Bridging Systems Engineering and Complex Systems Sciences

Sarah A. Sheard
Third Millennium Systems LLC
sheard@3MilSys.com (703) 757-7644

Abstract. This paper clarifies the state of systems engineering today and explains the necessity of more tightly connecting systems engineering with the sciences of complex systems. Systems engineering is having a very difficult time building the complex interoperating software-intensive systems that are possible and needed today in the timeframes and budgets that customers can tolerate. Part of the problem is that there is no underlying theory beneath the principles and heuristics according to which systems engineers do their jobs. Complex systems are discovering principles that directly apply in many ways to the problems of systems engineering, yet for the most part the bridge between these systems sciences and systems engineering is inadequate. It is now time to forge stronger ties between these two fields.

Introduction: Today's Systems Engineering Problem

Systems engineering, meaning our ability to engineer increasingly more complex systems, is in crisis at the start of this third millennium. Studies have repeatedly shown the difficulty of specifying and building the large systems we want and need today. [Standish 1994] The availability of smaller, higher power, and cheaper computer hardware, as well as off-the-shelf computer operating systems and other complex software, has vastly increased the amount of system complexity that has been allocated to software in a system. [Sheard 1998]. At the same time, the vulnerability of our systems to security attacks has skyrocketed, with their global interconnectivity, with increasingly sophisticated global crackers and economic incentives for cracking, and with instant internet publication of cracking scripts. It takes longer to understand these increasingly complex new systems in order to add to them, yet new technology is also needed faster. All of these issues result in few, if any, systems that are built to cost and schedule, and remarkably many that fail to be delivered at all.

Government Disillusionment with Systems Engineering. Although the US DOD Acquisition agency promised more focus on systems engineering about five years ago, the same agency is now expressing frustration with the results of that focus [Schaeffer 2005]. One of these reasons for this disillusionment is that the DOD was expecting that requiring capability maturity in systems engineering (see below) would force contractors to do systems engineering better; however, contractors on the whole have not shown vastly better systems engineering subsequent to such a requirement.[1]

Capability Maturity Disappointment. In 2001 the Software Engineering Institute released the Capability Maturity Model®, Integrated (CMMI®)[2], which merged the Capability Maturity Model® for Software with systems engineering. It was expected that requirements for contractors to demonstrate organizational maturity in software development processes in order to bid on large projects would now apply to require similar maturity in systems engineering. However, to date this has not had much impact on systems engineering practice. First, there are very few lead appraisers (who also serve as process im-

[1] Of course, it takes time for companies to believe the government is serious, to fund an improvement effort, and then to actually make serious changes. Within the five years or so that the government waited, not much was accomplished in many companies other than banner-waving. See "Life Cycle of a Silver Bullet", [Sheard 2003].
[2] Capability Maturity Model and CMMI are registered trademarks of Carnegie Mellon University.

provement consultants) who have experience in systems engineering to help these contractors...most have only software background. Second, systems engineering's value to a program does not have unarguable quantitative proof, and it is easier to cut budgets for something that looks like documentation and overhead than something that looks necessary, like a wing, or operational software. In addition, significant controversy about what systems engineering consists of has contributed to confusion about exactly what were the goals that the acquirers wished to achieve, and whether their means were appropriate to achieve those.

State of Systems Engineering Today

A complex subject, systems engineering must be explained in terms of multiple views. Sections below address the following three views: the people view, the documentation view, and the relationships view. Figure 1 shows many of these aspects. (Note that acronyms are spelled out at the end of this paper.)

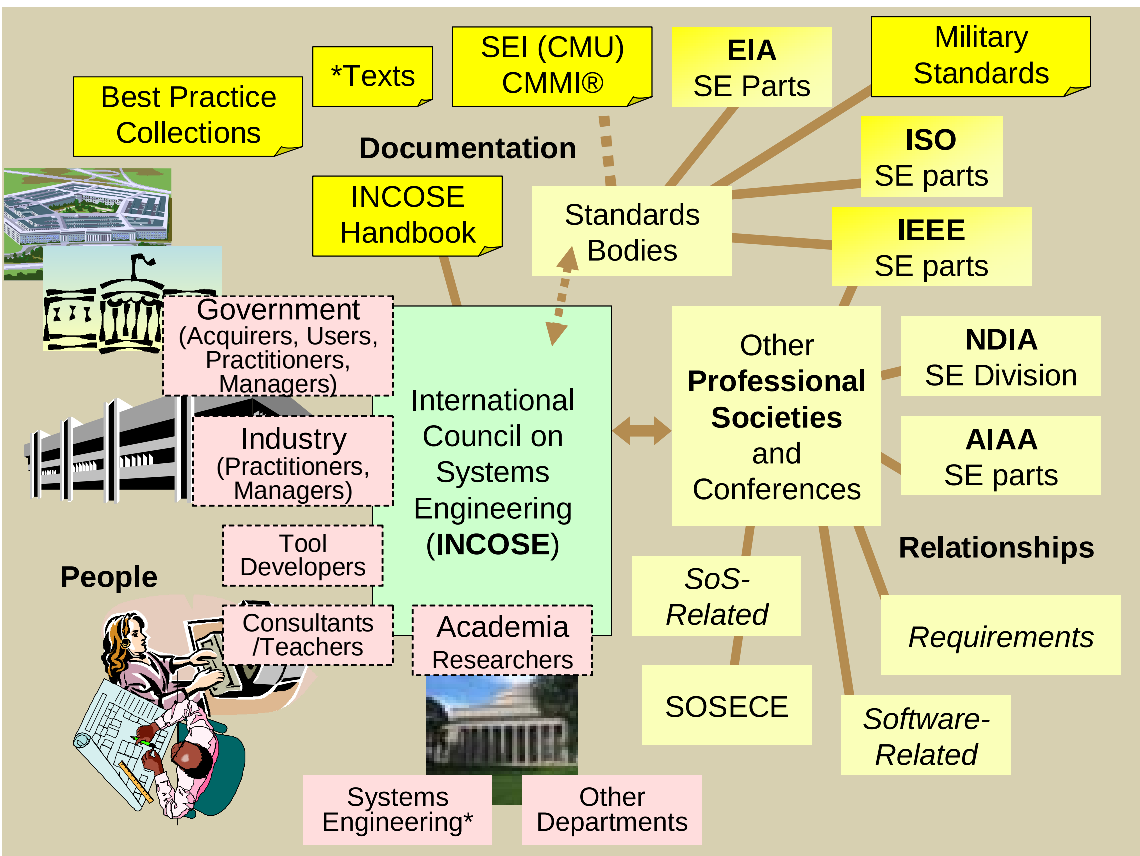

Figure 1. Systems Engineering People, Documentation, Relationships
(*Asterisk shows SE professors tend to produce SE Texts)

People: Six Types of Players on Systems Engineering Field Today

Within Systems Engineering one could enumerate many different sets of people, but for the purpose of analyzing how complex systems (CS) can relate to systems engineering (SE) it is useful to enumerate the following six sets. These are Practitioners (mostly in industry, but some in government); Managers of those practitioners; Acquirers and Users (Acquirers are mostly in government with some in industry, and of course many end-users are private citizens); Researchers (also called Academics); Consultants and

Teachers; and Tool developers. Table 1 shows their viewpoints, what they want out of systems engineering, what their goals tend to be, and what kind of points of view they tend to have of complex systems.

Table 1. Systems Engineering Players

SE Player	Viewpoint	Desire for SE	Goals	Point of view re CS
Practitioners	Implementer	Easy to do, effective, makes sense	Career and personal goals	Interesting but hard; is it relevant?
Managers	Manager, Controller	Cheap, efficient, effective, what's in it for me, now	Maintain control, don't overrun cost or schedule	May be useful in long run, but I have a schedule to meet now.
Acquirers and Users	Purchaser, User of end product	Effective, answers my problem, not hypothetical	Current and long term system goals, interoperability	Some hope CS is the answer; How do I make it happen for my system?
Researchers, Academics	Thinker, Boundary pusher	Principled, follows real truth, answers real problem not just a patch	Underly engineering with basic scientific principles	SE and CS not always working together today
Consultants, Teachers	Simplifier, Reducer to practice	Can bring new things to the masses	Continued utility to engineering community	Haven't yet figured out CS and how to apply to SE
Tool Developers	Implementer	Sellable to many people, maintainable	Continued use and upgrades	Apparently have not noticed CS yet

Documentation: Systems Engineering Standards, Guidance, and Best Practices

From the point of view of how the practice of systems engineering is currently documented, systems engineering standards bodies exist in the commercial, military, domestic and international arenas. Systems engineering standards relevant to US aerospace contractors have ballooned from one proposed US military standard in 1994 (Mil-Std-499B, never issued formally due to the Perry initiative), to a handful of specific systems engineering standards and capability models today (IEEE 1220, EIA 632, EIA 731, ISO/IEC 15288) plus a few standards and capability models that were initially software only but now are explicitly intended for both software and systems engineering (CMMI®, PSM, and ISO 15939, for example). [Sheard 2001] There are also numerous guidebooks in all areas, intended to convey good or best practices, ranging from a plethora of systems engineering textbooks to Military Handbooks such as the DOD 5000 series and the INCOSE Handbook, measurement guidebooks, guidebooks that accompany various standards such as the ISO/IEC series, and many others. Someone intending to stay on top of this rapidly expanding field could spend full time reading and still fall far behind.

Relationships I: Systems Engineering Relative to Other Professional Societies and Conferences

Systems engineering is not a new field, having made major strides in the 1950s and 1960s, but the only professional society dedicated exclusively to systems engineering, the International Council on Systems Engineering, is only 15 years old. Other societies such as the Institute of Electrical and Electronics Engineers (IEEE), and the American Institute of Aeronautics and Astronautics (AIAA), which have had systems engineering focus areas for some time, have found a vastly increased interest in those areas within the same time frame.

To address the issue of increasing complexity and increasingly interconnected systems of software-intensive systems, conferences and societies and think tanks have begun to focus on the "systems of systems" issue in the last few years. Two notable examples are the Systems of Systems Engineering Center of Excellence and its "First Annual" Systems of Systems Engineering Conference in 2005, and the IEEE Conference on Systems of Systems scheduled for April of 2006.

Relationships II: Systems Engineering Relative to Other Engineering and to Other Systems Disciplines

There have been many different depictions of where systems engineering fits relative to other disciplines. Perhaps the best for this article's purpose is reproduced as

Figure 2. [Hybertson 2006] Since systems are composed from pieces that require a variety of engineering disciplines systems engineering is composed of relevant parts of all the engineering disciplines. However, whereas the engineering disciplines are supported by the theory of, and provide data to, their corresponding sciences, and whereas systems science does have something of an interdisciplinary relationship with the single-discipline sciences, at the current time the expected complementary relationship between systems engineering and systems science on the left side of the diagram is somewhere between nonexistent and extremely tenuous.

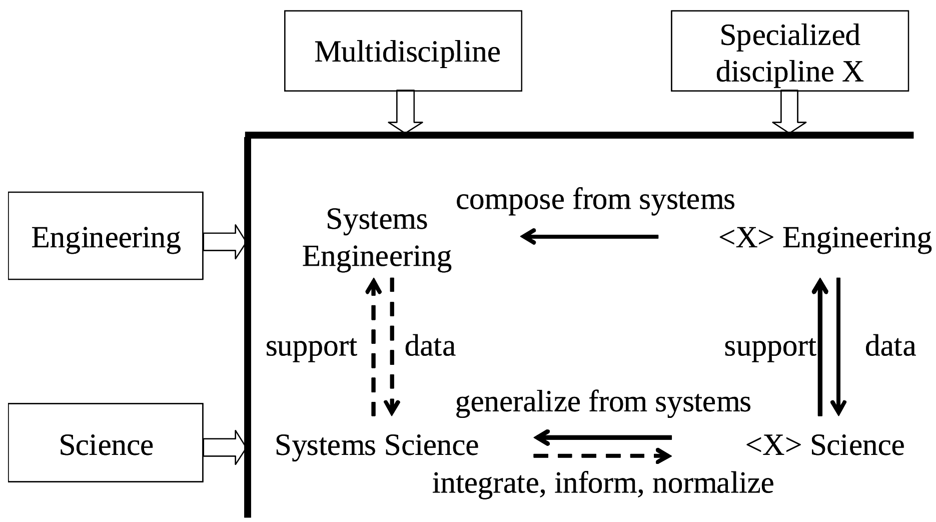

Figure 2. Science and Engineering, Multi- and Specialized Disciplines

Dilemma: Systems Engineering Research

Since INCOSE's inception in 1991 and since the inaugural issue of its journal *Systems Engineering* in 1994,[3] INCOSE's academics have contributed a small number of the papers published in the annual symposium proceedings. They have written a greater fraction of the papers published in the journal; however, there are only about 20 of these per year. This is a small number to drive or represent a discipline.

[3] Incidentally, this first series was cancelled after the first issue. The series was replaced in 1998 by the current Wiley publication of the same name.

What is missing, and whose absence has been lamented broadly within INCOSE, is a general systems engineering theory that provides a foundation for the principles and practices that systems engineers have collected experientially. Currently these are generally learned primarily by on-the-job training, experience, and systems engineering lore. There are a few collections of "best practices" or other guidance; two types of note are "Heuristics" and "Principles." Systems Architecting heuristics are found in a series of books by Eberhardt Rechtin and Mark Maier, notably *The Art of Systems Architecting* [Rechtin and Maier 1997]. Principles were collected in a draft report released by the Systems Engineering Principles working group of INCOSE in 1993 called, ironically, "An Identification of Pragmatic Principles – Final Report."

The author's experience for 25 years has been strictly experiential, not academic.[4] Personal interest has led her to explore recent advances in the fields of chaos, complexity, and Complex Systems in general. It became apparent that this is the field that not only can provide, but already has provided, guidance and potentially theoretical underpinning for some of the systems engineering heuristics which guide systems engineering stakeholders today. For example, look at the heuristics and principles in Table 2 below.

Table 2. Support Possible for Systems Engineering Heuristics and Principles

Heuristic or principle	**Source**	**Assistance possible from Complex Systems**	**Notes**
The principles of minimum communications and proper partitioning are key to system testability and fault isolation	[Rechtin 1991] Systems Architecting	What-if scenarios, fast modeling of a variety of ways to partition the system, ways to test and isolate faults	Much of this has to be done intuitively today because modeling is case-by-case, time consuming, and unwieldy
In any resource-limited situation, the true value of a given service or product is determined by what one is willing to give up to obtain it	[Rechtin and Maier 1997] The Art of Systems Architecting	Much simpler faster modeling, provided this can be communicated to analysts and projects.	Systems engineering often uses estimates of value of various alternatives in trade studies because it is too hard to model utility. Complex system models may make this much easier.
Don't depend on written specifications and statements of work. Face to face sessions with the different customer/ consumer groups are necessary	[Defoe et al. 1993] Pragmatic Principles	What questions are likely to cause the most issues, based on fitness landscape and the variety of alternatives? How can we tell when there is "enough" customer interaction?	SEs are uncomfortable with moving requirements and need mental models of fixed increments and agile overall progress.

It is easily conceivable that, with a little extra coordination, complex systems research can not only provide solid proof that the heuristics work and show why, it can also help us learn under what conditions, to what degree, when is enough, what is too much, and other questions for which our current guidance is only intuition and rules of thumb. The concept is very exciting.

However, few if any systems engineers are aware that Complex Systems theory is available to help with this[5]. Those who have become aware have been, on the whole, unsuccessful in communicating such

[4] Except for a graduate program in physical chemistry prior to systems engineering work.
[5] George McConnell of the United Kingdom is one exception, having shown that genetic algorithms lead to much faster scheduling analysis than deterministic algorithms, for example. See [McConnell 2003].

advantages to those actually out in the field, who are working "head down pencil up" on pressing problems without the luxury of down time to read and think about better ways of doing things, due to reduced budgets, complicated issues, and the need to do everything faster than yesterday. The fact that our current methods are insufficient to handle the complexity of systems to which we are applying them, with the resultant stream of unpredicted problems and continuous rework, only adds to practitioners' overtime, stress, and inability to reflect on better ways.

Steps Taken To Bridge the Gap

Some in both the systems engineering world, primarily through INCOSE, and the complex systems world, are trying to make this needed synergy more apparent. During 2006 a core of interested systems engineers, enterprise architects, and complex systems scientists are coordinating attendance at conferences that border the edges of these fields to spread the word. A small workshop is planned early in 2007 to coordinate efforts among some of the more senior players on each side, and a joint conference is in the planning stages for June of 2007 to bring together both sides in a 50/50 manner to explicitly focus on this communication issue and establish those bridges. It is not planned that this joint conference will be permanent; rather that it will initiate permanent connections among societies on both sides of the current chasm. The bridge can then disappear as an unnecessary and temporary artifact.

Conclusions

It is time. The need is there in the systems engineering world, in fact, it is baldly and sorely there. The ability is there, with many research results coming out every day and many graduate students seeking important thesis topics each year. It is time for systems engineering stakeholders to begin to learn enough about complex systems to see where current research has already provided answers and where future research can provide more and better answers. It is time for researchers to look at systems engineering to see what needs cry out for help. It is time for everyone to work together with all the stakeholders to help solve the world's problems.

References

Defoe, Joseph C. "An Identification of Pragmatic Principles – Final Report." Systems Engineering Principles Working Group, INCOSE, 1993.

Hybertson, Duane, "Ideas on Systems Science and Systems Engineering," presented to INCOSE Systems Science Enabler Group Workshop, Scottsdale AZ, January 30, 2006.

McConnell, George R. "Emergence: Applying the Principles – using Genetic Algorithms to derive Schedules." *Proceedings of INCOSE*. Las Vegas, Nevada, 2003.

Rechtin, Eberhardt. *Systems Architecting: Creating and Building Complex Systems*, New Jersey: Prentice-Hall, Inc., 1991.

Rechtin, Eberhardt and Mark W. Maier. *The Art of Systems Architecting*, New York: CRC Press, 1997.

Schaeffer, Mark. Discussion at the National Defense Industries Affiliation, Systems Engineering Division meeting; Arlington, Virginia; August 9, 2005.

Sheard, Sarah A. "Evolution of the Frameworks Quagmire," *Computer*, V.34, no. 7, pp. 96-98, 2001.

Sheard, Sarah A. Life Cycle of a Silver Bullet. July 2003.

Sheard, Sarah A., "Systems Engineering for Software and Hardware Systems: Point-Counterpoint." *Proceedings, Eighth Annual International Symposium of the International Council on Systems Engineering.* Vancouver, British Columbia: July 1998

Standish Group. The Chaos Report, 1994. http://standishgroup.com/sample_research/chaos_1994_1.php.

Acronyms and Abbreviations

AIAA	American Institute of Aeronautics and Astronautics	ISO	International Organization for Standardization
AT&L	Acquisition, Technology, and Logistics	LLC	Limited liability company
CMMI®	Capability Maturity Model, Integrated ®	MIL-STD	Military standard
CMU	Carnegie Mellon University	NDIA	National Defense Industrial Association
CS	Complex systems	PSM	Practical Systems and Software Measurement
DOD	Department of Defense	SE	Systems engineering
EIA	Electronics Industries Alliance	SEI	Software Engineering Institute
IEC	International Electrotechnical Commission	SOS	Systems of systems
IEEE	Institute of Electrical and Electronics Engineers	SOSECE	Systems of Systems Engineering Center of Excellence
INCOSE	International Council on Systems Engineering	US	United States

Biography

Sarah A. Sheard is the Principal at Third Millennium Systems LLC in Great Falls, Virginia. Ms. Sheard has been in systems engineering and process improvement for over 25 years. In 2002 she was awarded the coveted Founder's Award from the International Council on Systems Engineering (INCOSE) for her contributions to systems engineering and the Council, including over 30 publications, chairing the Measurement technical committee and the Communications committee, and serving as program chair and director of the Washington Metropolitan Area chapter. Ms. Sheard is a certified systems engineering professional (CSEP) with degrees from the University of Rochester and the California Institute of Technology.