# COMPUTATIONAL SUPPORT FOR P-CREATIVE SOLUTIONS

DARYL H. HEPTING
*Department of Computer Science*
*University of Regina*
*Regina, Saskatchewan, S4S 0A2*
*CANADA*
*e-mail: dhh@cs.uregina.ca*

**Abstract.** One can distinguish ideas that are historically creative (H-creative) and novel to society, and those that are personally or psychologically creative (P-creative) and novel only to their creator. Many current computational approaches to support creativity and innovation have focused on H-creativity. However, P-creativity is more important to individuals and potentially more valuable to society. In the end, the society whose citizens are supported in finding P-creative solutions is more likely to see H-creative solutions from those citizens. This paper discusses the design of a software system that is focused on the support of P-creative solutions and the democratization of creative thinking. The system allows an individual to work on a problem in his or her own terms of reference, by constructing a conceptual space that the system then populates. The user interface provides each individual many options for organizing this conceptual space in meaningful ways, without imposing a heavy cognitive burden. Four sample implementations are presented and the results of several usability studies are summarized.

## 1. Introduction

The assessment of creativity for a particular design or solution can be a difficult task. Although Boden (1995) makes the distinction between H-creativity (historical) and P-creativity (personal or psychological), the designer is likely to first be concerned with whether his or her solution satisfies the expressed need. Some of these solutions, according to Gero (1995), will be exceptional while others will be routine. Shneiderman (2000), with his *genex* framework, would direct computational support towards creation of the exceptional, and possibly H-creative, designs.

However, creativity at the psychological level is very important because one invents things for oneself. Wessel and Wright (2001) echo this sentiment with respect musical synthesis: a low entry fee with no ceiling on virtuousity.

Ideally, computer support for design and problem-solving will minimize the gulfs of execution and evaluation and provide users with the ability to interact directly with their designs in a representational mode that is most appropriate (Paivio, 1986). In the second option, the user will strive to become an expert in the software tool. However, it may be difficult to maintain expertise in all the facets of the software's operation that are required to realize solution alternatives. And, immersion in the command syntax of a tool may affect one's view of the semantic aspects of the problem.

A powerful tool is required to support the user both in creating an acceptable solution alternative and in its modification, which is related to one's ability to evaluate and compare different solution alternatives. Regardless of the expertise one has with respect to the syntax of the tool or to the semantics of the problem, generating solution alternatives can be difficult. As Bertin (1983) indicated, "to construct 100 DIFFERENT FIGURES from the same information requires less imagination than patience."

This paper describes an augmented system, called *cogito*[1], which enables combinatory play for users to explore and navigate complex (design) problem spaces.

## 2. Background

This work draws from several diverse sources for its foundation. This section first presents the background that inspired the *cogito* system and then reviews other work done in support of creative designs. A variety of application areas considered and illustrated with examples.

Figure 1 shows a sample visual representation where the parameters include plot-type, colour, data, and sorting. A user may understand each value for each parameter, yet he or she has likely been unable to assess each combination, each point in the conceptual space. P-creative solutions may exist within these unexplored alternatives. Although it is feasible to identify a parameter and change its value in order to change the design, one must know the permissible values and keep a record of

---

[1] The name of the system, *cogito* is taken from the Latin verb "to think", which etymologically means "to shake together". This is done to acknowledge the role of the combination of ideas in various models of human inventive thought.

which have been used in which combinations, if the activity is to be something more than random.

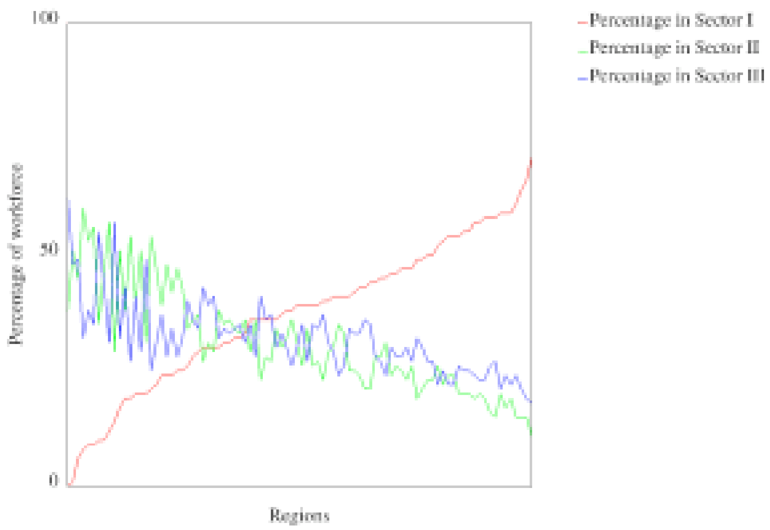



***Figure 1:*** ***A sample visual representation design that illustrates the description of each design as the combination of values, one from each parameter.***

When potential solutions are realized, as in Figure 1, they are interpreted and evaluated for effectiveness. Although a pattern may be clearly visible here, there are many questions that need to be answered before deciding if this solution is optimal for any particular application. Bertin (1983) contends that meaning can be communicated fully through the graphic and its legend. The more widely accepted view is that communication and interpretation occur, or are influenced by things, outside this realm. Casner (1991) argues that one must always consider the task when assessing any visual representation. Furthermore, it is important to understand that each viewer is unique in terms of the experience brought to the interpretation of any visual representation. Although some have suggested the use of "efficiency" as a means for choosing a visual representation, it cannot be generalized beyond a single user.

Many augmented systems require an initial, at least partial, specification of a design. Yet, the initial specification can be difficult if one is unfamiliar with the range of available options. Local searches using gradient-based methods are useful to refine a particular design, but they may trap the designer near a local maximum. Genetic algorithms are very effective at searching an entire space in a global fashion, and they permit what Gero (1995) has called *emergence* (of unexpected designs):

their combination of pieces from different known solutions may lead to something unknown. Interactive, computer-aided, searching of a large space has been available since Dawkins (1986) released *The Blind Watchmaker*. Sims (1991) presented a method for the use of artificial evolution in computer graphics that employed both a genetic algorithm (Goldberg, 1989) and genetic programming (Koza, 1992). Both of these "genetic" methods work by simulating the cellular-level processes of cross-over and mutation. The former does this as a means to search a space whereas the latter works to transform the space itself. For Sims, the goal was to evolve images and textures. However, because it is surprising to see images from different generations with no apparent connection between them, it can work to defeat the user's control. Todd and Latham (1992) also discussed a genetic approach to the generation of new images, theirs being more restrictive by not including genetic programming. Boden (1996) argues that the latter is more relevant to artistic creativity because it allows a more deliberate and disciplined exploration of alternatives. Shneiderman (2000) also focuses on the evolutionary creativity found in an iterative process, which he calls *genex* (generator of excellence).

Vass et al. (2002) study creativity by constructing a problem-solving environment (PSE) that supports flow ["being completely involved in an activity for its own sake"] (Csikszentmihalyi, 1996), and evaluating PSEs on that basis. The design of the *cogito* system, described next, was influenced by the same inquiry about what things support creativity.

## 3. Tool Design

The human-computer symbiosis first proposed by Licklider (1960) had the goals of bringing the computer into the "formulative parts of technical problems" which might be too difficult for humans alone; and to create a direct link between human and computer which would enable the computer's involvement in "real time" situations. The design of this software is based on five principles, described in this section, that are all facilitated by the parameteric representation of conceptual space. There is agreement between these principles, the activities that support the different generator-of-excellence phases (Shneiderman, 2000), and other problem-solving models. Ultimately, this work is concerned with empowering the user to work with problems in their most appropriate representational mode. Broadly, this means working visually with data plots. In terms of textual interfaces, though, it means choosing and naming parameters (or dimensions) of a conceptual space in the most appropriate way for individual users (for example, in the description of recipes or cleaning products with environmental impacts).

SUPPORT THE DESIGNER IN INTERACTING DIRECTLY WITH SOLUTIONS

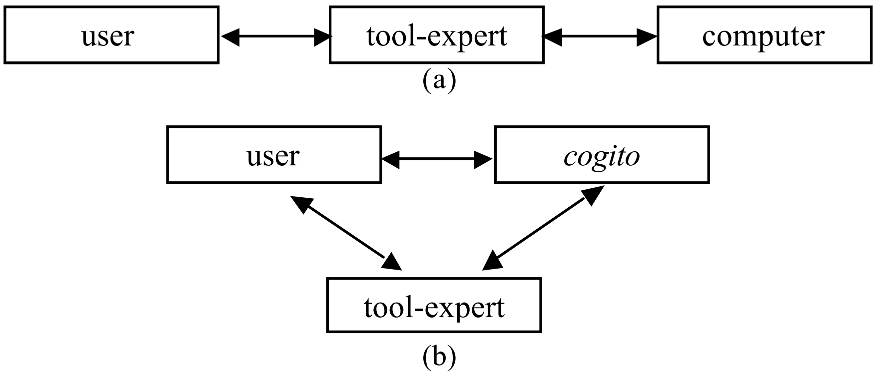



***Figure 2*: In (a), the user's access to the computer is mediated by the tool-expert. However, in (b), the user has direct access to the cogito software, and so fewer syntactic issues are anticipated.**

Although a user may have excellent semantic knowledge in relation to the problem, he or she may have poor syntactic knowledge in relation to the tool (Shneiderman, 1992). Requiring that a user upgrade his or her syntactic knowledge of a tool may not be a good solution. There is increasing evidence, in part reported by Fallshore and Schooler (1995), which indicates that attempts to verbalize descriptions of such non-reportable phenomena may overshadow the original information. For example, verbalizing the appearance of a previously seen face interfered with the ability to later recognize that face from a group of similar ones (Schooler and Engstler-Schooler, 1990). Likewise, the verbal specification of a desired solution alternative may ultimately impede access to it when the language of the user does not match the language of the tool. Remaining in the perceptual domain of images for visually-oriented tasks may alleviate this problem (Hepting and Arbuthnott, 2003). More generally, this suggests that representation modality be matched to the problem so that the need for syntactic knowledge can be minimized. In the area of decision support, users may have a more satisifying experience if the information is organized and named in ways that they would choose.

Simultaneous display of multiple solution alternatives is essential because it permits the user to see what is possible within the space and it facilitates navigation amongst alternatives, which is done by mouse clicking. In order for the user to concentrate on the semantics of the task, the interface must be simple, yet sufficiently powerful. The *cogito* system does this by presenting the user with a selection of solution alternatives, realized from potential solutions in the conceptual space, as illustrated in Figure 3. For example, a cook could conceptualize an American-style appetizer with beans as the main ingredient. At least one such a recipe does exist in the collection: “opposing-sides two bean dip”, which is then realized in the top left display cell. This interface, inspired by Sims (1991) and Dawkins (1986), presents the user with a sample of what is available in the space. The user sees the current space, with the current organizational view [examining American-style appetizer recipes by their main ingredient], one screen at a time.

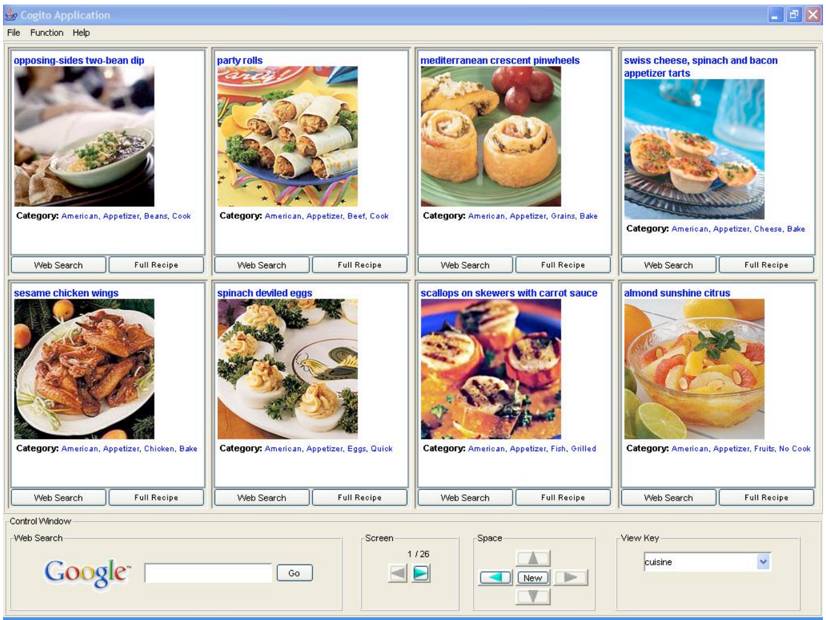



***Figure 3**: The cogito interface used for browsing a recipe collection. Each of the recipes shown is from cuisine “American” and course “Appetizer”, but the ingredients are varied.*

SUPPORT THE USER IN CHOOSING AND COMBINING SOLUTIONS

Any visual representation arising from data visualization is a combination of values to encode which data to depict, by what means to depict the data, which colours to use, and so on. In the general problem-solving case, identifiably desireable features may be spread over a number of combinations. A mechanism that permits combination would allow these features to be examined or explored more closely. Such an operation would very tedious if done manually but it is easily accomplished within *cogito*. Crossover is used to compute the subspace of alternative solutions that are consistent with the selections. The new subspace is sampled in the same way as the original: potential solutions are partitioned according to values of a parameter or parameters and then a representative potential solution from each partition is realized and displayed to the user. From the display in Figure 3, if the user selected "opposing-sides two bean dip" and "scallops on skewers with carrot sauce", the system would create a subspace of alternative American-style Appetizer recipes that shared the same characteristics as the selected alternatives (beans and cook, fish and grilled) as well as combinations of these: beans and grilled, fish and cook – if they exist in the collection. The cook could explore a wider range of recipes by indicating that similar recipes would be permitted (not matching on all parameter values). For example, an American-style recipe with beans and bake (differing on only 1 parameter value) would be considered similar.

## SUPPORT THE USE OF HEURISTIC SEARCH TECHNIQUES

Researchers have observed that, especially in design, problems and solutions co-evolve. One usually does not, perhaps cannot, have a clear statement of the problem without an idea of the solution. The advice is to begin, regardless of whether one is actually at the beginning. One's ability to explore the space from that starting point may affect his or her satisfaction upon completion. This searching permits an interactive articulation of the design, which has some similarity to participatory design (Muller and Kuhn, 1993) or evolutionary project management (Woodward, 1999).

According to Simon (1977), one may think of the search within the conceptual space for a solution as an instance of problem-solving. Even for small problems with relatively few alternatives, an exhaustive evaluation is almost always completely impractical. Instead, humans rely on heuristic search methods that are not likely to find an optimal solution, however judged, but rather to find acceptable (or *satisficing*, after Simon (1977)) solutions in a reasonable amount of time. In general, these search heuristics can be of two sorts. If the problem is well-understood, local search techniques may be employed effectively. If the problem is new, global search may be fruitful. Csikszentmihalyi and Sawyer (1995) also distinguish between problem-solving and problem-finding. One begins problem-finding with a less-clear problem at the beginning, and it can take longer to get through. Problem-finding has been connected to creative productivity and it may be distinct and more difficult than problem-solving.

Csikszentmihalyi and Sawyer (1995) trace many of the current three-stage models to the evolutionary epistemology of Campbell (1960). In these schemes, variations are created, then selected and possibly retained. Behind the flexible visual interface of *cogito* is an architecture that supports the problem-solving process of combination, selection, and retention. Combination is done by both user and computer: the user constructs the conceptual space and the computer samples the space to realize potential solutions for display in the interface. Selection is done by the user, who can interactively articulate his or her requirements in a powerful way (Wegner, 1997). The user indicates desirable parameter values or complete solution alternatives by selection (done by clicking directly on the desired cell). Once the user is satisfied with the selections made on a particular space, a new space consistent with those selections is generated by a genetic approach that performs crossover operations amongst selected combinations. Final solutions that meet the needs of the user are retained.

In this way, the space of all available solution alternatives can be navigated to achieve effective results. The possibility that the selected alternatives will come from combinations generated outside the user's experience is a very powerful aspect of this approach. It is therefore much more likely those fruitful combinations will be explored using this system.

In general, many existing software systems either limit user involvement and exploration by assigning search tasks solely to the computer, or encourage involvement by providing tools to support programming and ignore the issue of exploration. Spreadsheet interfaces (Jankun-Kelly and Ma, 2001) are powerful, but they are also limited by the fact that they require manual specification of the two parameters that are used for searching at any time.

### SUPPORT THE DETERMINISTIC MANIPULATION OF THE CONCEPTUAL SPACE

One's conception of the conceptual space relating to a problem may change considerably over time. According to Perkins (1995), a problem space can appear either as clue-rich (homing) in which the solution is evident, or clue-poor (Klondike) in which the solution must be found by prospecting. Before redesigning or respecifying a conceptual space, the designer may wish to restructure and reorganize alternatives in the space in an attempt to increase understanding and to find a transformation of the space from Klondike to homing. Choosing to view or organize the current space by a different parameter can do this: for visualizations, to look at alternatives based on "plot-type" or "data" or "colour" (with reference to Figure 1). The manipulations are deterministic because genetic programming is not used to increase the size of the space of available solutions.

Boden (1996) also discusses conceptual spaces defined by stylistic conventions, and how to expand those spaces with three rules: “drop a constraint”, “consider the negative”, and “vary the variable.” On the one hand, constraints may be necessary when the design space is large. However, to constrain or to rely on examples too much might lead one to fixate too soon on an ineffective design (Smith, 1995). Schank and Cleary (1995) made the following comments regarding computer systems: “We are not proposing that people simply loosen the constraints they use when searching for and applying knowledge. A system [person] that worked in this way would not be creative but would instead progress from schizophrenia (as it leaped from one random idea to another) to catatonia (as it found itself buried under a combinatorial avalanche of attempts).”

Figure 5 illustrates how organization is done within *cogito*: the conceptual space shown in (a) can be organized according to each axis. This affects how alternatives are presented to the designer through the interface. It provides, through views, the means to structure and examine the space according to a range of criteria. Successive generations can be used to either narrow or expand the search space, depending on the needs of the designer. The system does support “vary a variable” by allowing some or all values of a parameter to be returned to consideration when the designer might say “I like everything except the colour.”

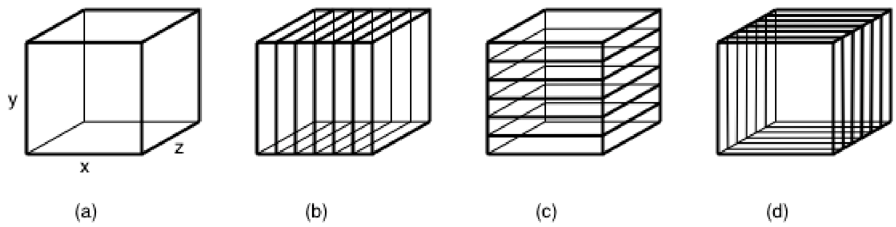



***Figure 5**: Consider a conceptual space parameterized by x, y and z as shown in (a). Figures (b)-(d) show how this space can be partitioned according to each of the parameters (first x, then y, then z).*

SUPPORT COLLABORATION AND SHARING OF IDEAS

The computer is well-suited to provide the external memory necessary to support decision-making processes that rely on the evaluation of a number of different alternatives, so that the user can know which potential designs have been realized without being responsible for the bookkeeping.

Both problem-solving and problem-finding are social activities, to which many people can contribute (Csikszentmihalyi and Sawyer, 1995). Yet, if novel designs are discussed too early, the group may discourage them. A record of exploration and of design rationale can help make these arguments more clear. According to Fischer and Boecker (1983), “design is concerned with how things (“artifacts”) *ought to be*, in order to attain goals and to function.” As the solution evolves, it is desirable to keep a record of changes.

## 4. Results

The function of the *cogito* system is separate from any application that may be written for it. Therefore, the interface presented to the user is always the consistent interface of the *cogito* system itself. This section describes results obtained with different applications written for the *cogito* system interface.

### INFORMATION VISUALIZATION

An information visualization application was developed to explore how people would go about selecting a two-dimensional data plot in order to answer some specific questions about the data being plotted. Figure 1 was created using this application. The conceptual space was defined by the author and it allowed users to manipulate data plots according to plot-type, data, data-sorting, and colour (for example). A user study was devised to test two variations of the interface: verbal-sequential (one alternative at a time) versus visual-simultaneous (several alternatives displayed at once) that respectively provided primarily-global and primarily-local searching and navigation capabilities. The two interfaces are shown in Figure 6. Over the course of two studies, fifty-seven undergraduate participants were asked to perform tasks using only one of the interfaces. Each participant was given a pre-task questionnaire, a training task with a simple *cogito* application, a main task with the *cogito* application, and a post-task questionnaire. The task involved creating two-dimensional plots based on data about the French economy from the early 1960’s (Bertin, 1983, page 100). For each départment in France, there was information about the workforce (in thousands of workers) for each of the three sectors (primary, secondary, and tertiary) in the economy; the total workforce (the sum of the three sectors); and the percentage of the workforce in each sector. The questions posedwere chosen to emphasize the three different types of reading one might use with a graph, identified by Bertin (1983): elementary, where one is interested in a specific fact; intermediate, where one is interested in characterizations of facts; and overall, where one is interested in relationships between different characterizations.

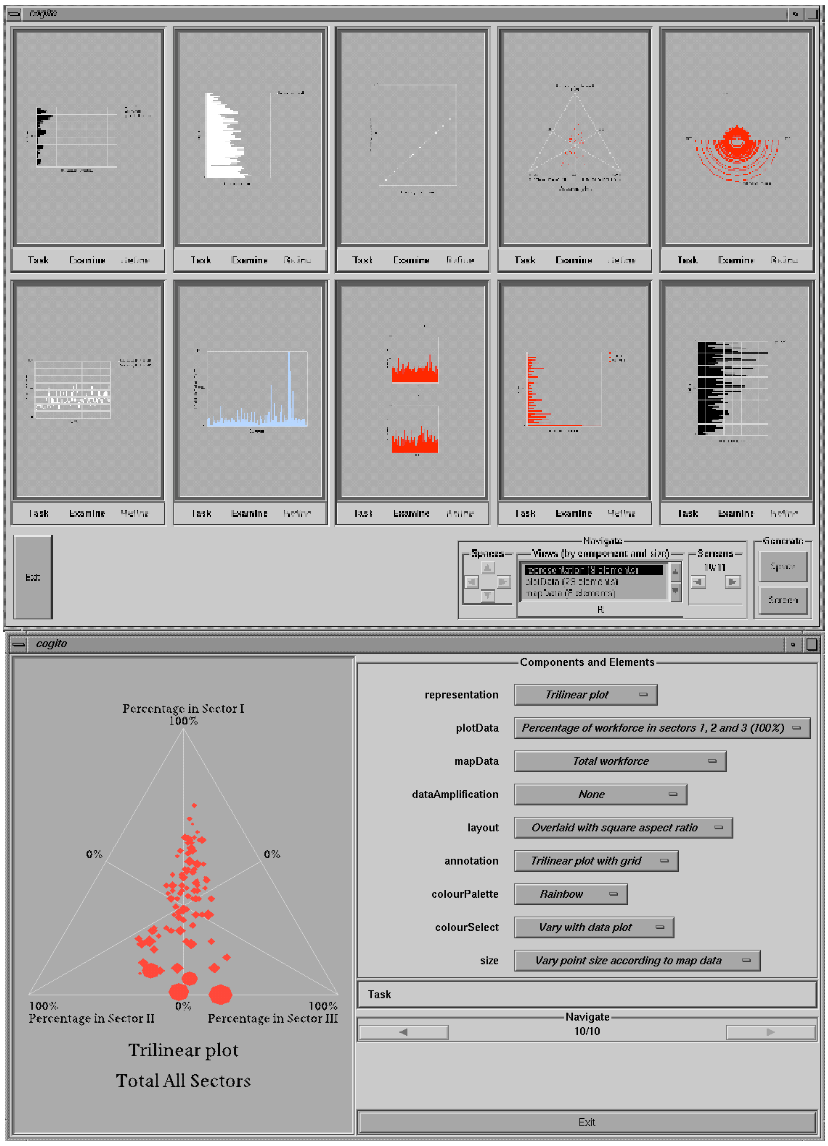



***Figure 6**: The global (top) and local (bottom) search interfaces for cogito. These interfaces were used in a study of the cogito software. The examples shown are from a data-plotting (information visualization) application.*

The data, in this case the percentage of workforce in each of three sectors, provided the necessary inputs to allow participants to use the tri-linear plot (bottom of Figure 6). A point that had values of 33.3%,33.3%, and 33.3% would lie in the middle of the triangle. If any of a point's values was 100%, it would lie at the corresponding vertex of the triangle.

A between-subjects design was used to gather qualitative data about the participants' impressions of the tasks they completed. It was found that users did not always find that the conceptualization of the problem (in terms to plot-type, data, colour, etc.) was not always liked. Therefore, performance from users was likely influenced by this potential mismatch. With respect to the interfaces, participants liked the speed with which they could manipulate alternatives (bottom, Figure 6) by selecting items from option menus. The participant comment "having the ability to play with the data, and see it in ways that were meaningful to me – and being able to build that meaningful representation" was indicative. These participants also missed the opportunity to see multiple plots at once.
However, there were many positive comments: "It seems to be a very good program for someone like myself, with little graphical software knowledge to use. It is easy to learn which would allow more time to focus on the data that the graph represents and less on actually making the graph." Users of the simultaneous-visual interface (top, Figure 6) also liked the organizational capabilities: "the way it organizes information trees, the hierarchy allows easy access to previous choices, the way many graphs are different options displayed at the same time lets it be easy to see which one is best for analysing data."

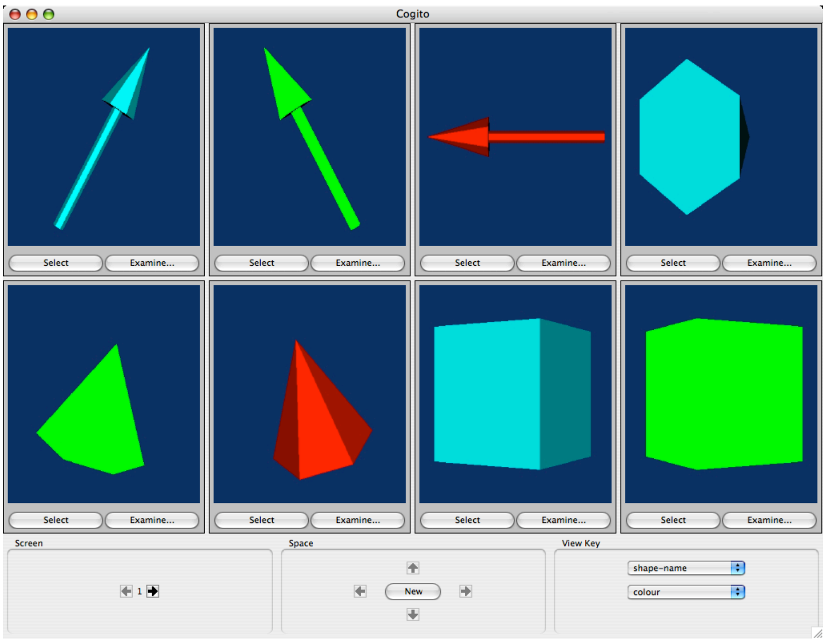
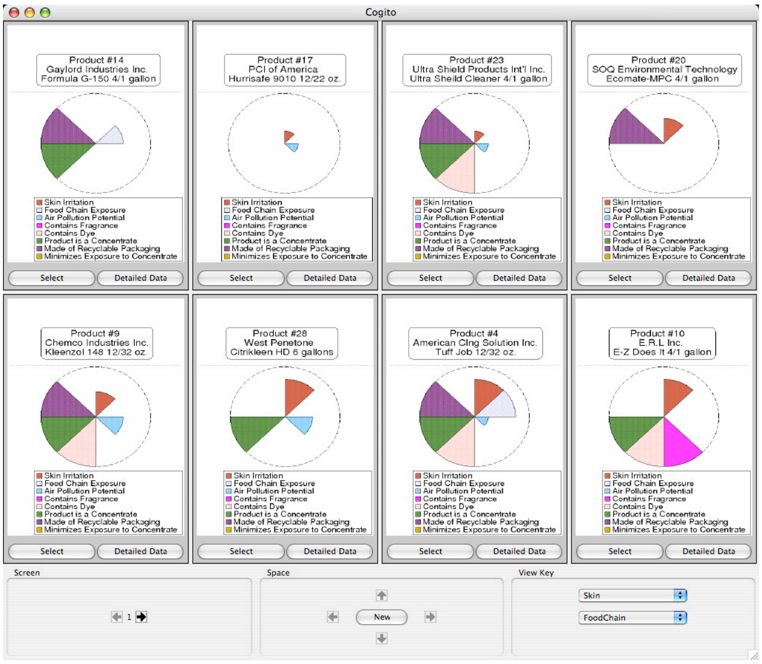



***Figure 7: The cogito interface exploring outputs from an application which uses the visualization toolkit (vtk).***

Overall, participants were not overwhelmed when dealing with a large space and that both local and global search features were appreciated. Although this global search interface may have a steep learning curve, it is also very novel. Further studies will be needed to better test for creative outcomes. However, that participants were not overwhelmed with the number of available alternatives may indicate the presence of flow (Csikszentmihalyi, 1996), in that people were able to stay focused their task, despite having over one million potential solutions in the conceptual space. It seems that *cogito* can provide the basis from which p-creative, and h-creative, designs (or choices or discoveries) may come.

A separate application programming interface (API) has been developed so that a variety of applications can be written to use the system user interface. This approach was adopted in order to provide easy access to the common capabilities for navigating and exploring a conceptual space. Figure 7 shows *cogito* being used to explore alternatives generated by an application written using the visualization toolkit (vtk), developed by Schroeder et al. (1996). The programming, as described below, that is required to interface with the application's syntax is done separately from the user's interactions and is hidden from the user.




***<u>Figure 8</u>*: The cogito interface used to provide decision support for environmetally-preferable purchasing.**

ENVIRONMENTAL DECISION SUPPORT

Figure 8 shows a decision support system for environmentally-preferable purchasing. It involves access to a database, and as such is different from the aforementioned applications (Figure 6, and 7). The database will not generally contain records for all possible combinations of parameters. For each product, there are eight attributes that are used to quantify environmental impact. This application has been studied in relation to more traditional web-based decision support alternatives, with positive results. Maciag et al. (2005) examined how people relate to the information presented through these attributes and explored ways that the presentation could be personalized without significant overhead.

RECIPE BROWSING

A recipe-browsing application, shown in Figure 3, was built for the *cogito* system. It is similar to the Environmental Decision Support in that it accesses a sparsely-populated database. For example, recipes do not exist for every combination of ingredients. Therefore, the application retrieves whatever available records match the current query and employs a concept of similarity to retrieve close matches when no or very few exact matches exist. A user study was conducted of the recipe browser using 20 undergraduate participants. The participants were asked a variety of questions involving 1, 2, and 3 parameter values (relating to cuisine, ingredients, season of year, and cooking method – for example). Comments were very positive, especially related to the ability to find interesting recipes (for example): "I like the idea of this software a lot. It would help me getting rid of leftover ingredients. I think the layout is good, and once learned, very fast, powerful and easy to use."

## 5. Conclusions

The prototype *cogito* system was implemented based on the design principles described herein, and various user studies of the software with different applications have validated those principles. The result that users were not overwhelmed by the size of the conceptual space presents significant encouragement for the architects of tools who seek to put the designers at the centre of the decision-making process regarding their own solutions. The development of the *cogito* system as a flexible interface to a wide range of applications is an important aid to personal creativity and understanding.

Although here is presented a single idea of how a user may proceed with the help of a programmer, there may be cases when users need to add to a conceptual space without a programmer so the means to enable that will require further development.

## Acknowledgements

I thank the University of Regina and the Natural Sciences and Engineering Research Council (NSERC) of Canada for financial support. This work was begun as part of my dissertation and I thank Bob Russell, Dave Fracchia, John Dill, and Joe Marks for their valuable guidance. My colleague, Katherine Arbuthnott has helped me to define new directions for this work. Students Timothy Maciag, Yuancheng Liu, Matthew McKague, Paul Schmiedge, and Orland Hoeber have helped with implementations, proof-reading, or both.

## References

R. M. Baecker, J. Grudin, W. A. S. Buxton, and S. Greenberg. *Readings in Human-Computer Interaction: Toward the Year 2000*. Morgan Kaufmann Publishers, San Francisco, Second edition, 1995.

J. Bertin. *Semiology of graphics : diagrams, networks, maps*. Univ. Wisconsin, 1983.

M. A. Boden. Creativity. In M. A. Boden, editor, *Artificial Intelligence: Handbook of Perception and Cognition*, Chapter 9. Academic Press, Second Edition, 1996.

D. T. Campbell. Blind variation and selective retention in creative though as in other knowledge processes. *Pyschological Review*, vol. 67, no. 6, pp. 380–400, 1960.

S. M. Casner. A task-analytic approach to the automated design of graphic presentations. *ACM TOG*, 10(2), April 1991.

M. Csikszentmihalyi, Creativity: Flow and the psychology of discovery and invention, New York, 1996.

M. Csikszentmihalyi and K. Sawyer. Creative insight: The social dimension of a solitary moment. In R. J. Sternberg and J. E. Davidson, eds, *The Nature of Insight*, pages 329–364. MIT Press, 1995.

R. Dawkins. *The Blind Watchmaker*. W. W. Norton and Company, New York, 1986.

M. Fallshore and J. W. Schooler. The verbal vulnerability of perceptual expertise. *J. Exp. Pysch.: Learning, Memory, and Cognition, 21(6), 1995.*

G. Fischer and H.-D. Boecker. The nature of design processes and how computer systems can support them, in *Integrated Interactive Computing Systems*, pages 73–86. North-Holland, 1983.

J. S. Gero. Computers and Creative Design, *CAAD Futures*, 1995.

D. Goldberg. Genetic algorithms in search, optimization, and machine learning, 1989.

R. B. Haber and D. A. McNabb. Visualization idioms: A conceptual model for scientific visualization systems. In *Visualization in Scientific Computing*, pages 74–93. IEEE Computer Society Press, 1990.

D. H. Hepting, *Towards a Visual Interface for Information Visualization*, Proc. 6th International Conference on Information Visualization, IEEE, pp. 295—302, 2002.

D. H. Hepting. Interactive Evolution for Systematic Exploration of a Parameter Space, in Proceedings of ANNIE 2003, pages 125--131, ASME Press, 2003.

D. H. Hepting and K. D. Arbuthnott, The Implications of Verbal Overshadowing for Computer Interface Design. University of Regina, Computer Science Tech Report CS-2003-10, November, 2003.

T. J. Jankun-Kelly and K.-L. Ma. Visualization Exploration and Encapsulation via a Spreadsheet-Like Interface, *IEEE Transactions on Visualization and Computer Graphics*, 7(3), pp. 275-287, 2001.

G. Kelly, *The Psychology of Personal Constructs,* Norton, 1955.

J. R. Koza. Genetic Programming: On the Programming of Computers by Means of Natural Selection *(Complex Adaptive Systems). MIT Press, 1992.*

J. Larkin and H. Simon. Why a diagram is (sometimes) worth ten thousand words. *Cog. Sci*. 11:65–99*,* 1987.

J. Licklider. Man-computer symbiosis. *IRE Trans. Human Fact. Elect.*, 4–11, 1960.

T. Maciag, D. H. Hepting, and D. Slezak. *Personalizing User Interfaces for Environmental Decision Support Systems,* Proc. Web Intell. Rough Set WS, 2005.

B. H. McCormick, T. A. DeFanti, and M. D. Brown. Visualization in scientific computing. *Computer Graphics*, 21(6), November 1987.

M. J. Muller and S. Kuhn. Participatory Design. *CACM,* 36(6), June 1993.

D. A. Norman. *The Psychology of Everyday Things*. Basic Books, New York, 1988.

A. Paivio. *Mental representations: A dual-coding approach*. Oxford Univ. Press, 1986.

D. N. Perkins. Insight in minds and genes. In R. J. Sternberg and J. E. Davidson, editors, *The Nature of Insight*, pages 495–533. MIT Press, Cambridge, MA, 1995.

B. B. Petchenik. Cognition in cartography. In L. Guelke, editor, *Cartographica: The Nature of Cartographic Communication*. Univ. Toronto Press, 1977. Monograph 19.

M. B. Rosson and J. Carroll, *Usability Engineering: Scenario-Based Design*, 2002.

R. C. Schank and C. Cleary. Making machines creative. In S. M. Smith, T. B.Ward, and R. A. Finke, eds., *The Creative Cognition Approach*, pp. 229–247. MIT Press, 1995.

J. W. Schooler and T. Y. Engstler-Schooler. Verbal overshadowing of visual memories: Some things are better left unsaid. *Cognitive Pyschology*, 22:36–71, 1990.

B. Schneiderman. *Designing the User Interface*, Addison-Wesley, 1992.

B. Schneiderman. Creating Creativity: User Interfaces for Supporting Innovation, ACM TOCHI, 7(1), pp.114-138, March 2000.

W. J. Schroeder and K. M. Martin and W. E. Lorensen, The design and implementation of an object-oriented toolkit for 3D graphics and visualization, in *Proceedings of Visualization 96*, pp. 93–100, 1996.

H. Simon. *Models of Discovery*. Reidel, 1977.

K. Sims. Artificial evolution in computer graphics. In R. J. Beach, editor, *Computer Graphics:* SIGGRAPH '91 Conference Proceedings, pp. 319–328. ACM Press, 1991.

S. M. Smith. Fixation, incubation, and insight in memory and creative thinking. In S. M. Smith et al. eds., *The Creative Cognition Approach*. MIT, 1995.

R. R. Springmeyer, M. M. Blattner, and N. L. Max. A characterization of the scientific data analysis process. In A. E. Kaufman and G. M. Nielson, eds., *Proc. Vis. '92*, 1992.

S. Todd and W. Latham. *Evolutionary art and computers*. Academic Press, London, 1992.

M. Vass, J. M. Carroll, and C. A. Shaffer. Supporting Creativity in Problem Solving Environments. Proc 4th conference on creativity and cognition. pp. 31—37, 2002.

C. Ware. The foundations of experimental semiotics: a theory of sensory and conventional representation. *J. Vis. Languages and Computing, 4:91–100, 1993.*

P. Wegner. Why Interaction Is More Powerful Than Algorithms, CACM, 40(5), 1997.

D. Wessel and M. Wright. (2001), *Problems and Prospects for Intimate Musical Control of Computers,* Proceedings of the 1st Workshop on New Interfaces for Musical Expression (NIME01), ACMSIGCHI electronic proceedings, 2001.

S. Woodward. Evolutionary Project Management, IEEE Computer, 1999, pp 49-57.

T. Winograd. *Bringing Design to Software.* Addison-Wesley, 1996.

T. Winograd and C. F. Flores. *Understanding Computers and Cogniton*. Ablex, 1985.