NEW SYSTEMS ARCHITECTURE VIEWS

Dr. Howard Eisner, The George Washington University

Abstract

Considerable attention has been paid in the last decade or so to the matter of architecting large-scale systems. The U.S. Department of Defense has been prominent in this activity through its C4ISR (Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance) Architectural Framework. This Framework has emphasized three views of an architecture, namely: operational, systems, and technical. The IEEE has put forth its views of an architecture as: functional, physical and foundation. Various other researchers and practitioners, including this author, have examined several architecting methods and procedures. In this paper, new architecture views are presented, based upon this author's architecting construction. i.e., Eisner's Architecting Method (EAM). These new views are derived specifically from the EAM, whose short form includes the three *critical* views of: synthesis, analysis, and cost-effectiveness. Areas of focus for these new views include system alternatives, requirements, risk, elements of performance, functional design features, interoperability, and others. A rationale for each of these is also provided. Since the EAM is a prescriptive procedure that leads directly to the three critical views, the new views are "spinoffs" that allow the architect to use data that is generated as part of the architecting process. These illustrative new views also set the stage for further elaboration, dependent upon the perspectives of different stakeholders, whether they be technical, management, user-oriented, acquisition-focused, or programmatic. These new views tend to fully integrate the architecting methodology and the specific architectures that are developed from the application of that methodology.

Introduction

In the last ten years or so, there has been an accelerating interest in the matter of architecting large and complex systems. Much of this activity has been related to the field of systems engineering. A major contribution was made by E. Rechtin through his book on Systems

Architecting [1], which contained many of his real world experiences as well as a long list of useful heuristics. The U. S. Government, principally in the form of the Department of Defense, recognized the need for a more coherent view of systems architecting. They led the charge by establishing a working group that would address this matter. The result was the Architecture Framework [2], a perspective regarding systems that defined three architectural views of systems, namely: operational, systems, and technical. At about the same time, this author included a chapter on systems architecting in his systems engineering book [3]. The focus of this chapter was to define a very specific procedure for architecting a system. This procedure is now called Eisner's Architecting Method (EAM). E. Rechtin, along with M. Maier, continued work on systems architecting, with the perspective that architecting helped to "create and build systems too complex to be treated by engineering analysis alone" [4]. Later, Dr. Rechtin took many of these principles and related them to the systems architecting of organizations [5]. Work in the arena of systems architecting was proceeding in parallel with the above, notably through professional societies. The IEEE, for example, briefly considered architecting in its P1220 standard [6], and also explored architectural descriptions (ADs) in its P1471 standard [7]. Finally, as systems have evolved into "systems of systems", many issues of design and development have been raised [8] which are likely to have an impact upon how we architect these mega-systems.

Rationale and Selected Methods

An appropriate place to explore the rationale for and nature of architecting is Rechtin's treatise on this subject [1]. Rechtin suggests that the "essence of architecting is structuring, simplification, compromise and balance", and that architecting has become more and more necessary with the increase in the complexity of systems. Architecting involves the conceptualization of systems which can be approached through four basic methods:

1. normative
2. rational
3. argumentative or participative
4. heuristic

Some six years later, Rechtin and Maier continued to examine the "art of systems architecting" [4]. These authors confirmed the above four methods and continued to insist that the architect needed to try to simplify as much as possible. However, the pure complexity of today's systems makes simplification a difficult task. We try to look at the system as a whole and attempt a system design that will serve a 'useful purpose at an affordable cost for an acceptable period of time". We also recognize that modeling of the system is usually a very important part of architecting as well as simplification. Further, there is a point at which we need to stop architecting and begin what might be called basic engineering. A long and very instructive list of heuristics for "system-level architecting is also provided by these authors.

In February of 1998, one year after the publishing of the above book, the Office of the Secretary of Defense (OSD) confirmed the need to "establish comprehensive architectural guidance for all of the DoD" [9]. The rationale for this important step was to try to ensure that our military systems are interoperable and cost-effective. This was also partly in response to a key recommendation of the Defense Science Board. A month after the above memorandum, the C4ISR Architecture Working Group (AWG) Final Report was issued [10]. This report confirmed the need for a unified C4ISR development process that would have the following features:

1. three specific architectural views
2. ease of comparison of architectures across organizational boundaries
3. leverage
4. early testing for validity and cost effectiveness
5. an audit trail for impact analysis

A key aspect of this work was to "institutionalize the C4ISR Architecture Framework". This placed that Framework in a central position in the world of architecting, not only for the DoD but also for many executive agencies of the government that tend to follow developments of this type with respect to building systems.

The C4ISR Architecture Framework, version 2.0, was issued in December of 1997 [2] and was broadly accepted as key guidance, and in consonance with government legislation requiring more emphasis on the following:

1. interoperability
2. integrated solutions
3. cost effective business practices and capabilities

The same Framework described a six-step architecture description process, as follows:

1. determine the intended use of the architecture
2. determine the scope of the architecture
3. determine the characteristics to be captured
4. determine views and products to be built
5. build the requisite products
6. use the architecture for the intended purpose

In an approach that was formulated independently of the work of Rechtin and Maier, but on a problem that was well recognized by many, this author suggested a method that would help to focus as well as simplify the architecting process. Part of the problem can be seen when one tries to define architectures from a series of design choices for each function of the system. An example is illustrated by the following set of design choices for eight functional areas of a communication system [3]:

Functional Area	**Design Choice**
Multiplexing/Demux	D11, D12
Modulation/Demod	D21, D22
Switching & Routing	D31, D32, D33
Encryption, Decrypt	D41, D42
Signal Formatting	D51, D52
Control & Monitoring	D61, D62
Recording & Playback	D71, D72
Satellite Comms	D81, D82

Where DIJ is the Jth design choice for the Ith function. We see from this formulation that the problem is a combinatorial one. That is, if all the design choices were admissable as parts of an architecture, a total of (2)(2)(3)(2)(2)(2)(2)(2) = 384 architectures would be generated by means of this process. Even if half of them were not acceptable, we are left with a complex problem and no apparent solution. This problem is "solved" by the procedure that will be examined as part of the later EAM discussion.

Another issue raised by Rechtin and Maier is that of when do you stop architecting and begin the engineering of a system. The answer lies, by analogy, with the tasking that is prevalent in an Architecting and Engineering (A&E) company. The architects do the architecting, and the engineers follow up with the engineering. In that field, years of practice have established the scope and boundaries of the two sets of participants. This author has attempted to establish the nature of this boundary in a systems engineering context [3] so that system architecting is appropriately viewed as "top-level" system design that concludes with the completion of a very specific architectural process.

Two other DoD documents are relevant with respect to their perspectives in relation to systems engineering as well as architecting. These are the so-called 5000 series in the DoD. The DoD Directive known as 5000.1 addresses the Defense Acquisition System [11]. This directive invokes Systems Engineering as the approach that is most likely to optimize total system performance and minimize total ownership costs. It also calls for a "total systems approach" that addresses such factors as:

1. human systems integration

2. operational effectiveness
3. suitability
4. survivability
5. safety
6. affordability
7. supportability
8. total system performance
9. schedule
10. total ownership costs

The DoD Instruction 5000.2 has as its subject the Operation of the Defense Acquisition System [12]. This Instruction is explicit about the need for integrated architectures. These architectures are to lead the development of integrated plans and roadmaps for the construction of systems. There is considerable emphasis on being able to place capabilities into the hands of the users quickly, which in turn places pressure on the services to develop integrated architectures as rapidly as possible. This Instruction also supports the basic results that were obtained by the C4ISR Working Group, as discussed above.

Another important aspect of systems engineering is dealt with in the International Standard known as ISO/IEC 15288 which is concerned with the life cycle processes within systems engineering [13]. This standard defines a series of life cycle processes, one of which is called “technical processes”. A subset of these processes is the “architecture design process”. Its stated purpose is to “synthesize a solution that satisfies system requirements”. Outcomes of the recommended process deal with:

1. an architecture design baseline
2. system elements that satisfy the requirements
3. a definition of the incorporated interface requirements
4. traceability between architecture and requirements
5. a basis for verification
6. a basis for integration of system elements

This standard confirms the need for systems architecting, and also defines the content of this process. Ten very specific architectural design process activities are defined.

If there is an anomaly in the matter of the need for a well-defined architecting process, it may lie in the Integrated Capability Maturity Model (CMMI) produced by the Software Engineering Institute at Carnegie-Mellon, and endorsed by the DoD [14]. This integrated model addresses systems engineering (SE), software engineering (SW), and integrated product and process development (IPPD). One might assume, given that the CMMI covers both systems and software engineering, that the topics of systems and software architecting would be visible at the top level of definition. This turns out not to be the case, as a total of 25 important process areas are defined, none of which is explicitly an architecting process.

Architectural Views

Along with the architecting of systems, the notion of “views” of these architectures has been set forth. In this paper, views of architectures will be a dominant theme, and are meant to subsume the idea of “architectural descriptions” (ADs), a term that is also used in the literature.

The Department of Defense’s (DoD) C4ISR Architecture Framework [2] defines three views, namely:

- the operational view
- the systems view, and
- the technical view

The meanings of these views, as represented by the DoD, are:

- the *operational architecture view* is a description of the tasks and activities, operational elements, and informational flows required to accomplish or support a military operation
- the *systems architecture view* is a description, including graphics, of systems and interconnections providing for, or supporting, warfighting functions
- the *technical architecture view* is the minimal set of rules governing the arrangement, interaction, and interdependence of system parts or elements, whose purpose is to ensure that a conformant system satisfies a specific set of requirements

As a way to fully describe the outputs that may be associated with these views, the C4ISR framework provides a list of the essential and supporting architecture framework products. The following is a portion of that list, showing the essential (but not the supporting) products.

Exhibit 1 – C4ISR Framework Essential Products

All Views – Overview and Summary Information
All Views – Integrated Dictionary
Operational – High-level Operational Concept Graphic
Operational – Operational Node Connectivity Description
Operational – Operational Information Exchange Matrix
Systems – System Interface Description
Technical – Technical Architecture Profile

The number of “supporting” products that complement the above are:

- operational: 6 products
- systems: 12 products
- technical: 1 product

It is to be noted that only one product of a total of 13 systems products is considered essential. The other twelve are supporting.

The systems engineering views presented by the IEEE [6] are: functional, physical and foundation, with the full definitions provided in the standard. The *functional architecture*, as expected, focuses on the functions, their decomposition, and their interfaces. The *physical architecture* addresses the arrangement of elements and components. The *foundation architecture* represents core elements that are “presumed to be invariant throughout the system life-cycle”.

The architectural descriptions for software-intensive systems, also produced by the IEEE, deal with “collections of products to document an architecture” [7]. As a recommended practice for architectural descriptions, it focuses on “activities of the creation, analysis, and sustainment of architectures of software-intensive systems”. Views of an architecture are also called architectural descriptions (AD). The practice appears to stop short of recommending specific views, as does the C4ISR approach.

The final area of consideration in this paper is the architecting process defined by this author, and referred to earlier as the Eisner Architecting Method (EAM) [15]. This method attempts to simplify the process, resulting in three *critical views*, as follows:

- the *synthesis view*: a matrix that shows exactly how the main functions and subfunctions of the system are to be instantiated by real world design choices, for a minimum of three alternative system architectures
- the *analysis view*: an evaluation framework, in the form of a spreadsheet, that numerically assesses at least three alternative system architectures against a set of specific evaluation criteria
- the *cost-effectiveness view*: a graphical plot that shows the numerical measures of life cycle cost and effectiveness for the (minimum of) three alternatives

There are several features of the above method that are to be noted. The first is that they parallel the notions that were presented in an earlier military standard dealing with systems engineering [16]. That standard, 499B, identified four critical steps in the systems engineering process, namely:

1. Requirements analysis
2. Functional Analysis/Allocation
3. Synthesis
4. Systems Analysis and Control

The first 3 steps lead to the “synthesis” view, the 4th step is the “analysis” view, and the cost-effectiveness view follows directly from plotting the analysis results. A second feature to be especially noted is the fact that the method implicitly considers several alternative architectures, with three as a minimum. The basic rationale here is that the architect needs to consider a cost-effectiveness curve that has at least 3 defining points. One is the low cost approach, which meets all the basic system requirements at minimum cost. This is a very practical alternative since it forms the basis for many acquisition approaches for systems. Another alternative is the high-effectiveness approach that seeks to have the best performance from among a series of possibilities. An example is the choice of a high performance fighter aircraft. Our position, in the defense world, tends to be that we insist upon fighter superiority, no matter what the potential enemy. Finally, the third point in the cost-effectiveness space is to seek the “knee-of-the-curve”, which represents getting the most effectiveness per dollar before the curve levels off, as it does for almost all conventional systems, based upon empirical data. Finally, we note that it leads us to choosing an architecture based upon numerical assessments in the cost-effectiveness domain. As such, the three critical views are very different from those represented by the C4ISR Architecture Framework or the IEEE approach discussed above. However, it does not preclude creating some number of *additional* views, as suggested by the DoD or the IEEE.

New Architecture Views

The new views defined in this paper are above and beyond the three critical views of the EAM, as cited above. They are also based upon the numerical data that is generated as part of the EAM. This paper presents a total of ten additional views, as described below, although other views have been developed and used by this author as well as his students .

Additional View 1: Requirements Satisfaction. We note that none of the three critical views deals explicitly with the systems requirements. Therefore, this view focuses on these requirements in the form of a bar chart that measures the degree to which each of the defined requirements is being satisfied. The chart has lists of requirements for each of the key functions and subfunctions (rows) mapped against each of the three system alternatives. A derivative or supporting view is one that aggregates these measures from subfunctions to functions.

Additional View 2: Risk and Requirements. This view is related to the above view in that it focuses upon the assessment of the risk of not meeting the requirements, as defined above. That is, this view is a bar chart showing, for each requirement (under each function), the current measure of the risk of not be able to satisfy the stated requirement. As above, a supporting view can aggregate these assessments of risk from the subfunction to the function level.

Additional View 3: Interoperability. This new view addresses the very important matter of system interoperability, which has been a key issue for several decades. Given the synthesis view, we are now in a position to examine the design choices from top to bottom. That is, we look at the matrix that lists the design choices for each of the three alternatives. These choices were generated from left to right, i.e., for each subfunction, across the three alternatives. If there are a total of “N” rows in this matrix, we check for interoperability taking the first against the second, the first against the third, the first against fourth, and so on.This generates a chart of N(N-1)/2 entries. This is replicated for the other two alternatives.

Additional View 4: Cost by Function. This represents a disaggregation of life cycle costs, by function. If a system has a total of “M” functions, this view would place in evidence the cost of each of these M functions, and possibly their subfunctions. In that way we are able to see what we might be paying to add or remove a particular function of the system. In today’s world of expanding costs, it is important to identify areas that are generating high costs, and areas that might be deferred until more dollars might become available.

Additional View 5: Cost vs Requirements. Related to the above view involving costs is the notion that some requirements may be major reasons why costs are as high as they might be. This leads to the “95% solution” as described by this author [3]. We want to be able to identify “high cost requirements” with an eye toward reducing or eliminating these requirements, but only if (a) the resulting system performance is acceptable, and (b) the cost savings are substantial. Such a view can then be envisioned as a matrix of estimated costs, for each of the 3 system alternatives, as related to each of the system requirements.

Additional View 6: Sensitivity to Changes in Criteria Weights. The EAM calls for establishing weights for each of the evaluation criteria, in the “analysis” view. This is presented in the form of a spreadsheet that aggregates the effectiveness score for each of the three alternatives. The use of weighting procedures is supported in the literature (eg., ref. [17]). The impact of postulated changes in weights can be illustrated by plotting the cost-effectiveness of the three systems, both before and after the weight changes are made. Multiple changes can all be shown on the same graphical output.

Additional View 7: Effectiveness vs. Risk. The measures of the effectiveness for the three alternative systems are based upon defining a series of evaluation criteria, starting with the “analysis” critical view. Normally, the evaluation criteria are established for a series of categories. For example, typical categories, that in the aggregate measure effectiveness, might include [3]:

1. risk
2. human factors
3. reliability-maintainability-availability (RMA)
4. residual performance factors

On this basis, we can see additional views that plot remaining effectiveness against each of the above areas. So, for this view, we produce a plot of effectiveness vs. risk, for each of the three alternatives.

Additional View 8: Effectiveness vs. Human Factors. This view involves a focused look at human factors, as measured by sub-criteria such as (a) ease of use, (b) operator safety, and (c) user safety. The aggregated scores for these factors are plotted against the remaining factors that measure system effectiveness, for each of the three system alternatives.

Additional View 9: Effectiveness vs. RMA. Using the same basic idea as for the above, RMA subcriteria are identified, such as (a) frequency of scheduled maintenance, (b) ease of maintenance, and (c) complexity of assembly. The aggregated score for these factors is then plotted against the remaining effectiveness measures, for each of the three system alternatives

Additional View 10: Effectiveness vs. Residual Performance Factors. This last-cited view is one that deals with other performance factors that might be important in any particular evaluation. Examples of such factors include: manufacturability, market potential/demand, and appearance/aesthetic quality. Any one or the aggregation of these factors can then be plotted against the remaining measures of effectiveness.

The ten additional views cited above can be thought of as derivatives of the EAM. As such, they are based upon the data generated by the EAM, and therefore focus on providing quantitative information. This type of information gives us a sound foundation for comparing the alternative architectures, and ultimately selecting a preferred architecture. Once a preferred architecture is chosen, these new views provide numerical information regarding this system, and therefore can be used to measure performance, effectiveness and cost throughout the system life cycle.

As noted previously, there is nothing in these new views that keeps us from also accepting the C4ISR perspective represented by the operational, systems and technical views. Rather than contradicting one another, they can be thought of as supportive of one another. The latter are more qualitative; these new views that flow from the EAM are all quantitative.

Summary

This paper has presented a series of new views that support the system architecting process. They are quite different from the views suggested by the DoD's C4ISR framework, and also different from the views implied by the IEEE work on systems engineering and architecting. These new views are derived from the EAM procedure which has at its core the three critical views of (1) synthesis, (2) analysis, and (3) cost-effectiveness. These three critical views also imply an architecting process. This is a very important feature of the EAM.

The suggested new views are:

1. Requirements Satisfaction
2. Risk and Requirements
3. Interoperability
4. Cost by Function
5. Cost vs. Requirements
6. Sensitivity to Changes in Criteria Weights
7. Effectiveness vs. Risk
8. Effectiveness vs. Human Factors
9. Effectiveness vs. RMA
10. Effectiveness vs. Residual Performance Factors

A brief and final overview of the rationale for each of these views is provided below.

1. **Requirements Satisfaction.** Requirements tracking is one of the system metrics that most agree is necessary as a project evolves. This view, however, is not the same as tracking over time. Rather, it deals with the extent to which the system architecture does or does not meet the stated requirements, as defined by the user. Virtually all recent inputs from the DoD emphasize the need to meet the system requirements, and other standards make the same point. In this view, we are accepting these inputs and responding accordingly.

2. **Risk and Requirements.** Here we are recognizing the need to relate levels of risk with the stated requirements. Some requirements will increase risk, as measured in cost, schedule or technical performance terms. This view gives us an explicit measure of this critical relationship. A program manager may wish to reduce or eliminate a requirement that carries with it more risk that he or she is willing to tolerate.

3. **Interoperability.** We are reminded of the fact that the matter of system interoperability has been a key consideration for system builders for many years. This is reflected in DoD guidance, as they experience problems with interoperability between the services and with systems produced and used by our allies. Here we have an opportunity to look at interoperability in an explicit way, as part of our architecting process.

4. **Cost by Function.** We are seeking a cost effective solution as part of our architecting process. If we are able to view the costs that are likely to be incurred for each system function, we will be able to trade off system functionality when our costs exceed budget. On the other hand, it may be possible to use other less costly approaches, such as greater use of COTS, for some of the sub-functions.

5. **Cost vs. Requirements.** This is similar to the risk consideration, except that it tells us which requirements appear to be driving the costs up. Here again, the project manager may be willing to give up some requirements so as to be able to build the system within the budget that has been allocated. Other lower cost approaches may be defined within the construct of the three postulated architectures.

6. **Sensitivity to Changes in Criteria Weights.** The selection of the preferred architecture can be a function of the weights that are assigned to the different evaluation criteria in the architecting process. By exploring these sensitivities we are able to see if our "answer" changes if some criteria lose or gain importance. This relates to the notion that it is the system user that ultimately establishes the importance of the various criteria. The system

architect can assist in this process by making the sensitivities explicit.

7. **Effectiveness vs. Risk.** This addresses the extent to which we are taking additional risk in order to achieve the required effectiveness levels. As such, it elucidates the nature of that trade-off.

8. **Effectiveness vs. Human Factors.** Through this view we are in a position to see explicitly how the human design features impact the system effectiveness. Such a view will allow us to redesign the system with more (or less) human interactions.
9. **Effectiveness vs, RMA.** RMA factors are a subset of the effectiveness assessment. Thus this view is giving us a measure of the degree to which RMA enhances or detracts from our overall effectiveness measurement.

10. **Effectiveness vs. Residual Performance Factors.** This view parallels the above view, but deals with other performance factors, such as those cited earlier.

Other new and quantitative views can also be developed from the EAM procedure, and these views can be supplemented by views set forth elsewhere, for example, the C4ISR architecture framework. It is noted that every one of the above views is quantitative, and in that sense supports the notion of architecting with measures (eg., technical performance measures – TPMs) or metrics. This results from the fact that the architecting process itself is quantitative, yielding specific sets of numbers that measure system effectiveness, performance and cost.

References

Rechtin, E., (1991). “Systems Architecting” Upper Saddle River, New Jersey: Prentice-Hall

C4ISR Architecture Framework, (1997), Washington, DC, Department of Defense, December 18

Eisner, H., (1997, 2002). “Essentials of Project and Systems Engineering Management”, New York, NY: John Wiley

Rechtin, E. and M. Maier, (1997).“The Art of Systems Architecting”, Boca Raton, FL: CRC Press

Rechtin, E., (2000). “Systems Architecting of Organizations”, Boca Raton, FL: CRC Press

Standard for Systems Engineering, IEEE P1220 (1994). New York: Institute of Electrical and Electronics Engineers

Recommended Practice for Architectural Description, IEEE P1471 (1999). New York: Institute for Electrical and Electronics Engineers

Keating, C., R. Rogers, R. Unal, D. Dryer, A. Sousa-Posa, R. Safford, W. Peterson and G. Rabadi, “System of Systems Engineering”, Engineering Management Journal, Vol. 15, No. 3, September 2003

Gansler, J, A. Valetta and D. Buchholz, *Strategic Direction for a DoD Architecture Framework,* Office of the Secretary of Defense, The Pentagon, Washington, DC Feb. 23, 1998

C4ISR Architecture Working Group Final Report, Washington, DC: Department of Defense, 14 April 1998

The Defense Acquisition System, DoD Directive 5000.1, Washington, DC: Department of Defense, May 12. 2003

Operation of the Defense Acquisition System. DoD Instruction 5000.2, Washington, DC: Department of Defense, May 12, 2003

Systems Engineering – System life cycle processes, ISO/IEC 15288, ISO, 11 Jan 02

CMMI, v.1.1, Software Engineering Institute, Carnegie Mellon, Pittsburgh, PA

Eisner, H., “Eisner’s Architecting Method (EAM): Prescriptive Process and Products“, Tutorial presented at INCOSE 2003, 13th Annual International Symposium, 29 June – 3 July 2003, Arlington, Virginia

Systems Engineering, Military Standard 499B, (Draft) (1992), Washington, DC: Department of Defense

Eisner, H. (1988). “Computer-Aided Systems Engineering”, Upper Saddle River, New Jersey, Prentice-Hall

Author

Dr. Howard Eisner came to The George Washington University in 1989 after thirty years in industry where he served in both R&D and management roles, and held the position of president of two systems and software engineering companies. His current position is Distinguished Research Professor and Professor of Engineering Management in the School of Engineering and Applied Science. He is a Life Fellow of the IEEE and is a member of Tau Beta Pi, Eta Kappa Nu, Sigma Xi, and Omega Rho, various research and engineering honor societies. He has written two books on systems engineering, and one on reengineering. He holds BEE, MS and Doctor of Science degrees from The City College of New York, Columbia University and The George Washington University, respectively.