Chapter 4

Beyond Formalisms: The Art and Science of Designing Pliant Systems

A Talk with Austin Henderson and Jed Harris

Austin Henderson has a Ph.D. in Computer Science from MIT and has been in the field of Human-Computer Interaction for more than twenty-five years. He has built applications in several areas including manufacturing, air traffic control, electronic mail, user interface design tools and workspace management. He has done research and user interface architecture for Xerox and for Apple Computer. In addition to being a principal in Rivendel Consulting, Austin is a co-founder of Pliant Research, a research consortium exploring the theory and practice of computing systems that move beyond the formal. Austin has been active in ACM/SIGCHI since 1983, including as conference chair and organization chair.

Jed Harris studied computational linguistics, computer science and the philosophy of science at MIT and Stanford University. He has worked in artificial intelligence and computer systems research for over twenty-five years, at Data General, Intel, and Apple Computer. He was a founder of the OOPSLA conference and a co-architect of the OpenDoc component software standard. For the last three years has been a member of a venture fund in Silicon Valley. He is currently exploring the feasibility of new technologies for implementing active patterns. He co-founded Pliant research with Austin Henderson in 1997.

KLAUS: I'd like to discuss the concepts and the framework that you present through Pliant Research, for instance at conferences such as CHI in Pittsburgh in '99. So let's start at the beginning. What is the problem with computing as you see it at Pliant Research?

JED: Well, one of the problems that we have with the whole Pliant story is that there are so many different ways to begin. It depends on the specific audience and it depends on the kind of interests involved. Maybe for this context we see the problem as rigidity in the existing computer infrastructure. That results in increasing tension over time between the social reality and the computer infrastructure because social relationships naturally evolve over time, meanings drift or are negotiated, whereas the computer infrastructure just stays static until there is a big investment to make a change.

AUSTIN: And a related problem is that the ideology of building computer systems is that you may talk to lots of users and if you really do your homework well you may get all the viewpoints. But then you settle on *one* way of having the things in the machine. They may have different meanings to different people, but you are still stuck with having only a single thing in there. It's not going to move and it is also singular. So the idea that the machine could be part of the process of dealing with the different perspectives bashing against each other and thereby discovering meaning is not thinkable within our current computing ideology, and as a result, not possible with our existing technology because the machine has only one view point and doesn't support multiple, local viewpoints.

KLAUS: At the CHI '99 presentation you argued that computing is very much tied to a certain perspective on organizations or, as I would call it, a certain "bureaucratic discourse". Could you expand on that?

JED: Well, I think that it's an example of how the audience shapes the way we tell the story. That certainly is one particular way of slicing it. Another way of slicing computing is that it is very tied to a certain view on meaning. You might say that it is tied to a truth-value kind of meaning or a predicate calculus style of meaning. There is another way at looking at it, as Austin says, a certain epistemology, that there is one world and one ultimate perspective on that world that makes sense and everything else is just a special case of that. But certainly we see those different ways of looking at it as interrelated in a social sense. They aren't just separate frameworks. And as you say, the bureaucratic discourse is probably one of the things that we regard as the source of that. In building organizations out of human beings, there has been an attempt to constrain people, to limit them and to keep them operating in a consistent way in order to prevent the organization from dissolving into chaos. That has tended to generate, in one way or another, all these different, more rigid perspectives.

AUSTIN: You can go back and look at it historically. I'm not an historian, but if you take a look at the bureaucracies prior to 1850, they were relatively small organizations. There might have been a reach which was larger, but still most of the management and the organizational components of the companies all happened in one room or in one building. Consequently management was something that could be done by a few people bouncing off each other and a lot of it not having to be very explicit. With the arrival of

the telegraph and the railroad - and the reach of big companies like Joanne Yates describes in her book[i] - you suddenly had a distributed organizational problem, which can be divided into three interrelated components: size, coherence and local realities. In certain social organizations, including large corporations, but I think much more broadly than that, you have these three components: You have the need to respond to local circumstances, to fit in and to couple with what is going on wherever the local presence of the company is; you have the need to keep those separate vectors coherent; and you have to do this in the face of getting big.

The way that organizations tended to deal with this prior to computers was by the invention of scientific management and the idea of having rigid, well-defined file systems and things like memos, which have particular forms and a bunch of other things which took a lot of inventing. In that sense modern organizations were invented between 1850 and 1920, as Yates describes. So what they had done was to get this single file system, these single answers lined up, all marching in exactly the same direction. They built this structure out of people. And suddenly with the arrival of the computer in the late fifties and early sixties there was a realization that here was just the tool they needed for keeping everything exactly aligned. That was the inevitable response to trying to deal with this.

Now, *maybe* you had to have only a single way of doing things when this management structure was being constructed entirely out of people, but the interesting thing to us is the thought that the computer could be used in a completely *different* way. The computer can be used to enable more than one view, to handle the local circumstances and provide some of the infrastructure that will allow you to produce coherence. Rather than enforcing a single view, you enable having more than one. That, to us, is a really interesting thought.

JED: You also enable very dynamic negotiation. Austin has given a good overview of what I would say is a relatively recent evolution of the bureaucratic mindset. But I really think it goes back to the very early phases of people living in large groups, for example Babylonian city states. And I think we can see the same kind of pattern in medieval theology. For example, as Austin says, there is a problem of coordinating and in maintaining some level of coherence in a very large and distributed organization and in medieval Europe it was the church. It certainly also goes back to the renaissance invention of bureaucracy and straight on through in military organizations. It is also no coincidence, for example, that the ideas of mass production and interchangeable parts were invented by the military in the production of weapons. I think it is also no coincidence that a lot of the computing technology was invented by wartime efforts under the direction of the military. I think it might possibly be a coincidence, but if so it is a startling one that our ideas of mathematics came out of large civil engineering projects in the fertile crescent and Egypt and got consolidated into an axiomatic format. And I think to a large extent our idea of meaning and truth comes out of theological arguments in the middle ages. So I really do believe that this is a set of issues that permeates our culture to a degree that we are completely unaware of. We can't even see it. It is the fabric of our enculturated reality to such an extent that it is almost impossible to imagine a world that is any different. And I think that Austin made a key point, which is that we don't believe that it is possible to have a world that is any different as long as

organizations are fundamentally only possible through enculturating people to a certain kind of ritualized rule-following process. But when the technology shifts sufficiently, it may be possible to get out of that.

KLAUS: Is technology one of the most important things that keep us within this world-view, so to speak?

JED: In a sense I would say the limitations of the technology.

AUSTIN: But I think that the technology is responding to something that is fairly deeply built into our organizational structure. I was thinking the other day about the incredibly hard work that somebody went through to create the idea of monotheism out of the collection of several different gods who do different things. The idea that you would create One and that all the others would be reflections of that One is the same notion in many ways. They could only be reasonable if there were One.

JED: And you can trace ideas like the universal laws of physics straight back to monotheism. There is no separation at all.

KLAUS: So you are saying that the kind of computing that we have today is an ideology rather than something intrinsic to the technology, however firmly embedded it is today. But a critique of this position might go something like this: The force of computers is primarily to crunch numbers and therefore also to perform and support activities that are precise and can be easily formalized, and thus support coherence for instance. It's not a tool that is particularly well suited for supporting some of the tasks that humans do well – pattern recognition, interpretation, empathy, situated actions etc. So wouldn't it be better to accept that the computer is good at some things but should be kept entirely out of matters related to things that humans do better. Do you not risk, in spite of good intentions, to support a colonization of the life-world, as Habermas would say, rather than creating pliant systems supporting local activities?

JED: I think that is an excellent question. First of all, as I am sure you have seen here in Silicon Valley and as people all over the world are seeing now, Habermas' project of separating the life-world from the technical world is not feasible anymore. People are buying books and groceries over the Internet and everybody has to interact with computers many times during the day and it's just going to keep getting more so. And that is not driven by some conspiracy. It's driven by economic efficiency and convenience for individuals and values like that, which people in a very diverse way are interested in.

I think that our concern is that in some sense this attempt to separate the two and to somehow contain technology within its own little sandbox is a counsel of despair. It is saying: "The character of technology is given, it's somehow autonomous, it's not socially determined, it's not socially modifiable so we just have to accept it." And we think someone who says that owes us an argument for why that's so. That is certainly our experience of technology today, but I think we should pry the cover up and ask *why* that is our experience of technology. To take an example, the experience of manufacturing, the experience of producing physical goods, was not an alienated

activity until someone came along and automated it with procedures that were borrowed from military models. It wasn't autonomously alienating or autonomously rigid. It was made that way by imposing a certain kind of social structure on it.

AUSTIN: I also think that in your question there was the suggestion that computers wouldn't be good at things that people are clearly good at. I am not sure how far computing is going to go if we change the nature of it. But I don't think that my aspirations for a new kind of computing are all in the direction of getting the computer to do everything that humans do. I think that computers are not "in the world" the same way that people are "in the world" and as such they will have different resonances. But that is not to say that the computer couldn't provide some mechanical help with aspects of things which go on as part of the beating together of ideas. I tend to want to *not* go the route of Artificial Intelligence, which says that computers have to do it all alone. My hope is to get a prosthesis for some of these things that the computer doesn't do at all now – namely the interplay of ideas and the working out of what meanings might be. Not that the computer would do it *for* you, but that the computer would do it *with* you in what we call a "co-productive" way. This would leave to humans, if you like, those things that are essentially better done by humans. But I think the line that we have drawn now is far short of what the computers could do to help us deal with the richness of ideas and meanings.

JED: I endorse what Austin says. I just want to push it even a little bit further in a certain direction. Even if we can't change the characteristics of the computer at all in terms of its basic rigidity, we can do a lot, with no special technical changes, to embed the computing process into the social process in a way that gives the social activities much more control over the local circumstances of computing. So even if you totally accept that people have to do all the empathizing and all the creativeness and all the pattern recognition and so on, and I think that is a very good point, we can take existing information systems and kind of add "joints" to them such that people can exert much more control over them and much more control over their work circumstances.

AUSTIN: I agree completely. If you buy the problem, then the agenda we see is two-fold. The first is to use the current, rigid technology in a pliant way. That would require a change in perspective on the part of all of those of us in the KMD's of the world to have the computer play a different role, even if it is the same technology that we know today. The second agenda is a much broader one: to get machines that can begin to push that mechanical edge further forward.

KLAUS: I would like to go more into your two agendas. So let's start with the first one. I'm interested in discussing some of the roads that we can all take in order to actually use computer systems as they exist today in a more pliant manner. I've heard you mention that developers can squeeze it a little here and fix it a little there and it actually would be a lot easier to use and it would support social practices a lot better. So could you go into some of your solutions?

AUSTIN: Well, we have different kinds of things. One of them is the observation that one of the brilliant inventions of the paper bureaucracy was the idea of the margin. The margin is a place on a paper form, which is designed for writing things down that are

outside, both physically and conceptually, the form that "the system" expects. The thing about the margin is that it is connected to the form in such a way that the form carries the stuff that goes beyond the form along with the form.

JED: So it is an unformed part of the form.

AUSTIN: Exactly so. But what was good about margins in the old paper bureaucracies was that they were uniformly there and that there were practices that meant that you knew where to look. You looked in the margin. Of course we have similar things now: You look at these little yellow stickies. That's something which has become an institutionalized way of getting beyond the form while still being solidly within the form. Unfortunately we haven't done the same thing in our computers. Not that we *couldn't*; we could easily do it. Put a field called margin, or add to every field the ability to tag it, the ability to say: "Here is some more stuff". The computer of course can't do anything more than carry it to another human being and present that it's there and that somebody had better look at it. It's part of the form, but outside the computational capability of the form, which addresses only what is expected *in* the form.

So the shift that could happen is that we just have to put those margin-like things in the form and change all the practices where computation deals with fields that have information associated with them such that it gets back and gets processed by a human to to see if there is something that needs to be done. So the system asks the user whether this marginal stuff, which is now carried electronically, makes a difference. So you genuinely put people back into the process in a way which is scary if you were hoping to automate it. This is a step away, philosophically, from the business of figuring it out once and for all, and then letting the machines do it. Instead, we say: "The designers of the machines will figure out something, people using them will figure out more, and then the users and the machines together will actually do it." This shift in the social relationship between the computation and the world it is serving is a big change and that will take a lot of shifting. So that is one potential solution to the problem of creating a more pliant use of current rigid computing: Margins.

JED: All the things that we have are examples. We think they are valid examples, but we don't have a taxonomy or some kind of architectural proposal that is somehow all-encompassing. I think this first one is an especially good example, because it is so obviously implementable. You can implement it with today's databases and in Cobol. It is something that could have been implemented from the very beginning of information technology. There is nothing at all demanding about it. And the fact that it wasn't done this way – even though in some sense it was prefigured by the actual practices of bureaucrats – is an indication to us that this was an ideologically dictated style of design, not a technically dictated style.

There is a whole other category of examples, which is derived from ecology. Instead of taking software as written and provided to the users, you give the users fragments of software that can be copied and co-aligned into different forms and you let the users essentially construct their own local software to do what they want. Then they pass pieces of software around and over time this population of fragments of software acts like an evolving ecology and adapts to demands of users. I think that maybe it is worth

referring back to your earlier question about human skills versus computing. Notice that in the first case – the example of margins on forms – the computer is doing absolutely no interpretation, no pattern matching (maybe some string matching but that's not even essential), no creativity. We are not trying to make the computer any more intelligent than it is in the most boring information processing environment. In this ecological example that is also true. The computer is simply keeping and copying and executing little pieces of code. There is no intelligence. We are not making it any smarter than it is in a normal course of events.

What is maybe a little scary about the ecological example is that in order to understand the overall development of the system, you have to do a certain kind of reasoning. This certain kind of reasoning is not so common for designers because it is no longer possible to predict the overall function of the system any more from the design. The design is a substrate that new things can grow on. We have three existing examples of that. One is the buttons that were implemented at EuroPARC[ii] by Austin and others. Another one is the HyperCard product that Apple provided, which established a very flourishing ecology of little pieces of code and buttons and forms that people passed around, copied and made new applications out of. A third one that's familiar to just about everybody is the World Wide Web, where you can always go look at the source of a Web page and in fact innovations propagate through the Web very quickly. There is this whole ecology of features and new mutations that sometimes propagate with blinding speed to many Web pages. So I think this a very real example and it could be driven deeper into the infrastructure. We could come up with much richer ecological software processes.

KLAUS: I would like to discuss these solutions a bit further. And I would like to do that by giving you a concrete example that I just now started to think of in terms of your concept of pliant systems - or pliant use of rigid systems might be more correct.

AUSTIN: Or rigid *technology*, the *system* being the larger "context" of both humans and machines.

KLAUS: Yes. I did a usability study of an electronic patient record for midwives that we developed at KMD. Often, what I find in such studies is an incongruence between different perspectives, e.g. between health authorities, which is one perspective, medical research, which is another, and there are different practitioners each with their own perspective. So we have a number of different perspectives, which sometimes are incongruent. But this example surprised me a bit, because even amongst midwives I found different perspectives and opinions. So, for example, the midwives would disagree about the rigidity versus the flexibility of the system. Some of the midwives were very unhappy with the fact that the system was constructed in such a way that, for instance, you could not go from one page to another without having filled out certain fields, which had to be filled out with the correct codes, not just free text for instance. They found that it obstructed their daily work, their daily routines. They said: "In my daily work I don't have time to remember these codes. I need to get back to the woman in labor instead of standing here figuring out the code". Other midwives were really happy about this, because it reminded them of what they needed to do and helped prevent them making errors. Some of them also mentioned that they were happy about

the fact that the computer functioned as a controller in this sense because it documented that things had been carried out in the right order in case of patients filing lawsuits for negligence or mistreatment. So they had different perspectives on this, which surprised me a bit. Of course there are different reasons for the different perspectives. The midwives that supported the rigidity were the younger midwives while the ones that hated it were typically the older midwives. So there is something about the different kinds of knowledge and experience that they have and there is something about their career trajectories probably, but this is just an example of different perspectives *within* the group of midwives and a lot of questions come along with this example. We can take them one at a time. One is that you seem to assume that people want flexibility, that people want pliant systems. How can you be sure?

AUSTIN: We assume that the world is a sufficiently rich place with enough pressures on different people that they want to be able both to respond in the way which the world is pushing them locally and yet at the same time to be able to be coherent with other people who are feeling different pressures. How do we get that to happen? At the moment, as we've said, we try to think it all out in advance and put a plan in place and then everybody marches to that drum. So people will say: "We'll all agree to fill out certain fields before we change pages", or they'll say: "We'll all agree that you can ask later" or maybe they'll consider a way to get some alerts. These are all design considerations that are thought out in advance. You might even have a switch such that you can individually customize it, but you are still thinking them all out in advance.

The thing that we are assuming that people want is not flexibility for its own sake. But they do want to be able to do their own thing. As the world changes and they suddenly see things differently, they want to be able to act accordingly. For example, a midwife will say: "Oh these ones I know about, but this field is particularly critical, I want an alert for me with respect to that." If you thought it all out in advance *either* it flips the page *or* it doesn't. Then you don't have that thing that maybe the logic in the workflow is differential, maybe it's more situated in the sense that it's based on which question is asked, when, by whom and how. In fact all those possibilities just enumerate forever. There are a million different circumstances so we can't plan the questions and answers in advance. The richness of life is such that, in the particular circumstance, I want to be able to say: "Well, the pattern for me is this" and then have the system be able to help at some level. So I am saying: "Do they want that flexibility for its own sake? No!" They want it in response to their need and what they notice about themselves and the way they work.

KLAUS: But how would you get that kind of local flexibility in an example like this?

JED: Let's work this through as an example, because I think that by putting together the two classes of examples that we had earlier, we can come up with something. First of all, let us assume something that you didn't explicitly state: at some point the data from the patient record is going to provide input to some less flexible system, like some kind of health statistics record or something of that sort. So ultimately the fields maybe have to be coded in order to adequately reflect the information. But in what order they get filled in and how that coding eventually gets done is open. So to begin with, we could provide the kind of margins that we were talking about earlier and allow, but not

require, that people can break out of this specific coding temporarily and move around more flexibly. But then the issue becomes: How do we lock things back down? How do we script it? And here it is important, I think, to realize that for the particular midwives you talked to, they only had a choice between having it completely scripted or having it open. But people might have, as Austin said, scripting needs that evolve over time. So maybe for convenience, they might want to start on a different page or want to rearrange what fields are on the page or what have you. Well, this is the sort of thing that you can relatively easily do with HTML for example and maybe a little Java-script.

The realistic problem in this kind of situation is that most people wouldn't be able to do that themselves. It's beyond what they should have to concern themselves with, it's beyond their skills. But in a social environment, typically you have people with different skills and they pass little pieces around to each other, they help each other out in various ways. So we would expect that if it was possible, if the basic material out of which this application was made lends itself to this kind of social intervention, people would adapt it. They would make friends that have similar needs and they would say: "Oh, that's just what I needed, could you give me a copy of that". Maybe the person involved wouldn't even know how it works, but they can make a copy and give it to you. So over time the base of software would evolve to fit the practices and not every midwife would have the same things.

But then at the boundary you have to put filters, so that once a record reaches the boundary, then if there is a mismatch or if the fields aren't coded or whatever, it gets the attention of the midwife, and says: "Help. This won't go into the statistics database. Please fill it in" and then maybe the midwife goes through the process of translating the annotations. So the system is requesting help from the user. It's a very flexible interplay back and forth.

AUSTIN: And what you have just done in that last step is to introduce this idea of the boundary; the boundary between before it's gotten to the records processing and after. But one can imagine other kinds of boundaries. So the idea that records will be sitting within these spaces, which are connected to people in certain ways, and the idea that that connection is the hook where the processing pieces could get applied, is something which we typically don't do at the moment because we don't have those kinds of boundaries. So this is the kind of invention purely within rigid computing that we need to make. They are social inventions that need to get added to the conceptual structure of rigid computing such that we can then allow this bricolage, this emergence and response to the needs of the local circumstances.

JED: Maybe this is a good point to tie this back to the study of social discourses, because what we are saying here is that the computer is not participating in the social discourse. It's not an actor. But it's enabling or supporting the discourse just the way that pen and paper does, except just in a more rich and diverse way. Now we can start to use the types of understandings that people use to analyze discourse: As an utterance or some discursive material moves further and further from the community in which it was generated, it tends to be reformulated in terms of broader and broader, more general languages. So it gets translated from a very informal genre typically in a local community to a more and more formal genre as it moves out into a larger community.

So there needs to be computational support for enabling and supporting and in some case enforcing that translation. But it's nothing exotic. It's something that fits into the normal social process.

KLAUS: I'm not sure that I hear you right. Are you saying that in this sense the computer is a neutral tool? You said it's not an actor in the social discourse.

JED: I don't mean that it is a *neutral* tool. I don't think that tools are ever neutral. They always have a grain to them. But it is not a person. It doesn't, for example, deserve respect or have rights in any personal sense. You don't expect it to make choices or have intrinsic values that it would enforce in a situation. That is all I meant to say. It isn't a social actor.

AUSTIN: Earlier I used the words "the mechanics of something", in an attempt to capture the idea that a lot of this stuff needs to be supported by putting mechanisms in place by which stuff can get gathered together and bits of this processing can get applied. Flags can be noticed and people can be called to take a look at this or that.

JED: People can be given an adequate context to decide what to do in a given case.

AUSTIN: Again, none of it is rocket science at that level.

KLAUS: Let me just stick with the example of the midwives. An obvious solution for any usability specialist would be to provide the user with the ability to decide whether she wanted this rigidity or whether she wanted to be able to move to another page without having filled out the fields for instance. But this then gives us another problem, a more traditional usability problem. The more flexibility you have, the more complexity you typically have and the more difficult it is for people, who are not skilled computer users, to use the system.

JED: That's why we go back to the ecological approach. Most people would probably copy something from a friend. If you brought all the complexity up to the surface and there was an affordance for every decision, then that would be catastrophic. People only want to see affordances for functions that they in fact use in practice. So it is always necessary to be able to hide most of the possible choices below the surface. Right now we hide them below the surface by having those choices made by the designers. The designers do some analysis and bring the affordances up to the surface that they think the community of users will need, or in some cases that the community of users should be allowed to have. What we are saying is not that all the affordances should just be floated up so that every user has to confront all possible choices, but that we should accept the social re-construction of those choices over time. So instead of a designer-controlled interface, provide a tool kit, an environment in which users can build very constrained sets of user interfaces and allow them to break those open and change the choices. Most people won't break them open in fact, so you have to make it easy for them to copy the ones they like. And then as part of that, you have to provide the underlying infrastructure to map values across the different systems that are constructed.

AUSTIN: This actually *is* a common practice. Take UNIX systems. They always have people breaking open and twiddling this file and that file. It gets completely out of their hands; it's hard to manage. If, on the other hand, you had an organized way of thinking about all of that in a systematic way, then the process of breaking things open may indeed be left to the individual user. You might start with certain things being openable by end-users. And you might at some point say: "Oh, the practice is changing such that something which was two or three levels deep, now suddenly needs to be up where people can get at it." So how does somebody reach down there and drag that up through the intervening layers and how does that affect other people? That whole process is beginning to move beyond the sort of thing that we know how to do in a straightforward manner. There is some work to be done there.

JED: I just want to make the point that we are making processes here explicit which aren't ordinarily explicit. But in fact if you closely observe any moderately large group of people, they are constantly engaged in negotiating these kinds of issues if the issues haven't somehow been locked down in software or otherwise taken out of their control— and even then they will often work around the attempts at control. People will invent terminology, they will come to agreements on certain kinds of conventions, other things will be left open and flexible. They'll negotiate times close enough, they'll establish routines and procedures in some cases and in other cases they won't. Then perhaps the lack of a procedure becomes a problem or perhaps the existence of a procedure becomes a problem. Sometimes things just change invisibly without conscious effort, other times there is a crisis and people discuss it. All of these things happen and what we are trying to do is to get the practice of software development to integrate with that natural social process. We are not trying to create some new set of skills that people don't have in the ordinary course of events.

AUSTIN: But having said that, we need to be careful because the usual practice of software development is to think it all out in advance. This is true, even when you iterate: things squeak, so you do a new specification, a new re-design and you roll it out. It's piece-wise, in chunks, whereas what we would like to see is the honoring of what happens in the *"real"* system, the social practices surrounding the rigid technical stuff, and then take some of that and move it down and get the technology to support it. If you can only do a re-design every six months, then what happens with those things that turn up a month later, or didn't get into the last round? What do users do? Well, they figure out something in the social world, which will handle it. We'll put some stickies on, or I'll write them down in my note-book and we'll remember. The system as a whole is of course "bubbling" along; it doesn't stop changing. Current, the computing part of it may not be bubbling along. All we are saying is: Let's admit to the fact that the world is bubbling along, honor those practices and get computation to help a little.

JED: I would like to just mention that in talking to a lot of small Silicon Valley companies, which are in the process of providing Web services, it is very clear that the ones that are going to be successful have a practice of putting their stuff out early. They are not trying to do a long design cycle. I have talked to people who are in successful Web-service organizations – like Amazon – and their system evolves quite rapidly and there is no real specification. The system is its own specification and there are many parallel development efforts changing it at once. But they are all on very short cycles so

that they can't become detached from the main line of development for very long or they become irrelevant. It is interesting also to watch open-source development projects like Mozilla because they are generating new releases every night. There are many different activities going on and they are all in open view, they are visible through e-mail and bug reports and check-ins and there are exactly these kinds of social processes. Sometimes things will break down or assumptions will fail, and then there will be discussions and sometimes practices evolve. So there are many ways in which this is actually visible in the software domain. I think actual software practices are now outrunning the classical computer system design ideology quite nicely.

KLAUS: One of the problems here is that while all this sounds quite innocent, you are actually asking people to give up power.

AUSTIN: Who?

KLAUS: Well, let's take the example with the midwives again. You are, in fact, asking the health authorities to put more faith in the local interpretations, the local activities. So I guess my questions could be formulated as this: Aren't you giving up on the coherence and promoting the local responsiveness on that behalf? And have you considered the implications in terms of power structures?

AUSTIN: I guess that my reaction is that the idea that we have been able to maintain the power of, say, the health authorities *through* the technology, that the technology has been a device by which we limit the practice, a) may be an illusion because what is really happening out there is not what actually happens in the technology, and b) is not a necessary thing. That is to say that what you can enforce through the technology is not the only thing that you can enforce. There are other social mechanisms. You can say to people: "Look, if you don't do it this way you lose your job or you lose your license". Just because we can't do it through technology doesn't mean that we cannot do it. I want to unload the burden that technology tends to have to bear of enforcing the law, and say: "Yep, it's going to loosen that up and then the mechanisms for coherence, of which enforcing the law may be one, are the subject matter of genuine debate rather than a statement that this just has to be that way because that is the way the technology is. I want to force that into the open.

KLAUS: My question would then be: Why should people, in this example the health authorities, want to debate this?

JED: Right. I agree with Austin about this. I think there are some more things I want to say along that line. But let me say, specifically, that the health authorities might not want to debate it, but they aren't autonomous either. They are embedded in the social process as well. Standard sociological discourses on power to some extent seem to treat power as somewhat autonomous. I won't go into the whole analysis but we see power as being produced partly because people feel a need to structure organizations as a highly coordinated system and someone ends up being able to take advantage of that coordination. But even when "the authorities" are taking unfair advantage, people tolerate the rigidity because in fact chaos really is worse. If you really ended up with

chaos it would be bad. People don't want that, so they are willing to pay quite a high price to maintain the coordination.

However, if it is possible to have coordinated, well behaved social processes without the rigidity, people will prefer that and they'll vote for it, in effect. They may vote economically: these processes, for example, will tend to be more efficient; they may vote simply by changing their membership in organizations; they may vote literally: they may vote for a new government that chooses to allow more flexibility. But there will be an overall social process that will change the rules.

Also, I think that there are other interesting things that Austin didn't mention about the nature of this rigidity that we have today and the way technology tends to enforce it. To a large extent the technology tends to put the burden, the cost of paying for this on the people, say, at the shop floor, or in your case the midwives. It's not accounted for, nobody counts up that cost. So it looks free, but it imposes penalties on the system as a whole if, for example, the people on the shop floor have to work around the computer system. In fact it takes more of their time, their productivity is lowered. But there is no explicit visibility into that so management thinks that it is free. If they could see the cost, if the costs became an explicit item and they were trading it off against higher productivity, they would actually be thinking much more carefully about that.

So our goal in this is not simply to open everything up and make everything flexible. It's to make the process of *choosing* a level of coherence and a tightness of coordination visible and float that up close enough to the surface so that people can see the cost trade-off – people on the shop-floor *and* the management. On that background we can then make reasonable trade-offs and negotiate as we do in our social situations and come to some workable consensus. And we believe that the consensus has the potential now to move considerably further toward flexibility because of a better technology.

AUSTIN: There are potentially some new points of equilibrium, some new patterns or new practices of coherence that we haven't explored yet. In fact, we might not even know the language to begin to talk about this. We have this almost binary idea of either having chaos or being "locked down". It's either the military or it's the market. And we are saying that there is potentially lots of space in between, and there is a lot of work to be done to understand that space. So if we were to spring, full blown on the world, a technology that could do all the things that we advocate – either the simpler ones or the much richer ones – we would be confronted with the fact that we wouldn't know how to talk about it. We would have to develop those techniques. And one of the stable points in this very rich technology, that we imagine, is the military style of thinking where everything is locked down. And if people want to go that route then they can still go that route. It's not as if you have the rigid systems on one hand and the pliant systems on the other. Rather, pliant systems include the rigid systems as a special case.

JED: That is an option. I think "locked down" has been stable historically, but I think it is becoming unstable, and I actually think you can see this in a very macroscopic sense with the positive terror that copier technology and now the Internet inspires in totalitarian regimes. They see right away that this is the death of their ability to retain

control. And I think they feel terror for this reason. So they try to suppress these technologies, but I think most of us believe they won't succeed.

KLAUS: Let's move more in the direction of your second solution which requires much more research than what we have been discussing up to now. Could you expand a little bit on what technologies are required to enable such systems? Also, I would like to have you expand more on what social forces will oppose and encourage them? These are large questions, so let's start with what technologies are required to enable pliant systems.

JED: Let me just provide a little background. We see that we can go a long way and we don't know how far with the pliant design of systems that use today's rigid technology. But ultimately that runs into inherent limitations in the technology itself. For example, computers tend to be based on very discrete values. I don't mean discrete just on the bit level, but the whole structure of computer programs is sequential and you end up making lots of discrete decisions. And the problem is that this kind of discrete choice is very unforgiving of any mistakes.

We believe that ultimately, in order to get full value from the pliant perspective, we will have to go to more radical approaches that will change the nature of computation itself. And right now there are technologies that do this such as neural networks and various kinds of Bayesian computations that are used in robotics. So we are not talking about something that nobody has ever seen before. The problem today is that those technologies are only useful in very restricted areas, like for example hand writing recognition or speech recognition or maybe some kind of process to decide whether a screw is defective based on visual inspection. They aren't really computational processes in the broad sense. You can't build big, complex systems out of them. They don't scale. What we've tried to do is to imagine what it would take to make those much more scaleable.

Essentially, the key thing that I believe is missing from these "softer" technologies today is compositionality. There are a lot of different names for it, but basically the wonderful thing about human language – and even more perhaps algebraic languages – is that you can take pieces and bind them together. This is of course also a key attribute of computer systems, both the hardware and the software. You can plug pieces together and they somehow combine their functions and if you do it in an appropriate way then the whole is greater than the sum of the parts. Right now, if you have a whole bunch of different neural networks, there is really no useful way to plug them together in the sense where you get a bigger, more complex, pliant system out of them.

There are a few examples, though, of compositionality for pliant systems. We think that one of the really interesting examples of that kind of compositionality, which appears to be pliant to us, is Christopher Alexander's pattern languages for architectural design.[iii] I want to make a distinction here. We don't mean the kind of patterns that have been used in software design. Alexander's approach is really much more flexible and continuous than the typical software patterns. So what we are interested in doing is finding ways to put Alexander's pattern languages on a computational footing. Not in the sense that computers would somehow gain the abilities that architects have using this language,

but just in the sense that computers would vivify those patterns, make them alive and responsive. So when you build something with those patterns, it would be active and responsive the way things can be when you build them out of code.

AUSTIN: One of the components of "active patterns", which really makes a sharp distinction with object oriented programming patterns, is the act of creating an instance of some pattern, or an instance of a class. In object oriented programming such an act in no way threatens to change the class or the conception that you are instantiating. The nature of the pattern language that we are imagining, is that the act of seeing some situation as a case of a particular pattern, an act which takes work to do, may in fact enrich your notion of what that pattern is and may cause you to adjust the pattern itself. So the very act of trying to think about something as a case of something you've seen before, threatens your understanding of what you had before. Now, *that* is terrifying to those who want to know what they have got. So again we come back to the question of how you are going to keep it coherent. But the technology, at the fundamental level of moment by moment "seeing something as" a case of a pattern, is already beginning to move, which then affects other things. And it's not that you are just seeing something as one pattern. You are seeing it in a structure of patterns so the process of bashing things together is happening not in the abstract, nothing happens much in the abstract. It happens in the very specifics of doing a particular thing. Let me give an example: In 1978, Eleanor Wynn observed a Xerox clerk taking phone orders for copiers supplies (paper and toner).[iv] As part of taking orders, the clerk got a shipping address from the customer. One customer had trouble providing the shipping address, because the copier was on an ocean-going barge: if Xerox could say when the supplies would be shipped, the customer could say where to ship them. The form wanted an address; the situation could not produce one. At this moment, Xerox (in the person of the order clerk) faced a conceptual shift: "Oops I see, an address can be time varying. I hadn't thought of that." The act of confronting this clerk with an address which was time varying caused the pattern which said "you gotta get the shipping address" to confront the fact that the address in this particular case is going to be time varying. How are we going to do that? So the pattern itself had to be extended to deal with the situation.

JED: I want to point out an implicit theme here. We are not talking about introducing some radical new technology, which is going to do some wonderful thing. We are talking about making it easier for people to do what they are doing anyway. This kind of pattern, the seeing-as, the incremental reshaping of practices through their application in particular cases is something that happens in the social context. Right now computers are very rigid. We are talking about making them easier to modify to begin with and then we are talking about incrementally over time developing technologies that make it easier and easier for the social process to reshape the computational process. We are also saying that in parallel that it is necessary to develop a better understanding of how the social process controls itself, how it controls the coherence of its structures and develop a taxonomy, a language for talking about it. And that will itself indirectly develop into computational techniques. So the reason we talk about the long term picture – the more continuous computation and Alexandrian patterns – is not so much to say: "Well, let's just start working on that and create some autonomous technology, which we can then throw into the social process", because it is not going to happen that way. It is to give us a vision for how this will develop over a longer period. And sure,

we can do the technology development, but it will have to be constantly re-integrated in the social process. The two will have to develop together.

KLAUS: Let me just step back a little. Before we end, I would like to take a really pragmatic look at this and see it from, for instance, my point of view or my company's point of view. What would be your advice to me and to companies like KMD? In many ways, I think a lot of people will agree with you: The midwives that we talked about before would probably agree with you (if they had any idea of what we are talking about) and I as a usability specialist agree with some of your fundamental principles. If this sells better, people at management-level in my company will also agree with you. But what should we do?

JED: Well, one thing you could do – and you have a more precise sense of how to execute this – is to develop a tool-kit for creating these applications of the sort we were talking about in the midwife example: a software infrastructure on which you could put these fragments together and allow the user community to copy them and recombine them on their own and you can lock down some of them. Underneath that there has to be a database and probably a conventional database would be adequate as long as it can store comments as well as other pieces of data.

AUSTIN: Even before you did that, there is the need to expose people who are building systems to this whole concept and the possibility of pliant systems and then letting *them* say: "Okay, I'm going to respond to it by adding a place for the margin", for example.

JED: I agree. Change the discourse first.

AUSTIN: Exactly. Get that going and then look at the regular patterns that are beginning to emerge out of that. You've got some really smart people. They are going to begin to see regularities: "Oh, I keep adding the same thing". Good, that's what the toolkit should do. Then let that follow. We can guess what some of these things might be, but from a pure practical point of view, if you could infuse people with the spirit of the thing, then you would be getting them to begin to design these things. Correspondingly, you would be having all those discussions with your customers, with the midwives and others, about what is actually needed. Is there a need for a new kind of specialist in the field etc.? That needs to be constructed because you have been arguing that you need not only the technical stuff but also the social stuff that goes with it.

KLAUS: One last thing: Could you think of – or have you actually met – social forces that oppose these ideas?

JED: Well, I think there are a lot of social forces that oppose these ideas at many different levels. In the United States there is a very strong social tradition, which interestingly is most clearly represented in religious thought, things like biblical inerrancy and opposition to teaching critical thinking and a very rigid idea of truth and falsehood. I think that forces like that – very broadly based social forces – will oppose this. I think that management in some companies will oppose this because of fear of loss of control. But I am not particularly concerned about that because I think the economic

and social forces will simply go around those companies and they will become obsolete and go out of business or change their policies. In a relatively free market that kind of process is pretty effective. I think that people who are socialized to existing practices of system design will probably be very uncomfortable, because this is not an easy transition for people to make.

AUSTIN: We have seen that regularly just in giving talks. People have trouble getting the idea of what it would be like.

JED: I think that some of these people will have a much easier time once there are more working examples. But other people will just fundamentally dislike it because they like very precise, crisp forms that can be combined in well defined ways, and they basically want to work in an environment where things are neat, clean and under control.

KLAUS: Imagine I was trained in computer science and spent 15 years learning these programming languages that are quite hard to learn because they are so detached from real-life problems. That gives me a certain position and in certain places of the world, like here in Silicon Valley, it enables me to write my own paycheck because programmers are hard to get these days. So I could imagine you would have some opposition there too.

JED: That's right. Information systems professionals profit from the obscurity of their technology. If you give more power to the users, then it undercuts this. So there are lots of constituencies that will oppose this.

AUSTIN: You will also find the deep social forces that Jed was talking about not only in the United States but in the whole western tradition of this idea of *the* one. A signal is the use of the articles "the" and "a". The moment you say "the" you are in that position of getting *one* answer. This is opposed to our view, which is that we need to have a coherent set of ways of looking at the world which can work together rather than "getting *the* solution". It is so deeply built into the way we think. Most of what we have been struggling with in doing this work for four years now, has been confronting those assumptions *in ourselves* even when we were consciously aware that this is what we are trying to do.

KLAUS: Have you experienced any differences between speaking about this here in the States and speaking about it in Scandinavia? Scandinavian countries have a history of taking different perspectives quite seriously in debating things before reaching consensus, probably more than in the States?

AUSTIN: I have given the talk in Denmark twice, once at Aarhus University and once at Danfoss[v], and I found a resonance. On the other hand, we have found a resonance most of the places that we have given the talk here. But we have also been very selective in finding those places where we thought there would be resonance. I think that there is somewhat less resonance in North America but we are talking to those who are interested in HCI or have this view that the unit of analysis is the socio-technical practice and *that* resonates.

JED: I have only given this talk in the United States so I can’t make the contrast. But I think that the biggest issue we’ve seen in talking about this is difficulty in getting people to overcome assumptions that simply make this idea incomprehensible. I don’t think the problem is that we have got people who understand it but push back and don’t like it. I think the problem is a failure of imagination. Once we can get people over that failure of imagination we often find that they get actively interested. So I think it is partly our own failure but I think it is largely an indication of how hard it is to get past the underlying assumptions that are so deeply built into our culture.

KLAUS: Well, this talk has certainly helped me understand some of these ideas. Thank you very much both of you.

---

[i] Joanne Yates: ”Control Through Communication: The Rise of System in American Management”, John Hopkins University Press, Reprint Edition, 1993.

[ii] Xerox’s European Research Center in Cambridge, England.

[iii] Christopher Alexander’s core books on architectural design and pattern languages are:
Christopher Alexander: “Notes on the Synthesis of Form”, Harvard University Press, 1964.
Christopher Alexander, Sara Ishikawa & Murray Silverstein: “A Pattern Language: Towns, Buildings, Construction”, Oxford University Press, 1977.
Christopher Alexander: “The Timeless Way of Building”, Oxford University Press, 1979.

[iv] Elanor Wynn: “Office Conversation as an Information Medium”. Unpublished Ph.D. thesis, University of California, Berkeley, 1979. For a paper that uses Wynn’s observations as a centeral driver to discuss the difficulty of supporting office procedures, see: See Austin Henderson & R. E. Fikes: “On Supporting the Use of Procedures in Office Work”, in Proceedings of the First Annual National Conference on Artificial Intelligence, American Association of Artificial Intelligence, Menlo Park, California, 1980.

[v] Danfoss is Denmark's largest industrial group with about 20,000 employees. Danfoss develops and produces mechanical and electronic components for several industrial branches worldwide.