On the Importance and Value of Case Studies

Introduction to the Special Session on Case Studies in System of Systems, Enterprises, and Complex Systems Engineering

Brian E. White
CAU←SES
Sudbury, MA, USA
bewhite71@gmail.com

S. Jimmy Gandhi
University of California
Northridge, CA, USA
sjg07646@yahoo.com

Alex Gorod
University of Adelaide
Adelaide, South Australia
alex.gorod@adelaide.edu.au

Vernon Ireland
University of Adelaide
Adelaide, South Australia
vernon.ireland@adelaide.edu.au

Brian Sauser
University of North Texas
Denton, TX, USA
Brian.Sauser@unt.edu

***Abstract*—Case studies provide a means for highlighting and extracting practical principles and methods for shaping and accelerating progress in solving wicked real world problems. They inform burgeoning theories such as those associated with complex systems engineering where people are considered part of the system to be conceived, developed, fielded, and operated; or extant systems targeted for improvement upgrades. This paper advocates for the value of case studies and summarizes some important examples. Motivations and ideas for increasing the use of case studies in the world's most difficult present and future systems engineering environments are suggested.**

Keywords—case studies; complex systems; complex systems engineering; enterprises; enterprise systems engineering; systems of systems; system of systems engineering; wicked problems

I. Introduction

Engineering is about solving human-made problems. In this endeavor scientific understanding, the laws of physics, chemistry, biology, etc., and the formulations of mathematics are applied to effect and appropriately as possible. Systems Engineering (SE) is multi-disciplinary, even trans-disciplinary [1], and like many other disciplines continually evolves to become even more relevant and effective as a practical approach to achieving worthwhile objectives.

Historically in education the underlying theory and the reasoning behind it is taught first. Then students are asked to apply the theory in practice, initially in solving homework problems and conducting laboratory exercises, for example, and later in their real world jobs. Another approach, as famously espoused by business schools [2][3], for example, is to teach through case studies, the philosophy being that established theories do not work so well in arenas where animate objects like humans exercise free will in unpredictable ways. This is the realm of trying to address "wicked" problems [4] at the "messy" frontier [5], in mega-system [6] SE environments.

This paper is another about emphasizing the importance of case studies in SE [7], where this technique has not yet been widely applied, at least not in traditional or conventional SE. The premise is that case studies are even more important in complex, System of Systems (SoS), and enterprise SE. Before proceeding consider how the terms are defined.

II. Some Definitions

A. Assumptions

Assume that certain terms, like system, engineering, and enterprise do not need to be defined. An extant collection of definitions of these and many other related terms exists [8] for those interested in delving further. However, the goal here is to simply explain how other terms like "complex", which are not so widely agreed to in their nuanced vernacular, are used in this paper, and then quickly return to the topic of case studies.

B. Complexity

Complex is more than complicated, a notion that is on the lowest rung of a discrete or even continuous scale of increasing complexity. Many people including engineers and systems engineers use complex and complicated interchangeably, or worse, use complex when they mean only complicated. Here complex refers to a range of difficulty that is more, and often much more, than merely complicated.

A complex system is often distinguished from simple, uncomplicated, or merely technologically complicated systems by intentionally including people, e.g., key stakeholders, as part of the system. It is much more difficult to successfully conceive, develop, improve, field, and operate such systems. This is fundamentally because this type of complex system evolves on its own in a self-organized fashion in response to its (usually many) internal and environmental interactions. Systems engineers whom work with complex systems are called complex systems engineers; they are not really in control of very much because of the nature of complex systems. They often can only influence the system and decide to intervene again at some point if the system moves in undesirable directions. So what about SoS and enterprises?

C. Relationships

Venn diagrams can be used to illustrate how one may think about the various kinds of systems that are to be system engineered. Fig. 1 depicts one way of viewing the relative SE difficulty. Here Enterprise SE (ESE) is considered more difficult because enterprises are constrained to have a degree of homeostasis that may be hard to maintain when performing ESE.

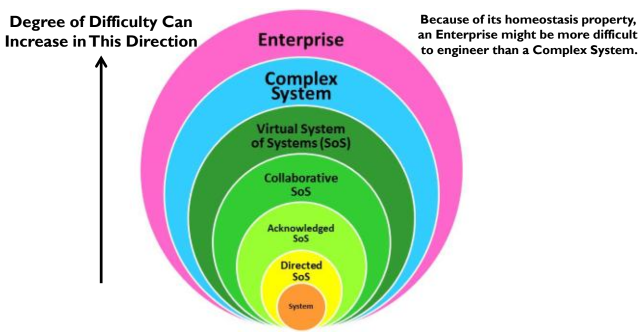

Fig. 1 A Perceived Venn Diagram Relationship Among Systems

Homeostasis: A property that exhibits relatively stable equilibria or behaviors among its interdependent component systems and its environment. Examples:

1) An agreed-to mission of an enterprise, like Ford Motor Company's Job One, the first new car off the assembly line.

2) Human body temperature.

Virtual Virtual SoSs lack a central management authority and a centrally agreed-upon purpose for the SoS. Large-scale behavior emerges—and may be desirable—but this type SoS must rely upon relatively invisible mechanisms to maintain it.
Collaborative In collaborative SoSs, the component systems interact more or less voluntarily to fulfill agreed-upon central purposes. The Internet is a collaborative system. The Internet Engineering Task Force works on standards but has no power to enforce them. The central players collectively decide how to provide or deny service, thereby providing some means of enforcing and maintaining standards.
Acknowledged Acknowledged SoSs have recognized objectives, a designated manager, and resources for the SoS; however, the constituent systems retain their independent ownership, objectives, funding, as well as development and sustainment approaches. Changes in the systems are based on collaboration between the SoS and each member system.
Directed Directed SoSs are those in which the integrated SoS is built and managed to fulfill specific purposes. It is centrally managed during long-term operation to continue to fulfill those purposes as well as any new ones the system owners might wish to address. The component systems maintain an ability to operate independently, but their normal operational mode is subordinated to the central managed purpose.

Fig. 2 depicts another way of viewing the relative SE difficulty. Here Complex SE (CSE) is considered the most difficult if complex systems are viewed as the most general systems that provoke the greatest challenges to systems engineers.

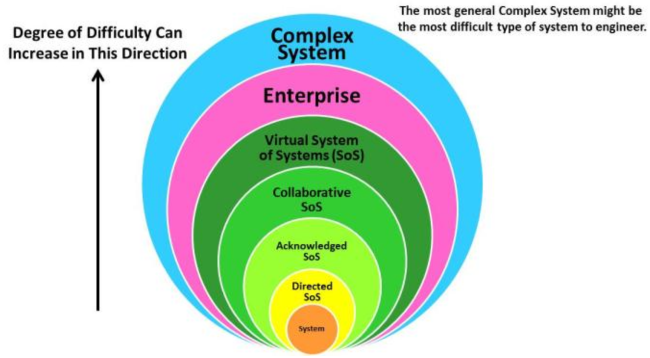

Fig. 2 Another Perceived Venn Diagram Relationship Among Systems

III. WHY DO CASE STUDIES?

Systems engineers can have the best ideas for promising ways forward in engineering difficult systems. They view and present their suggestions as abstract, general, theoretical, logical, reasonable, practical, rational, and emotionally compelling. Frequently they do not even think of case studies as being needed. Instead they simply draw upon their experience in performing SE. This learning is usually not acquired in an academic environment[1] but is accumulated on the job, hopefully while being mentored by a good systems engineer. And good program managers listen to their systems engineers and often take their advice – but not always!

A. Prove It!

Program managers, project leaders, bosses, and other key stakeholders, especially those that have already achieved a measure of career success in their current organizational positions by operating differently, want proof these SE ideas will work, especially those associated with CSE that may seem more radical. Before committing to, supporting, and promoting any suggested course of action, they ask the systems engineer "Prove to me this will work!"

In his/her defense the systems engineer might say "This is not a mathematical theorem. People are not predictable and their behaviors are difficult, if not impossible to model accurately. Therefore, you need to have some faith and take an informed risk in pursuing promising opportunities in trying these complex system interventions." To help alleviate these stakeholders' reluctance or fears, the systems engineer can promote modeling and simulation as a (perhaps lower consequence) means of exploring the ideas to discover what might happen in practice.

B. Modeling and Simulation

Agent based modeling (ABM) [10] is an established method that may show skeptics or naysayers whether the systems engineer's bright ideas are truly promising. This is done by giving several autonomous actors or agents that represent system constituents, including stakeholders, a few

[1] However, more and more, SE is being taught at the graduate level in college, and sometimes even at the undergraduate level.

simple rules of interaction. Then the model is exercised, with the agents collectively interacting with each other, usually over a large number of iterations. One advantage of this is that unexpected events are likely to occur that could not have been predicted even when each step of the simulation is completely deterministic, e.g., the appearance of a "glider" in game of life experiments involving simple two-dimensional automata [11].

Case studies can be used as initial conditions for modeling and simulation. What about connecting case studies to ABM? The results of case studies can be used to guide ABM through tailoring the rules of engagement among the actors. Then the simulation can be rerun to see how well the model mimics the behaviors of the subject system of the case study. Of course, this means time and effort must be invested in defining, selecting, and conducting suitable case studies unless one is lucky enough to already have a good and relevant case study on hand.

C. *Advantages*

What better way is there than to thoughtfully and mindfully detail real life in case studies to show what works and does not work? If this is done properly, the case study results will inform theory and cause it to be shaped or expanded to better validate good practice. In other words, theory follows practice in case studies. Paradoxically, many people tend to believe that applications must follow theory!

Various advantages of case studies come to mind. One can

- Build upon or enhance a body of knowledge
- Suggest things to try
- Highlight errors to avoid
- Learn[2] from others' mistakes
- Compare specific aspects across case studies
- Show the importance of including people in the system that needs to be engineered
- Dig into specific situations and extract ideas from participants that can be generalized into principles for others to apply.

D. *Story Telling*

Some extant case studies tout success focusing on the wonderful technology that is involved in the system. Unfortunately what made the system successful in human terms is largely ignored. The latter aspects are what need to be especially emphasized for the benefit of other practitioners.

In story telling one must first gain the attention of the listeners, usually by scaring them about bad things that might happen [13]. Once they are paying close attention, to the extent the story is compelling, the listeners will naturally translate the story into something more meaningful to them personally while they are listening. That's how the mind works.

IV. HOW TO DO CASE STUDIES

The authors' previous paper on this topic [7] emphasized case studies in engineering education and management and cited a classic work on preparing case studies [14]. A fundamental paper on case studies in more conventional systems engineering and systems management is also recalled [15]. Here the focus is on case studies in complex, SoS, and enterprise engineering. In addition, the authors are working on assembling a couple of dozen such case studies for a new book [16].

A. *Planning*

The first step in preparing a case study is the anticipation of doing one. So when starting an important project involving a complex system prepare an initial plan for gathering and recording data that can be used later. Recognize that some of the project participants will likely come and go so it is critical to preserve the project record and not rely entirely on people's memory. This record must be kept up to date continually, preserving the issues, decisions, actions, and results of each step or stage of the development or upgrade process. Even the plan should be updated and modified as required.

B. *Documentation*

Routine aspects associated with dates of occurrence can be documented in any convenient format. The more interesting developments should be described in concise essay form. Any insights that can be immediately gleaned from unexpected events, in particular, will provide valuable case study fodder later on. Every so often this chronological record should be reviewed retrospectively to gain addition insights for going forward. The ultimate objective is to end up with sufficient raw material to shape into a case study that would be of considerable interest to other practitioners.

C. *Guidelines*

Case studies must be written honestly, highlighting the realities of the challenges and the struggles in seeking solutions. Sometimes case study descriptions must be watered down to gain publishing approval of sponsors or customers. Because most organizations are risk averse and do not want to jeopardize their proprietary "crown jewels"[3], be criticized by outsiders, or lose their funding to others, obtaining public release properly can be a challenge.

Objectivity is important. But one must realize that emotions are also relevant. Humans cannot claim to be entirely rational, especially in decision making [18]. Descartes was wrong! [19]

To the extent an outline template is followed in preparing a collection of case studies, the comparing and contrasting of

[2] One only learns from mistakes, one's own or mistakes of others [12]. If one knows what to do already, there is no learning.

[3] Nevertheless, it is quite possible to share much information to one's own advantage and still protect the core business [17].

problems, and the methods and techniques of seeking solutions, across different case studies is possible. A suggested example outline that the authors are advocating in a forthcoming book [16] is provided in the Appendix. This version is quite similar but updated from the outline presented in [7] where the outline was explained in much greater detail than in the present paper.

V. EXAMPLE CASE STUDIES

The authors are anticipating at least twenty case study chapters in their aforementioned book [16]. They will be assigned to several potential domains: commerce, culture, environment, finance, health care, homeland security, military, and transportation. The book will be organized into several parts each covering one or more closely related domains. This 1st Paper of a Special SysCon 2013 Session on Case Studies in SoS, Enterprises, and Complex Systems Engineering leads by way of introduction. This session's subsequent papers (planned as of this writing) are now summarized.

2nd Paper. Case Study: SoS Engineering Applied to a Large IT Enterprise – Jeffrey Higginson, Timothy Rudolph, and Jon Salwen.[4]

This paper discusses what is claimed to be the minimum set of Systems of Systems Engineering (SoSE) processes needed to effectively manage an enterprise Information Technology (IT) SoS and provides an example of how these processes and methods may be applied.

The authors' model for SoSE implementation involves a continuous execution chain from a high-level strategy aimed at SoSE and associated capability delivery to system-level implementation and core SE processes.

The focus is directed at the interface between a set of core SoSE processes (including governance, architecture and analysis, test and evaluation, and management) and the existing SE processes already widely used at the system level. The SoSE processes are mapped onto the SE processes to complement and expand upon the traditional systems engineer's role toward changing his/her perspectives to embrace capabilities over requirements, collaboration over control, architecture over design, aggregation over decomposition, and acceptance of complexity [over complication].

Experience in applying these SoSE and SE processes in a supporting way have led to the successful implementation of an engineering framework and a sustainable IT SoS. More specifically, findings show that: 1) IT infrastructure can be operated, managed, and directed effectively as an Acknowledged SoS [refer to Fig. 1 above]; 2) centralized management and governance functions, shared across the stakeholder community, are effective mechanisms to establish requirements and deliver capability; 3) tacit acknowledgement and strong stakeholder support of the SoS is necessary to establish and execute strong SoSE principles; and 4) significant efficiencies in cost and manpower can be achieved when managing a large IT infrastructure as a SoS rather than individual, interconnected systems.

A federal IT enterprise was examined with specific examples provided to illustrate the challenges presented by a complex SoS, the methods applied to address those challenges, and the efficiencies gained via the judicious application of SoSE principles.

3rd Paper. Strategic Architecture Approach To Transforming Defense Acquisition: A Case Study in Moving From Formal Bureaucracy to Lateral Hierarchy – J. Dickmann.[5]

The U.S. Defense acquisition system is notoriously resistant to fundamental reform and improvement. Many studies and recommendations for acquisition reform focus on process factors and incentive structures. Few, if any, focus on how technical architecture can enable change and how enterprise architecture can help capitalize on technical advancements.

In the mid-1990s, the U.S. submarine force was losing acoustic advantage against Russian submarines. Post-Cold-War fiscal constraints and a cumbersome acquisition process made timely and economical sonar system upgrades impossible.

Using a layered architectural approach to separate hardware and software, innovative sonar signal processing algorithms were hosted on commercial processors to economically improve sonar system performance and to capitalize on increased processing power provided via Moore's Law rates of technical change. Phased introduction of commercial processing and a decentralized, layered decision making structure resulted in improved acoustic capability. Today the entire submarine combat system operates on commercial processors and is under continuous upgrade in technical and operational capability.

This case study chronicles the genesis and evolution of fundamental change in the technical architecture of the Navy's submarine sonar systems and its associated acquisition enterprise in response to this challenge. The author found that the genesis of the change was no different than many explanations from the innovation and change literature concerning cases of an organizational crisis, or "burning platform". But rather than focus on process, which is certainly a substantial contributor to the success, this case takes an architectural perspective, focusing on the inter- and intra-organizational relationships which were crafted by the key actors in the system and on the dynamics which enabled these actors to succeed in the effort.

The technical system architecture evolved from tightly integrated hardware and software to a layered form where hardware and software were decoupled via the use of middleware, enabling hardware upgrades and software

[4] A case study on this topic is being prepared by these authors for the case study book [16]. Unfortunately, the U.S. Department of Defense freeze on travel funds (induced by the "sequester") and the consequent unavailability of the authors may prevent this paper's presentation.

[5] A case study on this topic is being prepared by this author for the case study book [16].

upgrades to be implemented separately, increasing system flexibility. A layered enterprise architecture was critical to success, where multiple participating organizations were connected at different hierarchical levels. This enabled fleet operators to provide unfiltered feedback to development engineers, for mid-level managers to coordinate priorities and key decisions, and for senior leadership to provide consistent strategic guidance to the system. Mechanisms of success are rooted in areas not widely considered by most practitioners and scholars of defense systems development and acquisition.

This study highlights critical features of reform/transformation: 1) initiation and sustainment of change; and 2) creation of a flexible development and engineering process and enterprise. Lateral interactions across organizational boundaries at multiple levels of hierarchy enabled "opening up" of the process to new ideas and organizations – increased innovation through trust and information transparency across the enterprise. Crafting a flexible system development, testing, and fielding process requires establishing lateral connections at multiple levels of organizational hierarchy: the working engineer level, the mid-level program managers and the senior executive/Flag Officer level.

Change in the acquisition of a complex, high dollar value system becomes a battle of organizational elites at the senior and mid-levels of the stakeholder organizations. Change must be catalyzed across the government-industry boundary and there must be strong alignment at the senior levels within the government. Top level protection of middle-level innovators was crucial to overcoming incumbent stakeholder resistance.

4th Paper. Study Regulatory Institutions Competition and Collaboration Dynamics in Network Industries: Using Agent-based Modeling and Simulation – Hamid R. Darabi and Mo Mansouri.

In network industries a myriad of organizations compete and collaborate to deliver a product or service. More often than not the geographical extent of these networks places them under the influence of many various regulatory institutions. These institutions enforce rules and policies on network industries to further their own, and their shareholders' interests (citizens and companies of their sovereign entity).

The aim of this paper is to highlight this effect and to study its impact on the performance of the overall system. The dichotomy between competition and collaboration is modeled using game theory. A small subset of the air transportation network is used for this case study. The model is limited to the route between New York City and London to eliminate the biasing effect of network externalities, where the utility of a product or service is dependent on the number of users of that product or service. The conceptual model, its preliminary results, and the implications of work are presented.

5th Paper. Modeling and Elaborating a Conceptual Collaborative Framework: A Case Study for Wood Supply Chain in North-Shore Region in Quebec Province, Canada – K. Shahriari and A. G. Hessami.

A collaborative framework is proposed in this paper to increase the global productivity/product-quality and to improve the business competitiveness of the wood supply chain in North-Shore region of Quebec Province.

The supply chain consists of a number of wood industries performing the first, the second, and the third transformations from wood harvesting to lumber production, paper making, and panel board manufacturing.

The proposed solution is a four level hierarchal framework including business and marketing, scheduling and planning, quality control and supervision, and finally process levels.

6th Paper. Identifying characteristics of complexity in Mobile Emergency Medical Service: a case study in Brazil – A. W. Righi, P. Wachs, and T. A. Saurin.[6]

An approach to complexity in its different perspectives has been increasingly used to investigate the performance of socio-technical systems, mainly due to the characteristics which predominantly cognitive work activities have gained in recent times. This approach opposes the reductionist view in which the functioning of the whole can be explained by the functioning of the constituent parts. However, few studies provide a systemic analysis of the extent to which a socio-technical system is complex or explain which features of complexity are more or less present.

Mobile pre-hospital care service is a sector whose performance can possibly be improved by taking complexity viewpoints. In Brazil this service has as its main representative the Mobile Emergency Medical Service which provides care through specialist teams located at strategic points under command of a central regulatory body, responsible for receiving and processing emergency calls.

This paper aims to identify complexity characteristics and opportunities for improvement present in activities of a Medical Regulation Center in Porto Alegre, Brazil. This setting has 36 telephone attendants, 10 radio operators, and 33 medical doctor emergency regulators. Data collection was based on interviews with local professionals, three telephone attendants, three medical regulators, and three radio operators, during May to June 2010.

Characteristics of complexity stemming from this compilation and reflecting the understanding of the present authors were investigated. In this open system it is difficult to define system boundaries and interactions among the internal elements of the system (telephone attendants, radio operators, medical doctors, telephone and other technological devices) and external elements (bases, health institutions and supervisors), all consistently dynamic, interconnected, and interdependent entities.

[6] The authors were not able to secure funding for this conference so this paper could not be presented in this session.

These interactions are potentially rich and often trigger unexpected situations or emergent phenomena, e.g., difficulty in finding beds for patients in health institutions. The information is indirect, distributed, and is not always consistent and reliable. In addition, the training of professionals is lacking.

7th Paper. Biomedical Diagnostic System for Device Coding – C. Alvarez, D. Smith, and A. Agarwal.

Many issues exist in repairing faulty electronic devices in the biomedical engineering field, especially when problems are intermittent. When the device issue is a constant and verifiable problem, technicians can rapidly proceed with the repairing of the unit, replacing bad components and performing the required tests before returning the repaired unit to the customers.

However, the most relevant issue, which is subject of this paper, is the repair of a unit with an intermittent problem. Even long hours of testing does not often lead to any useful result. This results in returning the biomedical device to the customer (hospital, healthcare provider, or medical center) with a diagnostic note stating "no problem found". But frequently the problem recurs in practice. This leads to waste of resources such as tester and customer labor times, shipping and handling costs, and most importantly, customer and patient frustration and dissatisfaction.

Having a biomedical representative continuously using the equipment for a very long period of time is not a viable solution due to time and labor involved in such process. In this paper, a technique leading to a valid solution to this problem is described.

The solution is based on an internally generated error code in every biomedical device that characterizes a detected failure. A service oriented architecture with mobile based front-end is provided. An open source database of medical devices, their manufacturers and operating details, and meaningful error codes was developed.

Further the database would be open to the community to contribute more data in order to increase the number and veracity of error codes. Thus the system provides social media type functionality where users would be allowed to offer loadable data and comment on performance and possible improvements. The system administrator would control the data being entered into the system in order to ensure data sanity and integrity.

VI. CONCLUSION

The authors have advocated that case studies are important and a valuable means for understanding what works and doesn't work in addressing the most difficult systems engineering problems. They urge the systems engineering community to focus more on improving the practice of systems engineering by creating and learning from case studies. In their view the better theories of complex systems engineering will follow the successful applications of its case studies.

REFERENCES

[1] Andrew P. Sage, George Mason University, Editor, Wiley Series in Systems Engineering and Management, authoritative treatments of all foundational areas central to this *transdisciplinary* subject, New York: Wiley, http://www.wiley.com/WileyCDA/Section/id-397384.html?sort=DATE&sortDirection=DESC&page=3 (accessed 15 February 2013).

[2] "The Harvard Business School MBA – HBS Case Method," http://www.hbs.edu/mba/academic-experience/Pages/the-hbs-case-method.aspx (accessed 15 February 2013).

[3] "Paper Chase," (movie), YouTube, 1973, http://www.youtube.com/watch?v=qx22TyCge7w (accessed 15 February 2013).

[4] Jeff Conklin, Chapter 1 of *Dialogue Mapping: Building Shared Understanding of Wicked Problems*, 2005, "Wicked Problems and Social Complexity," CogNexus Institute, http://cognexus.org/wpf/wicked_problems.pdf (accessed 15 February 2013).

[5] Renée Stevens, "Profiling complex systems," Proceedings of the IEEE International Systems Conference, Montreal, Canada, 7-10 April 2008.

[6] Renée Stevens, *Engineering Mega-Systems –The Challenge of Systems Engineering in the Information Age*, Boca Raton, FL: Auerbach Publications, CRC Press, Taylor & Francis Group, 2011.

[7] B. E. White, S. J. Gandhi, A. Gorod, V. Ireland, and B. Sauser, "Application of case studies to engineering management and systems engineering education," ASEE Annual Conference, San Antonio, TX, 10-13 June 2012.

[8] "Systems Engineering Lexicon," Complex and Enterprise Systems Engineering Book Series, Boca Raton: Taylor & Francis, http://www.enterprise-systems-engineering.com/lexicon.htm (accessed 15 February 2013).

[9] Judith S. Dahmann, George Rebovich, Jr., and Jo Ann Lane, "Systems engineering for capabilities," *CrossTalk*, *The Journal of Defense Software Engineering*, November, 2008, http://www.stsc.hill.af.mil/crosstalk/2008/11/index.html.

[10] Eric Bonabeau, "Agent-based modeling: methods and techniques for simulating human systems," Proceedings of the National Academy [PNAS] of Sciences of the United States of America, Vol. 99, Supplement 3, 14 May 2002, pp. 720-7287, doi:10.1073/pnas.082080899.

[11] R. J. Abbott, "Emergence Explained: Getting epiphenomena to do real work," Department of Computer Science, California State University, Los Angeles, CA, 12 October 2005.

[12] Russell L. Ackoff, "Transforming the systems movement," *The Systems Thinker*, Vol. 15, No. 8, Pegasus Communications (October 2004) pp. 2-5, www.pegasuscom.com.

[13] Stephen Denning, *The Leader's Guide to Storytelling: Mastering the Art and Discipline of Business Narrative*, San Francisco, CA: Wiley, 2011.

[14] Robert K. Yin, *Case Study Research – Design and Methods*, Third Edition, Applied Social Research Methods Series, Volume 5, Thousand Oaks, CA: Sage Publications, 2003.

[15] George Friedman and Andrew P. Sage, "Case studies of systems engineering and management in systems acquisition," *Systems Engineering*, Vol. 7, No. 1, 2004.

[16] Alex Gorod, S. Jimmy Gandhi, Vernon Ireland, Brian Sauser, and Brian E. White, Co-Editors, *Case Studies in System of Systems, Enterprises, and Complex Systems Engineering*, Boca Raton, FL: Auerbach Publications, CRC Press, Taylor & Francis Group, Scheduled for Publication in 2013.

[17] Don Tapscott and Anthony D. Williams, *Wikinomics: How Mass Collaboration Changes Everything*, New York: Penguin, 2006.

[18] B. E. White, "Systems engineering decision making may be more emotional than rational!" 22nd Annual INCOSE International Symposium, Rome, Italy, 9-12 July 2012.

[19] Antonio Damasio, *Descartes' Error – Emotion, Reason, and the Human Brain*. New York: Penguin Books, 2005.

[20] A. Gorod, B. Sauser, and J. Boardman, "System of systems engineering management: A Review of Modern History and a Path Forward," IEEE Systems Journal. Vol. 2, No. 4, 2008, pp.484-499.

[21] A. Gorod, "System of systems engineering management framework: a 'satisficing' network approach," Ph.D. dissertation, Stevens Institute of Technology, Hoboken, NJ, 2009.

APPENDIX[7]

(Complex) System/System of Systems/Enterprise Case Study Outline

[optional items are indicated in brackets]

Case Study Elements (bulletized, for sorting purposes)
- Fundamental Essence (briefly, what's this about?)
- Topical Relevance (briefly, why does this matter?)
- Domain(s) (choose one)
  - Academia
  - Commerce
  - Government
  - Industry
  - Other (specify)
- Country of Focus (country most involved)
- Stakeholders
  - Passive/Active
  - Primary/Secondary
- Primary Insights (takeaways)

Key Words (alphabetized, separated by commas)

Abstract (no more than 200 words)

Glossary (abbreviations and acronyms, alphabetized)

Background

- Context (how did this arise, and why?)
- Relevant Definitions (define unfamiliar terms)
- Pertaining Theories (theoretical knowledge applied)
  - [Literature Overview]
  - Research Nuggets (past and present)
- Existing Practices (extant methods, available tools, and/or proven processes
- Guiding Principles (applicable principles, precepts, and/or tenets)
- Characterizations
  - Type of System
  - System Maturity (legacy, upgrade, or new)
  - Environment
  - Systems Engineering Activities (before and after)
- "As Is" System Description (before)
  - High-Level Diagram
  - [Performance Graphs]
- "To Be" System Description (after)
  - High-Level Diagram
  - [Performance Graphs]

[7] An earlier outline very similar to this was espoused in [7].

Purpose

- History (describe previous situation and evolution)
- Then Current Situation
- Known Problem(s)
- Mission and Desired or Expected Capabilities
  - [Vision
  - Goals]
  - Objectives
  - [Targets]
- Process for Achieving the Objectives Needed and Why

(Complex) System/System of Systems/Enterprise (describe each in sufficient detail) Complete a specific instantiation of Fig. A-1 below.

- Environment
- Scope
- Structure
- Boundaries
- Internal Relationships
- External Factors
- Constraints
- Other Descriptors
- Sponsor(s)
  - Customer(s)
  - Governing Body
    - Interactions With
    - Decision-Making Process
    - Methods/Tools/Practices/Principles Used
  - [Teams/Groups/Employers]
  - [Operators/Users]
  - [Countries/Organizations]
- Other Stakeholders

Challenges (what kept people awake at night?)

- Anticipated
- Actual

Development (emphasize non-conventional aspects)

- Project/Program Management
  - Planning
    - Contingencies
    - Information Management
      - Sharing
      - Security
    - Strategy
    - Resources
      - Staffing
        - Roles
      - Budget
    - Schedule
    - User/Operator Involvement
    - [Processes Instantiated]

Systems Engineering (in narrow sense)
Architecture
Alternatives Analysis
System Approaches
Description
Technology
[Technology Readiness]
Technologies Selected
Opportunity and Risk Management
Selected Approach
Design
Implementation
Integration
Testing
Fielding
[Sustainment]
[Retirement]
Management of Uncertainty (how implemented and integrated?)
[Philosophy]
[Policy]
Politics
[Organization]
Operations
Economics
Technologies

Results

Objectives Accomplished and Not Accomplished
Functions
Services
Other Assets or Capabilities
Final System Description
High-Level Diagram
[Performance Graphs]

Analysis

Analytical Findings
Activities (key tasks and their interactions)
Time Frame/Line
[Sequence of Events]
Significant Delays Incurred and Why
Methods Employed (and their efficacies)
Lessons Learned
How Were Biggest Challenges Met?
What Worked and Why?
What Did Not Work and Why?
What Should Have Been Done Differently?
To What Extent Were Lessons Applied to Subsequent Programs/Projects?
Best Practices (what would be recommended to others?)
Replication Prospects (how practical might this case study become?)
Necessary Conditions
Proposed Action Steps

Summary (provide concise overview of what happened after the fact)

[Epilogue] [what significant events have occurred since?]

Conclusions (construct an elevator speech)

Suggested Future Work

Questions for Discussion[8]
Additional Research

[End Notes]

References (primary and secondary)

[Appendices]

[Index]

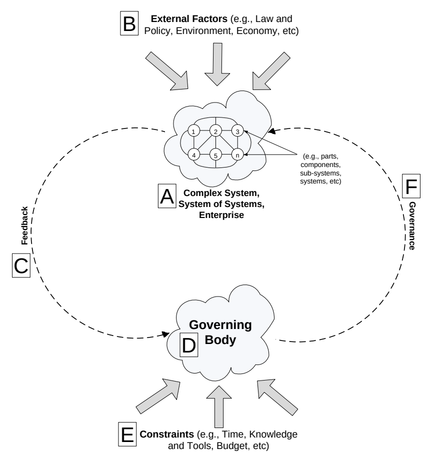

Fig. A-1. Context of Complex System/System of Systems/Enterprise (Adopted from [20] and [21])

With reference to Figure A-1 we co-editors request you to include the following sub-sections in your case study. The descriptions provided below correspond to the capital letters in the figure.

A – A short description about your Complex System, System of Systems (SoS) or Enterprise. Please explain the logic for your selection and also explain the parts, components, sub-systems, systems, etc., that make up your high-level choice. So if you have a SoS, please explain the systems that constitute the SoS.

B – Please provide a short description of the External Factors that could affect the parts, components, sub-systems or systems, etc., in your Complex Systems, SoS, or Enterprise. Examples of external factors include Law and Policy, Environment, Economy, etc. Explain briefly the effects and/or results on (especially the interconnectedness of) the parts under consideration.

---

[8] Three to five case-based questions that could be used for a discussion in a classroom environment are required.

C – Please include details of the Feedback provided by your Complex System, SoS, or Enterprise to the Governing Body.

D – Describe the Governing Body of your Complex System, SoS, or Enterprise.

E – Describe the Constraints that could affect the Governing Body. Examples of constraints can be Time, Knowledge and Tools, Budget, etc.

F – Based on the constraints affecting the Governing Body, there will be Governance provided by the Governing Body. Please describe this outcome in your case study as well.