

Roadmap for Design Rules

Carliss Y. Baldwin*

Kim B. Clark†

Version 1.0

January 21, 2006

*Harvard Business School
†Brigham Young University, Idaho

Direct correspondence to:
Carliss Y. Baldwin
Harvard Business School
cbaldwin@hbs.edu

Copyright © Carliss Y. Baldwin and Kim B. Clark, 2006

Roadmap for Design Rules

Carliss Y. Baldwin and Kim B. Clark

Abstract

This paper describes briefly how we came to write *Design Rules*, why we divided the work into two volumes, and what lies ahead in volume 2. It provides both a personal and an intellectual overview of our work.

Revision History:

v. 1.0 1/21/06 First draft (adapted from Preface to the Chinese edition)

We would like to take this opportunity to explain our objectives for this work, why it is the first of two projected volumes, and how it lays a base for further investigation. We would also like to talk briefly about what lies ahead in volume 2.

We began this project in the late 1980s. Both of us were interested in Sun Microsystems, a new upstart in the technical workstation market. In its approach to finance and technology Sun did not fit in with well-established patterns of competitive behavior. Indeed, many of its actions seemed to fly in the face of the “sound management principles” taught to MBAs and executives. Sun raised money in the public capital markets far more frequently than seemed prudent. It offered high-performance technical workstations, yet appeared to have no proprietary technology. The company built systems that were incredibly fast with off-the-shelf hardware and software. It outsourced much of its manufacturing. It developed a network file sharing protocol and a RISC chip architecture, and then, instead of exploiting these proprietary technologies, practically gave them away. Sun’s managers appeared to be doing everything wrong, and its success seemed at first glance to be a matter of smoke and mirrors. But, as it turned out, there was method in their madness, and genius in their approach to technology.¹

Technological mastery lay at the heart of Sun’s strategy. The hardware and software architects within Sun seemed to see technology in a new way—as a playing field for moves in a competitive game. Thus we set out to try to understand what they were doing, and in particular, how they viewed their technological opportunities. Our goal was to bring their “game” and its “moves” into the realm of formal economic analysis.

Then as now, the word “modularity” was in the air, surrounded by connotations of technical wizardry. Modularity was what allowed both Sun and its main competitor, Apollo Computer, to outsource key components of their systems. And Sun in large measure succeeded in driving Apollo out of the market because it was “more modular” than its rival—whatever that meant. But what was this thing called modularity, and how could we represent it within the formal analytic framework of economics?

The virtue of a modular system, we discovered, is that its components can be mixed and matched to achieve the highest-value configuration in a particular setting. The mixing and matching is possible because the designers do not have to know precisely how the modules will be arranged after the fact.

¹ Baldwin and Clark (1997b).

They only have to know generally what the module will do, how it will fit in, and what constitutes good module performance. Thus the essence of modularity, we felt, lay in the *options* it gave designers to postpone and then revise key decisions. Option theory is a well-defined field in economics with a wealth of formal models. It is the key to understanding the economics of design architecture and design competition.

Obviously, however, not all decisions about the design of an artifact can be postponed: some early decisions are necessary to provide a co-ordinating framework for the others. Those early decisions, in turn, would serve as rules—*design rules*. Design rules were needed to govern the building out of a modular system, ensuring that the respective parts did not clash and in so doing “kill” the system as a whole. Such rules, when well-constructed, provided harmony among the many different parts of a modular system.

We saw too that the construction of a modular system governed by design rules would not be a deterministic, pre-planned endeavor. It is in the nature of options to allow unexpected things to happen. Thus, within the framework of the design rules, module designers would be free to experiment and choose the best design from among many trials. Development could proceed opportunistically and incrementally, with each module on its own semi-independent trajectory. The overall process would be like biological evolution in the sense that it was incremental, parallel and not pre-determined. But what we came to call *modular design evolution* was unlike biological evolution in three important ways. First, the processes generating design change were search processes based on designers’ foresight and subject to their incentives (including economic incentives). Second, the survival of designs depended on designers’ own assessments and evaluations, which in turn were based on tests they devised to rate and rank designs. And third, the whole sequence needed to take place within an architectural framework that was itself designed for the purpose of “playing host” to an evolutionary process.

These then were our two key insights, which form the central message of this book:

- Modular designs create options; and
- Modular designs can evolve.

Similar insights expressed in different words can be found in seminal works on the science of design by

Ross Ashby, Herbert Simon, and Christopher Alexander.² And our two-pronged thesis echoed theoretical arguments that were being put forward in the 1990s by other researchers, especially, Richard Langlois and Paul Robertson, Ron Sanchez and Joseph Mahoney, and Raghu Garud and Arun Kumaraswamy.³

However, it seemed to us that all the arguments of that time (this was the early 1990s), including our own, were based on indirect evidence. We were depending on what other people said about designs. But second-hand reports are unreliable: in some circles every design is said to be modular. And modularity can be relative. Sun and Apollo could both truthfully claim to have used modular computer designs, but, in some fashion, Sun's were more modular. Thus we felt there was a great and looming gap in the literatures on technology, strategy and economics. What was happening to the designs themselves? It was this question that led us to write *Design Rules*.

The need to go "to the designs themselves" caused us to focus exclusively on one industry, the so-called "greater" computer industry. We knew that modularity was a general property of complex systems.⁴ Hence modularity of some kind would be present in almost any industrial context. But in order to "see" designs as the designers saw them, we needed to deepen our technological understanding in some particular industrial domain. We would have to learn one or more engineering languages to the level of comprehension, if not mastery. We would have to read descriptions and appraisals of specific designs and learn to identify the recurrent engineering tradeoffs and compromises. This was a daunting prospect, and we avoided making this commitment for a long time. But in the end, it was inescapable.

By then we already had a base of technical knowledge about computers from our prior work. We knew that most computer designs were highly modular, and we knew that the industry had gone through massive structural changes as a result of changing designs. (Indeed those changes continue through today.) Hence, in 1995, we chose the greater computer industry, comprising all makers of hardware, software, components and services that go into a computer system, as our research site. We then began the task of finding and valuing the options and charting the evolution of computer designs.

² Ashby (1952, 1960); Simon (1969), Alexander (1964).

³ Langlois (1992); Langlois and Robertson (1992); Sanchez and Mahoney (1996b); Garud and Kumaraswamy (1993). A set of seminal articles and commentaries by these and other authors has been reprinted in a single volume: R. Garud, A. Kumaraswamy, and R. Langlois, eds. (2003).

⁴ This idea is developed and explored in Schilling (2000).

We actually began with the designs of the 1980s—personal computers, technical workstations, microprocessors, and operating systems. But we were quickly driven backward in time to IBM’s System/360 and even older computer designs. We found that, while the designs and certainly the performance of computers have changed greatly, the desire for design options and the perception that modularity was the key that would unlock the door were constants. This seemed to be true no matter how far back in time we went. Some designers were always trying to create options through modularity, while others wanted to integrate all the components in order to achieve higher levels of performance.

Imagine our relief when we reached the beginning of this seemingly endless series of recurring debates. The origin, we discovered, was what might be called the first “architectural document” on computer design. It was written by the great mathematician, John von Neumann in a few weeks in the wartime spring of 1944. Although the original memorandum was not published until after his death, copies and redrafts were widely circulated and became enormously influential. Virtually all subsequent computer designs bear this document’s stamp—because it provides a way to think abstractly and systematically about the enormously complex artifact that is an electronic computer. (The story of von Neumann’s report and other precursors of modular computer designs may be found in Chapter 5 of this book.)

With 1944 and the von Neumann memorandum identified as our starting point, we were free to be moving forward in time again, tracking the lineages of actual designs from this common source. We sought to document both the evolution of designers’ perceptions and the parallel evolution of the designs themselves.

Our efforts received an unexpected gift in early 1996 through the work of the computer scientist and evolutionary theorist, John Holland. Holland has developed an overarching theory of complex adaptive systems, which encompasses the processes of biological evolution, neuronal and immunological growth, cellular automata, and importantly, complex games, like checkers and chess.⁵ Because of the broad reach of his theory, we could locate our theory of modular design evolution within his framework. And, by looking at how his theory was constructed, we could see what was needed to complete our own. From Holland, we got the idea of “operators” as primitive moves and sequences of operators as strategies

⁵ Holland (1992, 1995, 1998).

in a structured, multi-player game.

Thus, the first part of this book, (Chapters 2–8) explains what modularity is and how it came to be present in computer designs. The second half (Chapters 9–16) explains what can be done with modularity and its consequences for the surrounding economy. In these chapters, we catalog the “operator-moves” that are made possible by the fact of a modular design. In developing this catalog, our objective was to connect the abstract theory of operators with the nitty-gritty reality of actual designs and designers’ decisions. Thus we did not strive to come up with a complete list of operators. Instead, our rule was “a real example for each operator.” That is, each operator described in this volume had to play a documentable role in at least one important episode in the history of the computer industry.

For the operators *splitting* and *substitution*, we could point to the design of IBM’s System/360 and the subsequent emergence of plug-compatible devices (see Chapters 10 and 14). We could also cite Sun Microsystems’ decomposition of the design of a technical workstation in the 1990s (Chapter 11). For the operator *excluding*, we had the computer architectures and technology strategy of Digital Computer Corporation (DEC), once the second-largest computer systems maker in the world (Chapter 12).⁶ For *augmenting*, we had not only DEC’s strategy for minicomputers, but the example of Visicalc, the first spreadsheet program. Visicalc’s functionality drove the early growth in demand for personal computers, but, in successive rounds of *substitution*, Visicalc was itself unseated by Lotus 1–2–3, Borland’s Quattro Pro, and finally Microsoft Excel (Chapter 12). And finally, for the operators *inversion* and *porting*, we had the examples of Unix and C—respectively the first portable operating system and the language invented for the purpose of coding it (Chapter 13).

Describing how modularity emerged in computer designs and then documenting the existence of operator-moves made possible by modular architectures took us through the 1970s in terms of the history of computer designs. The history of the designs, in turn, we thought contributed to a better understanding of the great sea change in the structure of this industry that occurred in the 1970s. As we indicate in Chapter 1 and go on to explain in Chapters 14 and 15, between 1970 and 1980, the greater computer industry changed from a highly concentrated and vertically integrated oligopoly dominated by IBM to a fragmented and vertically disintegrated *modular cluster* of independent firms, linked by design

⁶ In the 1990s, DEC was acquired by Compaq Corporation, which in 2002 was acquired by Hewlett Packard.

rules governing the systems they make. The design rules in turn comprise the architectures of various “platforms,” plus common interface protocols and standards.

In 1980, the market value of the *cluster* surpassed the market value of IBM. Hence that year was a convenient way to mark the beginning of a new industrial order in computers. The workstation battles fought between Sun and Apollo, as well as all the intricate and fascinating moves and countermoves in the personal computer and operating systems markets, were part of this new order. But, there was more: new centers of economic activity, like Silicon Valley; new cultural institutions, like the Internet, the Worldwide Web and email; economic anomalies, like the Internet bubble and crash; new rules of property established via antitrust and intellectual property disputes; new social and political movements, like the Free Software movement; and even new theories of engineering and technological innovation, like the Open Standards and Open Source initiatives. All of these things, we believe, are tightly linked to the new, post-1980 industrial order. But the post-1980 social and economic order in computers also relies—indeed depends for its very existence—on the presence of modular, option-rich design architectures for hardware and software. Hence, if it is to continue, this complex adaptive system must continue to provide participants with the incentives and resources to create new, option-rich, modular design architectures in the future.

In 1997, we knew we were in a period of unprecedented ferment, growth and change in the industry we were studying. The breathtaking scope of change threatened to overwhelm us. On the advice of a good friend, we divided our work in half; relegated post-1980 designs and events to a planned second volume; and concentrated all our research efforts on finishing volume 1. In part, this was a pragmatic decision: Kim Clark was by then Dean of Harvard Business School, and Carliss Baldwin had taken on some administrative responsibilities as well. But we also had a strong sense that the industry, the designs, and the surrounding institutions were in a state of flux, and that more time was needed for the underlying patterns to become clear. Hence the volume you are holding ends with a description of the emergence of a *modular cluster* type of industry and some preliminary analysis of incentives and equilibrium behavior in such a cluster (Chapters 14 – 16). Thus Volume 1 tells of the existence of *modular options*, the possibilities of *modular design evolution*, and the emergence of a *modular cluster*. The story of the interactive evolution of designs, firms and patterns of competition *in the cluster* is left for volume 2.

Indeed, when we sent this volume off to the publisher in December 1998, we were at a loss as to

how we would approach the enormously complex and variegated developments we were observing. We were especially concerned that the phenomena we sought to understand were no longer bounded by engineering design on the one hand and economics on the other. Too many interesting issues and developments—the Internet, hackers, open standards, the free software movement, intellectual property law, the open source development process—were cultural, political and social in nature. And yet these things unquestionably interacted with and influenced both engineering designs and economic outcomes.

In 2001, we received a second intellectual gift in the form of Professor Masahiko Aoki's monumental work on *Comparative Institutional Analysis*.⁷ Here at last was a social and economic theory to complement John Holland's theory of complex adaptive systems. Here was an analytic framework in which we could comfortably nest our own efforts.

Professor Aoki conceives of institutions as equilibria of linked games plus self-confirming summary beliefs about how the games are played. Thus our task became to identify the games that can be played “within the walls” of a modular architecture and “between” architectures. For each game, we needed to explain how an equilibrium gets constructed (or, in some cases, does not); how value is created, captured and distributed; and what beliefs must be fostered and fulfilled. Describing and documenting these games is not a trivial task, but thanks to Professor Aoki's deep thinking and careful exposition, we knew where to begin.

As we worked on this agenda, we began to discover new things. We have always known that our theory of modularity and option value applies to only a subset of all designs—those with relatively high “technical potential,” and a low “cost of experimentation.” But as we worked on models of competitive interaction, it became clear that some “patterns of competition” were specific to this type of design.

Designs with *low* technical potential relative to the cost of experimentation give rise to classic oligopolies, in which a small number of firms compete on the basis of new product development and process-improving innovation. In contrast, designs with *high* technical potential (relative to the cost of experimentation) cannot be contained “within the walls” of a few firms. Our theoretical models show that, except in rare circumstances, such designs will attract entry by firms and venture capitalists and give rise to clusters of firms engaged in tournament-style competition to offer the best design. Because

⁷ Aoki, M. (2001).

they will not stay confined and may “behave badly” from the perspective of their creators, we have semi-humorously called this class of designs “unmanageable.” Unmanageable designs are the main focus of volume 2.

But what does it mean to have “high technical potential”? In Chapter 10 of this volume, we defined a parameter, denoted “ σ ” or “sigma”, as a measure of technical potential. Formally, sigma is the standard deviation of a probability distribution of design outcomes. *Our theory assumes that this probability distribution (with a finite standard deviation) exists.*

Unfortunately, although designs are manifestly uncertain, there are *as yet* no good data from which to measure probability distributions of design outcomes. Science relies on observable data, and thus the fact that our theory depends on an unobserved parameter means it can be criticized as “unscientific” because it is “not subject to scientific tests.”

We see no way around this dilemma. *Uncertainty lies at the very core of designs and design processes.* Some designs fail, some succeed, and some succeed beyond our wildest imagination. “E pur si muove,” Galileo said — loosely translated, “it is still that way.” Design outcomes *will* be uncertain whether we choose to think of them that way or not.

The parameter we call sigma represents one aspect of this intrinsic uncertainty. Admitting it into our theory allows us to model the consequences of design uncertainty in a more formal and rigorous way than would be possible otherwise. We do not want to lose this opportunity. But that means we must begin to develop ways of observing and measuring the probability distributions of design outcomes. From such distributions, we can infer sigmas (and higher moments). We need both *ex ante* measures to enable valuation and *ex post* measures to validate predictions. Developing such measures is an important part of the agenda for volume 2.

Our third discovery was that in our setting, the Aoki-style institutional equilibria differ greatly depending on what the participants believe about the designs and about each other. At first this seemed a nuisance.⁸ But then we realized that, in reality, participants’ beliefs would change as the “rules” of the game evolved and became clear. Thus, we saw, the institutional equilibria derived from our models would be *transient patterns* that appeared and then disappeared as the participants gained experience in

⁸ Indeed the nuisance has a name—it is called the “equilibrium selection problem” in game theory. See Samuelson, (1998).

competition. Initially, participants would react “blindly” to their competitors. At a later stage, some would develop strategies that recognized the existence of rivals but did not depend on the rivals’ beliefs. Still later, some participants (called “lead firms”) would see profit in attempting to influence the beliefs of others.

These then are the ideas we will develop in volume 2:

- “Unmanageable” designs (with high probabilistic technical potential relative to the cost of experimentation) give rise to clusters of firms engaged in design competition;
- To understand design competition, we must assess the probability distributions of design outcomes and measure technical potential (σ) both *ex ante* and *ex post*;
- In industries with “unmanageable” designs, there are successive “patterns of competition.” Each pattern is based on a different configuration of beliefs among industry participants.

In closing, we would like to renew our thanks to those cited in the Acknowledgments of the English-language edition and to thank others who have contributed to this edition and to our ongoing research program.

References (not included in the Bibliography of Design Rules, Volume 1)

- Aoki, Masahiko (2001). *Towards a Comparative Institutional Analysis*, MIT Press, Cambridge, MA.
- Baldwin, Carliss Y., Christoph Hienerth and Eric von Hippel (2005) "The Migration of Products from Lead User-Innovators to Manufacturers," MIT Sloan School of Management Working Paper No. 4554-05, Cambridge, MA, September.
- Garud, Raghu, Arun Kumaraswamy, and Richard Langlois, eds. (2003) *Managing in the Modular Age: Architectures, Networks and Organizations*, Malden, Mass.: Blackwell.
- Samuelson, Larry (1997) *Evolutionary Games and Equilibrium Selection*, MIT Press, Cambridge, MA.
- Schilling, Melissa A. (2000) "Toward a General Systems Theory and its Application to Interfirm Product Modularity," *Academy of Management Review* 25(2):312-334, reprinted in *Managing in the Modular Age: Architectures, Networks, and Organizations*, (G. Raghu, A. Kumaraswamy, and R.N. Langlois, eds.) Blackwell, Oxford/Malden, MA.