Gatemaker: Christopher Alexander's dialogue with the computer industry

This paper is a pre-print and extension from the proceedings of the 2013 PUARL conference at the University of Oregon, where this talk was first given.

Greg Bryant

Rain Magazine

Abstract

The Gatemaker project provides an enlightening lens through which to examine the interplay of ideas between Christopher Alexander and computing, over the last half-century. Interest in his work became intense by 1996, so his team secured funding for a research project: a computer-based design tool for improving the built-environment, which treats feeling *as an objective metric. The application made significant progress along these lines, unfortunately unmatched in any other computer project since. There is much confusion about the program, within both computing and architectural circles. This is because, like many of Alexander's projects, Gatemaker unveiled a battle between different systems of thought.*

Prologue

"The conclusion I draw is that the character of this program has a bearing not only on the future generation of programs in the building/environment field, but that all software (including software designed to help developers design software) may, in all probability, also benefit from the kind of sea-change which is anticipated in Gatemaker. One may describe this sea-change, broadly, by saying that there are reasons for thinking that the character of the computer environment of the future needs to become more childish, and more human, if it is to help human beings genuinely extract the best of themselves …"

-- Christopher Alexander, April 1997

Prehistory

In 1958 Christopher Alexander [CA] entered Boston's academic arena, then a pre-eminent global center for computing research. Technological metaphors permeated every field, some quite fruitful if used judiciously, such as *'the city as a machine'* and *'the mind as a computer'*. As part of his attempt at a technical understanding of what people meant by 'good', in art, architecture and urban planning, CA initiated several analytic computer programming projects, and published his results.

Another idea developed within this intellectual environment: ‘good structure’ as an *objective* quality. Instinctive, unconscious, genetically-endowed human faculties, shaped by natural laws, factor into the mind’s judgment of ‘good’ and ‘natural’, in the same way that the visual system judges height and distance. This judgment might, however, be obscured by experience and other instinctive mental faculties, analogous to the perception conflicts of an optical illusion. A rational theory of this ‘good structure’ mental-faculty could be constructed from empirical data, from experiments that explore its operation, using people as test subjects. Emerging cognitive sciences, linguistics and vision for example, were inching their way forward with this approach, and it was a thread running through *Gestalt* experimental psychology for decades.

CA’s analytical work ultimately led to the publication by 1964 of his *Notes on a Synthesis* of *Form* [*Notes*]. This was the real beginning of his influence on computing. Over the next decade, most prominent computer scientists engaged with ideas from *Notes.* For example, it was required reading among Marvin Minsky's colleagues and students -- the core of the Artificial Intelligence movement.

These were enthusiasts of automation. They were encouraged by the ambitious methodology in *Notes*, which presented a structured decomposition of complex real-world problems, with an approach to systematic construction of a solution. When I started programming in 1974, the engineering *zeitgeist* of "structured programming" was deeply entangled with the method of "hierarchical decomposition": of problems, models and programs. This was a direct influence of *Notes* and CA's other published work.

In 1963, CA moved to the San Francisco Bay Area, an important player in global commerce since the Gold Rush. It was then emerging as a major center for computing, which it would ultimately dominate.

While teaching at UC Berkeley, CA initiated building and planning experiments, notably those that increased the participation of non-professionals, the users themselves, in design and construction. He increasingly used the idea of a generic solution called a *pattern* to help participants understand the possible good they could do. CA adopted 'pattern', in this sense, from one of the primary meanings of *Gestalt*.

CA's first important project, in his own opinion, was a planning initiative for the University of Oregon, in Eugene. He reported on this in 1975’s *The Oregon Experiment*. In this book he quietly introduces site diagnosis using human *feeling*, as an *objective* measure and powerful tool, more powerful than the application of known patterns: *“our feelings for the life of the environment will always outstrip the current set of patterns.”*[p.156]

In 1977 he released *A Pattern Language* [*APL*], an eye-opening guide to good patterns found in good structures in the real world. It’s densely packed with interconnected ideas and observations.

The book inspired many devotees. It’s probably the best-selling, and most-read, book on architecture or planning in human history, setting a new standard for descriptions of complex human activity.

Among the readers chewing through its 1,171 pages were countless programmers, product designers and computer entrepreneurs. The book was infectious, and for Bay Area technologists it was, in part, considered a local brainchild. Among the many who promoted it was Stewart Brand, a quintessential local counterculture figure, founder of the *Whole Earth Catalog*, and chronicler of important early computer experiments.

The final book in the new trilogy, published in 1979, was *The Timeless way of Building*. In CA’s words, it was “the hardest to write" of the three. The most relevant bit for us begins on page 367: *Differentiating Space*, describing the smooth, stepwise creation of organic structure through increasingly specific global operations, whether performed by organisms, in ways we're just beginning to understand, or by humans using their innate faculties. The chapter introduces a long, site-sensitive narration of a design sequence, one that uses patterns so *lightly* that they nearly disappear from consideration. By page 464, there's a specific "from the outside, working inwards" building sequence for a home. I'll paraphrase it by providing titles:

1. Outside corner positions

2. Outside column Size and Place

3. Perimeter beams

4. Windows and doors

5. Room vaults

6. Outside walls

etc.

These are not patterns in the APL sense, but descriptive statements that explain clearly emerging global properties and structures. Still, most importantly, even this better tool is *far* less important than the feeling for natural structure, used while employing the tool.

Despite the shortcomings of the older approach to patterns, eight years after *The Timeless way of Building*, increasing interest in patterns among computer people led to a formalization, for an increasingly fashionable software engineering approach known as Object-Oriented Programming.

Almost a sequence

In their influential 1987 paper, two Oregon programmers, Kent Beck and Ward Cunningham, published what they called a small example 'pattern language'. It was actually a tested, ordered sequence. The intent was to empower users to define their own computer interfaces, and to demonstrate that something akin to CA's patterns could improve the communication of good engineering ideas.

The paper describes particular structures that should appear, or principles that should be kept in mind, at each of five stages during the design effort for a human-computer interface:

1. Window per task

2. Few panes per window

3. Standard panes

4. Short Menus

5. Nouns and Verbs

The importance of the right order, combined with global application of each principle at each step, was an aspect of sequences gleaned by reading the above passages in *A Timeless Way of Building*.

This part of the paper was also inspired by the 'user-as-designer' notion found in all of CA's work. Even though this sequence was only about particular aspects of certain kinds of interfaces, and was unrelated to the natural structures CA aims for, it was still an interesting interpretation.

But it wasn't the right interpretation, which is hardly their fault. This material is ripe for misinterpretation because, to understand it properly, people need to train to recognize natural structure.

It's not too extreme to say that the authentic transfer of Alexander’s ideas to the computer industry stalls because of this paper, rather than starting because of it.

The focus of the subsequent Software Patterns movement remained within the formal sciences, and so could not interpret CA’s work in the natural and human sciences: the application of human judgment using feeling, the smooth unfolding of natural geometry, and the task of helping people (programmers or users) to become more whole and alive. The criteria for ‘good’ were so different, that everything was misconstrued, from ‘pattern’ to ‘incremental’. Again, they aren’t to blame: this focus on abstraction, and the dismissal of feeling, and the reality beyond constructed formal systems, is endemic in the computer industry. Today, Software Patterns proponents, like most successful computer people, are not even interested in this cavernous disparity.

So what *is* a sequence?

A good unfolding sequence focuses the user’s unfettered concern upon nurturing the growth of natural structure. It encourages people to use their innate feeling for natural structure (which may require training to rediscover) to sensitize themselves to important issues as they emerge, so they will start to see them in any situation. It helps them to achieve organic growth: smooth, adaptive, harmonious, ever-improving and coherent. This unfolding growth should feel related to *morphogenesis.* A sequence is much like a genetic code, and drives increasingly complex structure, a 'complexity' that is still as simple as possible, coherent, and profound. A sequence is short, but accomplishes a great deal. Again, surprisingly little genetic information guides the development of incomprehensibly complex organisms. Sequence users put themselves into a certain state, a particular working modality. This is an actual, physical, easily perceivable phenomenon. It’s unfortunate that such phenomena are derided as ‘psychological’, even though that word literally means a *real* state within the brain, whose *actual* operation we should care about. Like most complex features of the human mind, we understand almost nothing about this phenomenon, yet. But we can *use* it, to do good work.

To evoke it, unfolding sequences and their presentations need to be created with great care and attention to effectiveness. In the 1980's, CA wrote and "debugged" increasing numbers of them, for clients and communities. These were presented mostly in workbooks, with trained assistants available to help.

Also in the 80’s, CA expressed disappointment with the effect of patterns on design in the built environment. He said architects use patterns in a shallow, unintegrated fashion, "like butter on bread". The fundamentals of cooking with love were somehow missing. Sequences were a possible key to unlocking people’s better judgment.

For example, in the early 1980's, for the *Moshav Shorashim* project, he produced a handbook with preliminary instructions, and 44 heavily-detailed steps, to allow a homeowner to design a house in harmony with a village. About the same time, for *Guasare, Venezuela*, he produced another manual, of 12 steps, also highly-detailed, for a very different environment. Here’s the ordered outline of required determinations:

1. size of house

2. limits of the house

3 - 10. layout of the *patio*; wings; opening; entrance; *sala*; *comedor*; kitchen; *verandas*

11. refinement of the patio

12. layout of the bedrooms and bathroom

Interestingly, these stages are huge efforts compared to the steps we'll see later in Gatemaker. But that's because they're *major* structures, and include exhaustive lists of considerations. But still, they are descriptions, rather than prescriptions, from which the user-designer is free to borrow. For example, step 10 includes:

** Verandas are covered open spaces where you can sit, have a drink, and they're also circulation spaces; the rooms open into them.*

** One of these verandas is very often located on the opening between the patio and the garden.*

The sequence workbooks were brimming with useful insights, but how much expert help *in person* was needed to use them?

This issue arises continually in CA's work. For example, the misinterpretations of the Software Patterns movement emerge from a lack of direct contact with CA's circle.

It's difficult to convey, through writing, the sensibility of seeking a particular subset of instinctive judgments. One needs exposure, in a real situation, to discover how to distinguish X from Y.

How can one consciously emphasize one part of one's mind over another? How does one even separate 'intense feeling' from other, equally innate faculties, that are counterproductive, such as thinking in terms of categories, properties, and objects? The last four centuries of scientific work have shown that the real world doesn't operate as the mind perceives it. And our brain doesn't work the way we think it does. Isolating a particular mental phenomenon cannot be done through introspection: we need to approach our own faculties experimentally.

One useful experiment uses the question 'what is alive?' Even if taken literally, that is, 'was this a biological organism?', we still seem to use the mental faculty of 'feeling' to answer it. It clearly factors into our unconscious internal definition of living morphology. So this question can be used as a self-education tool, *if* we take it seriously.

A complementary approach is CA's notion of a field of *centers*, which seems to help develop a perspective more in agreement with the modern natural sciences. This perspective is of a world made only of innumerable, mostly intangible fields with gradients of influence, called *centers*. There are centers between centers, centers consist of centers, and they interact with each other, sometimes reinforcing each other, sometimes not, creating centers that we might be part of, and might superficially perceive or measure. This may sound like a mystical worldview, but it's pretty close to current physics, chemistry and biology. The 'clockwork universe' was abandoned by science many centuries ago. And the 'computer universe' is just a related fiction.

So a world of centers is *not* a world of typical mental-constructs: no objects, properties, categories, connections, nor prepositions. Reality is outside of this normal conscious interpretation, and we become more sensitive to that, with the notion of centers.

Using the 'centers perspective', we get closer to the innate faculty that can detect life. That's an experimental result, which you can read about in *The Nature of Order*.

The 'unfolding sequence' also wakes up this part of the brain, reducing the facilitation necessary by CA's colleagues. There are many other ideas and techniques that can help. And even more that don't.

Good building requires this sympathy with the natural world for a simple reason: we live in this strange physical world, we are made of it, and we're impacted by it. Our logical faculties simply cannot encompass this everyday hyper-complexity.

Capturing good facilitation

Sequences also try to capture an effective client-architect interaction that Alexander uses.

He'll ask a client to close their eyes, and then to imagine, and answer, one step at a time, from large aspect to small, questions about the project they want. With the right facilitation, a keen sensitivity, and his personal ability to judge good structure, this is extremely effective at generating good results. A good written sequence is a very useful summary of this line of questioning. And almost anyone who worked with him could use it to achieve these results too.

But again, it's still an open question, many books later, whether or not users can figure this out *just* by reading. The record has not been so good. Besides the debilitating interference of the logical faculties, other forces are at work.

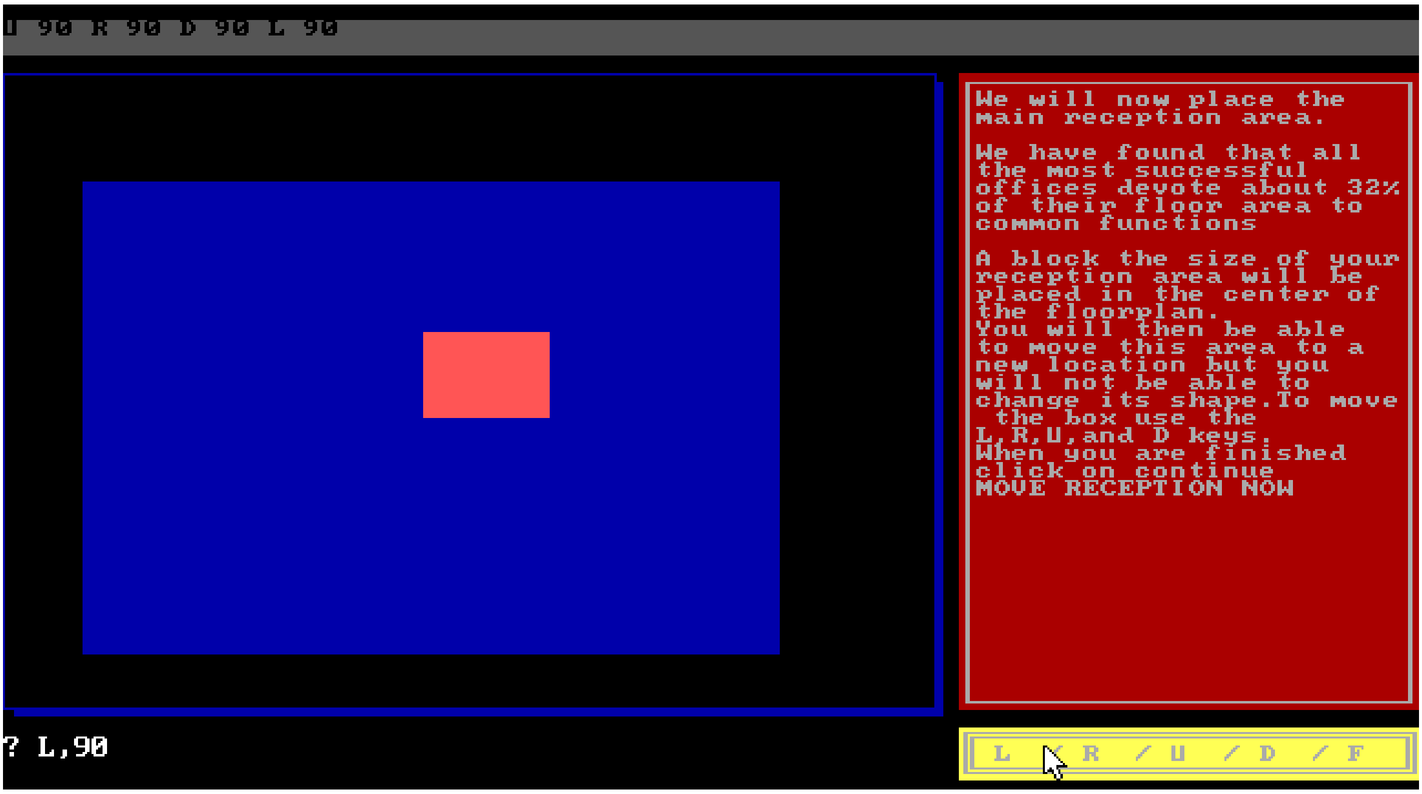
In the early 70's, one of Alexander's colleagues at the University of Oregon told him: unless you step away from the Oregon Experiment, and *see* if the process works without intervention, it won't really mean anything. He agreed with that.

Unfortunately, the Oregon Experiment *did* quickly lose its direction. Most technocratically-organized corporate or institutional bureaucracies will treat their own people with suspicion, and dismiss people's feelings as agents of anarchy.

A step further

Written sequences allowed a wider exposure to good structure. Why not try using a computer to *present* them, in ways that might achieve the right psychological state?

CA's office, the Center for Environmental Structure [CES], built a small machine, something like a flexible model, so the user didn't need to draw, but could just adjust the model while following a sequence. This was a sequence to design an *office*. By 1989 it was transformed into a program for a personal computer, using character graphics commonly available at the time (*fig. 1*).



```
U 90 R 90 D 90 L 90
```

```
We will now place the
main reception area.

We have found that all
the most successful
offices devote about 32%
of their floor area to
common functions

A block the size of your
reception area will be
placed in the center of
the floorplan.
You will then be able
to move this area to a
new location but you
will not be able to
change its shape.To move
 the box use the
L,R,U,and D keys.
When you are finished
click on continue
MOVE RECEPTION NOW
```

```
? L,90
```

Figure 1. 1989's Office sequence tool

By the early 90's, CA considered sequences a successful educational tool. After all, they were an ancient methodology. In *The Nature of Order*, he included his 24-step sequence for unfolding a Japanese Tea-house.

By 1996 the computer industry, flush with profits, and excited by their own interpretation of patterns, wanted to connect with Alexander, just to find out: what is he doing these days?

Computer People

As interest in patterns grew, we tried to nurture an opportunity to fund computer projects of our own, which would be focused on the built environment. We analyzed our options.

There were roughly two camps in the Software Patterns movement: (1) the original Hillside group led by Beck, Cunningham, Richard Gabriel, Jim Coplien and others; and (2) the 'Gang of Four' and the readers of their popular book *Design Patterns*.

Group (1) was interested in entire pattern languages, the applied order of patterns, combination strategies, and popularity with programmers. Group (2) was mostly interested in finding the best technologies -- patterns that should be enforced, and encoded into programmer tools such as frameworks. This was our impression, in any case.

Group (1) didn't want patterns to be forced, and didn't want to interfere with human thought. They promoted a workshop approach to writing patterns, which was pleasant and social. So we pursued connections with group (1).

I spent months, as did our business manager John Seamster, trying to determine which computer industry 'titans' might be helpful if approached for a grant or partnership. I felt that Bill Joy, co-founder of Sun Microsystems, who was a student at Berkeley for years, looked pretty open-minded. I wasn't sure how best to approach him, however.

In the midst of our search, in October of 1996, CA gave a speech to the major annual gathering of object-oriented programmers in San Jose, California. The OOPSLA speech is online. He made two important points:

1) The actual goal is to build *natural structure*.

2) Programmers could do *good* for people. They don't need be mercenaries to industry. They have more control than they believe.

The second point was forcefully made -- criticizing the industry's worship of success, and blind techno-positivism. CA implied a fight against monopoly capitalists that hire engineers only for their own ends. This part of the speech caused several people around me to wipe tears from their eyes. He spoke to a scattered idealism among computer engineers, which was then re-emerging as an undercurrent, because the world was turning upside-down during this time: the original dotcom bubble.

After the speech, I mentioned to Richard Gabriel that Bill Joy might be our best bet. He agreed, said he knew him, and so connected us.

Still, at this conference, I continued to see disturbing regressive intellectual trends within our preferred group -- mostly the kind of Behaviorist and pop-Darwinian dogmas common among Artificial Intelligence programmers, and incompatible with serious natural science. But we pushed on. People can change.

Days later in Berkeley, we chatted for a few hours with Bill Joy and his colleague John Gage. We found a great deal of common ground. Afterwards, I negotiated a direction for research with Mike Clary of Bill Joy's office. I said that CA and I would create something to help the built environment, but that I would report on the novel implications for engineering. Our business manager, John Seamster, then finalized the contract between Sun and CES.

With this budget, what should we build? Certainly, *some* kind of unfolding sequence tool. We couldn't know in advance what it would look like, because we needed to do hundreds of experiments, using ourselves and friends as subjects.

To make certain that I understood the potential and power of sequences, CA and I thoroughly discussed the mental toolkit in *The Nature of Order*: centers, properties, cycles, feeling. I sketched my way through sequences, with his feedback, until I became rather fluent in "applied feeling".

I remember some alternative cover designs then came into the office, for the first volume of *The Nature of Order*. One was nicely balanced, and became the leading cover candidate. But another design, which was probably just a mistake, reminded me of title pages from 16th and 17th century books. And then I realized: it was more *alive*. The title text was *pushing* against the borders. I pointed this out to him. He checked the pages, looked quite pleased, and the 'mistake' became the final cover. My first post-training contribution of judgment.

Recursion and the problem with fractals

At one point, we discussed 'sequences within sequences'. He was hesitant, because discussions of recursion tend to succumb to the wrong human faculties: logical systems over natural structure.

He read my article in RAIN Magazine from 1991, on 'the problem with fractals'. Although popular at the time, I was a dissenter, because fractals are such a poor idealization of the real world.

In nature, factors and interrelationships typically *change* at different scales. You can't find a Mandelbrot set in nature. L-Systems cannot be considered first-order approximations of trees. They tell us more about our perception than about trees.

However, we are stuck with our faculties, so idealizations are critically important. We must construct theories. Where would we be if Galileo hadn't dismissed friction while studying the inclined plane? But we must not confuse theories *with* nature. We don't understand nature. At best, we understand our theories.

I don't want to be hard on recursion. I was inspired by L-Systems and Phrase Structure Grammars to bridge the gap between operational and structural descriptions, with a simple programming language called *grogix*. I don't want to be hard on fractals: I believe that we might learn something about living geometry by studying some of the Hopalong equations.

We *see* recursion everywhere. That's partly because natural laws are universal, and so have effects we *find* similar at different scales. But humans possess a cognitive apparatus that's capable of cyclical combination, so we sympathize with it. But this sympathy can destroy our contact with other mental faculties.

CA knew that in a few months we'd never untangle these cognitive problems. But if we could expose normal people to just *one evocative sequence*, we'd really have achieved something.

Some time after discussing recursion, we sketched our way through his teahouse sequence. He drew one of the little gates. I prefaced my observation with something like "I'm not talking about recursive sequences, however …" I pointed out, "when you drew that *gate*, you definitely used a sequence within a sequence."

And so our target sequence was born.

He repeatedly drafted sequences for a gate, and I used them with paper and pencil. When it worked, we were ready to program.

We visited Doug Carlston, who built computer tools for home design. When we told him about sequences, he suggested we could brand them, like ‘Housemaker’, and ‘Officemaker’. We turned to each other and said *‘Gatemaker’*!

Gatemaker

We went to the UK to work, so Alexander could be at his new home, which was some 400 years old, and return to one of his childhood haunts: West Sussex. I stayed in the 2,000-year-old city of Chichester, a beautiful living structure in its own right. I worked in my 'office' in a B & B, and tried to implement some major aspect of the program each evening. In the morning I’d take a train-and-cab to his house, to show him the result.

CA was still finishing *The Nature of Order*. We’d look at my work, discuss it for many hours, find the most effective step forward, and then he’d go back to writing, and I to program.

These were experiments, so, when a feature didn’t improve the experience, we’d throw the code out. The improvements were hard to find: we disposed of 90% of the code I wrote.

Our workflow was his Fundamental Cycle. I’ll paraphrase:

1. Observe and evaluate the last thing you did, and its support for the goals, and its harmony with the existing work.

2. If it didn’t help, undo it.

3. Now, the group needs to think hard and decide what might be the next best step. That means the smallest effort that will make the biggest positive difference.

4. Build it.

Each cycle should be small but significant. We also incorporated this workflow into the user’s experience of Gatemaker.

No leaps

One of the overall design principles that emerged, in complete contrast with most computing: “don’t let the computer leap to conclusions” and change the screen. We grew increasingly skeptical of *any* automatic behavior, any perceptible ‘leap’ taken by the software, because it disrupted the mood necessary for harmonious work. The screen should be super stable, comfortable, reliable and quiet. It shouldn’t even seem to be there. Even word processors don’t have this level of gentle stability. We didn’t even let the application window change size, because this seriously changes the geometry of the work. The fact that this *does* effect people *should* give pause to designers of showy, kinetic, “entertaining”, and distracting, user interfaces.

Inside and out

We were certain of one principle, especially regarding computer programs with this extreme level of user sensitivity. It was a principle that the patterns people were *actively* rejecting.

Program structure should reflect the interaction with a user.

Most of the post-patterns world is quite confident and dogmatically *against* this, and there are patterns, such as MVC, which explicitly make it unlikely. Interaction with the user is considered an epiphenomenon, whose specific quality is far less important than the ability to generate *more* programs. People rarely gain from this underlying ‘flexibility’, and they lose, tragically, since the life of the user is not at the center of the work. It plays into industry goals. As a result, human-computer interfaces typically destroy our ability to do anything constructive, and put most programmers into blinkered specialist boxes.

Alexander writes that the whole is constrained by the parts. Interchangeable parts make an adaptive natural structure impossible. Programming environments, including those inspired by patterns, push this 'parts' view, which hampers sensitivity and true novelty. It's the worldview of the factory-builder, and these tools for mass-production logistics have little regard for people.

When underlying code is considered different, and more important, the tendency is simply to trash hard-won efforts to improve the user's experience. This happens all the time in the industry, and Google is among those pushing the lunatic idea that we may not need people to create user interfaces.

This is a complete divergence from CA's work, where the underlying structure is in *harmony* with human interaction. Computing is not 'somehow different'. It is still a human tool recruited for human purposes, and the principles still apply.

Comfortable setting

Using feeling as a measure, we did our best to make the program comfortable, playful, approachable, fun, child-like, joyful, and relaxed. Most importantly, it feels positive, *not* neutral.

We set the mood with this static opening screen, followed by a quiet replay of a gate unfolding with this sequence (*fig. 2*).

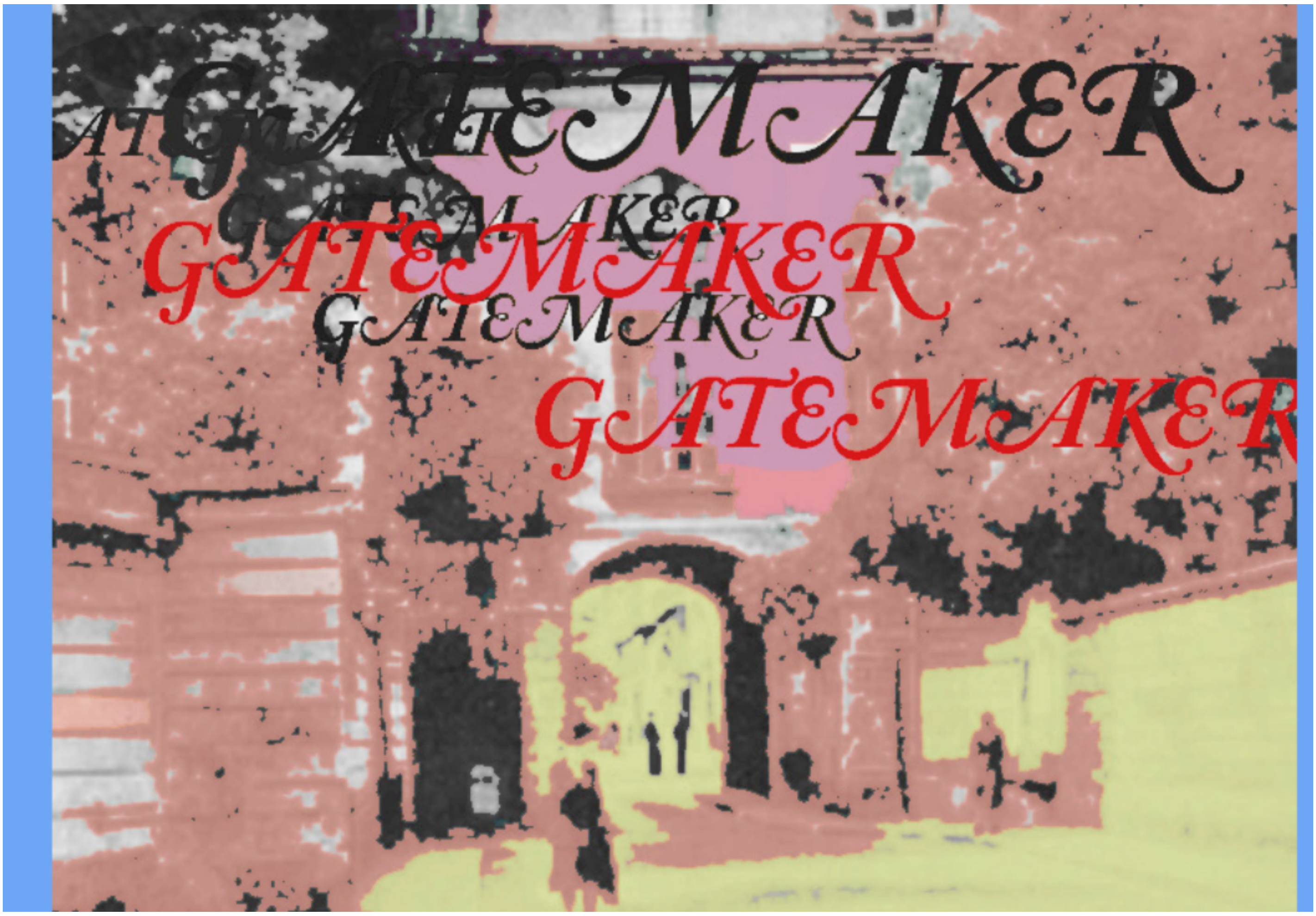
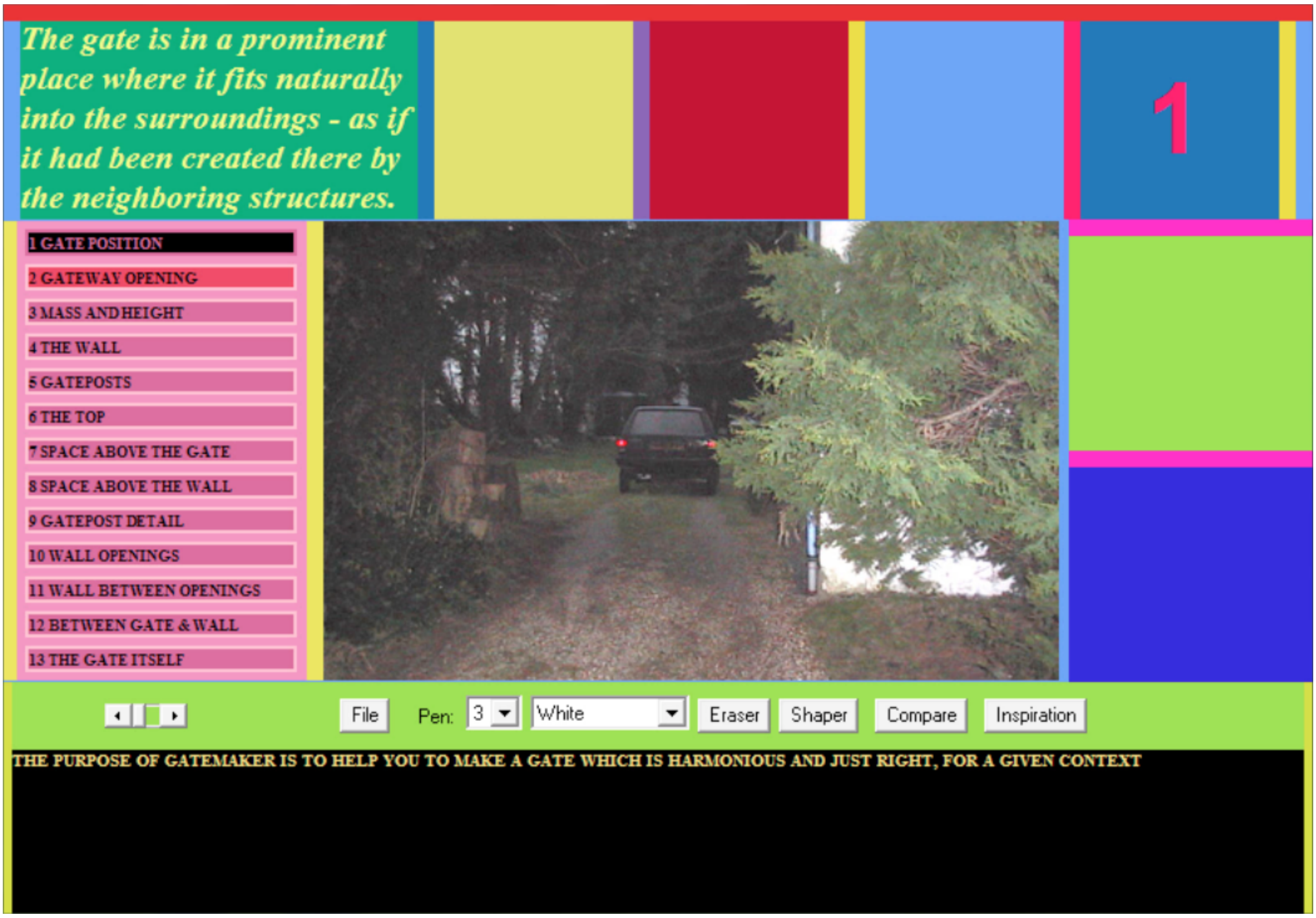


Figure 2. The initial screen.

We resisted anything that looked like a "professional" computer product, with all their attention to superficial style. Interestingly, considering this program was built almost 20 years ago, it still looks fresh. All the “professional” applications of 1997 -- ‘hot’, ‘fashionable’, ‘expensive’, and ‘leading-edge’ -- look terribly dated. Gatemaker is essentially a *timeless* computer interface. This wasn’t difficult. It was just *built* very differently (*fig. 3*).



The gate is in a prominent place where it fits naturally into the surroundings - as if it had been created there by the neighboring structures.

1 GATE POSITION
2 GATEWAY OPENING
3 MASS AND HEIGHT
4 THE WALL
5 GATEPOSTS
6 THE TOP
7 SPACE ABOVE THE GATE
8 SPACE ABOVE THE WALL
9 GATEPOST DETAIL
10 WALL OPENINGS
11 WALL BETWEEN OPENINGS
12 BETWEEN GATE & WALL
13 THE GATE ITSELF

Pen: 3 White

THE PURPOSE OF GATEMAKER IS TO HELP YOU TO MAKE A GATE WHICH IS HARMONIOUS AND JUST RIGHT, FOR A GIVEN CONTEXT

Figure 3. The work environment.

This timelessness and deep feeling were not much appreciated by the professional computer people, who criticized the interface as "amateurish", “unfinished”, "not 3d enough", “not slick”, etc. They didn’t understand that those properties were *intended* -- even after we told them. It’s the quiet *playpen quality* of the interface that makes it possible for the user to achieve something that makes them feel more *whole*.

Drawing on reality

The program asks the user to select a site photograph, ours or theirs, to draw upon. We wanted the user to judge the harmony of their emerging gate within an actual setting.

The photo is crucial because design-with-feeling feeds upon the complexity of reality. Computing technology, and blueprints, fight against our understanding of the site. Someone new to this cognitive problem will need the photo as a reminder.

Only a photo, though. We didn't want to drown in exact measurement issues, or a particular construction technology. If a good design resulted, the actual construction, on site, needs to recapitulate the process, using the drawn design as a guide.

It was also true that running the sequence more than once produced better results -- as long as the user remained motivated.

The Sequence Presentation

On the screen of this comfortable, colorful work environment, there were four position indicators for the 17-step sequence.

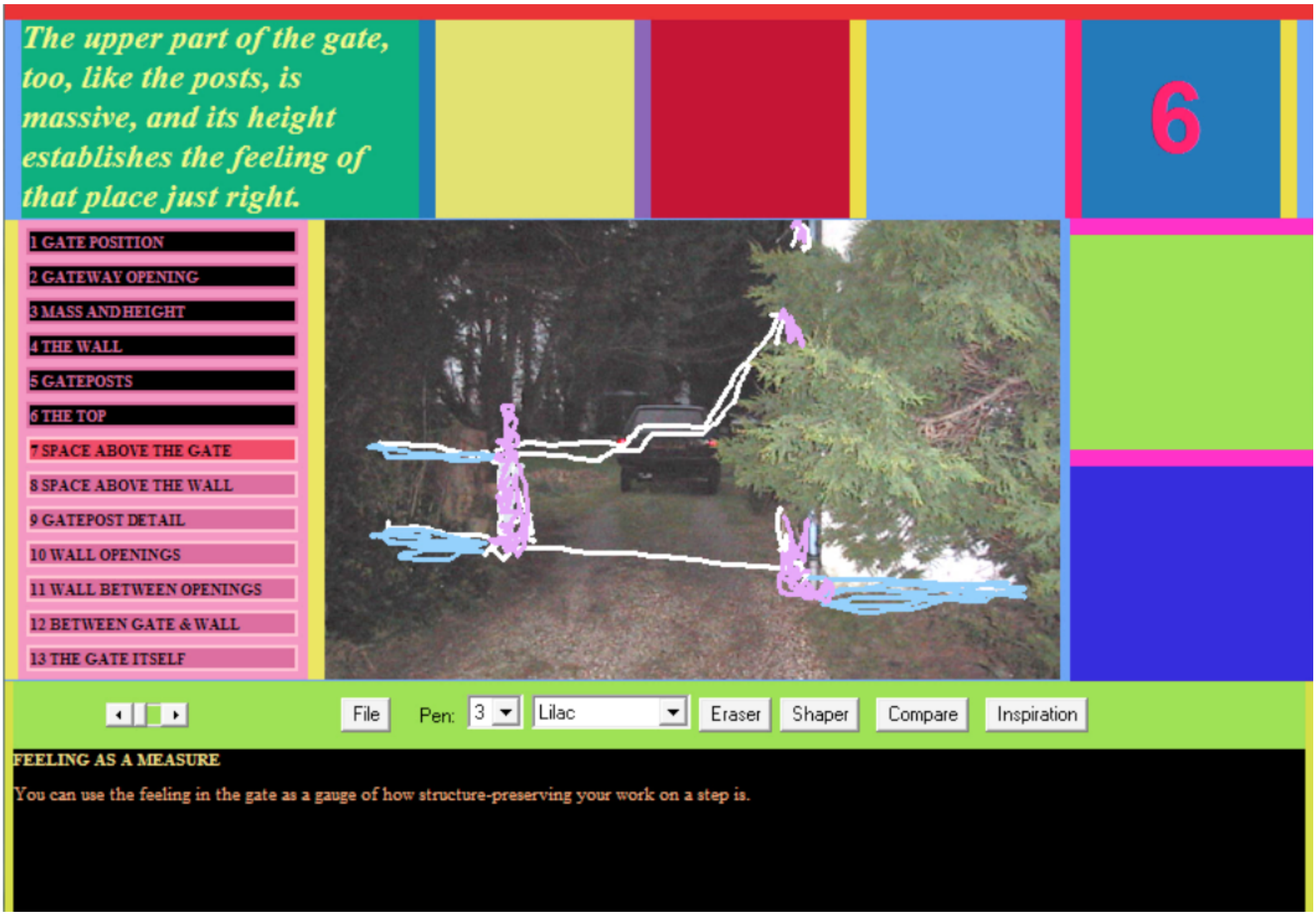
* the drawing itself

* the description of the step, in the upper left-hand corner

* a big, fun number in the right-hand corner

* a "sequence viewer" on the left which indicated, with colors: steps completed; the current step; and next steps.

A sense of moving forward helps the user maintain the effort needed to make something increasingly beautiful (*fig. 4*).



The upper part of the gate, too, like the posts, is massive, and its height establishes the feeling of that place just right.

Pen: 3

Lilac

FEELING AS A MEASURE

You can use the feeling in the gate as a gauge of how structure-preserving your work on a step is.

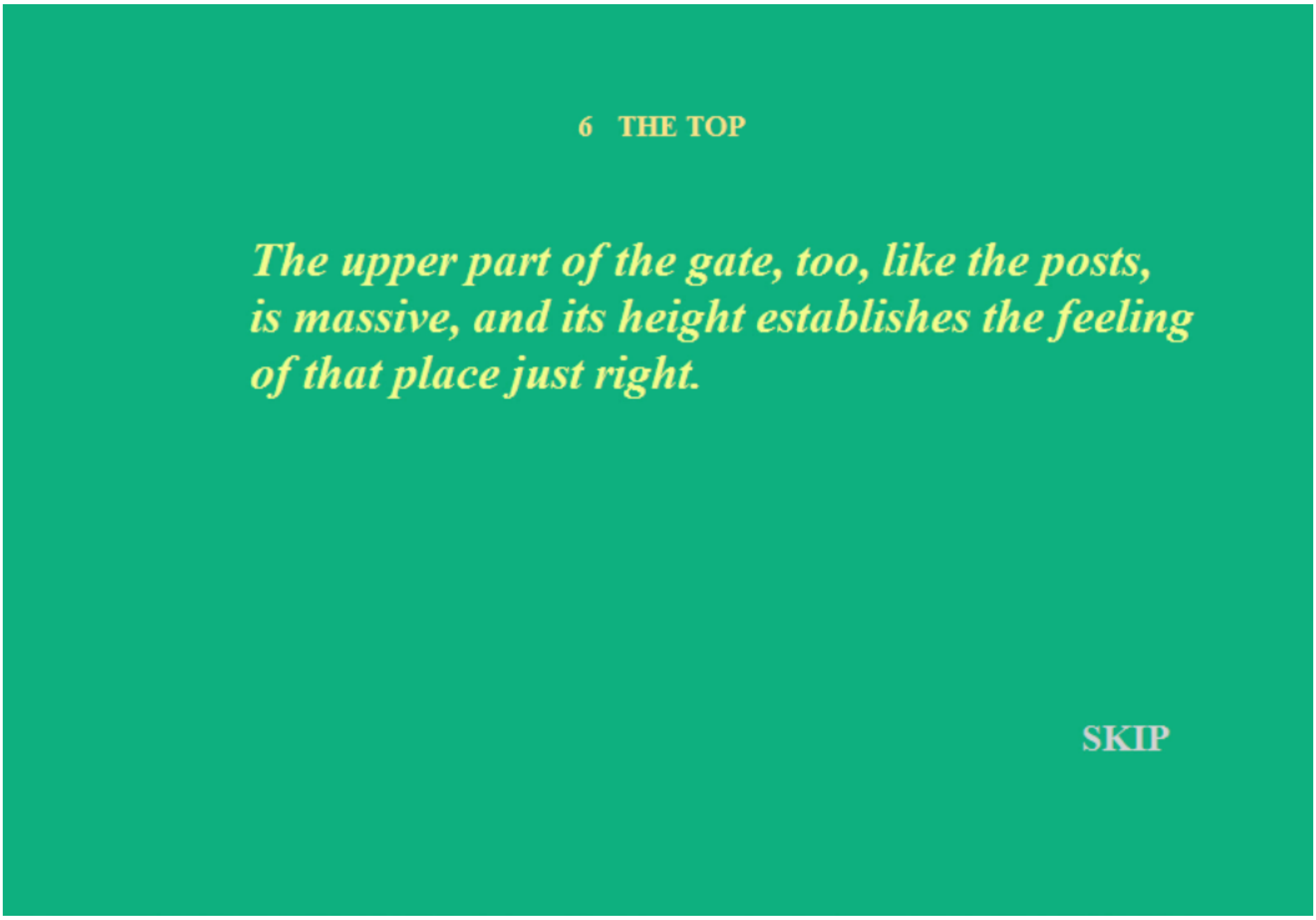
Figure 4. Sequences were visible.

Although this screen was a good work environment, we found it insufficient to attract focus to the *description* of the step. When our sketchpad is in front of us, it captures all of our attention.

We wanted people to be fully conscious of each proposed step, then let the work proceed. Otherwise the sequence can't help them to experience smooth unfolding. Of course, they also need to feel that the advice is optional. But they need to *see* that advice first.

So, at the start of a step, a full-screen description screen is presented, with nothing else. The user then clicks this screen, to get back to the photo-sketch. Or they skip it (*fig. 5*).

This flipping back-and-forth between 'imagination' and 'reality' has a profound effect, and begs further investigation.



6 THE TOP

The upper part of the gate, too, like the posts, is massive, and its height establishes the feeling of that place just right.

Figure 5. The step screen

Nouns and descriptions

The step-description screen does not work if the step, in any way, is oppressive, dogmatic, or an instruction.

No matter how gently we said "now do this" it sounded like a shouted command. Even words like "must" and "critical" sounded like unjustified dogma.

The screen needed to get its point across -- that at *this* moment in the sequence, the user should see that certain kinds of structures are *latent*, ready to emerge.

So the steps were written as if describing something the user might soon discover.

If the user had already created some of those structures, they could feel good about it. The steps sounded optional, more like really good suggestions, so the user felt free to apply the suggestion, everywhere in the drawing they felt it appropriate.

The Inspirations

One could always leave the program, and get away from the computer, for inspiration.

We also provided an inspiration *button*, an inspiration window, and sayings, to help people discover the point. These were unlike steps, and could be written loosely. There's also a full list of these inspirations, if you click the window. At the top we wrote:

The purpose of Gatemaker is to help you to make a gate which is harmonious and just right, for a given context.

Any time you clicked the inspiration button, one of them would slowly fade into view:

STRUCTURE-PRESERVING

To be successful, you must be trying, all the time, to do that thing which does the least harm, and the most benefit, to the structure of the world where the gate is going to be. We call each step you take in this spirit a structure-preserving transformation. If you want more information on what a structure-preserving transformation is, click on this sentence.

MOST STRUCTURE-PRESERVING

Was the step you have just taken, the step which is MOST structure-preserving? Can you think of another step, which would have preserved the structure better? If so, you can modify the gate, by using SHAPER or COMPARE.

During the process of unfolding the gate, you will make a series of steps. At each step, you are working on some one center. Your effort should be to make this center as wholesome and harmonious as possible. Above all, each center you make should be the simplest center, which can be made, to get the thing right.

FEELING

One way of deciding whether a step is structure-preserving, is to ask yourself if it has deep feeling. You should try, as far as possible, to do that thing which has the most feeling. Making a thing which has feeling, and making something which preserves the structure of the world as deeply as possible, are almost the same.

FEELING AS A MEASURE

You can use the feeling in the gate as a gauge of how structure-preserving your work on a step is.

STRUCTURE PRESERVING AS A MEASURE

You can also use the structure-preserving character of a step, and your estimation of just how structure-preserving it is, as a gauge of your own feeling. Gradually you will notice that the more structure-enhancing something is, the more it makes you feel at peace with yourself.

OVERALL AIM

Our aim, in the process, is to make a gate which, when you look at it in the context it is in, leaves you feeling as alive, and as whole within yourself, as possible.

CRITERIA

There are a number of other ways of paying attention to the structure-preserving character of a particular design: these include ALIVE, MIRROR OF THE SELF, FEELING, WHOLENESS. They are all equivalent in their deep meaning, but each one gives you a different view of the same underlying reality.

The Sequence Itself

1 GATE POSITION

The gate is in a prominent place where it fits naturally into the surroundings - as if it had been created there by the neighboring structures.

2 GATEWAY OPENING

The opening of the gate is a size which makes passage seem intimate: whether it is large, for trucks and cars, or small, for man and a dog.

3 MASS AND HEIGHT

The mass of the gate establishes its proper psychological weight in the place where it is standing.

4 THE WALL

The gate sits in a wall, which extends the feeling of the gate outward.

5 GATEPOSTS

Where the gate meets the wall, there are two massive gateposts: thin for wood, heavy for stone or masonry. These posts defined the opening of the gate.

6 THE TOP

The upper part of the gate, too, like the posts, is massive, and its height establishes the feeling of that place just right.

7 SPACE ABOVE THE GATE

The sky above the gate has a particular shape, which in turn helps to shape the solid material of the upper gate.

8 SPACE ABOVE THE WALL

The line along the top of the wall, too, makes the sky a beautiful shape.

9 GATEPOST DETAIL

Each gatepost is ornamented with small structure that makes the gate more meaningful and solid as a center.

10 WALL OPENINGS

The upper part of the wall contains, occasional openings, which allow a glimpse of what is on the other side.

11 WALL BETWEEN OPENINGS

The pieces of wall between the openings have a definite and beautiful shape.

12 BETWEEN GATE & WALL

The angle between gate and wall is filled with a definite central shape which connects the two.

13 THE GATE ITSELF

The gate fills the opening, fully, or there is a gap in the upper opening.

14 GATE LEAVES

The leaves of the gate have a definite repeating pattern which makes each leaf a beautiful thing in itself.

15 ORNAMENT

At the top of the gate there is an ornament or crest where the gate meets the sky.

16 GATEPOST BASE

At the base of the gateposts there is a very solid entity which helps the gate to meet the road.

17 WALL ORNAMENT

There is a modest repeating pattern along the top of the wall, which makes the wall harmonious.

Orientation

We wanted the program to work without guidance from a live person, but the number of experiments necessary to make this possible, began to eat away at our time.

We experimented with 'helpers', essentially post-it notes, that could help explain the interface. The common solution is a kind of tip-over or roll-over, but they were distracting, as our 'no leaps' principle predicts. A pretty good solution was a 'help' toggle button that would fade-up, and fade-down, *all* helpers across the screen. We had many more ideas to pursue, but ran out of time.

Hill-climbing

If you want people to really dig down and use feeling, they must feel free to 'undo' their work.

But it would be even better if they could do something like a two-part preference test. They take one step's worth of work, which they believe is *ok*, 'store' it, and try again. Then they *compare*. They click back-and-forth to see which direction they want to go. This is "hill-climbing", named after an ant's postulated "heuristic", or simple method, for determining which way is "up" and which is "down".

We used the 'compare' feature ourselves, quite often, but it took work to get our users interested. Although simple, a tool like this needs some kind of tutorial explanation.

Besides 'undo' and 'compare', we avoided any kind of elaborate back-tracking, like the features in document and source-control systems. CA felt that a really *good* sequence made 'backtracking management' unnecessary and counter-productive.

By the way: the famous inability of hill-climbing to find anything but the local maxima is irrelevant here. We are trying to build the best gate possible, in harmony with a specific place.

Studious

At the start, we play a previous user's experience. During the session, we auto-save every stroke. We replay this user's session after the last step. These features help them perform better, but also provide a sense that they're studying something serious about nature and people, and contributing to a larger conversation.

Shaper

The shaper helps you to shape negative space. It temporarily inverts color in your drawing (not the photo) so the negative is visible. You can then give it a beautiful shape. It worked, and we used it often, but required extra tutoring.

Aspen

We brought this program to Bill Joy's Aspen office, and he had a number of guests, who tried the program and discussed it for days.

The specific geometries of the user interface, and specific presentations to the user during Gatemaker, create a particular kind of user experience. No one had tried to facilitate this user experience on a computer before. And we felt it was important.

But it was hard to get computing folk to focus on the actual human effect of the program. In fact, they saw it, experienced it, momentarily agreed that it was surprising, but soon forgot it.

To be fair, they simply weren't equipped to discuss what we'd accomplished. They were not natural scientists, nor activists. They shared 'pragmatic' and fashionable industry viewpoints that make it nearly impossible for anyone to discover anything new or important about people & computers.

A proposal

At Aspen I presented a paper: what would good programming look under the influence of Gatemaker and *The Nature of Order*?

Unfortunately it was too far away from contemporary programming to be understood.

I suggested a research program using a Gatemaker-like development environment to facilitate the fundamental cycle. It would record steps, and ideas, and would allow us to study the impact, on the unfolding of a program, of different techniques, sequences, steps, and concepts. It would be an environment where the user could experiment upon themselves, while getting work done. This would lead to better understanding of the mind, of people, and of the effects of different aspects of computing.

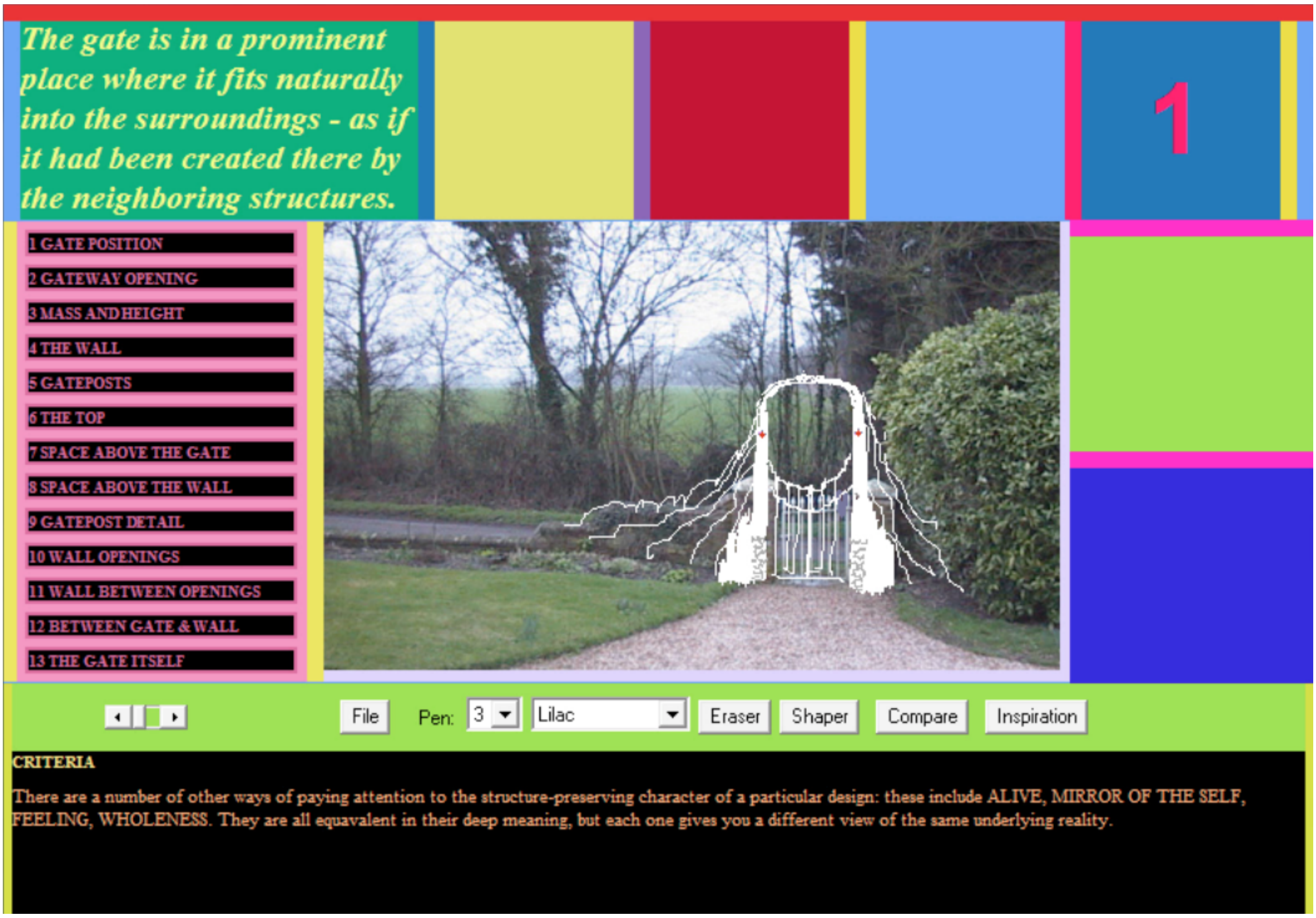
The software patterns literature has no such research initiative. Instead, they focus on objects, properties, types, lists, titles and categories. It looks like butterfly collecting, with no drive to build a theory with explanatory adequacy. It's Natural History instead of Natural Science.

Not everyone at Aspen was a computer industrialist. And there was one person, a rock musician with a strong interest in people, computers, and Alexander's work, who had something he wanted to say about Gatemaker.

Peter Gabriel

"I'll probably sound like an old hippy ... I guess because I *am* one ... But it was a very *Zen* experience... *'Man'*."

Peter Gabriel had long been affected by *A Pattern Language*. He tried Gatemaker, along with the tutoring we were giving everyone. He produced a very good gate, even though he'd never really drawn anything before (*fig. 5*).



The gate is in a prominent place where it fits naturally into the surroundings - as if it had been created there by the neighboring structures.

CRITERIA

There are a number of other ways of paying attention to the structure-preserving character of a particular design: these include ALIVE, MIRROR OF THE SELF, FEELING, WHOLENESS. They are all equavalent in their deep meaning, but each one gives you a different view of the same underlying reality.

Figure 5. Peter Gabriel's gate

Even on the first day, there was something else he wanted to say about the experience, but he couldn't quite put his finger on it.

He said he felt kind of compelled down a rabbit hole, where he was quite uncertain of what he was doing. He kept wanting to take a break and study. But this "compelling" didn't sound so bad. The tool was providing good motivation, which was kind of amazing.

A few days later, Peter Gabriel tried to express it differently. The *driven* nature of the sequence, he said, made him want to escape. "It's a little too *male*".

This struck a chord. I have a note from Alexander: "I think we should make a special effort to interest women in this process."

Whatever the phenomenon of the sequence *is*, it would be more effective, and more nurturing, to allow someone 'tutorial escapes' anytime they want, before jumping back into the whirlpool.

The Upshot

At the meeting, we produced business plans and proposals for moving forward. But there were problems.

First, the relationship between our goals and those of the existing Software Patterns movement were tenuous, and in many ways antagonistic. Sun was about to make patterns into a major product: the frameworks for Java. We didn't want to suggest support for this.

Second, there was a growing, intoxicated excitement about the web. Everything was changing. The bubble that would pop three years later was expanding at an accelerating rate. So nobody could focus on our subtleties regarding nature and people.

So, it did not work out. Over the next few years, I wrote proposals for Gatemaker as a web service, with no industry interest. At CES we built useful websites, but nothing with Gatemaker's ambition.

In retrospect, the possibilities for industry support were illusory, like much of the idealism of the first dotcom boom. The only way to really make progress, is to take these ideas to the people.

So, was Gatemaker used to build a gate?

Olga Volchkova is a talented artist that Christopher Alexander and I met in the Bay Area during the dotcom boom. She needed to build a gate, and wanted to try this 'Gatemaker' we talked about so much.

This was no normal test. It's important to mention that she is thoroughly, classically trained -- in proportion, observation, technique and process – with vast experience. She found Gatemaker's sketching tools a bit primitive, but she worked through its 17 steps, and simply memorized them, so she could follow them in her mind, while using normal art supplies. She printed a site-photo onto paper (*fig. 6*), and sketched through the sequence again (*fig. 7*). Marty Jones, a talented Berkeley carpenter and rock musician, agreed to build it, even knowing we only had Olga's sketch, a sequence, and *feeling*, to determine the exact measurements during construction.

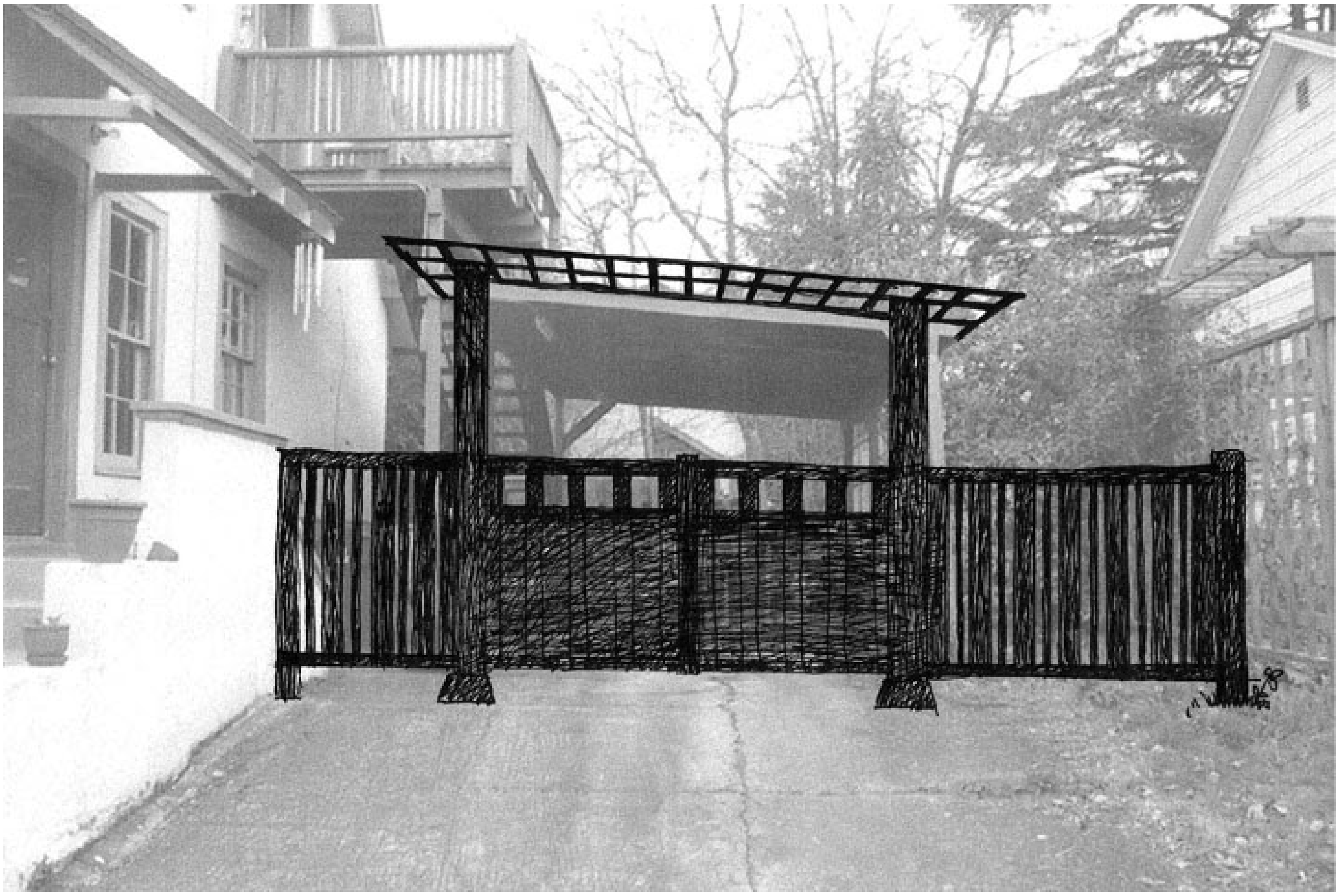


Figure 6. The gate site

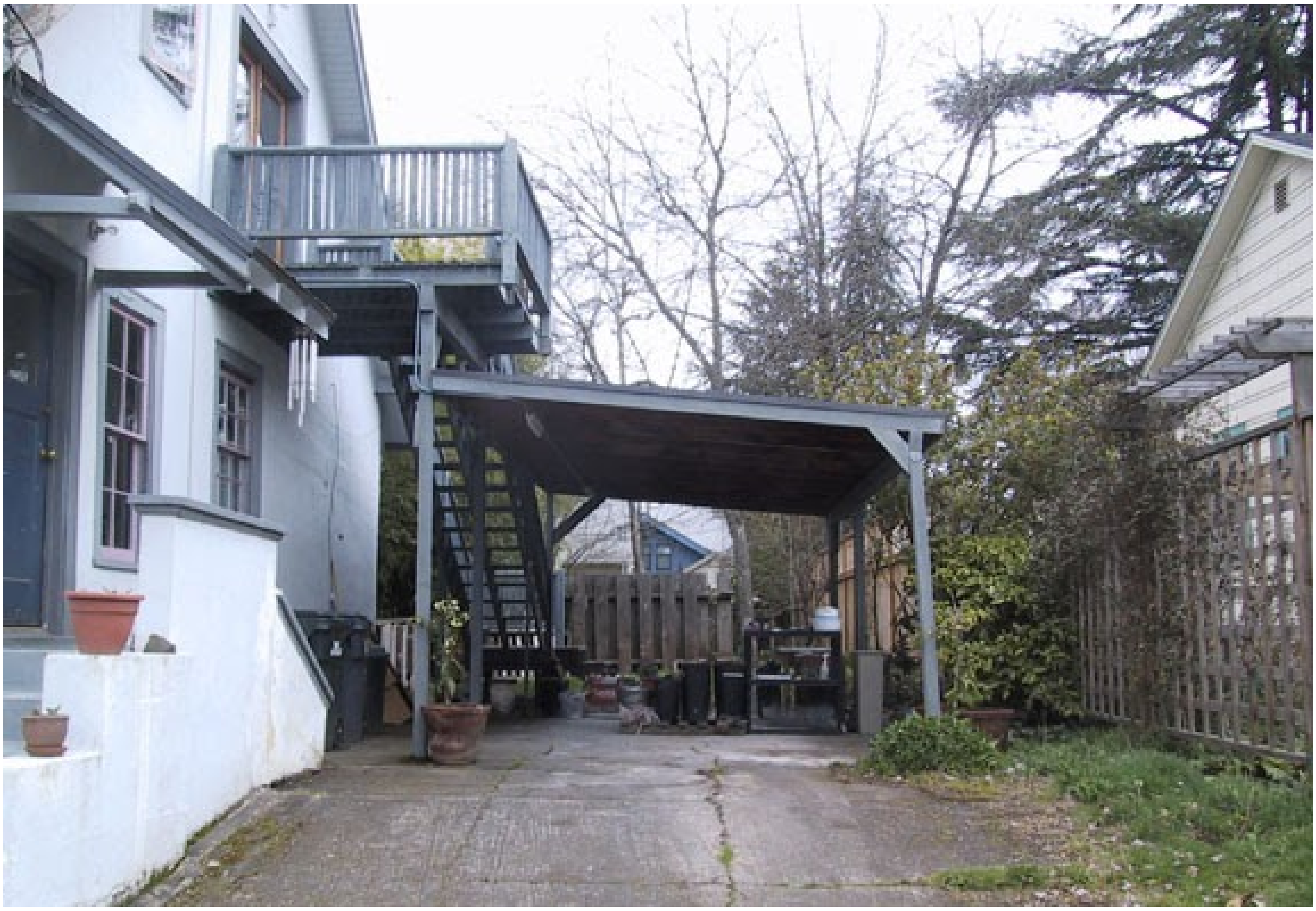


Figure 7. Volchkova's final gate sketch

Since the sketch was sequence-based, and we had the sequence itself, this was a breeze. Marty created mechanisms for each step so we could make small adjustments, to be harmonious to the actual site during each step of construction. We would rough-out Olga's clear response to each step, and then micro-adjust for feeling. With *surprising* speed, we built a completely custom, site-specific, harmonious gate (*fig. 8*).

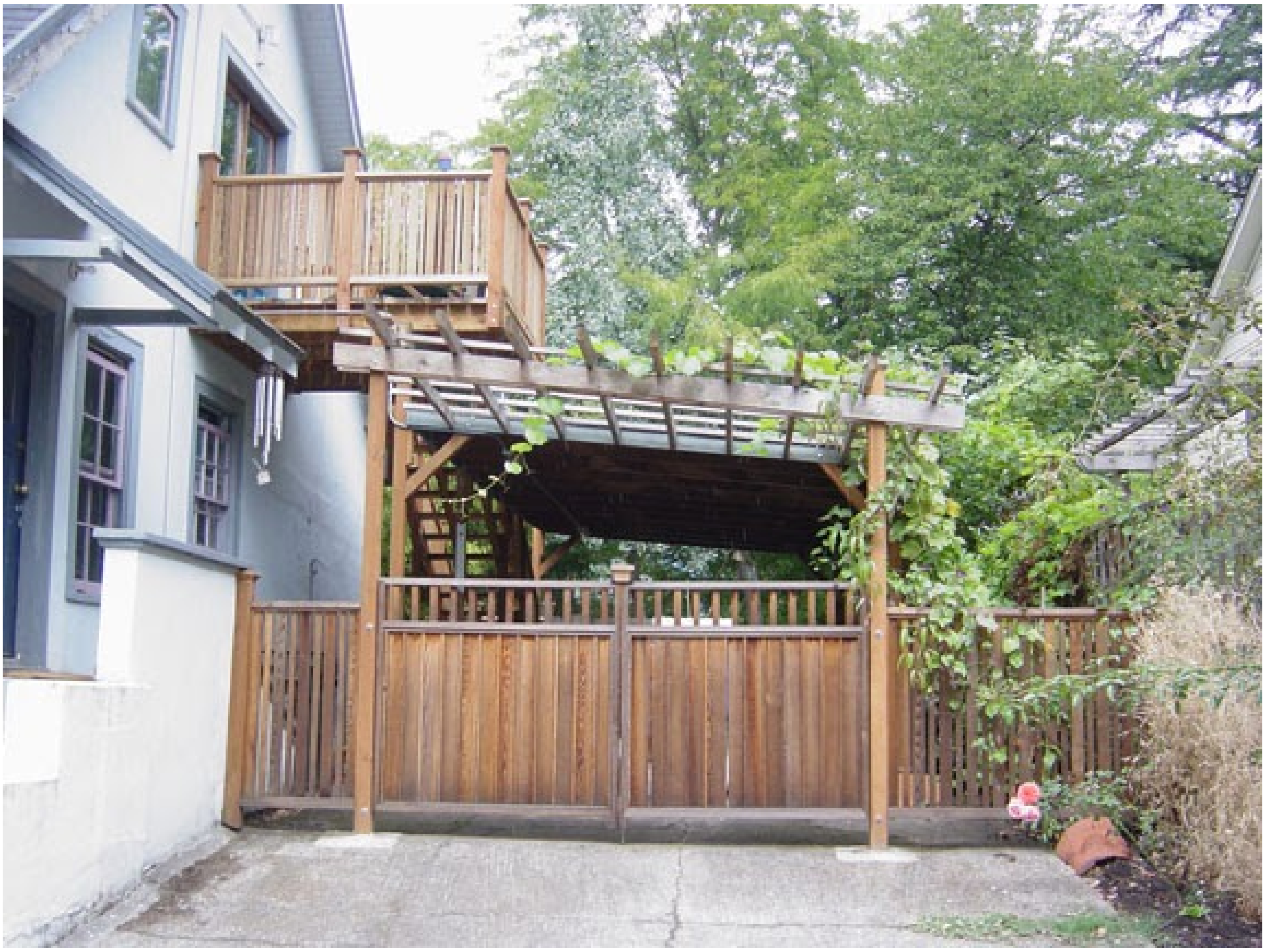


Figure 8. The resulting gate

Olga agreed with Alexander that the unfolding process in Gatemaker was very like sequences in traditional craft, and more generally in painting and sculpture.

She's also a trained restorer, and owns books of detailed sequences from the great Soviet restorer and painter Igor Grabar. Although meant for practitioners, shouldn't those sequences be published? CA once told me: "a good sequence is as valuable as gold." I'm thinking of the public good.

Olga felt publishing would be a *very* weird idea. She said the sequences are important, but almost useless by themselves. It takes training, by someone *already* trained, to fully understand them. How else could someone even *recognize* specific situations? How could they know *how* to do the right thing?

This goes back to the basic problem of quality, and the need for a culture of support. Peter Gabriel *knew* he needed training, assistance, media, information -- anything to help with self-education. He was familiar with real-world artistic endeavors, and knew that he didn't know enough. It's interesting that seasoned artists like Peter Gabriel and Olga Volchkova could treat Gatemaker as a tool for real work, but computer people couldn't.

The 'Same' Sequence

After the training needed to use this sequence successfully, the 'same' sequence can be used for almost anything.

Olga used it for trellises and buildings. I used it for several features in a dancehall, and for the first mobile device interfaces. I only need to substitute context, using gate and fence *centers* as metaphors for the *other* emerging centers. After a while, you can always see the proper scale of the next step that will improve the life in something.

Epilogue

"Bill Joy, at our meeting with John Gage and Greg Bryant (October 1996) stated his hope for a creative process, within the computer environment, that is more humane, and more like real creation than what is currently possible ..."

"None of the computer scientists present at the Aspen meeting fully grasped, I think, that the requirements of … a more deeply intuitive computer environment – necessarily imply that the resulting changes are likely to be suspicious-looking, disturbing, even offensive at first to the trained computer eye."

-- Christopher Alexander, April 1997

The Future

On the computing side, I'm trying to initiate two things:

1. A human-interaction-first toolset, that treats programmers like human beings. I tried to garner interest in using sequences directly in software, here.

2. A related study of the cognitive faculties recruited by people who program, to create a better foundation for computing, and tools that can help programmers to pull away from the total domination of formalism.

But in the built environment, I'm continuing to pursue neighborhood revitalization tools, based on my own community organizing.

The next experiment is available on *Urbanology.com*, and I'll present a paper on this at the next PUARL conference. *Urbanology.com* also contains a carefully-reconstructed web version of *Gatemaker*.

Most of my writing on this subject is published at *RainMagazine.com*, along with videos of Gatemaker in action, and videos of Alexander on related topics. My contemporary account of the meeting itself, along with some documents, may be found at *Gatemaker.org*. Post-Aspen memos on gatemaker by Christopher Alexander can be found here.

2020 update: My current initiative is *Beautiful Software*, launched under the purview of *Building Beauty*, a school run by a network of Christopher Alexander's former students and colleagues.

Olga Volchkova, Maggie Alexander, and Christopher Alexander in Berkeley, California.

Peter Gabriel and Christopher Alexander in Aspen, Colorado.

INVOICE

Village style: Christopher Alexander's Berkeley kitchen with casual flower arrangement by Olga Volchkova.

Photos © 2013 Greg Bryant