

Center for Environmental Structure Computing:
Christopher Alexander
Greg Bryant
John Seamster

Sun Microsystems:
Bill Joy
Mike Clary

Thoughts on the Aspen Meeting

Christopher Alexander

After listening to comments made in Aspen, and afterwards, I was very puzzled. I've come to believe that participants in the Aspen meeting must have found themselves unable to retain their experience with Gatemaker, or understand the experimental process that led to the program in its present early form.

Since this kind of misunderstanding may be having a material impact on people's assessment regarding where to go next, and what to do with the material, I felt it worthwhile to write down, in brief, an accurate summary of what was done to create Gatemaker, and what it has achieved, and what it has not achieved.

Our main aim, in January and February, was to find out what form -- that is, what form of interaction for the user -- in a computer version of a generative process, might come close to the generative processes which CES has experienced, and implemented in non-computer forms during the last ten years. The most critical issue was the psychology of the process. That is, what kind of realization, on a computer, might bring the user close to the psychology of a successful process for a designer, or inventor.

In our realizations at CES during 1985-1996 we have discovered that the process is extremely sensitive to nuance, atmosphere, and subtle matters of control, and volition. It was readily apparent from the outset of this project, that currently acceptable realizations such as the Windows environment, browser formats, and various formats typical of common software such as word-processors, spread-sheets, games, advice-takers, structural design programs etc., would simply not recreate the atmosphere which actually enables a person to succeed in designing something.

Real design requires an atmosphere of freedom, tranquillity and joyfulness, which is frighteningly rare in the computing environment. The question then,

was: what kind of computer realization, what kind of screen organization, what kind of temporal organization, and what kind of format might approach the successes we have had previously in the non-computer environment?

I therefore initiated a long series of experiments, designed, every day, to ask whether the necessary atmosphere was appearing, in any useful form, on the computer screen.

After writing a small bit of code, we would use it, and try it on others, asking all the time whether the experience had the necessary depth.

The answer, nine times out of ten, was that it did not. The format of the program, and of the interface, was usually too cold, too logical, too constrained, too sterile, too unbalanced, too upsetting, too intimidating, or too didactic. It would put the user in the wrong frame of mind, nearly always, for any realistic or inspired design.

Whenever we sensed even an inkling of positive effect in some aspect of the program, something which in some way produced the state of mind familiar to me outside the computer world, we kept it, and improved upon it.

Here are a few of these positive discoveries. It is rather unusual, I would guess, to say that a computer design tool absolutely needs the following elements to really work:

Balanced Windows

A centered workspace, with serious and balanced framing, which sets the workspace firmly in a very distinct but comfortable world of its own.

Settled workspace

Locked rectangles (windows) and separators around a *fixed* rectangle, to avoid the serious and disturbing annoyance of having to position windows all the time.

Natural Colors

Individual elements with truly natural colors, providing a calm, varied and harmonious environment.

Color interaction

Carefully chosen contrasting/complimentary colors for adjacent screen objects. The effect on restfulness or excitement engendered by the color of text and background is extraordinary, and should be used to enhance the right state of mind at the right time. For

example, the sequence screen is unusable in a red on black form, or black on grey or black on white. The natural yellow on green puts the user in an eager, considerate, "in-control" state for the reading of sequence descriptions.

Description vs. instruction

Changing sequence guidance from *instructions* to *descriptions* (1) allowed the user to imagine better, and (2) did not force them to make moves that their judgment resisted.

Applying everything to the small step

In all design, there is a moment when you are looking for information, analogous to the way a cell applies its entire genotype to deal with the problem at hand. One must be in an environment with a contemplative atmosphere at that moment to really draw upon yourself in this way.

Gradient and scale

We found that poorly chosen relative size and relative color of fonts could very easily make the user uncomfortable or distracted. For example, there is a gradient of font sizes from the top to the bottom in the main screen during work, and this gives the user a very definite feeling of being in a supportive, *structurally sound*, natural environment. Most computer applications look brittle, as if they'll fall apart any minute. No good work can take place in such an environment.

Comfortable introduction

The live tutor is our current method of easing the user into the designers seat. We can say definitively that many standard approaches do *not* help achieve quick comfort.

Screen changes

The position of tools and instructions within the main work area was extremely critical. Take sequence text. The positive "looking through a window" effect of the main screen was ruined by putting this text over it. The text could be put in no window which would draw the user's attention sufficiently. But eliminating the entire main work area worked perfectly: the transition is comfortable and clear. This effect should only be used for such a tightly coupled set of screens as the workspace and the sequence.

State space and access to tools

The phrase "context-sensitive" tools in interface design doesn't make much room for the idea that the

contexts themselves, the state space, must be sensibly organized, centralized with small sorties into other states. This makes it possible to have major, tranquil, wholeness-contemplative states with many possible short active journeys to minor states. This encourages small steps of activity, making it more likely that the resulting design will be well-structured.

Value of resources

A small handicap, the use of the mouse, had a tendency to make people use their effort more carefully. This distributes resources better over the whole. But the handicap wasn't some conceptual or mechanical deficiency, like a hard-to-use 3-D tool. It was perfectly understandable, so it simply made the user become more careful. With a computer program, one must consider carefully where to help the user, and where to give them a real dose of the consequences of their actions.

Structure-preservation

It is most important to put the user in a mental state where good things are easy to see and emphasize. The concentric emphasis on the real photo, as well as every supporting aspect of the interface, contributed to this. We found also that the less time which one allowed oneself between steps, the more structure-preserving was the result. This is certainly not true with the interface of a typical CAD tool, because there is nothing good to hang onto while moving quickly.

Balance

Designers need to look at every major aspect of the whole before making the next move, so that efforts are distributed evenly. Coherence in complex systems is impossible without this consideration at every step. A balanced screen, a "window on the world" in the middle, and other features, help the user to keep their eyes "pulled back" in contemplation of the whole.

These attributes of the workspace, though they sound "minor" in computer jargon, were in fact of primary importance in deciding the effectiveness of the software. The overwhelming impression I had (as a non-computer-scientist) was that good software, software that really works for *people*, is dominated by features for which there is not even a vocabulary in the computer world, let alone any formalized language for discourse or testing.

Thus, we were wandering in an unexplored domain, making discoveries very slowly. Sometimes, 24 hours of solid work, writing and rewriting hundreds of lines of code several times over, enabled us to take one tiny step forward on the appearance of a tool, or the form of its emergence on the screen.

When I heard computer scientists in Aspen discussing what we had done, it struck me that no one was really familiar with the fundamental experience under investigation in our study. I was rather distressed at the discussion of "rapid-prototyping". Certainly good task breakdown and small iterative loops are an indirect result we want. But this is miles from the experience we were directly studying: generating good results by helping people put themselves into their effort, body and soul. I still have difficulty with the fact that the variables actually under investigation have no names, nor descriptions within computer science.

Empirically, it is highly significant that in the environment of Gatemaker, people did in fact manage to behave in a truly childish and naïve fashion. Although this was accepted as obvious by all the participants, in fact it is not a feature shared even by such award-winning design tools as Kai Krause's "Bryce2".

This simple, everyday naiveté is absolutely necessary for serious design; but the conference in Aspen failed to recognize that this small step had been achieved by immensely hard work, involving a daily 18 hour schedule of non-stop experiment, trial, modification, trial, discussion, experiment. No one seemed to recognize that what was achieved, though simple, is a significant breakthrough for the format of such a program.

For most of December, January and February, I was not convinced that this was possible at all. I became convinced only 10 days before our Aspen meeting: convinced that indeed the psychological conditions necessary to a generative process, and to the successful implementation of a computer based generative process, had been achieved in rudimentary form in Gatemaker

The lack of response to this rather substantial accomplishment, was, to me, the most peculiar aspect of the Aspen meeting. It was not changed by my subsequent meeting with Dick, who again solemnly assured me that a good programmer could have written that program in about two days!

Real design, on a computer and generally, is a far more serious matter. What we did may seem, to most computer scientists, to be simple "user-interface" issues. But only "users" can bring *life* to a thing. Sensitivity to the user's natural abilities should reverberate throughout a computer's design.

Going into the meeting, I thought everyone understood this. Certainly everyone talked about user-directed design. And I believe everyone had the

correct experience with Gatemaker. But then it was somehow forgotten and perhaps dismissed. Perhaps the user's importance is not actually felt strongly.

This makes me worry. Computer engineers cannot truly understand coherent design unless they become more sensitive to experiences like those in Gatemaker.

Those brief moments of generating life must be searched for actively, with real feeling and passion. When found, one must explore them, seek their juice, pursue them further and revel in them.

This is the most effective method I know.

It's not easy. But it's worthy of attention.

After discussing this paper with the SUN partners, Christopher Alexander wrote further about [the implications for computing](#).