Interactions Among Components in Complex Systems

Wilson N. Felder[1]
Stevens Institute of Technology, Hoboken, NJ 07030

A key attribute of all Systems-of-systems is the dynamic connectivity among their component member systems. This effect requires a higher level of attention to the inter-system interfaces on the part of designers, developers, testers, and program managers. Specifically, during the life cycle of such systems, we must be prepared to react flexibly to changing requirements, opportunities for design modification, and insertion of new technology. With complex systems, we also must be prepared to react to emergent properties of the system that are "essentially unpredictable" even given a complete knowledge of the functioning of the component systems. This reality requires us to modify our approaches across the board, including, in part, processes for requirements definition, system design, system decomposition, verification and validation, configuration management, interface control, and program management, as well as staff education and training and supply chain management. The challenge for managers of complex system development lies in being able to recognize unambiguously when aberrations are the result of complexity, as opposed to just bad management. The challenge for systems engineering theorists is to develop the mathematical framework that allows such discrimination.

I. Synopsis (introduction and background)

The purpose of this paper is to provide a framework for discussion of the proposed theme for the AIAA Complex Aerospace Systems Exchange (CASE) 2013: interactions among member systems in a system-of-systems, and potential design, development and management approaches to mitigate the difficulties these interactions cause to the system engineer.

Big, complicated aerospace systems (like those described by Feiler, et al[1]) – and not just Complex ones, either – have attributes that cause problems in design, development, test, and operations. To begin with, these systems inhabit a world in which technology is evolving at breathtaking speed, which exacerbates the natural effects of complexity. Secondly, some systems which meet the definition of complexity may do so by exhibiting the particularly vexing phenomenon known as emergence, which is unique to complex systems and which appears to invalidate many best practices of traditional systems engineering. Finally, complicating the picture is the reality that engineered systems such as aircraft and weapons systems are embedded in socio-technical systems which are in and of themselves complex. So, when we talk about problems with complex systems, we are usually responding to three different phenomena; first, impacts from the rapid evolution of technology; second, issues of coordination that arise across the broad spectrum of systems-of-systems; and finally, unexpected results, which may be a special feature of a subset of complex systems.

II. The Nature of the "Complex System" Problem

Examples of difficulties with system development are easy to find, but poorly documented. While Program Managers are willing to talk about their experiences in system development, there are few documented and rigorous studies of system development failures. Upon reflection, it seems that these difficulties fall into three main areas.

A first and obvious condition involves the extreme speed of development of new technology. As reviewed by Felder and Collopy[2], the pace of technology is unprecedented in history, and the technology refreshment cycle exceeds the system development cycle by an order of magnitude. As a result, systems designed to take advantage of the latest technologies are doomed to deploy antiquated equipment and software by the end of the typically decade-long acquisition cycle for most major aerospace systems. It is therefore critically important to provide for innovation, sometimes radical in nature, as the system is designed and developed. There is a natural tension between the need for control of design at early stages in development and this need for adaptability, and the tools of classical system engineering, in particular the mantra that requirements must be fixed and changes in requirements controlled from the very beginning of a project, act to prevent flexibility.

[1] Distinguished Service Professor, School of Systems and Enterprises, Fellow.

This is what killed the FAA's Advanced Automation System (AAS) in 1992, perhaps the first nationally important System-of-systems to fall prey to the information revolution's explosive innovation cycle. The original AAS design was developed (1982-1984) before the widespread availability of local area network technology, consequently the vendor was forced to develop a one-off LAN architecture, which was obsolete by the time the system entered test[3]. Luckily, this failure directly influenced the early SoS work of a number of authors[4,5]. Needless to say, but highly frustrating, the "lessons learned" from that failure were completely erroneous, failing to realize that it was the direct result of technology overtaking the ponderous acquisition process, and not a failure to correctly apply management techniques!

Second, as if the speed of technology development were not enough of a challenge, we have also apparently entered a period where, because of the dynamic, asynchronous, and numerous interactions among the component systems in a SoS, certain aerospace systems produce results that are not fully predictable based on a complete knowledge of each of the component systems. There is a paucity of carefully controlled experiments designed to characterize these phenomena definitively, but there is much anecdotal evidence, including reports of interactions between the Flight Control System and Flight Management System on certain commercial aircraft; interactions between the propulsion control system and other aircraft management systems on some military fly-by-wire aircraft; and interactions among the autonomous and diverse systems brought together to constitute today's most advanced air traffic control systems. Indeed, it has been pointed out that given enough software brought together in one place, and when chunks of that software are designed to operate independently as well as collaboratively, then unpredictable interactions can occur even in what might appear to be a single system[6]. Unfortunately at this point most of the evidence is anecdotal.

Third, the concept of socio-technical systems, which originated in the management literature, but has been eloquently summarized by Miller and Page[7] as well as others, further complicates the job of the aerospace system engineer, because engineered systems such as aircraft and weapons systems are embedded in these socio-technical systems. It is in the very nature of socio-technical systems that they are complex, indeed, complex adaptive, because of the presence of human operators.

The Air France 447 accident, dissected in the CASE 2012 Executive Workshop[8], is a prime example: it illustrates how even a thoughtfully designed and well behaved system can produce catastrophically negative results when immersed in a socio-technical system context. Briere et al[9,10] clearly lay out the careful fault tolerant design of the Airbus family of fly by wire aircraft; on the other hand, the Bureau d'Enquêtes et d'Analyses (BEA – the French National Transportation Accident Investigation Board) reports on the Air France 447 accident[11] and an earlier similar accident[12] and Charette[13,14] lay out in chilling detail the impact of an engineered system poorly matched to its socio-technical environment.

As it turns out, all Systems-of-systems possess dynamic connectivity, with interaction among the component systems happening asynchronously and freely, and sometimes this leads to unexpected results. It goes without saying, as already noted earlier, that we need to be able to react to rapid changes in technology by being open to dynamic changes in requirements and solutions during the lifetime of a system. However, complex systems demand a different kind of preparation: to be ready to react to emergent phenomena ***exclusive*** of any change in requirements or technology! Indeed, the signature of a complex system is that unexpected results emerge ***even though*** the standard disciplines of system engineering (system decomposition, configuration management, risk mitigation) are skillfully and effectively employed. Useful approaches to addressing the demands of complex systems have been suggested by Sheard and Mostashari[15], among others[16,17]: we need to continue to refine these approaches in response to our growing understanding of the theory behind complex system behavior.

Interaction among component systems in a complex SoS seems to be the key ingredient that drives the unique challenges of these systems. Unexpected results from this interaction are a defining characteristic of complex systems, and manifest themselves across all dimensions of engineering and management related to the development, test, and operation of these systems.

III. The Question of Emergence: a Brief (or maybe not so brief…) Digression

Up to this point, I have used the term "complex system" in a very generic way to describe any system with component parts that are systems in their own right. I have left the definition of a system, and of a complex system, purposely vague, in order to be able to concentrate on the effects from system-to-system interaction that are so vexing to aerospace system engineers. However, there are in fact different kinds of systems out there, and their definitions are both unsettled and important. It is both appropriate and instructive to start with first principles as laid out by Ludwig von Bertalanffy, whose fundamental understanding about the properties of open systems should serve as a foundation for any work on complex aerospace systems[18].

A useful working definition of a complex system is one that exhibits the property of *emergence*. Emergence is defined as any result from the system that cannot be predicted from a complete knowledge of the system's component parts. A system is defined as a set of component parts brought together to achieve a goal, an attribute we call *autonomy*. Under this set of definitions, a subsystem is any logical assembly of components that cannot achieve its own particular goal alone (i.e., does not possess the attribute of autonomy). Think of the propulsion system on an aircraft, which serves an important purpose, but cannot do so fully unless it is attached to the aircraft itself. Systems of systems, under this definition, are systems whose component parts are systems themselves. In a system of systems, the component systems receive a benefit from membership in the whole, and also contribute to the goal of the larger assemblage while simultaneously achieving their own goals. We call this attribute *belonging*. Systems of systems also possess the attributes of *diversity* (the component systems are not simply replicas of one another), and, necessarily, are connected to one another, either in a fixed manner (static connectivity) or as required to perform their function (*dynamic connectivity*). These are the definitions set out in a series of papers by Maier[19], Boardman and Sauser[20], and Baldwin and Sauser[21], and developed into a system taxonomy by Baldwin et al[22], which is consistent with work done on complex adaptive systems described by Boulding[23], and Miller and Page[7]. The definitions and taxonomy are also consistent with the emerging view of complex systems as embodied in the DoD handbook on system-of-systems engineering[24].

It is an open question whether emergence can actually exist in a system made up entirely of engineered components. Edmonds[25] has eloquently posed this as the question of emergence versus ignorance – in other words, is an unpredictable result truly a feature of the complex system, or merely an artifact of our lack of understanding? Recent work has been undertaken that seeks to develop a mathematical test of this hypothesis, but the effort is still in its infancy[26].

It's also important for us to recognize how the science of complexity in other disciplines relates to the complexity problem in aerospace. So, we need to understand, for example, how Snowden's taxonomy[27], Edmond's definitions of complexity[25], and Jackson's use of the concepts of systems thinking with respect to management[28,29,30], relate to our problems of design and development. There is a rich literature on complexity in management theory[31] and social science[32]. Since the objective of such work in general relates to human behavior, however, it is only tangentially related to our aerospace systems problem, and the similarity of the terminology used should not be taken as evidence that the conclusions are applicable to our problem.

Finally, there is a tantalizing sense that the existence of dynamic connectivity in a system is what creates the environment in which emergence can occur: there seems to be something about the dynamic, ansynchronous and unpredictable flow of digital data across the interfaces among systems (and even among the subsystems inside a sufficiently rich digital system[6]) that triggers unforeseen behavior. As additional effort is applied to these open research questions, we will come closer to understanding the mechanisms underlying the behavior of complex systems and an ability to design for mitigation of their unpredictable emergent behaviors.

IV. Reflection and Challenge

An important "to do" for system engineering theorists is to settle the question of emergence in complex aerospace systems: is it a real and fundamental feature? Is it an excuse for ignorance? Does its existence subdivide the set of all systems into two subgroups, one complex, and the other not? How do we detect its presence *a priori*, in order to make wise design decisions for its mitigation? Most interestingly, is it a result of how the systems in a SoS are interconnected, and therefore, is it a provable consequence of certain types of system architecture? If so, what are the mathematical relationships that allow us to describe the effect?

This goes beyond just fixing an interface, because the nature of the connectivity is such that we cannot predict exactly how the results will go haywire, only that they are likely to do so. Indeed, if it is a symptom of complex systems that they don't fail very often, but when they do, they do so in spectacular ways, that may well be the result of timing: for most of the time, the asynchronous and dynamic connections among component members is of no consequence, but sometimes it is of very great consequence, with probabilities much larger than expected for a well behaved Gaussian distribution.

We can imagine several important impacts that come from these sorts of considerations:

- The design process must allow for adjustment based on the distributed nature of the system, both to take advantage of potential solutions that come to light during the process, and to tackle unexpected outcomes as they appear;
- System architectures must be designed in a way that provides a graceful degradation to a stable, if less capable, state in the inevitable situation where inter-system interactions disable the system;

- Test plans must be developed that wring out the distributed, dynamic, and unpredictable nature of complex systems;
- Staff must be educated, hired, and trained to work across the dynamic connections that are the signature of complex systems;
- and, Management tools and techniques over the life cycle must be adapted to allow adjustment as these impacts become visible.

The goal of this paper, of course, is not to propose solutions, but to pose a challenge: as systems engineers we must develop tools that help developers, testers and managers of complex aerospace systems tell the difference between unexpected behaviors that occur because of a lack of management discipline, and those that result from the fundamental nature of the system itself.

References

[1]Feiler, Peter, Richard P. Gabriel, John Goodenough, Rick Linger, Tom Longstaff, Rick Kazman, Mark Klein et al. (2006) "Ultra-large-scale systems: The software challenge of the future." *Software Engineering Institute* (2006).

[2]Felder, Wilson N., and Paul Collopy (2012) The Elephant in the Mist: What we don't Know about the Design, Development, Test and Management of Complex Systems, *Journal of Aerospace Operations*, IOS Press, v.1, no.4, pp.317–327.

[3]Boppana, Krishna, Sam Chow, Olivier L. de Weck, Christian LaFon, Spyridon D. Lekkakos, James Lyneis, Matthew Rinaldi, Zhiyong Wang, Paul Wheeler, Marat Zborovskiy, and Leonard A. Wojcik, (2006), "Can Models Capture the Complexity of the Systems Engineering Process?," *International Conference on Complex Systems,* (pp. 25-30).

[4]Carlock, Paul and Steve Decker, (1998) "Development and Implementation of an Architecture for the National Airspace System," *Systems and Information Technology Review Journal*, Spring/Summer 1998, pp. 6-16.

[5]Carlock, Paul G. and Robert E. Fenton, (2001) "System of systems (SoS) enterprise systems engineering for information-intensive organizations," *Systems Engineering*, Vol. 4, No. 4, pp. 242–261.

[6]Serna, Frank (2012), Draper Laboratories, Cambridge, MA, Comments made during CASE 2012 panel discussion on non-technical obstacles to system development, personal communication.

[7]Miller J. H. and S.E. Page (2007), *Complex Adaptive Systems: An Introduction to Computational Models of Social Life,* (Princeton Studies in Complexity), Princeton University Press, Princeton, NJ.

[8]American Institute of Aeronautics and Astronautics (2012), CASE 2012 presentations: https://www.aiaa.org/Secondary.aspx?id=12866, [accessed 28 June, 2013]

[9]Briere, D., C. Favre, and P. Traverse. (1995) "A Family Of Fault-Tolerant Systems: Electrical Flight Controls, From Airbus A320/330/340 To Future Military Transport Aircraft." *Microprocessors and Microsystems* 19.2 (1995): 75-82.

[10]Brière, Dominique, Christian Favre, and Pascal Traverse. "Electrical flight controls, from Airbus A320/330/340 to future military transport aircraft: A family of fault-tolerant systems." *Spitzer [190], chapitre* 12 (2007).

[11]Bureau d'Enquetes et d'Analyses, (2012) *Final Report On the Air France 447 Accident*, (in English), 219pp.

[12]Bureau d'Enquetes et d'Analyses, (1994) *Rapport Preliminaire, Commission d'Enquete sur l'Accident Survenu le 30 Juin 1994 a l'Airbus A330 Immatricule FWWKH*, 28 July 1994 (in French).

[13]Charette, Robert N. (2009) *Automated to Death*, IEEE Spectrum online, December 2009.

[14]Charette, Robert N. (2012) *Air France Flight 447 Crash Causes in Part Point to Automation Paradox*, IEEE Spectrum Blogs, 10 July 2012.

[15]Sheard, Sarah A., and Ali Mostashari. (2008) "Principles of complex systems for systems engineering." *Systems Engineering* 12.4, 2008: pp.295-311.

[16]Lane, Jo Anne, and Barry Boehm (2008) "System of Systems Lead System Integrators: Where Do They Spend Their Time and What Makes Them More or Less Efficient?", *Systems Engineering*, Vol. 11, No. 1, Feb. 2008, pp. 81-91.

[17]Weinstock, Charles B., Goodenough, John G., Hudak, John J. *Dependability Cases*. CMU/SEI-2004-TN-016, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA. May 2004.

[18]Von Bertalanffy, Ludwig (1950) "The Theory of Open Systems in Physics and Biology", *Science*, v.111, January 13, 1950, pp. 23-29.

[19]Maier, M. W., (1998) "Architecting Principles for Systems of Systems", *Systems Engineering*, vol. 1, no. 4, pp. 267-284,1998.

[20]Boardman, John, and Brian Sauser, (2006) "System of systems – the meaning of of", Paper Presented at the Proceedings of the 2006 IEEE/SMC International Conference on System of Systems Engineering, April 2006. Los Angeles, CA.

[21]Baldwin, W. Clifton, and Brian Sauser, (2009) "Modeling the characteristics of system of systems," Paper presented at the 2009 IEEE International Conference on System of Systems Engineering (SoSE), 1 June 2009, Albuquerque, NM.

[22]Baldwin, W.C., W.N. Felder and B. Sauser, (2011), Taxonomy of increasingly complex systems, *International Journal of Industrial and Systems Engineering* **9**, 298–316.

[23]Boulding, K. E. (1956) General Systems Theory – the Skeleton of Science, *Management Science*, v.2, pp.197-198.

[24]Office of the Secretary of Defense (OSD), (2008) DoD System Engineering Guide for System of Systems Engineering, vers. 1.0. Washington, DC: DoD, August 2008.

[25]Edmonds, B., (1999) *Syntactic measures of complexity*, PhD Thesis, Department of Philosophy, University of Manchester, 1999, 245pp.

[26]Baldwin, W. Clifton, Stephanie Hostetler, and Wilson N. Felder, (2013) "Mathematical Models of Emergence in Complex Systems-of-systems," *23rd Annual INCOSE International Symposium, 1–15. Philadelphia, PA: INCOSE.,* June 2013, pp.1-15.

[27]Snowden. David J. (2000) "New Wine in Old Wineskins: From Organic to Complex Knowledge Management Through the Use of Story," *Emergence*, Vol.2, No.4, Copyright © 2000, Lawrence Erlbaum Associates, Inc., Mahwah, NJ, pp.50-64.

[28]Jackson, M. C., (2003), *Systems Thinking: Creative Holism for Managers*, Copyright 2003, John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England, 352pp.

[29]Jackson, M. C., (1990), 'Beyond a System of Systems Methodologies', *Journal of the Operational Research Society,* vol. 41. no. 8**,** pp. 657 - 668, 1990.

[30]Jackson, M. C. and P. Keys, (1991) 'Towards a System of Systems Methodologies', pp. 139 - 158. In R. I,. Flood & M. C. Jackson (eds.), *Critical Systems Thinking: Directed Readings*, Brisbane: John Wiley & Sons, 1991.

[31]Johnson, Jeffrey (2010). "The future of the social sciences and humanities in the science of complex systems", *Innovation: The European Journal of Social Science Research*, 23(2), pp. 115–134.

[32]San Miguel, Maxi, Jeffrey H. Johnson, Janos Kertesz, Kimmo Kaski, Albert Díaz-Guilera, Robert S. MacKay, Vittorio Loreto, Peter Erdi, and Dirk Helbing. (2012) "Challenges in complex systems science." (2012).