6.2.1 Complexity Types: From Science to Systems Engineering

Article *in* INCOSE International Symposium · June 2011

DOI: 10.1002/j.2334-5837.2011.tb01235.x

Sarah Sheard
Carnegie Mellon University
76 PUBLICATIONS **911** CITATIONS

Complexity Types: From Science to Systems Engineering

Sarah A. Sheard and Ali Mostashari
Stevens Institute of Technology
1027 Challedon Rd Great Falls VA 22066
sarah.sheard@gmail.com; ali.mostashari@gmail.com

Abstract. How to measure complexity for systems engineering is currently receiving great attention, including such efforts as DARPA's META program. However, complexity is a highly varying concept with many different views. This paper discusses some of those varieties of meaning, types of complexities, and entities which may be measured for complexity (qualitatively or quantitatively). First discussed are attributes that make a system complex and difficulties in dealing with the system because of the complexity (i.e. complexity causes and effects). Next types of complexity are reviewed from previous work. Finally those types of complexity are applied to four types of systems engineering entity: a project building a technological system, that system itself, the environment into which the system will be inserted, and the cognitive load on humans involved in the system. Ongoing and future work is also described.

Introduction

Engineering marvels appear every year that are considered more complex than any previous type of technology. Every new program expands the boundary of what has been done before. Since the internet began defining every system we use and build, new systems put globally available information together in new ways to exploit and create new capabilities.

No longer can systems be created that one mind can understand. In fact, there is probably no one in the world who has the ability to make (completely from raw materials) something as ubiquitous as a computer mouse. (Ridley 2010) Information fusion creates superimposed maps upon request, for military use, advertising, even for consumers. (An Android smartphone with a ShopSavvy application can list nearby stores' prices for a product based on integrating a photograph of its universal product code with photo interpretation, price search and global positioning algorithms.)

It is clear that today's systems are far more complex than yesterday's, both the systems that consumers use every day, and the systems that companies are trying to develop and evolve. The question is, what does it mean to be complex? What makes a system more or less complex? How can the complexity of different systems be compared? How much does complexity "cost" and how can it be reduced? If it is reduced, does that translate to cost, schedule, or other benefits?

This paper separates attributes associated with complexity into causes and effects of complexity, and applies previously-published types of complexity to a number of systems engineering entities whose complexity can be typed and measured.

Ongoing research is investigating which measurements of complexity (of which type, for which entities) correlate to program success. Future work can focus on recommending strategies for dealing with specific types of complexity, including complexity reduction.

Background

What is complexity? Definitions of complexity include a variety of concepts and terms. Many dictionary definitions circle among complexity, complex, complicated, and intricate, in that the definitions for each word use the others. Such definitions are descriptive but not useful in this context.

Dictionary Definitions. The American Heritage 2000 definition of complexity combines structural and cognitive aspects: "*consisting of interconnected or interwoven parts; composite*" and "*having parts so interconnected as to make the whole perplexing*." The Oxford English Dictionary 1992 also includes concepts of diversity and hierarchy: "*...formed by combination of different elements*" and "*parts or elements not simply co-ordinated, but some of them involved in various degrees of subordination*." The Oxford English Dictionary also suggests inherent difficulty in understanding or coping with complexity: "*not easily analyzed or disentangled*."

Software and systems. Specific software definitions address interfaces, loops (conditional branches), nesting, and types of data structure. (Read 2008) Approaching systems engineering, the concepts that arise include emergence (Abbott 2006), abstractions, layers, and internal networks (Moses 2002). Maier (2007) addresses systems development efforts in particular and notes number of sponsors, whether users are the same as sponsors or different, low- or high-technology, feasibility of meeting expectations given cost and schedule, centralized or distributed control, clarity of objectives, required quality, total size (money), organizational experience, whether the system is stand-alone or embedded within an "assemblage of products and enterprises," and how much the operators must be able to adapt the system.

Warfield. Warfield (2007, 2001) takes a different approach: his method requires group construction of a "problematique" or a model of interacting problems, then counting and combining various measures such as how different the opinions are of the top 5 problems (Spreadthink Index), how many problems and how connected they are ("Miller Index" and "DeMorgan Index"). In particular, he decries any notion of complexity that is external to the observer.

Variety of definitions is a problem. There is general agreement that complexity is a problem (or even "the" problem), but there is virtually no agreement on the definition of complexity. Most agree that complexity is associated with difficulty of understanding, difficulty of teasing apart the problem (or system) without destroying the emergent functionality, and difficulty of prediction and control. Complexity is also associated with large size, lots of parts, things that are densely interconnected, things that have many different types of parts. And certainly complexity is blamed for many systems engineering and program management problems; indeed, complexity is even portrayed as the "enemy" of engineers. (Swartz 2006; Shin and Williams 2008)

Causes and Effects of Complexity

The problem with these varying definitions, aside from confusing the engineer, is that they are talking about different things. "Difficult to understand" is a different sort of attribute than "highly interconnected." Figure 1 sorts attributes of complexity into two types. The attributes on the left are attributes of the system that cause the system to be considered complex. The attributes on the left are the resulting effects, often on the human mind, that cause the system's complexity to be considered a problem. Interestingly, all types of complexity mentioned above can be placed on either the left or the right, aside from compound definitions of the form <something on the right> because of <something on the left>.

Changing complexity. Note that changing the complexity of the situation requires changing something on the left, as the things on the right are consequences of complexity, not its cause. If you change the uncertainty of a situation by gathering more data, you did not change the situation at all, you just changed what you know about it. Things like instability or costliness to build are not changed by using different modeling tools; something fundamental about the system has to change for the complexity to change.

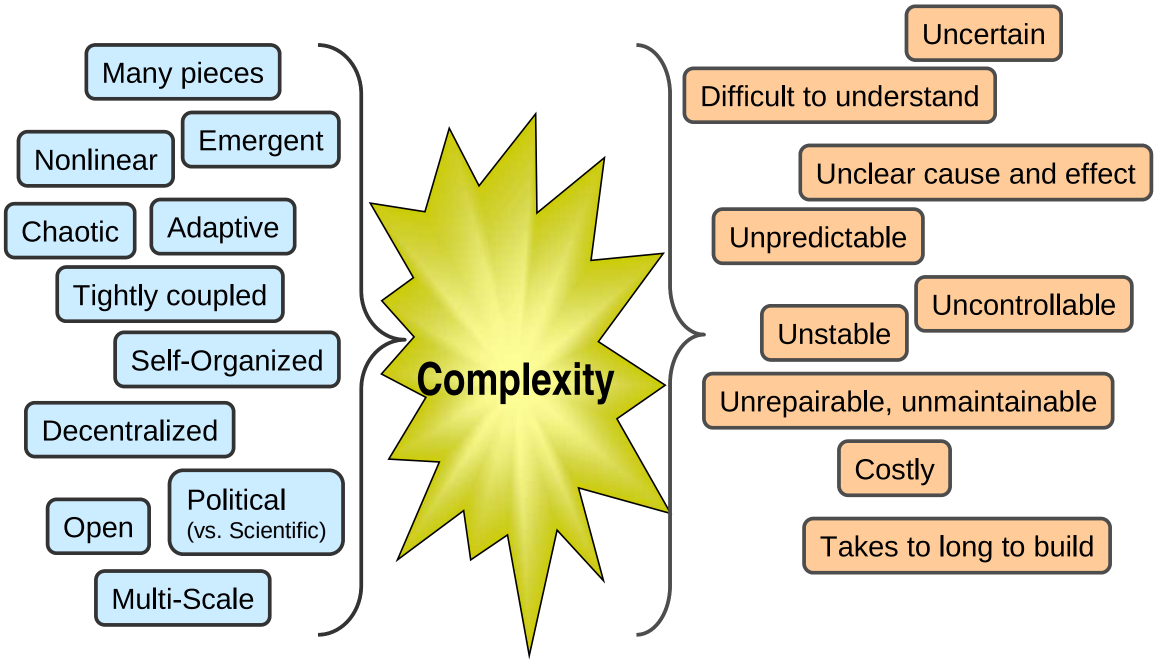


Figure 1. Causes and effects of complexity

In contrast the items on the left are fundamental characteristics of the situation. If you change a system from being decentralized to centralized, then the change is so great that you are really not talking about the same situation any more. It is possible, likely in fact, that therefore the situation's complexity will have changed. Similar things can be said for whether a system is adaptive or not, whether it is tightly coupled or not, or self-organized, etc.

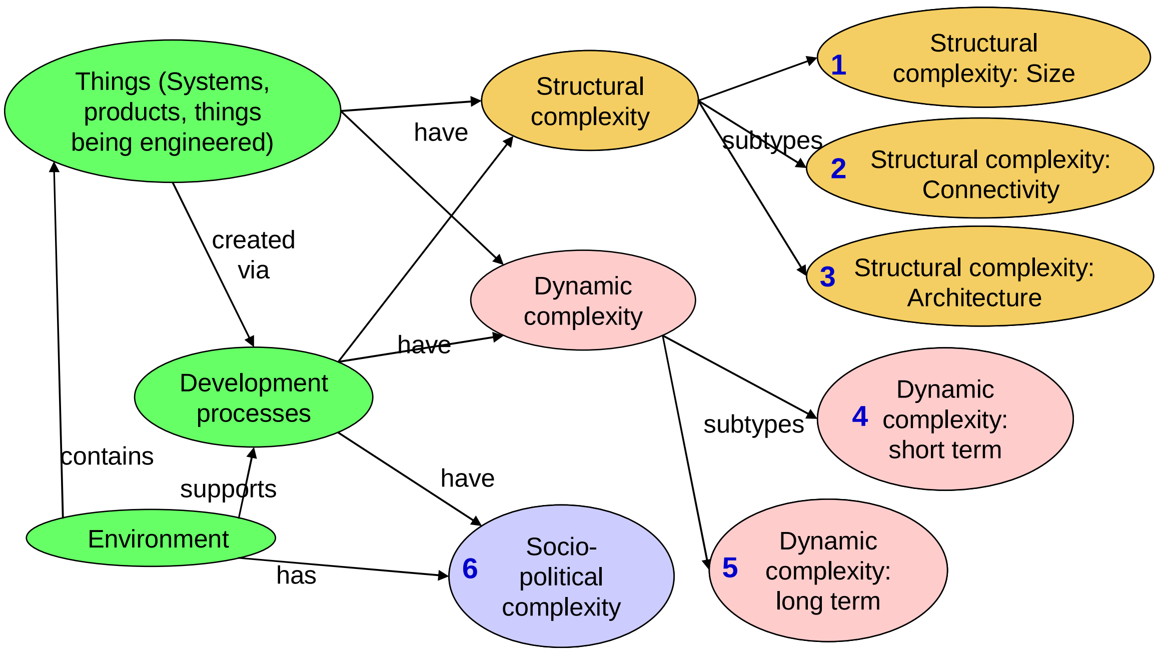

Things (Systems, products, things being engineered)
have
Structural complexity
1 Structural complexity: Size
subtypes
2 Structural complexity: Connectivity
3 Structural complexity: Architecture
created via
Dynamic complexity
have
Development processes
subtypes
4 Dynamic complexity: short term
contains
supports
have
Environment
has
6 Socio-political complexity
5 Dynamic complexity: long term

Figure 2. Types of Complexity

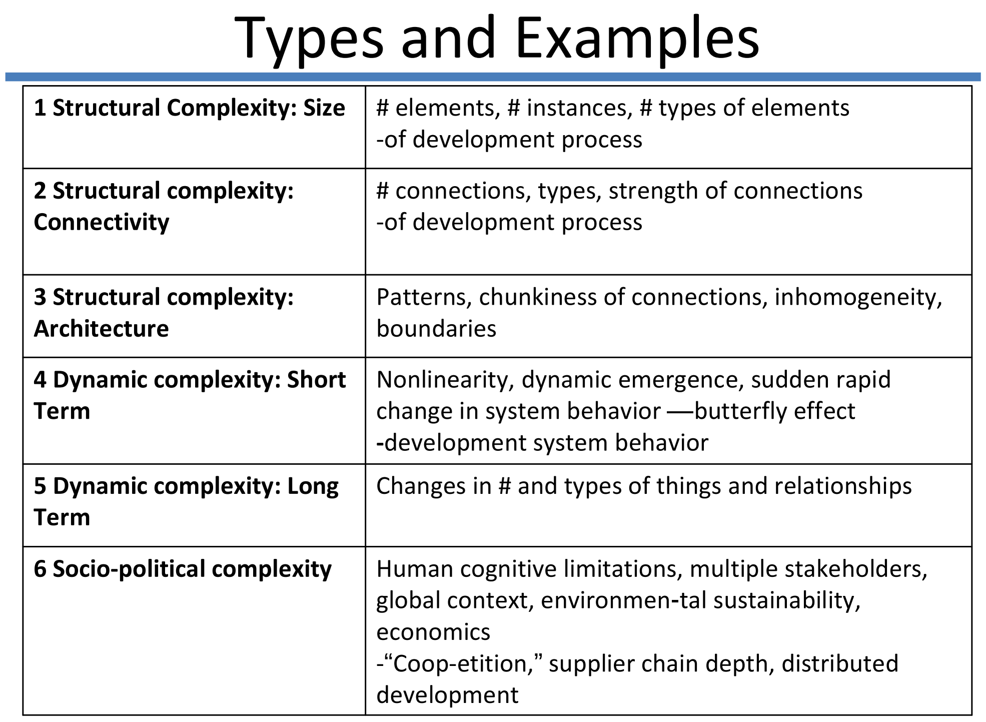
Types of Complexity

Sheard and Mostashari (2010) showed six types of complexity; see Figure 2 and, for examples,Table 1. These types were based on a literature review of complexity within the system sciences and in theoretical systems engineering.

Structural complexity comprises three types: Size, Connectivity, and Inhomogeneity (previously called Architecture). Size refers to number of pieces (a system with more pieces is likely to be more complex, although many authors...e.g. Renee Stevens, Linda Vandergriff...have noted that size alone should be considered complicated, not complex). Connectivity refers to the average or total number (or density) of interconnections (or interfaces) among the pieces. Most people who define complexity insist that connectivity must be at least fairly high for a system to be considered complex.

Dynamic complexity occurs over a range of time scales, from instantaneous to an evolutionary scale. Considering the two ends of this scale brings up the two types called Dynamic Short-Term (how quickly can things get unmanageable, on an operational time scale) and Dynamic Long-Term (what will a system evolve into, on a time scale where an entire ecosystem changes). System behavior involves short-term dynamic complexity. Both short- and long-term dynamic behavior get more complex the more nonlinear the behavior is. Nonlinearity can lead to chaos, which has an essential element of unpredictability due to measurement errors.

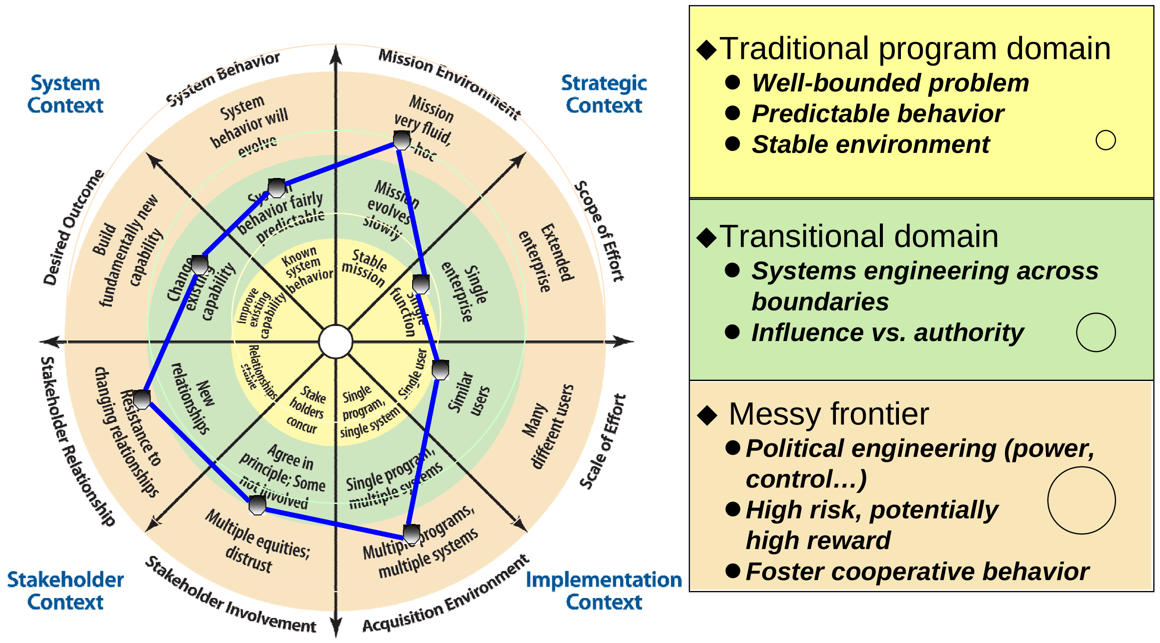
The sixth type is a grouping of factors called socio-political complexity. This has been best abstracted for systems engineering by the Enterprise Systems Engineering Profiler (Stevens 2010), shown in Figure 3.



Types and Examples

1 Structural Complexity: Size	# elements, # instances, # types of elements -of development process
2 Structural complexity: Connectivity	# connections, types, strength of connections -of development process
3 Structural complexity: Architecture	Patterns, chunkiness of connections, inhomogeneity, boundaries
4 Dynamic complexity: Short Term	Nonlinearity, dynamic emergence, sudden rapid change in system behavior —butterfly effect -development system behavior
5 Dynamic complexity: Long Term	Changes in # and types of things and relationships
6 Socio-political complexity	Human cognitive limitations, multiple stakeholders, global context, environmen-tal sustainability, economics -“Coop-etition,” supplier chain depth, distributed development

Table 1. Types of Complexity: Examples



◆Traditional program domain
- *Well-bounded problem*
- *Predictable behavior*
- *Stable environment*

◆Transitional domain
- *Systems engineering across boundaries*
- *Influence vs. authority*

◆ Messy frontier
- *Political engineering (power, control…)*
- *High risk, potentially high reward*
- *Foster cooperative behavior*

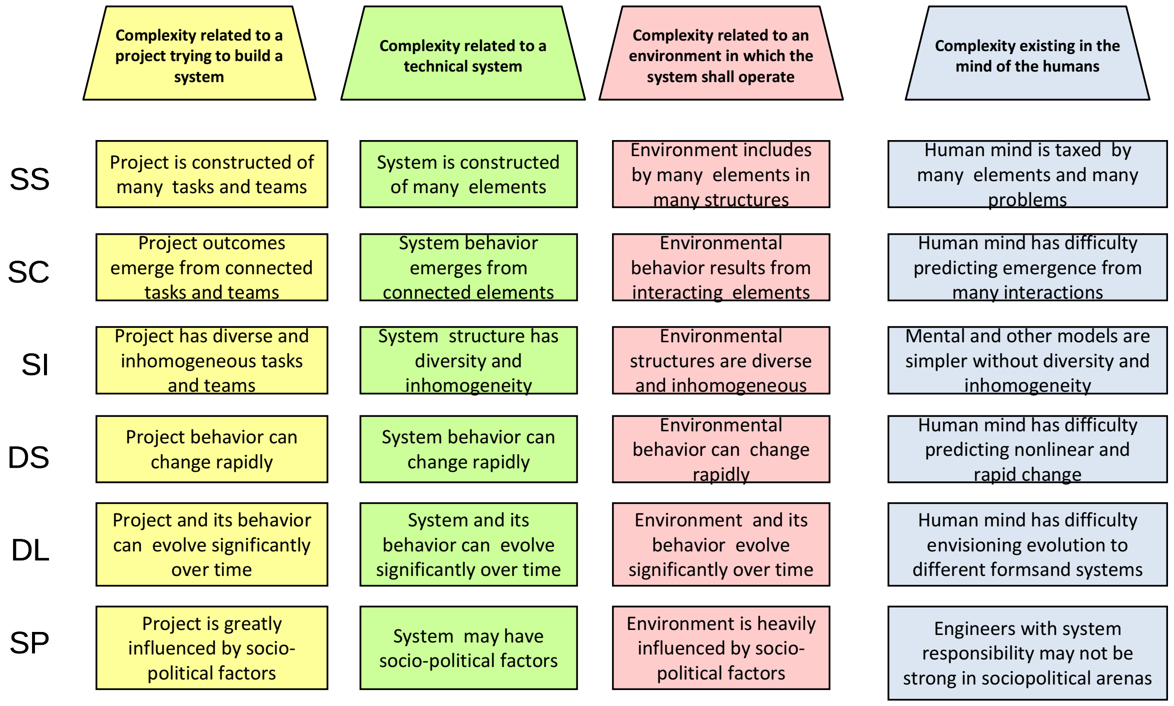
Figure 3. MITRE’s Enterprise Systems Engineering Profiler

TYPES APPLIED TO SYSTEMS ENGINEERING

We have recently been applying these six types to systems engineering. The first way to get more specific about engineering application is to address exactly what entity is to be evaluated as more or less complex. Three kinds of entities are the technological System being built, the Project doing the building, and the Environment into which the system will be inserted. The fourth, implied above, is the cognitive load on the human involved with system development and operational tasks, here called “Cognitive complexity.” Figure 4 shows the four kinds of entities across the top.

Next, for each of these entities, what attributes can be measured to represent the six types of complexity? A project decomposes into a number of tasks (SS), for example, which are connected by dependencies (SC), and some tasks are bigger than others or grouped separately (SI). The tasks change rapidly (DS) and over time the project can evolve into something completely different (DL). A project has many socio-political spects as well. (SP). In similar ways the system, environment, and cognitive complexity can be viewed in terms of the six types of complexity.

Looking at the second column (System), we see the system has been implicitly decomposed into elements and the “size” has been counted in terms of number of elements. There are many other ways to look at a system than just an element view.



	Complexity related to a project trying to build a system	Complexity related to a technical system	Complexity related to an environment in which the system shall operate	Complexity existing in the mind of the humans
SS	Project is constructed of many tasks and teams	System is constructed of many elements	Environment includes by many elements in many structures	Human mind is taxed by many elements and many problems
SC	Project outcomes emerge from connected tasks and teams	System behavior emerges from connected elements	Environmental behavior results from interacting elements	Human mind has difficulty predicting emergence from many interactions
SI	Project has diverse and inhomogeneous tasks and teams	System structure has diversity and inhomogeneity	Environmental structures are diverse and inhomogeneous	Mental and other models are simpler without diversity and inhomogeneity
DS	Project behavior can change rapidly	System behavior can change rapidly	Environmental behavior can change rapidly	Human mind has difficulty predicting nonlinear and rapid change
DL	Project and its behavior can evolve significantly over time	System and its behavior can evolve significantly over time	Environment and its behavior evolve significantly over time	Human mind has difficulty envisioning evolution to different formsand systems
SP	Project is greatly influenced by socio-political factors	System may have socio-political factors	Environment is heavily influenced by socio-political factors	Engineers with system responsibility may not be strong in sociopolitical arenas

Figure 4. Complexity Types Applied to Project, System, Environment, and Cognition
(SI=structural inhomogeneity = Structural, architecture)

For example, there is a requirements view. **Error! Reference source not found.** shows, next to the second column, a nearly duplicative set of boxes that look at requirements, for example. The system has many requirements; the requirements interact; the applicability and difficulty of meeting the requirments vary; the requirements specify dynamic behavior, the requirements will change and evolve, and many of the requirements relate to sociopolitical factors (implictly or explicitly). Similar additional columns could be made for test paths, for example, or maturing technologies.

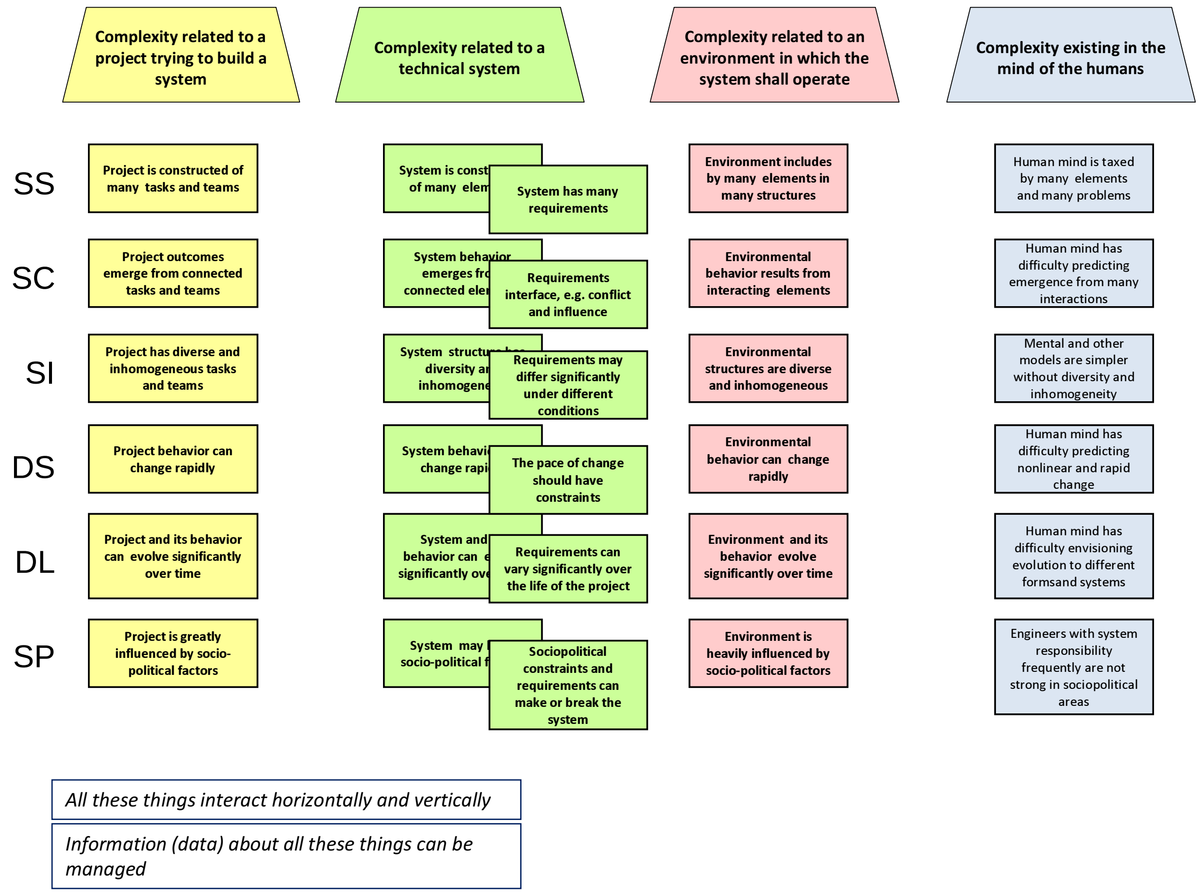

Complexity related to a project trying to build a system
Complexity related to a technical system
Complexity related to an environment in which the system shall operate
Complexity existing in the mind of the humans
SS
SC
SI
DS
DL
SP
Project is constructed of many tasks and teams
Project outcomes emerge from connected tasks and teams
Project has diverse and inhomogeneous tasks and teams
Project behavior can change rapidly
Project and its behavior can evolve significantly over time
Project is greatly influenced by socio-political factors
System is const of many elem
System has many requirements
System behavior emerges fro connected ele
Requirements interface, e.g. conflict and influence
System structure has diversity ar inhomogene
Requirements may differ significantly under different conditions
System behavi change rapi
The pace of change should have constraints
System and behavior can e significantly ove
Requirements can vary significantly over the life of the project
System may socio-political f
Sociopolitical constraints and requirements can make or break the system
Environment includes by many elements in many structures
Environmental behavior results from interacting elements
Environmental structures are diverse and inhomogeneous
Environmental behavior can change rapidly
Environment and its behavior evolve significantly over time
Environment is heavily influenced by socio-political factors
Human mind is taxed by many elements and many problems
Human mind has difficulty predicting emergence from many interactions
Mental and other models are simpler without diversity and inhomogeneity
Human mind has difficulty predicting nonlinear and rapid change
Human mind has difficulty envisioning evolution to different formsand systems
Engineers with system responsibility frequently are not strong in sociopolitical areas
All these things interact horizontally and vertically
Information (data) about all these things can be managed

Figure 5. Two examples for System (by elements and by requirements)

ONGOING AND FUTURE WORK

Another paper is in work that shows how a system that is being built is intended to cure some problem in the environment, also called The Way Things Are. Stakeholders wanting the cure fund a project to create a [solution] system. All of these (Environment, project, and system) also have aspects of cognitive complexity and information. When this standard systems engineering sequence is broken out into tasks and attributes, then all 33 definitions of complexity in (Young et al. 2010) can be mapped onto this chart. Such an accomplishment should help to clarify what is being talked about regarding complexity, and how the varying definitions relate.

As mentioned earlier, ongoing work includes correlating measurement of complexity to program success or failure. This will be published as the PhD dissertation of the first author.

Future work should also determine what about measurement of complexity is due to the representation of the situation (the model, of the environment, the system, the project, or all three, and modeling assumptions), and in contrast, what is due to something inherent about the situation. This could help give heuristics for when a model of a complex system is detailed enough.
It would also be useful to identify which of the varying columns mentioned in Figure 5 are useful for specific programs, and derive generalizations about what about the system or the program should be measured when the question of complexity measurement is asked.

If complexity is the enemy, then clearly a most useful research thrust would be to identify specific ways to reduce complexity, based on the most predictive measures, and identify whether changing the program so that the complexity measures reduce actually helps a program manager obtain better predictability and control of the program. Breaking up “what complexity is” into types and views helps understand how that reduction must occur.

Of course, complexity is not really the enemy, for complexity also allows capability. It is not possible to do a spell checker in hardware; still, we need spell-checkers. More complexity means both more capability and more trouble, both benefits and drawbacks, so a tradeoff has to be made between the known benefits and the risks. Future work should help with the cost and benefits tradeoffs of additional complexity.

Also, future work should look at the change in perception of complexity over time. Whereas the first spell checker was unimaginably complex when it was being developed, now it is considered not complex, and instead we are focusing on map fusion and many other capabilities that involve far more elements, far more computing time, probably far more lines of code if that is at all meaningful any more, and certainly far more developers to make or change the code. What about time inherently reduces the perception of complexity? Can this process be speeded up in some manner?

CONCLUSION

The problem with complexity is that, not being simple, it is not easy to pin down. Rather than artificially simplifying complexity by establishing an isolated definition of something manageable, we have attempted to understand what the various aspects of complexity are and how they relate. First we presented a distinction between what is inherent about a situation that causes complexity, and what are the often cognitive results of that complexity. Then we brought in the previously published six types and applied them to four types of systems engineering entities. Using figures 4 and 5 we can locate complexity to an entity and a type, and then establish means of measuring that type of complexity. It is conceivable that a rolled-up measure of complexity can be determined, but more likely that the various parts of complexity will need to be looked at individually.

References

Abbott, R. 2006. Emergence Explained: Getting Epiphenomena to do the Real Work: California State University, Los Angeles, 68.

Maier, M. W. 2009. *Dimensions of Complexity other than "Complexity"*. California State University, Los Angeles 2007 [cited Oct. 14, 2009]. Available from http://cs.calstatela.edu/wiki/index.php/Symposium_on_Complex_Systems_Engineering.

Moses, J. 2002. The Anatomy of Large-Scale Systems. In *ESD Internal Symposium*: Massachusetts Institute of Technology Engineering Systems Division.

Read, C. 2008. Complexity Characteristics and Measurement withing Engineering Systems, Loughborough University.

Ridley, M. 2010. *The rational optimist : how prosperity evolves*. 1st U.S. ed. New York: Harper.

Sheard, S. A., and A. Mostashari. 2010. A Complexity Typology for Systems Engineering. In *Twentieth Annual International Symposium of the International Council on Systems Engineering*. Chicago, Illinois: INCOSE.

Shin, Y., and L. Williams. 2008. Is Complexity Really the Enemy of Software Security? In *Fourth Workshop on Quality of Protectyon (QoP'08)*, edited by A. Ozment.

Stevens, R. 2010. *Engineering Mega-Systems: The Challenge of Systems Engineering in the Information Age* Edited by CRC: CRC Press.

Swartz, F. 2010. *Java: Complexity is your enemy* 2006 [cited **29 Oct 2010** 2010]. Available from **http://leepoint.net/notes-java/principles_and_practices/complexity/complexity_enemy.html**

Warfield, J. N. 2001. Measuring Complexity. In *The John N. Warfield Digital Collection*. Fairfax, Virginia: George Mason University.

———. 2007. The Aristotle Index: Measuring Complexity in the Twenty-First Century: George Mason University.

Young, L. Z., J. V. Farr, and R. Valerdi. 2010. The Role of Complexities in Systems Engineering Cost Estimating Processes. In *8th Conference on Systems Engineering Research*. Hoboken NJ.

Biography

Sarah Sheard has 30 years of experience in systems engineering, process improvement, and curriculum development and implementation. An INCOSE Fellow and winner of the INCOSE Founder's award, Ms. Sheard is working toward a doctorate at Stevens Institute of Technology with a focus on measurement of complexity for systems engineering. At Third Millennium Systems, Ms. Sheard teaches courses ranging from basic systems engineering to advances for the future, including application of the sciences of complex systems to systems engineering. She consults with private companies and government agencies regarding systems engineering processes, systems engineering curriculum, and interfaces to software and project management.

Dr. Ali Mostashari is the Director of the Infrastructure Systems Program and the Director of

Center for Complex Adaptive Sociotechnological Systems (COMPASS) at Stevens Institute of Technology, where he also serves as an Associate Professor at the School of Systems and Enterprises. Dr. Mostashari serves as the Co-Chair of the Global Conference on Systems and Enterprises and is a member of INCOSE and the AAAI. Dr. Mostashari holds a Ph.D. in Engineering Systems/Technology, Management and Policy from MIT, a Master of Science in Civil Engineering/Transportation from MIT, a Master of Science in Technology and Policy from MIT, a Master of Science in Chemical Engineering/Biotechnology with a minor in Business Administration from the University of Nebraska, a Graduate Certificate in Negotiation and Conflict Resolution from Harvard University and a Bachelor of Science in Chemical Engineering/Energy Systems from Sharif University of Technology.