Documenting Complex Systems in the Enterprise

Carlos Troche
The MITRE Corporation
ctroche@mitre.org

1.0 Introduction

Documenting how complexity informs systems engineering should be done in a systematic, repeatable form that is widely recognized and accessible to an audience of systems engineers, analysts and managers. While research is on-going in complexity, it is important to find ways of visualizing this information in practical ways that can enable its use and further development by the researcher and the uninitiated alike. MITRE supported the DOD in adopting and adapting a set of modeling practices – named architectures - that can serve the need of informing systems engineering about complexity and its effects. This paper will introduce architectures, a tool to facilitate documenting and understanding complexity.

1.1 Relevance

Finding practical ways of informing engineering and other disciplines on complexity has become increasingly important as the level of apparent complexity of information must be maintained in sufficient detail to inform future generations or conveyed to decision-makers.

Two particularly stark examples illustrate the consequences of this lack of tools for communicating and visualizing complex information.

- A review of different bridge designs over 150 years showed each design degenerated into a bridge collapse incident within roughly 30 years of it first being instituted [Sibly, 1977]. It seems the lack of a complete appreciation of the whole system, its components and their interactions (including failures), resulted in the progressive elimination of safety elements of the design until it precipitated a complete failure, sometimes with tragic results.

- The Space Shuttle Columbia Accident Investigation Board Report related how NASA's penchant for using business graphics tools to present and discuss detailed engineering data led to an oversimplification of the problems raised by the engineering staff, so what were critical technical concerns quickly became management toss-ups as they were reviewed by decision-makers. What's perhaps more tragic is that this same problem was noted in the Space Shuttle Challenger Accident Investigation Board report 20 years ago (back to fading memories).

2.0 Architectures – Tool to Visualize the Enterprise's Complexity

Architectures have been formally defined as the "structure of components, their relationships, and the principles and guidelines governing their design evolution over time" [IEEE, DOD]. A more practical definition is that an architecture is a model (but not necessarily a single model) that details what a system does (activities), the relationships among those activities, how a system performs those activities (processes), the *functions* of those tools that are used to perform the activities in total or in part (*system functions*), the actual *tools* themselves that *perform* those system functions (*systems*), the skills and experience of those performing the activities (*roles*), all *related* through a documented, repeatable, and agreed-upon structure (*taxonomy*), and rendered through an equally approved, documented and agreed-upon rule set (*framework*). Each of these activities, processes, system functions, systems, and roles – in turn - are described in terms of *attributes*. Depending on the scale, state, and emergent properties of the model, the model may turn out to be a collection of models, some related, others nested.

For the sake of simplicity and to overcome human cognitive limitations, architectural models are often rendered as a set of different perspectives or *views*, allowing a more focused look into one or more aspects of interest, in much the same way that models of the human body often have separate views of the circulatory, skeletal, muscular system, etc. The key, however, is that all of these views are integrated, e.g., the rendition of the circulatory system must acknowledge the shape, curves, and special structures the skeleton gives the cranium to accurately show blood flow to the brain.

Let's apply these two definitions.

In traditional Systems Engineering, architectures inform the engineering process by decomposing, dissecting the system into its components, and evaluating the role of each component within the integrated whole of the system. In TSE, architectures serve as the equivalent of a human anatomy text, carefully identifying the organs, tissues, bones, sinews, etc, their functions and relationships.

In Complex or Enterprise Systems Engineering (C/ESE), the architecture describes capabilities manifest in the enterprise that may or may not be evident simply from the component systems. Thus, enterprise architecture exists at multiple scales. And, at each scale there are architectural artifacts that describe the properties (capabilities) at the scale, as well as details of the components and their interactions at the next finer scale. An example of emergent potential discovered through the use of architectures may be seen in the study conducted some years back on the Copernicus Architecture [Levis, 1994]. In that study two modes of responses to queries – one very terse, the other verbose – were compared against the availability of communications capability. The study showed, counter intuitively, that the more verbose communications mode was more effective as the communications capacity dropped.

To sum up, architectures inform complex systems engineering by providing a multidimensional description rendered at various scales and scopes. In traditional systems engineering, architectures decompose the "system" under study into its component elements maintaining and further specifying the relationships between the components. For Complex or Enterprise Systems Engineering, architectures decompose within and across the system's environment, identifying and synthesizing the opportunities for emergence and – by extension – innovation.

3.0 Architectures at the Enterprise Level: US Air Force Application

This section illustrates the use of architectures in documenting and informing an organization about the complexity of its enterprise, i.e., its collection of information technology capabilities used in weapons, weapons systems, intelligence, cryptology, communications and business operations. For the U. S. Air Force (USAF), as well as for all other exécutive branch agencies in the Federal government, having this level of understanding is required by law (the Clinger-Cohen Act) as a pre-requisite to the rationalization of the investments required for their development and upkeep.

To handle the scale typical of an enterprise, enterprise architectures are often developed as a nested set of models. The following illustrations offer two such views, one from the Open Group (Figure 1) [Blevins, 2005], the other from the Air Force's Operations Support Enterprise Architecture (Figure 2). Note the notion of the "Architecture Continuum" in the Open Group's graphic, showing how each architecture scale tends to support products corresponding to that scale. Foundational enterprise architectures,

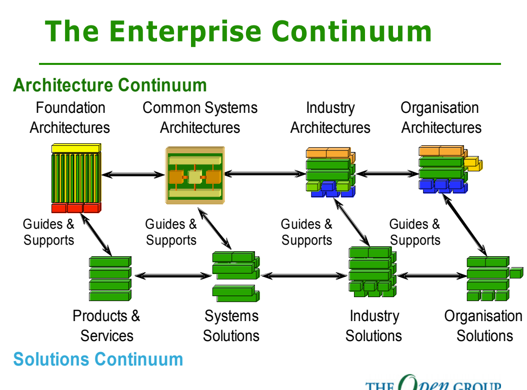

Fig 1 – The Enterprise Continuum

such as those developed by the Federal Government to guide the provision of services to the citizenry, support the definition of key products and services.

The Air Force's Operations Support Enterprise Architecture (OSEA) shown in figure 2 [McFarren, 2005] illustrates the multi-dimensional nature of architectures as they are

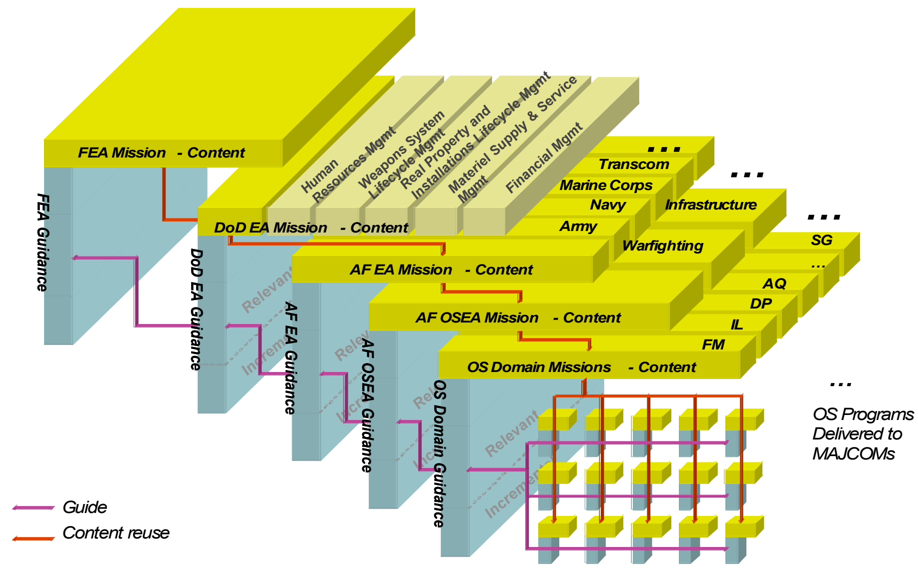

Fig 2 – OSEA Taxonomy

"scaled up" to address enterprise-wide effects. Referring to figure 2, the Federal Enterprise Architecture (FEA) defines the services the government provides its citizens, including those associated with national defense, of which operations support is part. The FEA "bounds" the enterprise of interest, guiding lower-level, more detailed architectures in adding further detail. Note that the next level represents a definition of the Department of Defense (DOD) Enterprise Architecture (EA) highlighting many of the functional areas representing operations support (e.g., Financial Management). Subsequent levels apply these functions to the Air Force at large, the Air Force's operations support domain, and it then breaks down this domain into its different functional members as they are implemented within the Air Force. At the bottom of the cascade are the individual Operations Support programs that realize in hardware, software, personnel, and procedures the requirements specified in the architectural models depicted in the previous levels. Of particular interest in this rendition are the concepts of guidance and content reuse, both of which ensure traceability as you travel up and down the scale. Each level's guidance constrains the one below, focusing the architecture further. The constraints are more that just semantic, as each level's content is reused to the degree necessary for further detailing by the level below. So, for example, the Air Force EA cannot cover missions that are outside of the DOD EA.

Nested architectures, scale and constraints hint at an underlying set of rules or framework that defines the structure and its relationships. Similar to an anatomical map, the framework indicates what the relationships of interest are and – sometimes – how to document them. Most frameworks tend to stop there, similar to the human anatomy text that stops at the documentation of which bone, sinew, etc, is connected to which other member. In architectures, there are various frameworks that can be used (e.g., Zachman, DoD Architecture Framework, etc); the critical point is that one should be declared and used consistently if architectures are to be useful for the Enterprise.

Figure 3 is an illustration of the framework used by the USAF.

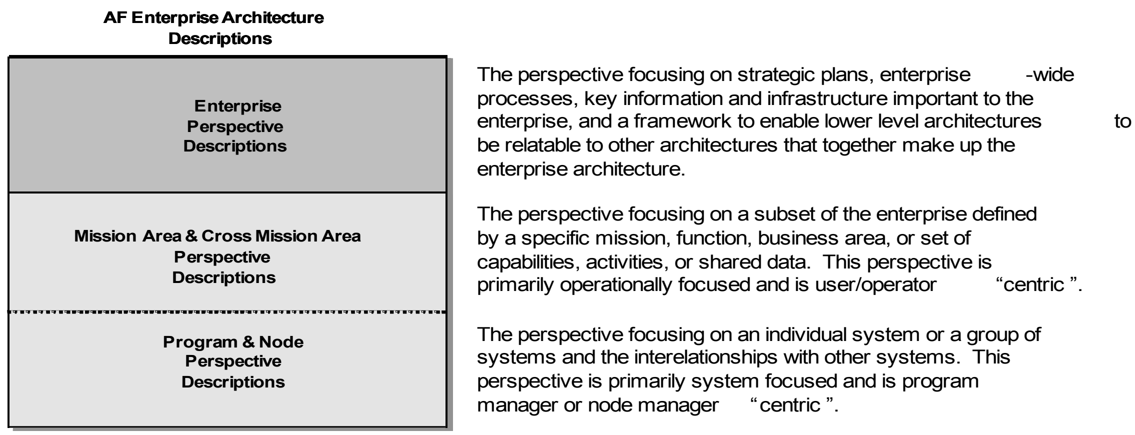

Fig. 3 USAF Enterprise Architecture Descriptions

At the top of this framework are those perspectives that cut across the entire Air Force especially in a way that assists in the discovery of emergent relationships and potential. The top level – as well as the subsequent one, tends to constrain, guide the development of the lower ones, and must acknowledge its place in a larger "framework" or context.

For the Air Force, their Enterprise Architecture Framework defines its context as shown below in the next two graphics (Figures 4 and 5) [Baehre, 2005]. In the first one, the context is simply generalized as the set of inputs and drivers that trigger the content of the architecture descriptions and the ultimate uses and impacts of the architecture.

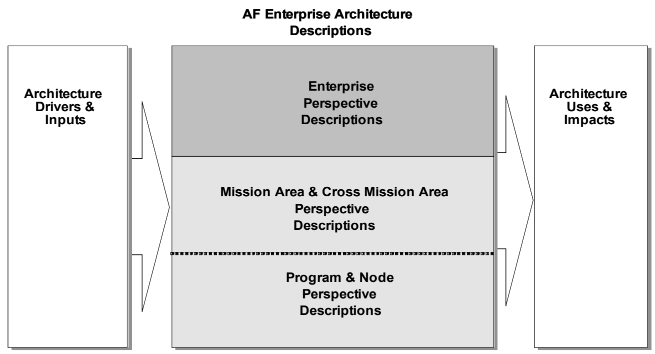

Fig. 4 – USAF Enterprise Architecture Context

In the second one, all components of the first are detailed. While it is beyond the scope of this paper to discuss each subcomponent, note that many of the drivers and outputs are what would be expected of a public service like national defense (recall previous discussion on the Federal Enterprise Architecture and its definition of services to the citizenry). On the "driver" side of the flow are statute and policy guidance from the DOD and other executive agencies; as well as other architectures and architecture-like guidance. On the output or "uses" side are the development and budgeting of new Air Force systems and the planning and operations of Air Force forces.

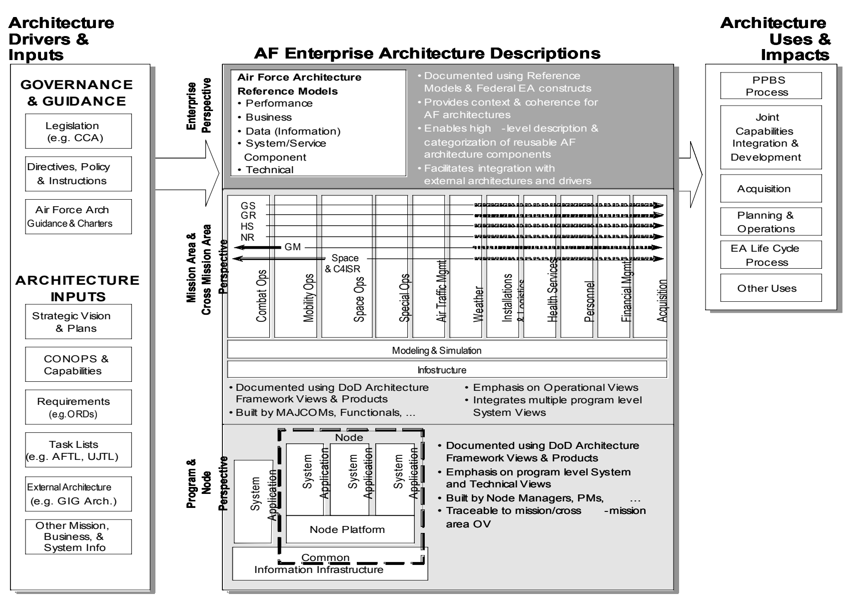

Fig. 5 – USAF Enterprise Architecture Context (Detailed)

Consider for a moment how the Fig 5 architectural depiction informs Complex or Enterprise Systems Engineering. First, it unequivocally encapsulates the enterprise, defining its nature, without constraining the possibility for emergent behaviors to surface. It does not prejudge the degree of dependencies or influences of the drivers and uses (horizontally) or the different levels (vertical). Next, the vertical integration is loose enough to allow autonomous development within each level, development that may be prompted horizontally by the integrating concepts of operations shown as GS, GR, HS, NR in the case of the Mission Area perspective ; and/or by the vertical interaction with the complex solutions developed within the Program/Node perspective of the enterprise Finally, the architectural construct in Figure 5 establishes a set of outcome spaces [Kuras, White, 2005] for the outputs of the architecture without prejudging any one of them.

So, architectures support the enterprise and – specifically – Complex/Enterprise Systems Engineering by enabling the visualization and exposition of the enterprise's contents, and in the process highlighting capabilities manifesta t the enterprise level that show potential for emergent behavior. This visualization and exposure are not just graphical artifacts, but reflections of an underlying rule set or framework defining the contents and spaces of interest at each level. If developed correctly, architectures can document in data and graphics enough sufficient detail about complexity that can assist the engineering of complex systems.

4.0 Conclusion

Informing engineering and decision-makers about complexity in a reputable, repeatable form, is a critical task that must evolve in parallel and at the same time as the rest of the discovery of complex behavior and systems. Unless this is done, there is an increasing likelihood the discovery and harnessing of complex behavior will be retarded with potentially tragic consequences. Documenting complexity with the aim of informing engineering will do for complexity what the printing press did for language – facilitated understanding by normalizing its representation. Architectures – nested models structured to tease emergent behavior in complex environments – may be used to document this relationship within a larger framework that provides for their repeatable application.

References

Baehre, Mark, The MITRE Corporation, provided this illustration and associated description

Blevins, Terence, The MITRE Corporation, provided this illustration and associated description

IEEE Standard 610.12, and Department of Defense Architecture Framework, 2004

Kuras, M. L., White, B. E..,*"Engineering Enterprises Using Complex-Systems Engineering,"* as published by INCOSE, 2005.

Levis, Alexander, Hearold, Susan L., Perdu Didier M.,*"Effectiveness of Two Modes of Information Pull in the Copernicus Architecture,"* as published in "Science of Command and Control: Part III Coping with Change," 1994.

McFarren, Michael, The MITRE Corporation, provided this illustration and associated description

Sibly and Walker, *"Structural Accidents and Their Causes,"* (1977) as discussed in Petroski's *"Design Paradigms,"* p. 169. Petroski goes on to postulate that absent the systematic study of failure (which, in turn, relies on comprehensive documentation of the system and its components), this pattern will continue.

Saunders, Thomas, Study Chair, *"System-of-Systems Engineering for Air Force Capability Development,"* Air Force Scientific Advisory Board, 30 June 2005