Press.

Yaner P. & Goel A. (2008) From Design Drawings to Structural Models by Compositional Analogy, *AI in Engineering Design, Analysis and Manufacturing*, 22(2), pp. 117-128.