CREATIVE SAILBOAT DESIGN

JACK PARK
ThinkAlong Software, Inc.
Box 359
Brownsville CA 95919 USA

Abstract. Creativity in the design process is explored by means of a sailboat design exercise. A rule-based design system is controlled by a directed evolution algorithm such that the design space for sailboats is computationally explored. Directed evolution provides heuristic guidance for a genetic-like approach to mutation of design rules. This mutation may result in improvements to the design rules. Some results from an early trial of the system are discussed.

1. Introduction

When Garry Hoyt discussed his ideas about a sailboat capable of speeds in excess of 50 knots (Hoyt, 1986), he opened, for many, a kind of pandora's box, an area of sailboat design which had largely been ignored by designers. Indeed, empirical evidence suggests sailboats are, with some exceptions, limited to their so-called hull speed, typically in the 5 to 20 knot range. Hoyt's ideas, for some, were a form of creative magic, setting individuals off on design projects in completely new (to sailing) directions.

Multihull sailboats (e.g. catamarans with two hulls) and "non-traditional" designs, most notably the sailing surfboards, easily violate the hull speed maxim. Thus, with Hoyt's stimulation, there began the evolution of designs which, to say the least, are not at all traditional. But, they are fast.

If hull speed—the limitation that, regardless of wind speed, the boat is restricted in speed by the length of its hull—is the traditional constraint, one simply builds a longer hull to go faster; the relationship is nearly linear. If, instead, one follows Hoyt's prescription and looks to the non-traditional, a step toward creative design is taken. An example of this is Greg Ketterman's novel two-masted boat (Ketterman, 1990). Riding on hydrofoils, this fast little boat may eventually achieve Hoyt's 50 knot benchmark.

A new relationship is defined by the creative leap from traditional to non-traditional sailboat design: the ratio of boatspeed to windspeed. Ketterman's boat achieves between two and three times the speed of the wind. The boat design we explore here may perform at around three times the speed of the wind. This boat design effort orginated in support of our interest in molecular design; the sailboat project is a comfortable domain in which to explore the construction of the design engine we discuss here.

Once the initial decision to look to non-traditional approaches is made, it remains to explore the newly-unconstrained possibilities. That is, the traditional hull speed constraint is presumed canceled, as evidenced in the performance of, say, the sailing surfboards. Those boards have reached speeds exceeding 40 knots. New constraints, however, may be discovered; we discuss such discovery below.

We focus this discussion on an application of a discovery tool to the exploration of the design space as conceived by an individual (e.g. the author) operating in a liberated constraint space. The project is a team effort, shared by the program and the individual using the program. This is a multi-space search effort; the search through design rule space, and the search through—indeed, the extension of—the constraint space. Exploration of the design space (c.f. Bradshaw, et al., 1991), traces some roots back to George Kelly and his personal construct theory (c.f. Kelly, 1955; Shaw, 1980; or Shaw & Gaines, 1991). One designs what one visualizes, and the templates of personal construct theory lend some insight into the process.

The work reported here exploits an analogy between airplanes of different configuration, and sailboats of different configuration. Specifically, the performance of an airplane increases as one approaches the configurations of high performance gliders—sailplanes. By analogy, it may be that sailboat performance may be improved by moving the design away from conventional sails to a design similar to a sailplane somehow "attached"[1] to the water. In this project, we apply design rules originally intended for sailplane design to our sailboat design space; we apply the physics of sailplane aerodynamics to the design of a sailboat. Additional rules cover the design of components of the boat intended to stay in the water.

The computational design exercise is one of synthesis. An initial design is proposed—either by randomly selected design parameters or as the individual's best first guess. A design tree is bloomed along best-first lines; boats of higher performance receive more attention from the system. At each stage in the design, a simulator models the performance of the evolving design.

The project is implemented on a discovery system shell called The Scholar's Companion (TSC) (Wood&Park, 1992) running on a Macintosh workstation. TSC provides a frame-based memory structure and tools for building directed graphs. These graphs may be interpreted as design trees, as we discuss

[1]It seems a reasonable defining constraint that a sailboat somehow remain in contact with the water.

here, or as envisionments—histories of some set of physical processes. Thus, the TSC graph tools enable projects in explanation, prediction, and design. As we illustrate below, the application of TSC to the exploration of a design space for a high-speed sailboat uncovers an unanticipated opportunity for explanation and possible discovery.

We now look at the TSC knowledge base as we apply it to the design of a high-performance sailboat.

```
If you notice the performance is not improving
   in this branch
Then terminate this branch of the design tree
```

This particular example is an instance of a search pruning rule. As noted by (Navinchandra, 1991), we often do not want to prune a search for a design since we may truncate a path which leads to an innovative result.

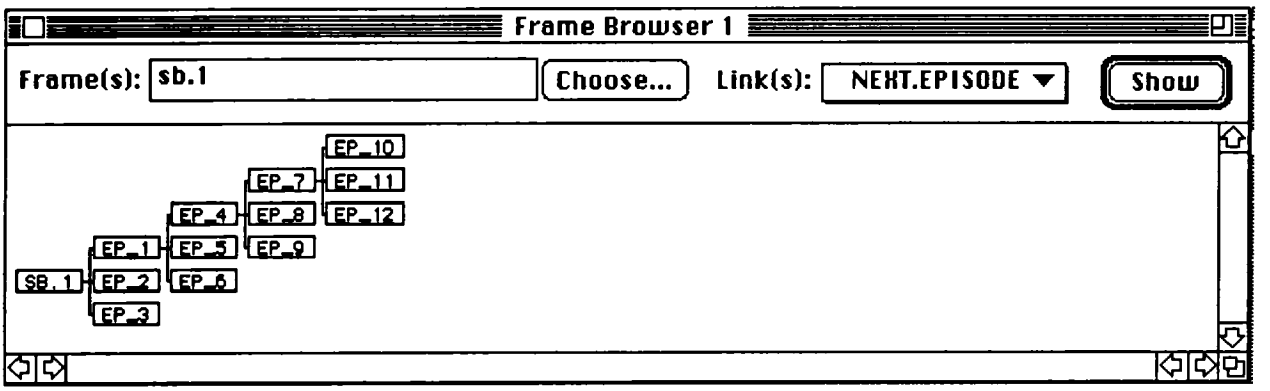

Figure 1. TSC design tree.

2. Sailboat design knowledge base

A TSC knowledge base consists of several classes of rules. Some of the rules implement TSC's behaviors during a project. Some of the rules represent some aspect of the physics of the project. By constructing these physics rules such that they contribute to the design of a project, design rules are implemented. An example design rule is:

```
If you want to improve the performance of the
   vehicle
And the vehicle includes an aerodynamic
   structure (e.g. a wing)
Then consider increasing wingspan
```

The initial sailboat design, the first node in the design tree, initializes all the design parameters. These include wingspan, boat weight, rudder area, and so forth. A design cycle involves "firing" design rules on the current (in the first case, initial) design. Each rule creates a new node with some small change made to one of the design parameters. From the example rule above, wingspan would be increased about ten percent. A simulator is then run on each new design and the sailboat performance recorded at the node. In this project, the only performance parameter noted is maximum speed of the boat. Since the simulator finds the maximum boat speed at the same windspeed conditions for all designs, maximum speed is a convenient comparison value between designs. Figure 1 illustrates a short exercise of TSC building a design tree. Nodes along the top of the illustrated tree tend to represent the highest performance boats. A total of three design rules were included in this run.

TSC includes a body of rules called "common sense" or sense rules. The tree is studied by such rules of the form:

Another form of sense rule is the "special sense" rule, the highly specialized rule created specifically for the project. Included in this group are rules which analyze the results of a simulation and determine if the performance has been improved. Such rules "reward" the design rules which create the improved vehicle. A typical reward rule looks like

```
If a design rule has consistently improved the
   performance of the project
Then conjecture that rule will always improve
   the performance of the project and give the
   rule a small boost in worth
```

TSC further includes a body of rules which direct learning and knowledge base improvement. This body also includes rules which form hypotheses, study hypotheses, collect data, and so forth. These rules play an important role in the act of design creativity since they may be directed at the design process itself. As we discuss below, creativity occasionally involves "breaking the rules." This may be done either by some random rule mutation process, such as in a Genetic Algorithm (Holland, 1992), or by heuristic mutation means.

Formation of hypotheses centered on rule mutations becomes an interesting area of research related to computational creativity. We now briefly look at creativity as it relates to this design project.

3. Computational Creativity

An early benchmark for discussion of creativity has been the 4-step model of creativity due to (Wallas, 1926): preparation, incubation, illumination, and sum-up. The individual involved in design work of this nature is responsible for preparation and incubation, while TSC assists in incubation and illumination by conducting a search through design space.

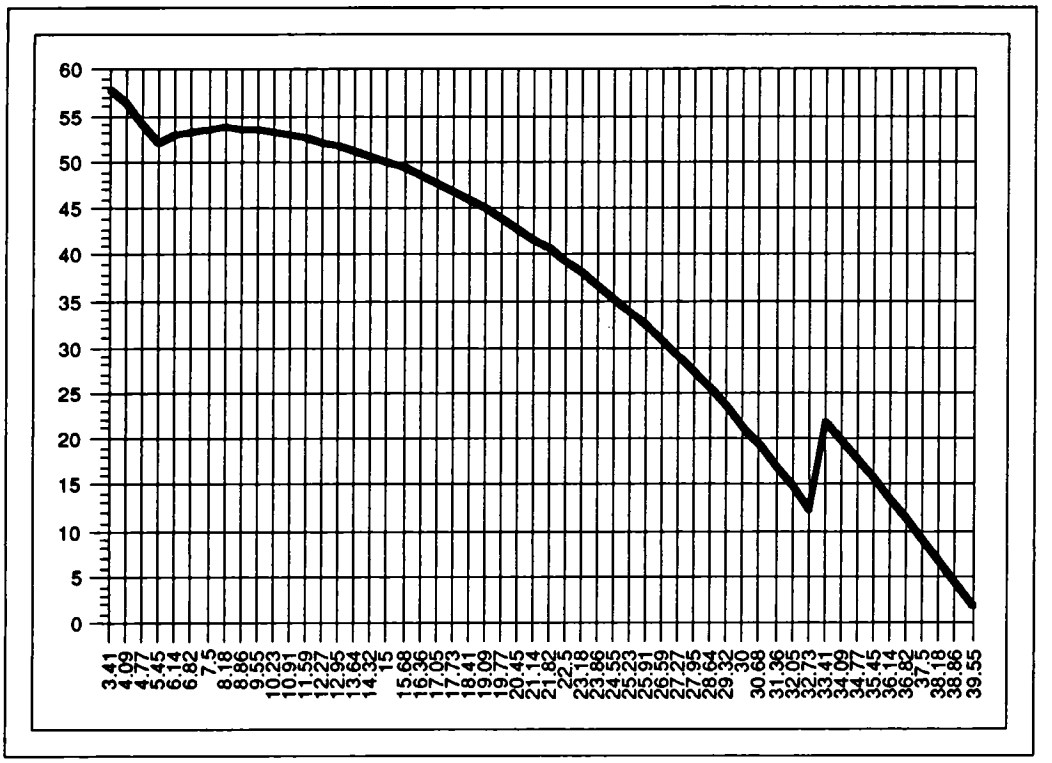

Figure 2. Speed vs. net forward thrust

The individual then performs any sum-up required to execute a final boat design. Occasionally, the combination of computational and human intellect being applied to a design task illuminates some aspect of the design not anticipated at the outset.

One acquires, through rote and experiential learning, a corpus of design rules. These rules allow one to map goals to ideas and concepts. A body of such rules in an expert system does an adequate job of taking an initial design and mutating it until some goal is met. This is a direct analog to optimization.

We believe an important aspect of creativity is missing from the expert design system. Creativity, we suspect, often means "breaking the rules." The Wallas 4-step model provides a convenient "outer loop" but, often as not, we simply ignore some constraint and plow on as though trying to escape some local minimum—hill climbing to reach a design goal. The results of this are occasionally useless, but also occasionally quite useful. The TSC architecture encourages mutation of rules such that the design search space is potentially expanded. The source of creative power in the TSC system is its rule mutation system, running under the direction of directed evolution[2]. Directed evolution is a mix of genetic algorithm and heuristic guidance.

Breaking the rules involves machine learning approaches to design rule mutation. In the next section, we describe two related projects which motivate the work described here.

[2]The term is borrowed from molecular biology (Abelson, 1990).

4. Related Work

Much of this work is inspired by the work of Douglas Lenat (Lenat et al., 1982) and by the work of Peter Karp (Karp, 1990). Lenat's work deals with the generation of design alternatives in the domain of VLSI. In his Eurisko program, Lenat applies heuristic rules to the combination and mutation of other rules. This rule mutation scheme provides an opportunity to explore notions of computational creativity.

Karp's work deals with the formation of hypotheses in the domain of molecular biology. His approach to the hypotheses formation based on analysis of experiments is directly applicable to the design work discussed here. As a design tree evolves, analysis of the ongoing results presents opportunities for hypothesis formation, rule mutation, and design creativity. We have explored other aspects of the hypothesis formation task in an immune system response project (Park & Wood, 1992).

5. Results and Discussion

Figure 2 illustrates one of the designs explored by TSC. The plot shows the relationship between net forward thrust and boat speed for a 20 mph wind speed. The concave downward curve satisfies our intuitions that the faster the boat travels, the less surplus thrust it will have to accelerate. Ideally, one reads the maximum speed as the point where the curve crosses the x-axis.

The curve offers a pair of interesting points worth pon-

dering: two discontinuities are noted. The upper left—occuring at low speed—is easily explained by the boat lifting a balance ski out of the water when it is no longer needed to maintain, as sailors would say, an even keel. The second discontinuity[3], occuring at much higher speed, is a bit more interesting. In fact, the author found this second point an inspiration for discovery: design rules to keep the boat in the water. In fact, the problem of the boat lifting clear of the water remains a partially unsolved problem at this writing; the effective maximum speed of the boat is therefore the speed at which it tries to fly. The problem suggests a counter-intuitive solution: the heavier the boat, the faster it can go.

Every aspect of the design tool discussed, except the rule mutation activity, has been implemented for this project. Rule mutation has been implemented in a project on protein structure (LeClair et al., 1992). It remains to integrate all the tools.

6. Summary & conclusion

We have illuminated an architecture we are implementing for design. The project used as the trial case is that of a high-speed sailboat. By mapping knowledge from the domain of sailplanes to the domain of sailboats, we introduce new performance criteria to boats—boat-to-windspeed ratio—and we introduce opportunities for new (at least, new to the author) discoveries.

[3]These are, indeed, discontinuities, masked by the curve plotter used.

Acknowldegements. This project has benefitted greatly from the work of many others. In addition to the workers noted above, we happily acknowledge productive conversations and help from Dan Wood, Steve LeClair, Scott King, and Dave Foster.

7. References

Abelson, John, "Directed Evolution of Nucleic Acids by Independent Replication and Selection," *Science* 249, 3 August, 1990, pp 488-489.

Bradshaw, Jeffrey M., David Shema, and John H. Boose, "Canard: An Alternative Generation Tool Based on Possibility Tables," in *Creativity: Models, Methods, and Tools*, 1991, AAAI.

Holland, John H., *Adaptation in Natural and Artificial Systems*, 1992, Cambridge MA, The MIT Press.

Hoyt, Garry, *READY ABOUT!* , 1986, Camden Maine, International Marine Publishing Co.

Karp, Peter D., "Hypothesis Formation as Design," in Jeff Shrager and Pat Langley, eds., *Computational Models of Scientific Discovery and Theory Formation*, 1990, San Mateo, CA, Morgan Kaufmann.

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

Ketterman, Greg, "Experiments in High-Speed Sailing: TriFoiler," *Multihulls* , May/June 1990, pp 63-65.

LeClair, Steven R., Scott King, Dan Wood and Jack Park, "Using A Directed Evolution System To Predict Protein Secondary Structure," 1992, manuscript in preparation.

Lenat, Douglas B., William R. Sutherland, and James Gibbons, "Heuristic Search for New Microcircuit Structures: An Application of Artificial Intelligence," *The AI Magazine*, Summer, 1982.

Navinchandra, D., "Innovative Design by Exploration," in *Creativity: Models, Methods, and Tools*, 1991, AAAI.

Park, Jack, and Dan Wood, "Getting the Model Right: Coupling a Cellular Automata to a Discovery System," in Jan M. Zytkow, editor, *Proceedings of the ML92 Workshop on Machine Discovery*, 1992, Aberdeen, Scotland, University of Aberdeen.

Shaw, Mildred L.G., and Brian R. Gaines, "Mapping Creativity with Knowledge Support Tools," in *Creativity: Models, Methods, and Tools*, 1991, AAAI.

Shaw, M.L.G, *On Becoming a Personal Scientist: Interactive Computer Elicitation of Personal Models of the World*, 1980, London, Wiley.

Wallas, G., *The Art of Thought*, 1926, New York, Harcourt.

Wood, Dan, and Jack Park,*User's Manual for The Scholar's Companion*, 1992, Brownsville, CA, ThinkAlong Software.