MASSACHUSETTS INSTITUTE OF TECHNOLOGY

BOOK REVIEW - THE ART OF SYSTEMS ARCHITECTING

BY MARK W. MAIER AND EBERHARDT RECHTIN

November 2000

Sebastian Fixson, fixson@mit.edu

Technology, Management and Policy Program, MIT

Course: ESD.83 Research Seminar in Engineering Systems

Prof. David A. Mindell

For the last 50 years engineering has attempted to separate itself from research approaches and methodological tools seen as non-quantitative, sometimes even non-scientific. Emphasizing rationality, engineering has focused on phenomena that can be measured, calculated and optimized. In contrast, art has been categorized as an endeavor beyond scientific scrutiny. Art involves always non-rational criteria like emotion, senses or pure instinct.

Two events at the end of the twentieth century seem to have begun changing this perception of separation between engineering work and the social world with which the systems interact and in which they operate. First, the end of the cold war has dramatically altered the financing options for almost all large-scale systems, in particular military and space projects, enforcing a stronger focus on heterogeneous customer demands. Second, the rise of the Internet has reformulated questions of who actually plans, builds, and controls many of these of systems. Both phenomena demonstrate that in today's world the success of any large system depends not only on its technical performance, but also on the level of acceptance that it achieves from different constituencies, i.e. customers, government, the general public, etc.

Recognizing these developments and the fact that the most far reaching decisions are made in the very early stages of system conceptualizing, Mark W. Maier and Eberhardt Rechtin propose to reconcile 'artistic' approaches with 'classical' engineering. Their book, *The Art of System Architecting*, calls for allowing 'fuzzy' approaches to support the early system development steps, which they call architecting. Distinguishing between engineering as a deductive process and architecting as an inductive process, the former being more science than art, the latter being more art than science, their book builds a bridge to reconcile these two worlds. Even though, their own strong background in systems engineering in the aerospace arena shines through most of the examples and their view on public decision making processes shows a strong grounding in the engineering thought world.

The authors direct their book towards experienced engineers, others "concerned with creating, building, or using complex systems," and various other constituents of graduate studies. Consequently, the authors have structured the book in a textbook style.

The book's content is grouped into four parts. The first lays the groundwork and positions the book, the second presents a number of different application domains for the architecting approach, the third introduces various tools helpful for architecting systems, and the fourth explains how democratic decision making processes work and looks where systems architecting stands as a profession.

The two chapters of part one form the foundation for the remainder of the book. In chapter 1, the authors locate one of the major reasons why systems architecting becomes increasingly important in the growing complexity of the systems being engineered. They define complex as being "composed of interconnected or interwoven parts" and a system as "a set of different elements so connected or related as to perform a unique function not performable by the elements alone." To expand the classical engineering view, Maier and Rechtin argue for a purpose orientation that goes beyond mere systems performance, but rather includes explicit value judgments: "a useful purpose, at affordable cost, and in acceptable period of time." They see the architect as an "agent of the client," that works "jointly with the client on problem and solution definition. System requirements are an output of architecting, not really an input." To tackle these often ill-defined problems, the authors propose in the second chapter a list of heuristics as tools for the systems architect. Abstracting experience, humans can use heuristics as a means to transfer knowledge that cannot be expressed in mathematical equations or hard logic. Chapter two presents an extract of the authors' twenty-year effort of collecting and sorting heuristics helpful for systems architects. (Appendix A offers an extended list of heuristics.) Maier and Rechting find four types widely applicable. All four remind the architect of the

limited capacity of his own understanding of the situation. In essence, they recommend allowing for the possibility that the original problem statement may not have been the best, or even the right one, and in turn, argue for system clarity, simplicity, and flexibility.

In its five chapters, the second part of the book presents five application domains for systems architecting. Chapter 3 is devoted to the authors' home field; what they call builder-architected systems. The term implies, that builder and architect are identical. This situation usually applies for large-scale high-technology projects like new weapon systems or space programs. Having used systems engineering approaches since WWII, this industry had to cope with shrinking government funding after the end of the cold war. Forced to compete in more open markets, the term customer focus has become more important, and with it a new definition of project success: customer value. A *Higher, Faster, Farther* approach had to be exchanged with a *Better, Faster, Cheaper* way of doing business.[i] This leads the authors also to remind the reader that a good architecture, in addition to good system performance, can have strategic value in market competition. Chapter 4 shifts the focus from developing a product-like system to operating a manufacturing system. Maier and Rechtin point out that four major innovations over the last twenty years have changed the architectural rules for manufacturing systems. Ultraquality requirements, dynamically run manufacturing systems, lean production, and flexible manufacturing systems have added tasks for the manufacturing systems architect such as making connections to the product waterfall or determining and helping control the system parameters for stable and timely performance. In the fifth chapter, the authors turn to the domain of social systems. One of the fundamental differences compared to the systems previously introduced is the separation of client and user. In order to cope with the different constituents' value assessments involved in evaluating the merits of projects like buildings, bridges, and public power systems, Maier and Rechtin suggest to ask the four who-questions: "Who benefits? Who

pays? Who provides? And, as appropriate, who loses?" The authors also point out another dimension that is particularly problematic for technically trained engineers, that is the tension between facts and perception. They propose the following painful heuristic: "It's not the facts, it's the perceptions that count."

In chapter 6, the authors discuss software and information technology systems. They predict that "software is rapidly becoming the centerpiece of complex system design, [..]." Software features like the ease for upgrading which fuels a higher pace of change, and software's inherent flexibility that allows to create intelligent, high value systems, will ultimately result in a shift of priorities. In the future, system flexibility will become more important than technical (but narrow) perfection in the first attempt. Due to its non-physicality, software overcomes classical hierarchical design restrictions and allows new architectures, e.g. object orientation. "Architecture in software can be definitions in terms of tasks and modules, language or model constructs, or, at the highest abstraction level, metaphors. Because software is the most flexible and etheral of media, its architecture, in the sense of defining structure can be equally flexible and etheral. The most famous example is the Macintosh desktop metaphor [..]."

Collaborative systems are introduced in chapter 7. These systems are not under central control, either in their conception, their development, or their operation, the Internet being the prime example. Here the authors blur the lines of their own definition given earlier, in that they allow the parts of the collaborative systems having purpose in their own right. The management is also assumed to be at least partially autonomous. For this reason the author's focus is on specific parts of the system: the interfaces. "When the components of a system are highly independent, operationally and managerially, the architecture of the system *is* the interfaces." For the same reason Maier and Rechtin identify standards as having tremendous strategic importance for these type of systems.

Part three of the book proposes tools for the architecting process. Chapter 8 sets the stage by specifying terms like roles, views, and models. The most important role of models is assigned to communication with client, users, and builders. Maier and Rechtin provide a list of model types useful for architecting. Objectives and purpose models, models of form (scale, block, etc.), behavioral models (data and event flow, mathematical systems theory, autonomous agents, etc.), performance models (analytical, simulation, judgmental), data models, and managerial models (waterfall/spiral, PERT/CPM, etc.) are discussed regarding their merits and drawbacks. The authors summarize three key characteristics of working with models: "(1) Models are the principal language of the architect. [..] (2) Architects require a multiplicity of views and models. [..] (3) Multidisciplinary, integrated modeling methods tie together the various views. [..]"

In chapter 9, the authors present their view on the design progression in systems architecting. They suggest a three-part approach to the process of system architecting. The first is a conceptual model "connecting the unstructured processes of architecture to the rigorous engineering processes of the specialty domains or disciplines." The second is an introduction to and review of the general concepts of design, and the third poses that "[t]here is a guide to the place of architecting and its methods with the specialized design domains and the evolutionary development of domain-specific methods."

Chapter 10 combines the two previous chapters and introduces integrated modeling methodologies. Maier and Rechtin focus mainly on two models, the Hately-Pirbhai (H/P) integrated models for computer-based reactive systems, and Q^2FD, quantitative quality function deployment, for performance driven systems. In addition, the authors introduce a number of integrated software models (ADARTS, OMT, UML, etc.)

For chapter 11, the authors have collected three major descriptions of architecture frameworks. They understand architecture frameworks as "the standards for the description of architectures." To demonstrate possible differences, the authors list three different frameworks: the U.S. DoD C4ISR architecture framework, the ISO's RM-ODP standard which is more computation and software centric, and the IEEE's P1471. The approaches differ with respect to the required rigor and detail. The authors are skeptical about the possibility to develop a general architecture framework, partially because of the differences between the hardware and the software world.

The final part of the book, part four, is devoted to the systems architecting profession. Chapter 12, contributed by Brenda Foreman, discusses the role of the political process for systems architecting. This chapter reads like an explanation of the 'real world' to an engineer used to focus on his 'hard numbers.' Maier and Rechtin point out that the political decision making process is intertwined with the technical decision making process and is in fact often the determining part. They frame these insights as *facts of life* like "Politics, not technology, sets the limits of what technology is allowed to achieve" or "The best engineering solutions are not necessarily the best political solutions." Chapter 13, the last chapter, investigates the current state of systems architecting at North American Universities (plus Tel Aviv) and proposes a curriculum for graduate studies similar to the one introduced at USC[ii].

The book closes with three appendices (A: Heuristics for systems-level architecting, B: Reference texts suggested for institutional libraries, and C: On defining architecture and other terms) and a glossary. Packed with material, the book is a good reference book and fuels the thinking with many cross-links. Unfortunately, its dense information also makes it sometimes difficult to access it on the sub-chapter level. A numbering below the chapter level would certainly improve this.

Although the book makes a big step toward its goal to allow systems architecting being an art, its own structure reveals the author's own prioritization of the views. The fact that the chapter about the political decision making process, chapter 12, appears (a) almost at the end of the book and (b) separated from the technical domains demonstrates that its impact on systems architecting is acknowledged, but its meaning is not integrated – at least not yet. Democratic decision making processes are typically neither perfectly efficient in terms of invested time or resources nor are they always rational, but they provide the environment most of us consider as fundamental for a free and somewhat fair world. To view them only as constraints, rather than as an integral part of our life, is the first step of allowing architecting being an art, but should not be the last.

Reference:

Maier, Mark W., and Eberhardt Rechtin, 2000, The Art of Systems Architecting, 2nd Edition, CRC Press, Boca Raton, Florida

---

[i] The terms Higher, Faster, Farther and Better, Faster, Cheaper are borrowed from a paper by Murman et al. about the challenges for aeronautical design, engineering and manufacturing and MIT's Lean Aerospace Initiative as an answer strategy. See: Murman, Earll, Myles Walton and Eric Rebentisch, 2000, Challenges in the Better, Faster, Cheaper Era of Aeronautical Design, Engineering and Manufacturing, MIT, Lean Aerospace Initiative, http://web.mit.edu/lean

[ii] University of Southern California