

# PREFACE

## THE CONTINUING DEVELOPMENT OF SYSTEMS ARCHITECTING

*Architecting, the planning and building of structures, is as old as human societies  
— and as modern as the exploration of the solar system.*

So began this book's 1991 predecessor.<sup>1</sup> The earlier work was based on the premise that architectural methods, similar to those formulated centuries before in civil works, were being used, albeit unknowingly, to create and build complex aerospace, electronic, software, command, control, and manufacturing systems. If so, then still other civil works architectural tools and ideas — such as qualitative reasoning and the relationships between client, architect and builder — should be found even more valuable in today's more recent engineering fields. Five years later, at the time of the first edition of this book, judging from several hundred retrospective studies at the University of Southern California of dozens of post-World War II systems, the original premise has been validated. With another five years of perspective the observations only holds more strongly. Today's systems architecting is indeed driven by, and serves, much the same purposes as civil architecture -- to create and build systems too complex to be treated by engineering analysis alone.

Of great importance for the future, the new fields have been creating architectural concepts and tools of their own and at an accelerating rate. This book includes a number of the more broadly applicable ones, among them heuristic tools, progressive design, intersecting waterfalls, feedback architectures, spiral-to-circle software acquisition, technological innovation, and the rules of the political process as they affect system design.

Arguably, these developments could, even should, have occurred sooner in this modern world of systems. Why now?

### ARCHITECTING IN THE SYSTEMS WORLD

A strong motivation for expanding the architecting process into new fields has been the retrospective observation that success or failure of today's widely-publicized systems often seem pre-ordained; that is, traceable to their beginnings. It is not a new realization. It was just as apparent to the ancient Egyptians, Greeks and Romans who originated classical architecting in response to it. The difference between their times and now is in the extraordinary complexity and technological capability of what could then and now be built.

Today's architecting must handle systems of types unknown until very recently; for example, systems that are very high quality, real-time, closed-loop, reconfigurable, interactive, software-intensive, and, for all practical purposes, autonomous. New domains like personal

---

<sup>1</sup> Rechtin, E., *Systems Architecting, Creating & Building Complex Systems*. Englewood Cliffs, N.J.: Prentice Hall, 1991. ISBN 0-13-880345-5. Hereafter called Rechtin 1991.

computers, intersatellite networks, health services and joint service command and control are calling for new architectures — and for architects specializing in those domains. Their needs and lessons learned are in turn leading to new architecting concepts and tools and to the acknowledgement of a new formalism — and evolving profession — called systems architecting, a combination of the principles and concepts of both systems and of architecting.

The reasons behind the general acknowledgement of architecting in the new systems world are traceable to that remarkable period immediately after end of the Cold War in the mid-1980's. Abruptly, by historical standards, a 50-year period of continuity ended. During the same period, there was a dramatic upsurge in the use of smart, real-time systems, both civilian and military, that required much more than straightforward refinements of established system forms. Long range management strategies and design rules, based on years of continuity, came under challenge. It is now apparent that the new era of global transportation, global communications, global competition, and global turmoil is not only different in type and direction, it is unique technologically and politically. It is a time of restructuring and invention, of architecting new products and processes, and of new ways of thinking about how systems are created and built.

Long-standing assumptions and methods are under challenge. For example: for many engineers, architectures were a given; automobiles, airplanes and even spacecraft had had the same architectural forms for decades. What need was there for architecting? Global competition soon provided an answer. Architecturally different systems were capturing markets. Consumer product lines and defense systems are well-reported examples. Other questions: how can software architectures be created that evolve as fast as their supporting technologies? How deeply should a systems architect go into the details of all the system's subsystems? What is the distinction between architecting and engineering?

## DISTINGUISHING BETWEEN ARCHITECTING AND ENGINEERING

Because it is the most asked by engineers in the new fields, the first issue to address is the distinction between architecting and engineering in general; that is, regardless of engineering discipline. Although civil engineers and civil architects, even after centuries of debate, have not answered that question in the abstract, they have in practice. *Generally speaking*, engineering deals almost entirely with measureables using analytic tools derived from mathematics and the hard sciences; that is, engineering is a deductive process. Architecting deals largely with unmeasureables using non-quantitative tools and guidelines based on practical lessons learned; that is, architecting is an inductive process. At a more detailed level, engineering is concerned with quantifiable costs, architecting with qualitative worth. Engineering aims for technical optimization, architecting for client satisfaction. Engineering is more of a science, architecting more of an art.

In brief, the practical distinction between engineering and architecting is in the problems faced and the tools used to tackle them. This same distinction appears to apply whether the branch involved is civil, mechanical, chemical, electrical, electronic, aerospace, software or

systems.<sup>2</sup> Both architecting and engineering can be found in every one. Architecting and engineering are roles, distinguished by their characteristics. They represent two edges of a continuum of systems practice. Individual engineers often fill roles across the continuum at various points in their careers or on different systems. The characteristics of the roles, and a suggestion for an intermediate role, are shown in Table 1.

Table 1: Characteristics of the roles on the architecting-engineering continuum

<b>Characteristic</b>	<b>Architecting</b>	<b>A &amp; E</b>	<b>Engineering</b>
<b>Situation/Goals</b>	Ill-Structured	Constrained	Understood
	Satisfaction	Compliance	Optimization
<b>Methods</b>	Heuristics	←→	Equations
	Synthesis	←→	Analysis
	<b>Art and Science</b>	<b>Art and Science</b>	<b>Science and Art</b>
<b>Interfaces</b>	Focus on “Mis-Fits”	Critical	Completeness
<b>System Integrity Maintained Through</b>	“Single Mind”	Clear Objectives	Disciplined Methodology and Process
<b>Management Issues</b>	Working for Client	Working with Client	Working for Builder
	Conceptualization and Certification	Whole Waterfall	Meeting Project Requirements
	Confidentiality	Conflict of Interest	Profit versus Cost

As the table indicates, architecting is characterized by dealing with ill-structured situations, situations where neither goals nor means are known with much certainty. In systems engineering terms, the requirements for the system have not been stated more than vaguely, and the architect cannot appeal to the client for a resolution as the client has engaged the architect precisely to assist and advise in such a resolution. The architect engages in a joint exploration of requirements and design, in contrast to the classic engineering approach of seeking an optimal design solution to a clearly defined set of objectives.

Since the situation is ill-structured the goal cannot be optimization. The architect seeks satisfactory and feasible problem-solution pairs. Good architecture and good engineering are both the products of art and science, and a mixture of analysis and heuristics. However, the weight will fall on heuristics and “art” during architecting.

One way to clearly see the distinction is in the approach to interfaces and system integrity. When a complex system is built (say one involving 10,000 person-years of effort) only absolute consistency and completeness of interface descriptions and disciplined methodology and process can suffice. When a system is physically assembled it matters little whether an interface is high-tech or low-tech, if it is not exactly correct the system does not work. In contrast, during architecting it is necessary only to identify the interfaces that cannot work, the mis-fits. Mis-fits

<sup>2</sup> The systems branch, possibly new to some readers is described in Rechtin 1991 and in Chapter 1 of this book.

must be eliminated during architecting, and then interfaces should be resolved in order of criticality and risk as development proceeds into engineering.

One important point is that the table represents management in the classical paradigm of how architecting is done, not necessarily how it actually is done. Classically, architecting is performed by a third party working for the client. In practice the situation is more complex as the architect might be done by the builder before a client is found, might be mixed into a competitive procurement, or might be done by the client. These variations are taken up in chapters to come.

*Systems* architecting is the subject of this book, and the art of it in particular, because, being the most interdisciplinary, its tools can be most easily applied in the other branches. From the author's experience, systems architecting quickly and naturally abstracts and generalizes lessons learned elsewhere, not only for itself but also for transfer and specialization in still other branches. A good example is the system guideline (or heuristic), abstracted from examples in all branches, of **Simplify. Simplify. Simplify.** It will appear several times in this text.

It is important in understanding the subject of this book to clarify certain expressions. The word "architecture" in the context of civil works can mean a structure, a process, or a profession; in this text it refers only to the structure. The word "architecting" refers only to the process. Architecting is an invented word to describe how architectures are created much as engineering describes how "engines" and other artifacts are created. In another, subtler, distinction from conventional usage, an "architect" is meant here to be an individual engaged in the process of architecting, regardless of domain, job title, or employer. By definition and practice both, from time to time an architect may perform engineering and an engineer may perform architecting — whatever it takes to get the job done.

Clearly, both processes can and do involve elements of the other. Architecting can and does require top-level quantitative analysis to determine feasibility and quantitative measures to certify readiness for use. Engineering can and occasionally does require the creation of architecturally different alternatives to resolve otherwise intractable design problems. For complex systems, both processes are essential.<sup>3</sup> In practice, it is rarely necessary to draw a sharp line between them.

#### CRITERIA FOR MATURE AND EFFECTIVE SYSTEMS ARCHITECTING

An increasingly important need of project managers and clients is for criteria to judge the maturity and effectiveness of systems architecting in their projects — criteria analogous to those developed for software development by Carnegie Mellon's Software Engineering Institute. Based upon experience to date, criteria for systems architecting appear to be, in rough order of attainment:

- \* A recognition by clients and others of the need to architect complex systems
- \* An accepted discipline to perform that function; in particular, the existence of architectural methods, standards and organizations

---

<sup>3</sup> For further elaboration on the related questions of the role of the architect, see Rechtin 1991 pps 11-14; on the architect's tools, Parts One and Three of this book; and on architecting as a profession, Part Four of this book and *Systems Engineering*, the Journal of the International Council on Systems Engineering.

- \* A recognized separation of value judgments and technical decisions between client, architect and builder
- \* A recognition that architecture is an art as well as a science; in particular, the development and use of non-analytic as well as analytic techniques; and,
- \* The effective utilization of an educated professional cadre; that is, of masters-level, if not doctorate-level, individuals and teams engaged in the process of systems-level architecting.

By those criteria, systems architecting is in its adolescence, a time of challenge, opportunity and controversy. History and the needs of global competition would seem to indicate adulthood is close at hand.