

## Is There a Design of Design?

**Design.** I have been privileged to participate in the design of five computer families, three programming languages, an operating system, a variety of graphics applications, and six building projects—four as principal designer. I am struck by the degree to which the design process seems to be the *same*, regardless of the medium.

What is this process? How has it changed over the years? What are its challenges today?

Why should systems designers spend any energy studying the design process *per se*?

Three hopes drive us:

- Can I design better by looking at the design process?
- Can we better teach others to design?
- Can we better organize and manage design?

Sir Francis Bacon asserts that studying the process across design media will bring about new ideas.

“by a connexion and transferring of the observations of one Arte, to the uses of another, when the experience of several misteries shall fall under consideration of one mans minde.”

*The Two Books of the Proficiency and Advancement of Learning*, book 2, p 10

As computer scientists, we can tap the results of the decades of study of the design process that has been done in architecture, in mechanical engineering, and in industrial design. What observations of their arts can we transfer to the uses of ours? Unlike David Parnas, I’m convinced there are some. Unlike David, I used Chris Alexander’s patterns in a building project and am very pleased with the results of the (unconventional) decisions he guided.

Of the many aspects of the design process, three seem to me to be the central 21<sup>st</sup>-century design issues:

- What mental model of the design process most accurately reflects real practice and is most fruitful?
- How do we organize collaborative and team design efforts to yield designs that are conceptually integrated?
- How do we practice, teach, and manage design so we get great designs, not merely competent ones?

**Models of Design—How Engineers Think of Design.** We each carry in our heads some implicit model of the design process. Most of our models are, I suspect, highly rational model, yet rather simple-minded.

In my naive model, one has a clearly defined goal or problem. One has a set of desiderata to be achieved by the designing. Some sort of utility or goodness function describes how these desiderata are weighted and combined into an overall measure, the measure to be optimized. This utility function is generally non-linear.

Together with the utility function one has a variety of constraints, including resources with fixed limits. Invariably there will be one of these which becomes the controlling budget. Often this is the implementation or product cost; but this is not always true. Sometimes it is schedule—time to delivery. In space-borne vehicles, it is usually weight. In computer architecture, format bits must be allocated—the bit-budget.

Finally, in this model one has an always-implicit tree of design decisions, the structure of options that define the space of possible designs.

Given all these, the design algorithm is straightforward. One explores the design tree depth-first, backtracking when constraints make a path infeasible. At each node, one chooses the descent that seems

most promising. One finally picks the design with the highest utility function, and voilà! Or, one runs out of design time, so one picks the best design found so far. This algorithm is, of course, the same as the brute-force algorithm for playing chess or go or other games of skill. Much AI work aimed at automating design has begun essentially here.

**What's Wrong With This Model?** We don't really know the goal. Design is quite unlike chess. In design, we rarely have a clear, sharp, precise definition of the object to be designed. We rarely have a precise, correct description of the problem to be solved. Indeed, very often the most useful thing the designer does for the client is to force and aid in the crystallization of what is to be designed.

“Why doesn't he know what he wants? Why doesn't he make up his mind?” My rapid prototyping, my realization of my client's ideas, is in fact the forcing function enabling him to understand what he needed. Often the hardest part of design is deciding *what* to design. And it is precisely here that experts designers are apt to go wrong.

Not only do we not know the goal, in fact we rarely know the design tree, the detailed option set. Moreover, both the desiderata and the constraints typically change during the design process.

So the rational model of the design process, that generally taught in our engineering textbooks (such as Pahl and Beitz, 1984) is dead wrong and seriously misleading. Lawson, Schön, Maher, and Cross have contributed to the development of an iterative co-evolution model that much better describes the practice of real designers. There is still plenty of work to do on this model. In software, we must exorcise the waterfall model in favor of co-evolutionary models.

**Team Design—Collaboration and Telecollaboration.** Since at least the start of the 20<sup>th</sup> century, large design projects have been the work of teams, rather than solo designers. This change creates a new and crucial challenge, how to maintain *conceptual integrity* in such designs. I have elsewhere argued that this is a most important property of a design, and that separating the design of the user interface from the design of the means of implementing it enables one mind to master the user interface. I elaborate this into a discrete functional specialty, system architecture, whose importance I am prepared to defend fiercely.

Team design is occasioned both by the increasing sophistication of design specialties, so that one mind cannot master them all, and by the need to partition work to meet ever-more demanding schedules. I think it very important to differentiate clearly among aspects of design where collaboration positively helps, aspects where the work is partitionable so that collaboration may help schedule but not result quality, and aspects where collaboration positively impacts the conceptual integrity of the design.

A late-20<sup>th</sup>-century development that we barely begin to understand is telecollaboration, joint effort by non-collocated teams.

**Great Designs Come from Great Designers, not from Great Design Processes.** What makes a design *great*, or *elegant*, or *cool*? This is fundamentally an esthetic judgment, and we need more theory of engineering esthetics. We need to know how to teach it.

But I will defend the propositions that design process improvement, and even the development of a science of design, will raise the floor of design practice, but not the ceiling. Now raising the floor is superb, and very important. But what our hearts desire and our minds acclaim are great designs, not merely competent, sound, trustworthy ones. I am convinced that great designs come only from great designers. Hence a major

part of our challenge is to figure out how to identify, train, and nurture those to whom the innate gifts for great design have been entrusted.