SAND REPORT

SAND 2011-8793
Unlimited Release
Printed November 2011

Complex Adaptive System of Systems (CASoS) Engineering Framework Version 1.0

Arlo L. Ames, Robert J. Glass, Theresa J. Brown, John M Linebarger, Walter E. Beyeler, Patrick D. Finley, Thomas W. Moore

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the United States Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND 2011-8793
Unlimited Release
Printed November 2011

Complex Adaptive System of Systems (CASoS) Engineering Framework Version 1.0

Arlo L. Ames, Analytics and Cryptography

Robert J. Glass, Systems Research, Analysis & Applications

Theresa J. Brown, Policy and Decision Analytics

John M Linebarger, Systems Research, Analysis and Applications

Walter E. Beyeler, Policy and Decision Analytics

Patrick D Finley, Operations Research and Knowledge Systems

Thomas W. Moore, System Research, Analysis and Applications

Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-0826

Abstract

This document defines a framework for engineering changes within complex adaptive systems of systems (CASoS). The process is defined as a series of steps devised to focus attention on obtaining real solutions while respecting the aspects of CASoS that are uniquely different from subject areas that have been addressed by more traditional engineering disciplines. This is a living document; the framework will grow as the science and practice of CASoS engineering evolves.

PREFACE

This is a living manuscript documenting the evolving capability of CASoS Engineering from its beginnings within multiple programs at Sandia National Labs. One of four living documents, this report summarizes the theoretical Framework that drives our efforts to engineer change within complex adaptive systems of systems; the others provide greater detail on the history and guiding principles of Phoenix, CASoS Engineering Applications, and Engineering Environment. Additional living manuscripts will be added to this list in coming years.

At the time of this printing, people who have contributed to this effort are listed as co-authors; they have taken leadership roles and are encouraging others to join the growing Phoenix effort. People who join the project contribute directly to this and other manuscripts: thus, as with all living documents, some sections are slightly out of sync with others, there is some redundancy, and content is presented in a variety of styles. We accept all of these idiosyncrasies for the benefit of increased ownership.

Periodically, more concise documentation of Phoenix and its projects will be distilled, such as *Complex Adaptive Systems of Systems (CASoS) Engineering: Mapping Aspirations to Problem Solutions*, written for the New England Complex Systems Institute's 8th International Conference on Complex Systems, and also presented as the keynote at the 6th IEEE International Conference on Systems of Systems Engineering, both in June 2011.

ACKNOWLEDGEMENTS

Following the favorable reception of the 2008 *Sandia National Laboratories A Roadmap for the Complex Adaptive Systems of Systems (CASoS) Engineering Initiative*, initial funding was provided in November 2008 through Sandia's Laboratory Directed Research and Development (LDRD) from the Energy Resources and Nonproliferation (ERN, reconfigured to be ECIS or Environment, Climate and Infrastructure Security in 2010) Strategic Management Unit (SMU) to develop a pilot for the initiative, Phoenix, in context of analysis for the Global Energy System. Reported on in *A General Engineering Framework for the Definition, Design, Testing and Actualization of Solutions within Complex Adaptive Systems of Systems (CASoS) with Application to the Global Energy System (GES)*, this initial development has continued to evolve with additional contributions from Sandia LDRD within both ERN-ECIS and Homeland Security and Defense (HSD, reconfigured to be IHNS or International, Homeland and Nuclear Security in 2010) and from projects funded by a wide range of institutions:

- National Infrastructure Simulation and Analysis Center (NISAC), Department of Homeland Security (DHS)
- Science and Technology Division (S&T), DHS
- Public Health & Environmental Hazards (OPHEH), Veterans Health Administration (VHA), Department of Veterans Affairs (DVA)
- Center for Tobacco Products (CTP), U.S. Food and Drug Administration (FDA) Department of Health and Human Services (HHS)
- Department of Defense (DOD)

- Air Force Office of Scientific Research (AFOSR), DOD
- Office of the Secretary of Defense (OSD), Human Social Culture Behavior Modeling (HSCB) Program, DOD
- Center for International Security and Cooperation (CISAC), Stanford University
- New Mexico Small Business Administration (NMSBA), New Mexico Livestock Board (NMLB)

This work benefited greatly from the support of the following individuals within Sandia National Laboratories administration: Les Shephard (retired, previously 6000), Steve Roehrig (retired, previously 6300), Margie Tatro (6100), Rush Robinett (6110), Richard Griffith (6130), Pablo Garcia (6920), and Steve Kleban (6132). While members of Phoenix have come and gone, all played critical roles in the Initiative's development.

The development of the CASoS Engineering Framework required the efforts of the many people and significant stretching from traditional work processes, as well as significant time and resource commitments in order to frame and begin to test the ideas represented herein.

CONTENTS

FIGURES

ACRONYMS, INITIALISMS, AND ABBREVIATIONS

CASoS	Complex Adaptive System-of-Systems (concerning a specific system) or Complex Adaptive Systems-of-Systems (concerning the field of systems)
DHS	Department of Homeland Security
GES	Global Energy System
NISAC	National Infrastructure Simulation and Analysis Center
SoS	System-of-Systems or Systems-of-Systems
U.S.	United States

1 CONTEXT

This document defines a framework for engineering[1] change within Complex Adaptive Systems of Systems (CASoS).

From the beginning of our effort at Sandia to understand CASoS, we struggled to ensure that our efforts would be meaningful, vibrant, effective and fun. While our work with CASoS might have focused entirely on studying the properties of such systems, we chose instead to focus on engineering change to them. The difference is well expressed in the following quote from von Kármán:

> *Scientists study the world as it is; engineers create the world that has never been.*

CASoS embody the world's most important systems, the biggest problems, and the largest opportunities. It is important that, in creating the world that has never been, we are careful and thorough, but also that we enjoy the creative process, become immersed in it, thereby thinking more broadly and deeply about the possibilities and creating the best possible world.

This document is written from an engineering perspective, with the intent of conveying understanding about what issues might be encountered in dealing with CASoS, and what levers might be applied to influence their actions. It is likely that extreme positions are taken, that we spend time worrying and warning about potential pitfalls. Any particular real-life example may be easier to address (the problem isn't terribly complex, the system-of-systems aspects are easier to work through) than we might imply here. Our purpose is to convey how CASoS engineering differs from more traditional engineering pursuits and to provide a scaffolding of theory and practice sufficient to begin systemization of the world's biggest problems. We want to be successful in this, so we at least need to be aware of what can go wrong.

The theory and practice of CASoS engineering is currently emerging. This document is, therefore, of necessity a living document that will be periodically revised as we expand and revise the theory and practice.

Defining this framework has involved consideration of CASoS and engineering, how to extend engineering approaches and processes to CASoS problems, and what might be missing.

A CASoS is a complex adaptive system of systems. Our current definition is cast in terms of the definitions of the component terms. Complex refers to complex behavior, encompassing behavior that is greater than a sum of the parts, exceeds expectations, difficult to understand, emergent, results from interactions of components. Adaptive refers to the ability of the system to change itself in response to external stimuli. System of systems refers to the structure of the system, being made of parts that are themselves systems. Further definition follows below. For purpose of developing the framework, we treat the CASoS terms as a

[1] Engineer: v.t., contrive or plan out, usu. with more or less subtle skill and craft [Webster].

kind of catechism – we examine the ramifications of each term in every new context, in order to understand the impact of the CASoS notion in that context.[2]

In developing a framework for engineering change to CASoS, we focus on those aspects of CASoS that fundamentally change the engineering process. To that end, we treat the aspects of CASoS in terms of how they might impede changes, or make things difficult, or, on the flip side, make solutions simpler. Thus, complexity means behavior that is hard to understand, hard to model, and emerges. Adaptivity can seek to co-opt attempts to change the system, rendering attempted changes useless, or worse, destructive. System-of-systems means the system is composed of components that are changed (or change themselves) independently, cannot be directly manipulated, behave willfully, or are unfriendly. There are certainly positive and neutral opportunities with respect to these aspects of CASoS (e.g. system-of-systems are simple to model as a hierarchy, complexity or adaptivity might be co-opted for our purposes), but we must focus on the difficult aspects in order to create a maximally effective framework. We can't prepare to address the world's biggest problems by soft-pedaling the issues that make such problems hard.

The American Engineers' Council for Professional Development defines engineering as:

> "[T]he creative application of scientific principles to design or develop structures, machines, apparatus, or manufacturing processes, or works utilizing them singly or in combination; or to construct or operate the same with full cognizance of their design; or to forecast their behavior under specific operating conditions; all as respects an intended function, economics of operation and safety to life and property."[AECPD]

Here, we are interested in application of scientific principles, related to CASoS, to design, develop, and field changes to CASoS, respecting their intended function, economics of operation and safety to life, property, and their environment. For purpose of developing the framework, we consider engineering processes for development and fielding of engineered solutions, and the ethical ramifications of our processes.

There are many important subfields of engineering, for example electrical, mechanical, chemical, biological, and civil engineering. The engineering subfields currently viewed as most relevant to our work process include systems engineering [c.f. Kossiakoff] (which focuses on notions specifically related to development of systems consisting of many components, and on topics such as requirements and interface management) and system-of-systems engineering [c.f. Jamshidi], a more recent subfield, focusing on systems where the components are themselves systems, which might be subject to independent development and modification, might be black boxes, and/or might possess interrelationships that are unknown and unexpected. Approaches to dealing with nonlinear behaviors are present in all engineering subfields and are also relevant. Because CASoS include people and organisms, understanding their emergent behaviors requires more holistic treatment and a somewhat different set of analytical skills. We frequently rely on biology, ecology, and social science to fill in theory and strategies for influencing these systems.

[2] At least once we will include "system" by itself within the catechism, because the behavior of the CASoS as a system is itself particularly relevant.

A fundamental difference between this work and much of the theory and education in other engineering disciplines is that we focus our engineering activities on efforts to change existing artifacts (i.e. retrofitting), rather than creating new artifacts from scratch. Many of the greatest challenges to humanity involve acting to improve existing systems; much of the challenge involves determining what changes are desirable, and what path should be followed to implement those changes, given all the investment in creating current systems and all of the potentially unexpected ramifications. While retrofitting solutions can significantly limit the possibilities for change, it is a practical reality. We don't have the luxury of redesigning CASoS from scratch.

The process portions of the framework are engineering processes (ways of getting things done) and the things the processes are applied to are CASoS (the things we want to change). Their interplay is the substance of this work.

It is important up front to consider Murphy's Law: "anything that can go wrong will go wrong" [Bloch]. Murphy was an engineer. Since we are engineering changes to CASoS, we would be wise to be careful.[3]

Equally important is an understanding that, historically, changes to CASoS have been directed by political pressures, which may not involve such detailed processes as the present work attempts. If our process is useful, it will help us to discover better answers than have been developed by historical approaches. Fielding the process may entail uphill battles against people who have built careers on less-careful processes.

Be bold and careful in fielding change.

The following section introduces the CASoS Engineering process, the details of which are explored in this document.

Possible methods, theories and fields of contribution are discussed in **Appendix A**. In **Appendix B** we have assembled a set of questions and answers for the uninitiated in CASoS and CASoS Engineering to help with articulating these concepts.

The CASoS Engineering Framework is a work in progress. It is an attempt to capture the essence of past and current successful efforts, while avoiding pitfalls common to many modeling and simulation efforts. Development and refinement of the CASoS Engineering Framework must be accomplished through focus on high priority, specific applications while keeping an eye on general patterns that can provide high leverage to future activities.

[3] Freshman engineering students are routinely shown video of the Tacoma Narrows Bridge collapse (e.g. http://www.youtube.com/watch?v=3mclp9QmCGs) in order to encourage them to be careful in what they ignore.

2 INTRODUCTION TO COMPLEX ADAPTIVE SYSTEMS OF SYSTEMS ENGINEERING INITIATIVE

The concept of CASoS Engineering is described in the CASoS Engineering Initiative Roadmap[4] which was developed in 2007-2008 at Sandia National Laboratories to find ways to understand and solve the world's greatest problems. The Roadmap defines CASoS, CASoS Engineering, and the process for building the discipline of CASoS Engineering. The classical linear compartmentalized progression from Research (R) to Development (D) to Application (A) is replaced with an intrinsically integrated R&D&A process that develops CASoS Engineering theory and principles in context of solving high-impact (national and international) problems. The theoretical approach for CASoS Engineering outlined in the Roadmap emphasizes the importance of treating the Initiative itself as a CASoS and an object of CASoS Engineering. This means that, as we proceed, we must apply the developing CASoS Engineering principles reflexively to the Initiative's organization, development and growth.

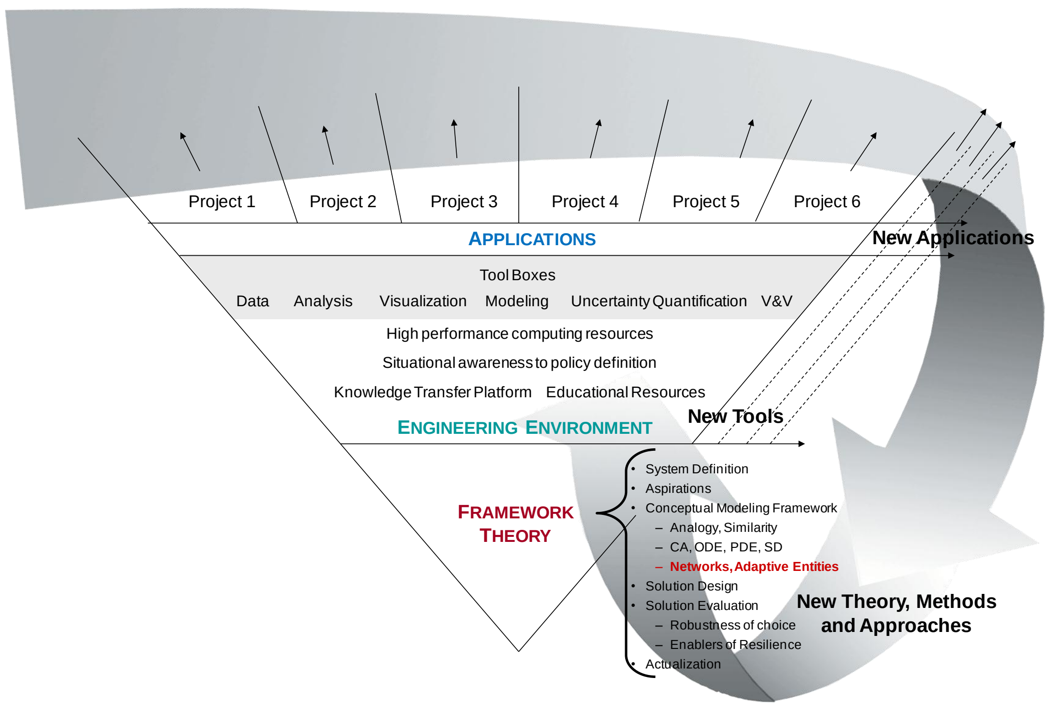

Figure 1. Integrated Research, Development and Applications Structure for Phoenix CASoS Engineering Initiative

As illustrated in **Figure 1**, the functional structure of Phoenix has formed to fundamentally integrate Research, Development and Application:

---

[4] Glass, RJ, AL Ames, WA Stubblefield, SH Conrad , SL Maffitt, LA Malczynski, DG Wilson, JJ Carlson, GA Backus, MA Ehlen, KB Vanderveen, D Engi, 2008, "Sandia National Laboratories A Roadmap for the Complex Adaptive Systems of Systems (CASoS) Engineering Initiative," Sandia National Laboratories SAND 2008-4651.

- **Application**: High-impact CASoS Engineering Applications[5] having problem and system orientation that meet CASoS criteria are chosen from the newly forming as well as established projects for which we have funding. The choice, sequence, and integration of applications are critical to the success of Phoenix and the growth of CASoS engineering research and development; we must learn to walk before we can run. Here, application drives the need for Research and Development and the requirements for CASoS engineering.

- **Research**: The ever-evolving CASoS Engineering Framework systematizes the theory and practice of CASoS engineering across wide ranging domains and diverse aspirations for affecting CASoS behavior. The Framework integrates three components:
  - Defining the CASoS, problem and approach
  - Designing and Testing solutions that are robust to uncertainty while identifying critical enablers of system resilience
  - Actualization of the solution within the CASoS.

  Here, Research is defining the science of CASoS engineering.

- **Development**: A CASoS Engineering Environment[6] that supports the Framework by providing:
  - A modeling, simulation and analysis platform in which modular computational tools can be assembled in many ways and for many purposes
  - A knowledge facilitation platform for the capture, integration and evolution of the theory and practice of CASoS engineering, providing for the education and training of newly emerging CASoS Engineers.

  Here, Development is creating the tools of CASoS engineering.

Through implementation within and across these components, we seek the next steps and episodic transformations whereby intrinsic connections between research, development, application, insights, and breakthroughs in any area can inform the others. Accelerating this process will allow Phoenix to develop and use CASoS engineering principles (e.g., simplicity, analogy, dimensional analysis and similitude, verification and validation) and methods (e.g., networks and adaptive networks, agents, hybrid approaches that blend discrete with the continuous, uncertainty analysis, experimental design and high performance computing) effectively and efficiently and thus more rapidly pose and provide innovative solutions to problems currently beyond our reach.

---

[5] Brown TJ, Glass RJ, Beyeler WE, Ames AL, Linebarger JM, and Maffitt SL, 2011, "Complex Adaptive System of Systems (CASoS) Engineering Applications Version 1.0," Sandia National Laboratories SAND 2011-8032.

[6] Linebarger JM, Detry RJ, Glass RJ, Beyeler WE, Ames AL, Finley PD, Maffitt SL, 2011, "Complex Adaptive System of Systems (CASoS) Engineering Environment Version 1.0," Sandia National Laboratories, in review.

3 WHAT'S UNIQUE ABOUT CASOS ENGINEERING?

CASoS engineering attempts to design, analyze, and field changes to complex adaptive systems of systems. While the overall engineering process itself is essentially the same as for other engineering disciplines (e.g. iterative design, test, field loops, amenable to concurrent engineering), the systems being affected are significantly different. Here we enumerate these differences, contrast our approach to them with that of other engineering disciplines, and suggest how these differences distinguish this engineering discipline from others.

Complexity. Complex behavior emerges from relatively simple relationships between entities. Such behavior can't be ascribed to individual entities, but emerges from the system as a whole. Complex behavior can be difficult to understand, produces noisy signals, and may not amenable to classical reductionist approaches: the CASoS is more than the sum of its parts. Since behavior is a result of myriad interactions, interactions with the environment often have to be included to a much greater degree than for traditional engineering activities.

Adaptivity. The system adapts its behavior due to external influences. Whether those influences are outside of our control (e.g. a tsunami hits Japan) or are due to our own influences (e.g. we attempt to field some change to the system) the system might change its behavior in ways that are beneficial or detrimental. Adaptive behavior means the system can seek to co-opt and misdirect efforts to change the system (watch your back), and adaptations emerge from an infinite space of possibilities (you don't know where they're coming from).

Systems -of-Systems behavior. An overall CASoS is composed of individual systems (e.g. countries, ecosystems) that behave according to their own rules and are subject to influences, both internal and external, that may defy our attempts to understand them. Systems-of-systems behavior also means that even if the overall number of entities might be sufficiently large to support mean-field behavior, entities are grouped in a small number of larger interacting entities, which themselves can support complex behavior.

CASoS are rarely designed. They are the result of local systems (which themselves may be designed or may also be emergent) emerging to form complex structures such as nations, ecosystems, and the internet. CASoS Engineering focuses almost exclusively on making changes to existing systems (i.e. retrofitting, designing small changes), rather than designing complete systems.

CASoS embody life- and lifestyle-sustaining systems; experimenting with them can cause large-scale disruptions. We frequently focus on modeling, simulation, and analysis to provide environments in which we can experiment on cartoon surrogates of real systems in order to evaluate possible changes to CASoS.

Compounding the difficulty is a lack of examples draw upon. In traditional engineering disciplines, we have a wealth of example designs from which to take inspiration: prototypical transformers, filters, linkages, controls, fasteners, standards, designs, etc. abound in traditional engineering practice. CASoS Engineering can look to disciplines such as public policy, biological sciences, and various social sciences for inspiration. Many of the potential approaches we glean from such theory bases lack formal testing in a CASoS environment: concerns such as implicit assumptions and conflicting theories frustrate attempts to apply them.

We are generally attempting to reverse engineer noisy, counterintuitive, willful systems, invent potential changes based on a currently-inadequate base of exemplars, and test our changes on cartoons of the real world.

4 CASOS ENGINEERING PROCESS

This section describes the process, as currently defined, for CASoS engineering. The process is documented to achieve the following purposes: to illustrate how we solve a “problem,”[7] to teach the process, to identify specific tools and methods that the framework needs to provide at each step to enable the process, and to show where those tools and methods are used.

Recent engineering approaches [Simon, Pahl], as well as traditional systems engineering theory, focus on beginning an analysis at the system level before working toward the details. Current thinking also suggests that engineering isn’t merely multidisciplinary, but omnidisciplinary [Hazelrigg]. The process we are developing follows these notions. We begin at a high level and attempt to think carefully before moving to the details. We attempt, in that thinking, to address issues broadly, in terms of understanding the behavior space of the CASoS as well as in understanding potential changes to the CASoS. In defining the process in this manner, we are essentially following traditional systems engineering processes. Previous work in systems-of-systems engineering [Jamshidi] follows similar reasoning: the effects of complexity and systems-of-systems behavior tend to dictate changes in the details of the process (e.g. organizational changes to address system-of-systems concerns, differences in modeling processes to address the kinds of behavior the CASoS may exhibit), while the essence of the overall process (i.e. top-down process for understanding the CASoS and architecting change) follows traditional practice.

While the process is described as a series of steps, remember that this is a creative endeavor. Any step may be revisited as often as is required to accomplish the goal of providing high-quality results.

Note that complexity science is still developing, and that engineering processes related to CASoS are still in their infancy. Even for “classical engineering,” many engineering processes are still in the process of being formalized; for example, theories of design are still emerging (c.f. ASME DTM, [Kelly]). The CASoS process as defined below is partially based on our best experience, partially based on speculation. Where gaps exist in complexity theory, we rely largely on modeling experience. Where gaps exist in engineering theory, we reference traditional engineering practice. Our best classical engineering analogies may be political policy making and experience in delivering engineered products that include human interfaces, because humans exhibit many CASoS behaviors, even if the overall product appears to lack CASoS focus.

Figure 2 is a diagram showing the overall CASoS Engineering process. Detailed discussion of each portion of the process follows. Note that the process diagram includes plentiful opportunities for iteration and/or concurrent activities. Iteration is fundamental to doing the job right: lessons learned from attempting any downstream activity may encourage new thinking that influences upstream activities.

---

[7] We have struggled with the use of the word “problem”. Our process is intended as a means of finding means of effecting change, whether change is targeted at specific problems, or at opportunities to improve. Common practice, both in general engineering and among the current CASoS team, is to use the word “problem,” so we follow suit here.

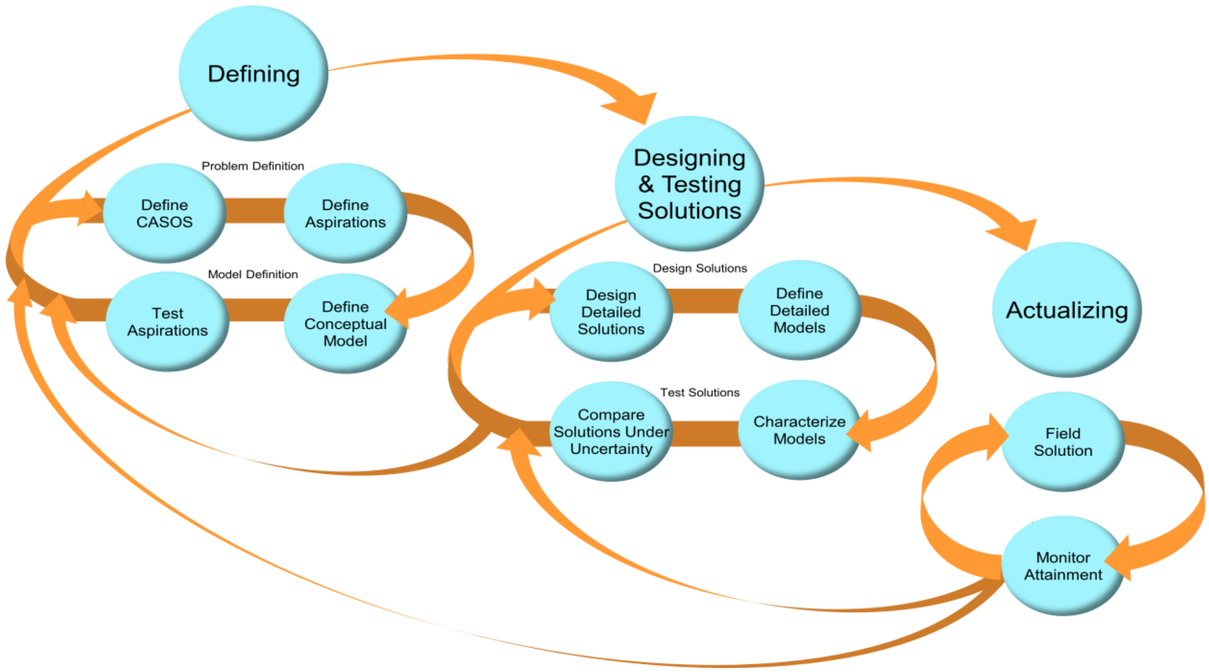

Figure 2: Diagram of CASoS Engineering Process

4.1 Defining

The Defining phase (see **Error! Reference source not found.**, below) addresses definition f the CASoS of interest, choice of aspirations, construction of a conceptual model that captures an appropriate level of detail to facilitate understanding, and testing of the aspirations relative to the conceptual model. Defining is a process of gathering information, and organizing it, of scoping the effort, of focusing efforts to understand the important issues that make the problem hard (the complex, adaptive, systems-of-system aspects), and of focusing efforts on making progress (aspirations and actualization). The defining steps are given in a logical sequence but in practice they must be addressed with a fair amount of internal iteration.

Traditional engineering disciplines might describe this phase of the process as high level conceptual design. We collect information regarding the CASoS (with particular focus on CASoS-related unique, difficult aspects of the problem), consider high-level possibilities for solutions (aspirations), construct a high-level conceptual model, and test the aspirations relative to the conceptual model both to enable down-selection to a reasonable collection of possibilities for further detailed design, and to confirm and expand our understanding of the relationship between the CASoS and our aspirations.

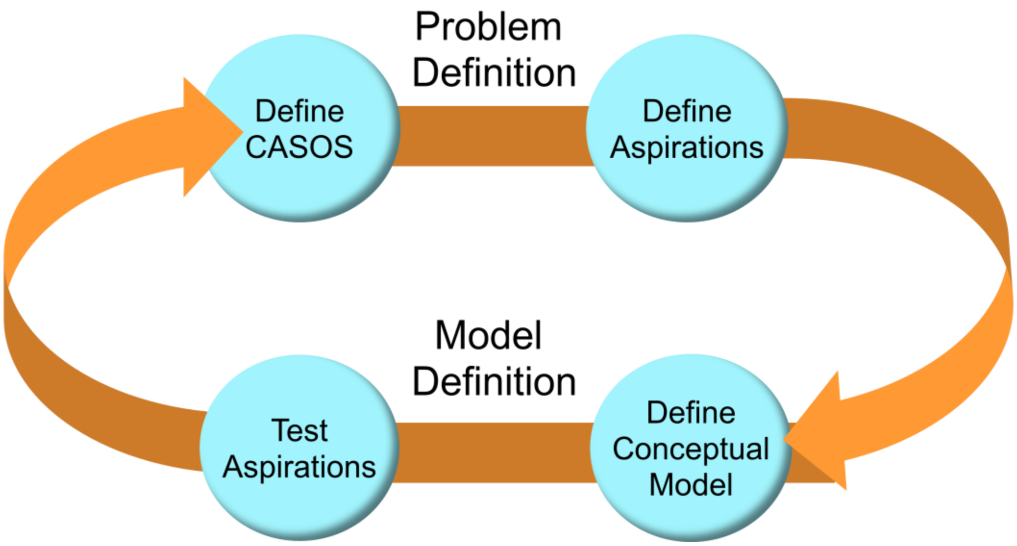

Figure 3: Diagram of the Defining Phase in the CASoS Engineering Process

4.1.1 Define the CASoS of Interest

The first step in the process is to define the CASoS, the entities that comprise it, and their interactions. Classical engineering/scientific theory provides ample examples of such a process. While we certainly engage in such capture, we focus on aspects that differ from those of classical engineering, aspects that can produce behaviors which are very difficult to manage. Our definition of CASoS is a practical one,[8] written in terms of the individual elements of the CASoS catechism (i.e. complex, adaptive, system-of-systems). The order in which we address these aspects of the problem space is not the ordinary order, but is, rather, based on practical experience – we find it easier to first consider the overall CASoS, then entities that make up the system, then complex behavior, then potential adaptive and system-of-systems behaviors. Of course, the process is iterative (and potentially recursive), so ultimately, the order in which these aspects are addressed is a matter of convenience.

Throughout this discussion, we will refer to an example CASoS, the Global Energy System, in order to illustrate the various steps in the process.

[8] And, ironically, reductionist (defined in terms of its components), given that CASoS don't necessarily lend themselves to reductionist approaches.

4.1.1.1 System[9]

A system is a set of entities, real or abstract, comprising a whole in which each component entity interacts with or is related to at least one other component entity and all entities serve a common objective. Any object which has no relationship with any other element of the system is not part of the system. From a modeling standpoint, the entities of interest within the system comprise a basic collection to be modeled, and their interactions comprise a fundamental collection of behaviors the model must capture. Other internal behaviors of entities are only relevant to the extent to which they impact these interactions.

Concepts for working with systems have been emerging for as long as Systems Engineering, a rather mature field, has existed. Current tools and processes for understanding systems are likely sufficient for our needs, although managing the amount of detail in which we attempt to model the system is a considerable hurdle.

Capture of the entities in a system and their interactions is fairly typical engineering practice. With CASoS, the kinds of entities tend to be people or systems managed by people. The interactions between people can be very subtle; we have to content ourselves with incomplete representation of the interactions. It can be useful to consider motivations of the various entities within a system – what improves their condition (relative to the concerns of the CASoS)? Capturing motivations can help in capture of interactions; relating motivations and interactions constitutes the bulk of model building.

Example: The Global Energy System (GES) encompasses the physical components of the atmosphere, lithosphere, and hydrosphere, as well as the components of the biosphere which prominently include human economic and socio-political activities in addition to energy generation, storage, control, and interdependent global distribution networks. All of these entities contribute to the behavior of this CASoS. Humans in the GES are motivated by food, shelter, warmth, transportation, entertainment, etc. Atmospheric activity is motivated by gradients of various kinds (e.g. thermal and density gradients). Interaction between humans and the atmosphere include both breathing (support of basic life function) and exhaust from various energy-generating technologies (support of transportation and higher desires among humans).

4.1.1.2 Environment

The system functions within an environment. Entities that possess very large capacity compared with the entities of interest can be considered "environment."[10] Frequently in modeling CASoS, entities that have historically been treated as the environment become important entities in their own right (e.g. when modeling transportation energy, the earth's

[9] Here we augment the typical CASoS catechism to include system and environment (an aspect of system) because they have particular relevance to understanding the CASoS. It is helpful to consider the entities that comprise the CASoS, separating them from the environment, and consider CASoS-environment issues early, before focusing too much attention on the system-of-systems aspect, which tends to focus attention inward on the CASoS.

[10] Be careful to distinguish between "larger" and "infinite."

atmosphere has historically been treated as having infinite capacity to absorb exhaust gases such as CO_2; this changes when we include pollution/global warming concerns, and the environment becomes a part of the system of interest).

Current theory and processes for separating systems from environment are likely sufficient for our needs, but can require some finesse. Rather than simply treating the environment as an infinite resource, it is important to capture interactions between the environment and the system, and represent constructs and behaviors within the environment (such as resource limitations) that might impact those interactions.

A specific class of such important interactions that occurs frequently in CASoS modeling is the notion of "threat." Threats are interactions between the system and the environment that, if implemented, cause harm (negative impact). The environment can threaten the CASoS and the CASoS can threaten the environment. Consideration of threats can rapidly generate a representation of essential interactions.

An equally important class of important interactions to consider is the notion of "promise."[11] Promises are interactions between the system and the environment that cause positive impact, for example the potential existence of (or ability to identify) new sources of energy. Promises are perhaps less frequently considered than threats in modeling processes, but the systematic identification of promises significantly improves the likelihood of successful engineering activity, both by providing balanced perspective and by focusing on additional opportunities for applying leverage.

Consideration of threats and promises can help in establishing the separation between the system and the environment. A well-placed, crisply defined boundary between the two will result in clearer definition of interactions.

Example: The boundary of the GES has often been drawn at the interface between human-created energy systems and natural systems; however, ignoring natural systems has proven short-sighted, e.g., with respect to greenhouse emissions. A more encompassing environmental boundary considers all (many?) sources and sinks for energy, natural or man-made, and their effects in both physical and social realms (for example, the economy, air and water quality, war, etc); the environment includes the physical world outside the CASoS energy system, as well as aspects of human life that aren't work-energy related. This further boundary is more difficult to draw; it's difficult to imagine what *doesn't* belong to the system. For practical purposes, we might choose to use the first boundary (at the interface between human-created energy systems and natural systems), but include significant detail in the environment to ensure that we don't make short-sighted decisions regarding the finite environment.

If we draw the boundary at the interface between human and natural energy sources, we determine that threats include natural and man-made events that would disrupt the supply of natural energy sources to man-made energy-transformation devices (e.g. hurricanes are a

---

[11] We originally considered calling these "opportunities," but the notion of opportunity carries additional meaning regarding exploitability. Since we find that we sometimes use opportunity interchangeably with the notion of "problem" (and maybe "solution") for issues of interest to the CASoS at large, we will reserve the use of opportunity, and use "promise" as an antonym for "threat".

threat that disrupts the flow of oil to refineries). Promises include new potential sources of energy from the environment and changes to the parameters governing exchanges with sources and sinks to make them more efficient. It is possible for global warming to be a promise if increased energy can be effectively harvested.

4.1.1.3 System of Systems

A CASoS is composed of component systems ("of systems"). While from a modeling standpoint, we might imagine the notion of system-of-systems as a trivial requirement for hierarchical modeling, we are, instead, interested in what makes a system-of-systems difficult to deal with in terms of its behavior, from the standpoint of engineering solutions for them.

Systems-of-systems have the following properties[12] [Jamshidi, pp 50]:

1. Operational independence of the individual systems
2. Managerial independence of the individual systems
3. Geographic distribution
4. Emergent behavior.[13]
5. Evolutionary and adaptive development.

What is clear from the above is that many of the difficulties in working with systems-of-systems are administrative or logistical in nature: the independence of individual systems and their geographic distribution means that management doesn't happen in a simple top-down fashion. Making component systems work together to achieve something more than they can accomplish independently (creating a functional CASoS) is highly difficult; such capability requires the generation of emergent behavior through the juxtaposition of various components while retaining the ability to develop in an adaptive way, evolving the overarching capability over time.

The component systems are natural to think of as systems in their own right, systems which can't be replaced by a single entity and which may be enormously complicated. Otherwise we would be dealing with a single system, rather than a system-of-systems. These other systems may have behaviors that cannot be directly observed (thus requiring modeling as a kind of black box in which behaviors are inferred or reverse-engineered), may behave independently (requiring special handling regarding "influence" rather than "control" when trying to devise solutions), or may be modified/re-architected on independent schedules (requiring continual observation and model updates).

The current state-of-practice regarding systems-of-systems considerations is less well-developed. Systems-of-systems (SoS) theory is in its infancy, and the practice is largely focused on engineering processes related to classical systems engineering practice. Complex

---

[12] While CASoS Engineering differs from System-of-Systems Engineering, we have similar views in the properties of Systems-of-Systems that are important for our respective disciplines.

[13] Not necessarily as per the CASoS/complex systems notion – Jamshidi defines emergence as a design goal of combining disparate systems, rather than as behavior to guard against.

Adaptive Systems (CAS) theory ignores these concerns. Much of current CASoS practice has its roots in CAS practice, so we need to pay special attention to system-of-systems capabilities with our framework.

The systems-of-systems perspective focuses attention on aspects of the problem where influence, rather than direct manipulation, is required to bring about change. In ordinary systems, each component of the system can be modified as required to achieve requisite performance, frequently is thoroughly understood, and can be replaced with some other component if necessary. In a system-of-systems, the components are themselves independent systems with their own inherent complexity and are not likely to be under direct control. The component systems cannot be modified without risking fundamental behavior change, may be poorly understood, and likely cannot simply be replaced. These constraints strongly limit the nature of engineering efforts that can be applied to CASoS. Attempts at change are likely to be more tentative and exploratory (due to incomplete understanding of the CASoS), and will generally focus more on influencing the CASoS to change itself rather than on directly causing change.

Example: In the Global Energy System, a component system might be a sovereign nation, such as China. China is certainly a sufficiently complex entity that it is hard to think of it as a simple component, but rather, as a system in its own right. In modeling terms, we might simply treat individual countries as entities, possibly hierarchically decomposing them. The more significant consideration, from the standpoint of devising changes, is that China *behaves* like a system. China cannot be simply exchanged for a different component; China makes internal changes at will, without announcing them ahead of time; China can sometimes be influenced, but may oppose or co-opt attempts to control it.

4.1.1.4 Complex

The system has behavior involving interrelationships among its component entities and these interrelationships can yield emergent behavior that is nonlinear and of greater complexity than the sum of behaviors of its parts. Such behavior is due to the complexity of certain interactions and is not necessarily due to system complication (a very small system can exhibit complex behavior, and a large, complicated system might exhibit very simple behavior). Complex behavior can be counterintuitive and difficult to understand, model, and/or validate.

In cases where systems exhibit complex behavior, it is highly likely that attempts to model the system will be incomplete, that some portion of the system that generates complex behavior will be missed in the modeling effort. Further, efforts to increase model complexity in an attempt to capture more sources of complexity are often misguided; they can produce much bigger models without capturing the essential generators of important complexity.

Given that complex behavior may be exhibited by a system, we are faced with deciding how to treat it in modeling and engineering activities. The following are possibilities:

- A model may have complex behavior whether or not the real system does, and for reasons unrelated to the real system. Before relying on complex behavior that a model exhibits, it is important to determine how much of the complex behavior reflects the real system and how much of it is merely a modeling artifact. Removing the effect of the modeling artifact may be difficult or impossible (sometimes a change of integrator or time step can make a difference, but complete model reformulation might be required); such behavior should at least be characterized, but is best eliminated.

- A model may have apparently correct behavior, even if it emerges from the wrong origins. Determining the correct sources of complex behavior is very difficult. Generally the behavior of a CASoS cannot be determined by study of component parts, making it difficult to determine the actual sources of complex behavior. If any engineering result depends on affecting the sources of complex behavior, the sources of complex behavior in the model must be validated (and this, in general, is not yet possible).
- Complex behavior can be superfluous to engineering activity. If the behavior is unimportant, complex behavior can be treated as noise. This may not be possible if the magnitude of complex behavior is large relative to general trends (i.e. low signal-to-noise ratio) or if the behavior is fundamental to an engineering objective. If the behavior is important to the engineering activity (e.g. control theory that addresses chaotic or complex behavior), the theory to deal with complex behavior must exist. In some cases the necessary theory has been developed, but general mathematical theory involving such behaviors does not yet exist, so care must be exercised.
- Complex behavior exhibited by a model can be correct and important to the engineering activity. This is a rare, happy state of affairs. Please report this.

The engineering opportunities involving complexity span a range of possibilities:

- Ignoring complex behavior.
- Controlling it: reducing or increasing its magnitude, frequency, content.
- Encouraging it: adding complexity generators in order to obscure underlying structure, to insert new dynamic behavior possibilities, to seek interesting emergent behaviors.

Approaches to controlling (influencing) unwanted complex behavior within models or in the real world include:

- Reducing the magnitude of discrete steps and pure time delays in a process
- Eliminating discrete operators/operations
- Reducing interactions between model components
- Inserting control systems
- Profoundly increasing the size of the system. In some cases, mean-field behavior emerges as systems grow large – the effects of individuals in such systems are increasingly less relevant, and an average, or mean, behavior characterizes the

> system. In such cases, complexity doesn't disappear, it merely becomes less relevant.[14]

Example: The Global Energy System is built of many interacting entities, including digital control systems, transmission hardware, and people using the system. The system involves generally simple behavior, many digital and discrete operations, and many interactions. The behavior of the real system is likely to be complex.

Example: A model of a transmission network may exhibit complex behavior, even though the real system may not, due to real-world control systems that are not present in the model. The model exhibits complex behavior due to discrete time integration. If an engineer attempts to learn how to reduce complex systems behavior by tuning certain parameters and those parameters interact with the time integration algorithm, the predicted change to complex behavior is entirely fallacious.

4.1.1.5 Adaptive

The system's behavior changes in time in response to external stimulus. We generally think of adaptivity as a kind of goal-seeking behavior, intended to improve the system's internal state.

The possible modifications a system might span a range from small localized change (as would be modeled as a parameter change within a model) to broad sweeping change (corresponding to fundamental changes to a model). These changes may occur within entities or among their interactions, within sub-systems or their interactions, and may result in a change in the overall system's behavior relative to its environment. Correctly modeling adaptive behavior can be difficult, particularly if the CASoS is truly a system-of-systems (meaning that subsystems are black boxes, interdependent, possibly hostile, etc.).

The earlier note regarding the difficulty of capturing essential sources of complex behavior is relevant here: capturing the sources of adaptive behavior in a CASoS can be extremely difficult, and building larger models as part of such a search can produce ever-larger models without capturing the essential sources of adaptation.

Possibilities for modeling adaptive behavior include:

- Modeling the adaptive behavior and engineering solutions in the presence of the adaptivity
- Modeling the adaptive behavior and co-opting the adaptation to achieve aspirations. It is important to remember that, because of the complex and system-of-systems behaviors of the CASoS, it may not be possible to completely capture the adaptive mechanisms. Attempts to co-opt adaptive behavior may themselves be co-opted.

---

[14] Consider the behavior of atoms in a gas. Mean field behavior might be ideal gas behavior, while individual atoms exhibit Brownian motion. As the number of molecules grows large, the Brownian motion becomes less important for most engineering purposes.

- Excluding adaptive behavior from the model and engineering a system known to be robust to the likely range of adaptive behaviors (i.e. treating adaptive behavior as noise)
- Excluding adaptive behavior from the model, engineering a solution that is not robust against the adaptive behavior, and continuously monitoring the ability of the solution to deal with the CASoS (thus including a human being in the engineered solution to implement an additional layer of adaptive behavior)
- Inserting adaptive behavior, possibly to guard against the potential for system adaptation

It is possible for multiple adaptive behaviors to exist within a CASoS, and engineering solutions may be required to deal with various adaptive behaviors differently.

Adaptation can be represented in a model as either or both of two kinds of change: parametric (changes to the numbers) or structural (changes to connections or equations). While current modeling technology and practice is likely sufficient for dealing with parametric change (corresponding to simple adaptation), a significant challenge exists in dealing with structural change (corresponding to large-scale, fundamental change in the system).

Example: The Global Energy System adapts to a variety of changes. Increases in gasoline prices encourage the development of alternative energy sources; complete lack of convenient energy sources (due to everything running out) might result in a fundamental restructuring/collapse of the energy system (as well as other systems).

4.1.1.6 Further Possibilities for CASoS Definition

The above discussion has focused on the behaviors of a CASoS that make it unusual and different from typical systems – that is, the aspects that make it complex, adaptive, and a system-of-systems. Some attention has been focused on its system behaviors, including the environment. Consideration of these aspects should produce an understanding of the essential structure and nature of the CASoS.

We cannot, at this time, provide specific recommendations regarding how detailed the definition of the CASoS should be, however, by parsimony, the simplest definition possible that allows critical entities and behavior to be represented and possible influences designed is required. The overall process for engineering changes to CASoS is iterative: the return to any previous step to revise results as understanding grows is implicit. Given the possibility of returning to system definition later, we encourage moving forward with other steps in the process in order to gain a broad understanding of the overall problem before focusing too much on details.

We provide here a list of tools and methods (beyond those listed above) that can be used to expand the system definition and encourage broad conceptual modeling. Use them as judgment dictates.

- **Components and influence charting**. Thinking about the components in the system and their respective influences can highlight gaps in thinking. Organizations tend to capture much of their thinking about components in artifacts such as organizational charts, which can readily inform such thinking. Beware, if using such pre-existing charts, to consider the purposes for which they have been created; such artifacts can limit thinking to previously-considered approaches, and engineering frequently requires novel new concepts.

- **Analogy**. Comparisons with other CASoS can reveal analogous structures, leveraging system definition effort as well as possibly leveraging downstream effort. Albright's system framework [Albright] provides a list of aspects common to system problems; consideration of a wide range of commonly-occurring aspects can rapidly flesh out understanding of a system.
- **Dimensional analysis and similitude (similarity theory)**. Dimensional analysis provides a formal method for determining the minimum number of equations necessary to describe a physically meaningful set of relationships.[15]
- **Measurement definition**. Attempting to define what measurements of a system can be taken, and how, can significantly impact the form a model takes. While focus on available data can overly limit the scope of an engineering activity, it is useful to understand measurement limitations. For example, the human element of most CASoS is notoriously difficult to measure, and early knowledge of existing limitations can either suggest alternative courses of action or provide lead time for development of measurement techniques.
- **Focus on less-robust (fragile) elements of the system**. [Colbaugh] suggests that robustness in engineered systems tends to be constructed on a feature-wise basis, that the system will tend to be increasingly robust in most features, but brittle in a few, and that modeling efforts focused on the robust portions of the system might be mostly wasted (noting that system behavior is insensitive to the robust portions of the system). He suggests identifying less-robust features and focusing on interrelationships there in order to concentrate effort where it matters most.

4.1.2 Define Aspirations

Once we've thought about what the CASoS is, we need to think about what to do with it: what are our goals, expectations, and hopes? We call these high-level general goals *aspirations.*[16] In this section, we discuss the process of generating aspirations and possible strategies for accomplishing them.

4.1.2.1 Generating Aspirations

It is important to understand, at a high level, what our goals for the CASoS are. Remember that our purpose is to think big, to imagine changes that would create fundamental

---

[15] C.f. http://en.wikipedia.org/wiki/Dimensional_analysis.

[16] The following discussion centers on goals. While we could have used the word "top level goal" rather than "aspiration", aspiration tends to encourage us to think in perhaps loftier terms than goals. If we're going to attempt to make a change to a CASoS, tampering with a difficult, potentially high-consequence system, we should ensure that it's a good one, and important.

differences that make life better, and to think broadly, in order to include possibilities that are good for the whole system (rather than being good for some and bad for others).

Sources for aspirations include:

- Examination of the system definition previously completed to identify ways of reducing threats and/or expanding promises.
- Views of competing political entities. Political entities tend to derive their existence from having strong views of how to change the world and connecting those views to constituents. Somebody cares about the views of each political group. Input from competing entities tends to provide a much more balanced view than focusing on the views on any particular group.
- Complaints about the behavior of the CASoS. "Gripe sessions" tend to include significant content regarding what's wrong with a system. Such sessions can be counter-productive if they don't also consider what's right with the system, and possibilities for how to make improvements. Any such input, however, can be quite relevant in thinking about potential aspirations.
- Interactions. Complex and system-of-systems behavior both derive from interactions among the components of the overall CASoS. A systematic study of interactions can provide thorough understanding of why the CASoS behaves as it does and a partial map of places to potentially change the system (other likely changes include parametric change and insertion of new structure in the system).
- Analogy. Aspirations that have succeeded for similar CASoS might provide similar advantage for the CASoS under consideration.

Consideration of input from even one of the above sources can produce a list of aspirations that might seem infinite. People tend to enjoy finding fault with the big systems in their lives, and most people have opinions they are willing to share. Expect a divergence of opinion, however. The strong opinions about what to do with a CASoS tend to include contradictions, opinions that appear to be directly at odds.

Given a large list of aspirations, a number of possibilities exist for managing them:

- Prioritization. To the extent that agreement can be reached on prioritization, or at least on measures of what is important, some of the possible aspirations might be set aside as being less important.
- Separate the "what to dos" from the "how to do it," goals from subgoals. Frequently, opinions regarding what should be done (e.g. deal with global warming) are intermixed with how they should be accomplished (e.g. sign a treaty, tax carbon emissions). The high-level goals are aspirations, while approaches to achieving them are strategies (see below).

Avoid down-selecting at this point. Deeper understanding of the CASoS and of strategies for achieving aspirations might uncover means of accomplishing multiple aspirations.

4.1.2.2 Commonly Occurring Classes of Aspirations

One of the patterns we have recognized in CASoS engineering is the remarkably small number of classes of aspirations we encounter. Even though the number of possible aspirations is potentially very large, we tend to find a small number of ways in which people address these aspirations.

The currently recognized[17] classes of aspirations include *predict*, *prevent or cause*, *prepare*, *monitor*, *recover or change*, and *control (influence)*.

Example: If the CASoS involves natural disasters, such as hurricanes, we would be interested in any approach that would improve our ability to predict, prevent, prepare (for), monitor, recover (from a disaster), or control impacts. The same list of possible classes of aspiration applies equally well to world hunger, energy shortages, disease.

The fact that these classes are consistent across multiple CASoS implies that the tools and methods for addressing them are likely to be consistent from problem to problem. We can leverage this consistency, and provide such tools as methods in the form of an Engineering Environment, so they can be readily applied to many problems.

Understanding the system enables us to identify appropriate aspirations. We are cautious of including understanding here as one of the classes of aspiration because it doesn't specifically change the CASoS and we have been part of too many modeling efforts that treated understanding as an end to itself.

A fairly common modeling error in treating CASoS involves creating endlessly growing models. Since a complex system derives behavior from interactions and a wide variety of interactions can create connections among entities (especially people), complex systems modeling can be endless. Adaptivity and system-of-systems behaviors additionally confound efforts to fully capture behavior. We devised CASoS Engineering as an engineering discipline partly to avoid this modeling pitfall, and instead to encourage modeling for the purpose of comparing, choosing, and fleshing out approaches to change CASoS. By focusing on problems, we can focus modeling on finding solutions, rather than on finding yet more irrelevant details to capture.

Predict the evolution of the system and, in particular, the results of events (e.g., perturbations of a variety of qualities and quantities) with direct and consequential changes in system health.

The complex behavior of a CASoS means that prediction is difficult (due to the difficulty in reducing the system to component parts, difficulty in accurately identifying sources of complex behavior, the immaturity of complexity science). Adaptive behavior means that predictive models must incorporate reasonably accurate adaptive behaviors or be continually updated to keep abreast of the changes; else the predictive model will lose its predictive capability over time. System-of-systems behavior means that interactions within the CASoS

[17] We are happy to add more classes as necessary; the current list, derived from experience dealing with large-scale disasters to critical infrastructures, has served our needs well so far.

may be hidden, and that changes to the CASoS may happen without warning, hampering the modeling effort.

Prediction requires construction of a model that accurately captures the behaviors of interest. Constructing a model that appears to reflect structures thought to be present is inadequate and misdirected, because the apparent structures may not reflect the dominant behaviors at all. Accurate prediction is difficult for CASoS due to:

- difficulties in obtaining sufficient data to validate the model
- complex, adaptive behavior
- difficulties in observing sufficient system structure
- difficulties in producing adequate models (both providing sufficient fidelity and avoiding unnecessary fidelity)

Predicting the weather, for instance, continues to prove difficult. Other classes of aspiration may be easier to implement and more effective (e.g. bringing an umbrella if it looks like rain or being prepared to get wet if it rains and the umbrella was left home are both easier than accurately predicting the weather).

Availability of accurate prediction (although costly) can significantly improve all other aspirations (because prediction permits attention to be focused on a smaller set of concerns). Prediction is synergistic with monitoring; monitoring provides data that enables predictions to be validated. Note also that while understanding a CASoS can assist prediction, it is not required (e.g. we might use correlations between a groundhog’s behavior relative to its shadow and weather patterns to predict the weather, while not understanding the weather or the relationship between the groundhog and the weather).

Note that feasibility of prediction can be significantly improved by constraining the scope of the system whose behavior is being predicted. Prediction is about determining how the CASoS will respond relative to external influences (events). Predicting world events is truly difficult; predicting how a CASoS would respond to certain classes of event is more manageable.

Monitor important aspects of a system to record the response of the system to events.

Complex behavior can make monitoring very difficult – signals are noisy, it’s difficult to localize causes (making it difficult to know what/where to measure), and complexity forces large numbers of sensors to be required in order to capture sufficient system state. Adaptive behavior means that the CASoS can change its structure to hide behaviors, requiring the monitoring system to adapt to system changes. System-of-systems structure means that monitoring may be prevented within some subsystems, complicating the task.

Monitoring requires ability to field sensors (which can be embodied in people or technology or a combination) that measure important variables of interest. Note that many variables (e.g. panic), while perhaps easy to recognize, are difficult or impossible to measure; sufficient social theories may not exist to provide mapping from measurable variables to the variables of interest. In order to be effective, this aspiration must address data collection, aggregation, display, and the potential for disasters to impact the data collection, aggregation, and display systems. Data representation and fusion are important. No model may technically be required, but various modeling constructs can be useful in data manipulation.

Monitoring is synergistic with Prediction – improved prediction can focus efforts to monitor, both reducing the need to monitor unimportant information and driving need to increase monitoring where important information for prediction is lacking.

Prevent or Cause an event to occur.

It's interesting to note that prevention and causation are not symmetric – causing an event requires that a series of events at a specific location occur, while prevention requires that at least one required event not occur at any possible location.[18]

Complex behavior complicates cause and prevention by being diffuse; because behavior arises from many interconnections, causes and effects have to account for the aggregation and the many possible locations for effects to connect. Cutting one connection doesn't prevent an event if the CASoS can use any of a multitude of other connections. Pushing on one connection doesn't cause much change if the particular connection only plays a small part in overall behavior. Adaptive behavior can complicate cause or prevention – the CASoS can adapt to work around preventive measures, and can adapt to prevent causative measures from working. System-of-systems structure means that preventive or causative measures can be difficult to implement because the local business rules obscure or limit causation.

The design of prevention and causation requires a model capturing cause and effect, so that reasoning can occur to determine when a sufficient set of causes have been addressed.

Prevent and Cause are synergistic with Predict – the ability to predict behavior can help ensure that efforts to prevent and cause actually work and don't produce unexpected behavior.

Prepare elements of the CASoS for impending events (e.g., minimize/maximize influence).

Complex behavior can affect preparation by distributing cause and effect – it's difficult to know what preparations to make if cause and effect are not obvious. Adaptive behavior can impact preparation in cases where adaptations circumvent the preparations. System-of-systems structure affects preparation by imposing internal boundaries within the CASoS that might impact the kinds of preparations being attempted, and can defeat attempts to create effective preparation.

Preparation doesn't necessarily require a detailed model of the anticipated event; it does require a representation of risk, resources, and possible implementations in order to maximize effects (e.g. determining which buildings are most critical, the costs and effectiveness of various preparations and available labor and materials sources, can inform preparation events, even if the exact location and magnitude of a coming disaster are

---

[18] For example, a model of the chain of events to cause a (gasoline-powered) car to start include obtaining access to the car, ensuring the presence of adequate gasoline, air, and spark, insertion of a key in the ignition switch, etc. A sufficient model for preventing the starting of a car might simply involve preventing access to one or any combination of the required components (i.e. without access the car doesn't start). Causing a computer failure requires breaking in at some location; prevention requires watching every possible location to be guarded.

unknown). Modeling support is mainly in terms of costs, options, and some form of discrete optimization.

Preparation is synergistic with Predict; where predictions can be made, preparations can be more focused, less costly.

Recover or Change in response to events.

This class differs from Prevent or Cause in being reactive, rather than proactive. Change is a process of modifying the CASoS in some way that (hopefully) improves its behavior. A German proverb reminds us that "to change and to change for the better are two different things". Not every change will result in global improvement, and people are not consistent in their interpretation of "improvement." The process of change involves measuring current status, determining what the desirable new status might be, defining possible implementations, and planning a process to achieve the goal state. Recovery is a special case of change where a negative event damages the CASoS and the *expectation* for change is that the change will restore the CASoS to something like it was before the event. The actual recovery can be very different from expectation – the CASoS might be restored to something less, something more or something rather different than it was before the event. The hope for recovery is that the result is better than before.

Since the CASoS is complex, changes can affect system behavior in unexpected ways, with potentially unexpected behavioral effects (both positive and negative), including the possibility that various elements of the CASoS might push back against change (even if the change ultimately results in global improvement). Adaptivity means the CASoS can change in response to changes being made, nullifying or misdirecting their impact. System-of-systems structure means changes might not be possible at the level where they would be most effective (within component systems), and the component systems might change themselves independently of changes intended for the CASoS.

Change and recovery don't necessarily require a detailed model of the CASoS. A recovery model, for instance, might seek only to capture what has been destroyed, and replace it with the hope that CASoS recovery will occur once missing functionality is restored, without further effort. Because this is a CASoS, however, system behavior is a function of state and can be hysteretic, so the CASoS might not return to previous behavior without significant effort; return to previous behavior may not even be possible (due to adaptation).

Change and Recovery are synergistic with Prediction; good prediction provides a means of understanding possible undesirable consequences.

Control aspects of the CASoS to steer it towards specified regimes, and/or away from others.

Theory that addresses control of (influence over) complex and chaotic systems has been recently emerging (c.f. Scholl 2007, Ioannou 2007, Kiss 2007, Tabuada 2009). The particulars of complex system control are outside of scope of this work; the reader is referred to the references. It is interesting that current thought on chaos and complex system control seems to emphasize control through influence; the references suggest nudging the system to get what's desired, rather than rigidly trying to control every aspect of the system, or applying constant forcing functions. Perhaps subtlety is the key.

Depending on the situation, complex behavior can engender the following engineering approaches:

1. Treat the complex/chaotic behavior as a dominant behavior, and apply the appropriate control theoretic technique. Models created for such treatment may not need to be particularly accurate; approximations of model structure and parameter values might be quite suitable for the control theoretic exercise, as long as appropriate means are taken to permit adjustment of the control system when fielded.
2. Treat the complex/chaotic behavior as noise to be ignored or filtered away, and control some other behavior (e.g. large scale trend).[19]

Adaptivity might require that the influencing force be adaptive, that the control system actuate change to the system via its adaptive mechanisms, or the adaptation might be sufficiently small that the control system can effectively ignore it. System-of-systems structure hides some of the structure of the CASoS, but control systems are capable of working well even if they are designed using approximate models of the real systems, and control theory directly addresses lack of observability of state variables. System-of-systems structure can affect one's ability to control the CASoS, by preventing or hindering control inputs from accessing system components; control may have to be indirect (as influence) rather than direct (as control).

Applying control requires a reasonable behavioral model of the system, and adequate control theory to apply to the problem. Interestingly, models used for design of control systems can often be very coarse abstractions of the real system (e.g. [Kiss 2007] provides control theory appropriate for treating heart and brain function issues, developed from phase models – models that only capture time relationships between variables, ignoring magnitude); parameter values need not be precisely known if a calibration process is used in implementing the control. The chief concern is that the control system is adequate for the purpose (that it deals with the dimensionality of the CASoS and its range of possible behaviors), not that the model used for its development be particularly detailed.

Control is synergistic with prediction; an ability to predict, even roughly, improves ability to control.

Often classes of Aspirations cannot be addressed independently. Each class influences the others and all may be interwoven in a comprehensive engineered solution. As an example, consider prediction. Prediction of time series within socioeconomic-technical CASoS is in general limited to short horizons. However, in combination with another class such as control prediction can often transcend this limitation as control places bounds on system behavior and thus constrains the realm of possibility. As another example, consider prepare. Independently, preparations could be made for all contingencies. However, why prepare for an event if control is designed to significantly reduce the risk of the event occurring or if the system is modified so that it is resilient, significantly reducing the cost of recovery from the event?

[19] This option is certainly counter-culture relative to current modeling trends. In a surprisingly large fraction of the models Ames has examined, discrete and continuum behavior have proven to be easily separable, with the resultant continuum behavior being linear. Whether this is a result of modeling practice or a result of the real-world CASoS being less complex than we might think is unresolved, but it is certainly possible that the system is simpler than we are prone to think, and, if not, that our best attempts to model such systems will be simpler than we might expect. While it may seem heretical to assert the possibility of a trivial solution for a large complex problem, it is important to be practical here – if there's a simple solution, don't ignore it.

Compared with other aspirations, control is perhaps the strongest. If you can make the CASoS do what you want, you're likely finished. (Of course we are still faced with the question of who gets to decide what we want the system to do…) We often see the other aspirations applied in the real world, rather than influence. We speculate that this might be because approaches to control are just emerging, because people prefer incrementalism, or because control is too difficult to achieve.

We believe (all other considerations being equal) that emphasis should be on design of influence (control) that increases CASoS robustness (keeps it from breaking) and/or CASoS resilience (permits it to recover quickly). Where control is not yet feasible, consider focusing on classes of aspirations that serve immediate needs.

4.1.3 Define Conceptual Models

A conceptual model provides high level representation of CASoS behavior and entity interactions. It should also enable us to represent the impact of potential actions to achieve our selected aspiration(s).

At this point, it is appropriate to focus on high-level concepts, avoiding large amounts of detail. It is important to create some tangible representation of the conceptual problem space (a chart, a computational model, a sub-system diagram, etc.) so that it can be reviewed, and tested relative to the behavior and relationships of entities in the CASoS and relative to proposed interventions.

At the conceptual modeling phase, we are likely to uncover a variety of gaps – gaps in interrelationships among entities, gaps in our understanding of how CASoS work, gaps in understanding how proposed actions could impact real entities in the system, inadequate ideas of how aspirations might be implemented. Identifying these gaps is likely much more important than attempting to fill them at this point. Characterizing the gaps and their likelihood of being filled can provide important information about likely robustness of proposed interventions to achieve our aspiration(s). Understanding the extent of gaps provides an opportunity to balance effort on them (such as focusing effort on those gaps that are most influential in creating robust solutions).

4.1.3.1 Modeling CASoS Behavior

A central concern in defining models is what to include. A model must include sufficient detail to capture those behaviors that need to be tested; entities and interconnections must be sufficient to answer questions. A spare model is frequently better than a complex one. We are aware of modeling approaches that only capture time-lag relationships (phase modeling), that focus mainly on elements of the system that are likely to be non-robust (Colbaugh), and that work only at high levels (e.g. entity-relationship diagrams with perhaps 10 entities). We are also aware of approaches that attempt to capture everything. The more terse conceptual models tend to be completed (at least to a level where they can be used), and tend to answer questions.

Ultimately, CASoS behavior must be modeled well enough to support engineering decisions. Know the aspirations, and carefully limit behavior models to those aspects that permit interventions to achieve the aspiration(s). The conceptual model of CASoS behavior does not have to be a computational model, but can be. It might merely list entities and some of their important relationships. It must support decision making.

A variety of concerns can complicate the modeling process – multiple time scales, multiple paradigms, interconnecting models having fundamentally different representations. As in other engineering disciplines, these are generally implementation concerns, and modeling activities should be managed to minimize their impact. It is noteworthy that many formulational and numerical issues associated with these concerns are unsolved research problems; if you must go here, tread carefully (c.f. [Peters]). Try to keep it simple.

Whether a single model or multiple models are produced is largely a matter of convenience. Separate models fail to capture interrelationships between representations, but are also significantly easier to build and maintain. Especially at conceptual design, we're interested in simple models that permit ready exploration and coarse decision making among many possibilities. Simplicity and agility may be much more important than depth of coverage of interactions at this stage.

The process of modeling the system involves selection of appropriate methods, construction of appropriate models, and obtaining data to support model development, analysis, and testing.

Choose appropriate methods and theories from the full space of those possible based on aspirations chosen. Possible methods, theories and fields of contribution include analogy, dimensional analysis and similitude, experimental design, system dynamics, non-equilibrium thermodynamics, complex adaptive systems, game theory, percolation phenomena, agent-based modeling, networks, system optimization and control. The discussion of CASoS aspects and aspirations above has provided some general suggestions for important features to be captured within a model. While methods and theories are frequently chosen based on familiarity, it is important to realize that model representation can have a fundamental impact on the reasoning you will perform with it, that every modeling method and theory potentially enables biases to enter the model. Breadth of knowledge of methods (tools and formalisms) and theories (things to represent) encourages better modeling practice, better thinking about the solution. Whatever methods are chosen, it is important that the resulting models capture an appropriate level of fidelity to support the problem, support analysis of potential engineered solutions, and are sufficiently simple to be understood. Appendix A lists a variety of potential methods that can be applied to development of appropriate conceptual models.

Construct appropriate conceptual models. It is important to capture salient features of the system, while avoiding getting bogged down in details. Since this is invariably a spiral development, and more details will be added as appropriate and necessary, it is useful to focus on the simplest representation possible, and seek to construct representation in general terms that might be able to be extended and re-used easily. In our experience, it is likely better to avoid attempting to reuse large pre-existing models, as they are likely targeted toward different problems, and it's easy to fall in the trap of repeating other's mistakes. We don't want the conceptual model to limit our conceptual thinking.

Conceptual models will include the essential behaviors of the CASoS as well as representation of the impacts of various strategies. It is important to ensure that the effects of strategies on the CASoS can be represented in the model, and that measures of the impacts of strategies must be identified so that positive and negative effects of strategies can be estimated, in order to evaluate individual strategies relative to the CASoS in the absence of change, and to compare strategies to each other. Keep in mind the possibility of unexpected impacts. Measuring unexpected impacts is always a difficult, uncertain problem.

Obtain data required to support conceptual model development and validation. Data to support the conceptual model comes in a variety of forms: parameter values, example time series, and directions of likely trends. Information about the data that should be captured includes source and uncertainty. Note that data frequently is valid over a limited time interval, so an estimate of expiration data and likely means of replacing data when it expires may also be important. Data may be difficult to obtain and of marginal quality. Gaps in data coverage can be addressed through use of uncertainty analysis, parameter space exploration and behavior exploration. Good data, when available, makes problems significantly easier to address.

The detailed process of defining a conceptual model is explored in [Beyeler]. An appendix in [Miller] is helpful in addressing system modeling issues.

4.1.3.2 Modeling Aspirations

Aspirations are modeled as attachments to the system model; they include simple changes to parameters, as well as structural additions (new interconnections and/or new equations). How to model them is typically straightforward; what to implement can be much more difficult.

Modeling aspirations can be a more creative endeavor than modeling system behavior. The system model captures what *is*, while aspirations are what *might be*. There are many more possibilities of what might be, but discovering possibilities that really work can be difficult.

Aspirations might be implemented in a variety of ways. Controlling automotive emissions might be implemented by taxation, by edict, by redefining work environments. We use the term strategy to refer to a potential implementation of an aspiration.

Historic strategies to dealing with CASoS issues tend to focus heavily on searches through parameter space (varying numeric parameters, such as tax rates), and trying variations of the same strategies people have tried before. If the problem is that easy, do it.

In others, the core problem is determining how to implement the aspiration.

Classical engineering approaches tend to focus on reductionism – write down the requirements, define components that achieve the requirements, and subdivide until the problem is solved. CASoS systems tend to defy reductionism; it seems in our experience that reductionist approaches to changing CASoS haven't historically been particularly successful. Try it if you will.

We are seeking engineering approaches that focus on the whole system. In seeking approaches, we have found significant food for thought in biological sciences. Nature tends to deal with incremental solutions that are robust and resilient for the entire system. Notions such as competitive exclusion, multi-level selection, anti-trust, ecologically stable systems (ESS), distributed non-localized control, evolution (especially being careful what you select for) are currently providing inspiration, and seem to be leading to novel approaches.

For CASoS systems, we know a lot more about how to create models of system behavior than about potential ways of changing their behaviors. Because of this, we have to work to ensure a balance between modeling and strategy design. If we aren't careful, we may find we spend too much on modeling, and we may be caught flat-footed, with too few ideas for how to fix systems.

4.1.4 Test Aspirations

At this point in the process we have a very simple conceptual model: a definition of the CASoS structure, a conceptual model of behavior, and conceptual models of one or more strategies for implementing aspirations. We test strategies to understand where strategies work well (applicability/robustness), to determine which to pursue (down-selection), to determine where we need more information to make decisions (learning) and to confirm our models for decision making (validation).

The decision process can begin with the conceptual model, and will likely require models of increasing fidelity in order to improve the information that informs decisions.

It is possible to make early decisions about strategies based on minimal formality (e.g. based purely on logical decisions about potential benefits, costs, unintended consequences). We include suggestions regarding more formality both to encourage more rigor – even if the process is informal, it should be informed by these considerations.

Classical decision theoretic technologies can provide support for making decisions. Strategies, benefits, costs and multiple objectives are all handled using classical theory. We list some specific considerations here that might be specific to CASoS engineering. Note that at conceptual design time we might be less sensitive to these issues than in later, more detailed steps; we list these considerations here to encourage as many as practical to be considered, to improve the likelihood of success of detailed design.

Measuring strategies. Complex and system-of-system behavior means that outputs are frequently noisy and might change in unexpected ways. Adaptivity means the system may change itself. The decision theory must accommodate these concerns, through a combination of careful definition of metrics, decision algorithms, and possible finesse (e.g. removing certain realistic behaviors from the model so decisions can be made).

Growth of understanding. This is an iterative design process, and understanding CASoS and interactions with strategies is a learning process. It may be wise to avoid postpone down-selection until more is known.

Uncertainty. We invariably have great uncertainty in the values of parameters and the structure of the systems we deal with. During conceptual design, we can often ignore much of the uncertainty by focusing on dominant differences in the behavior of various strategies.

Robustness. Given significant uncertainty and potential high impact to changes in CASoS, robustness of strategies is potentially as important as other benefits. Complex, adaptive, and system-of-system behaviors can all work to undermine, misdirect or embrittle strategies, so be sure to measure robustness and include in the decision process. Seek to understand the potential for unknown unknowns and design to be insensitive to them wherever possible.

Also note that optimality often comes at a cost in robustness. Seek good behavior and robustness before optimality.

Apparent Failures. Many strategies will fail, or appear to fail, at first attempt. We have encountered multiple CASoS activities where every strategy they have implemented has resulted in failure. Keep a log of failures, consider patterns in them, and seek to understand potential alternative formulations. The strategy may not be fundamentally flawed, but might require modification in order to succeed. For example, the social distancing strategy for pandemic influenza fails if applied to the entire populace; when applied to key individuals (children) it is entirely practical.

Interactions. Strategies may interact, either in relying on the same inputs or producing related outputs. Whether early or later, it might be useful to consider these interactions to determine whether they are synergistic or competitive, and whether something can be do in design to achieve synergistic strategies.

Multi-strategies. Strategies do not have to be implemented individually. For many CASoS, implementation of strategy is inexpensive, relying more on political capitol than on expenditure of large amounts of time or effort. Consider combinations of strategies.

We have not yet devised/chosen a decision-theoretic framework. The considerations above should drive whatever decision process is used.

Be aware of the possibility of aspiration and implementation choice being driven by political concerns. Decision makers and solution providers tend to be part of the CASoS themselves, and can be affected by a variety of motivations, including profit and by concerns about how a given solution might affect them personally. It is possible to take advantage of these effects, both as a part of the solution and as a part of convincing people to field the solution.

Time, budget and personnel constraints can all impose limits on what actions might be performed, either individually or in combination. In extreme cases, such limitations (particularly time and budget constraints) can directly impact the engineering process, preventing detailed design and reducing the process to "shooting from the hip." Such constraints must factor into the decision theoretic process; ideally, we should develop a scalable CASoS engineering process that provides gracefully degrading results in the face of such constraints.

Finally, choice of strategy isn't simply a discrete process of enumerating combinations of strategies. The effect of a strategy depends also on continuum logic associated with elements of the strategy: how much of each element to engage in (in cases where continuum amounts of effort are possible) and relative timing of actions (dynamical effects can influence behavior) mean that the search for good strategies is a hybrid (continuum + discrete) search.

4.1.5 Review and Potential Iteration

At this point, it is useful to review the CASoS definition, possible aspirations, strategies, and conceptual modeling methods, to ensure that they form a cohesive whole. It is possible that the conceptual models have shown that none of the strategies that have been developed work, that the aspirations are too high or too low. Inadequate data and inadequate understanding of the CASoS may invalidate many of the possible strategies that have been devised. It is also possible that the process has identified alternative possibilities that are worth exploring. Returning to the drawing board early is strongly preferred to proceeding with poor design options. Early iterations are cheap[20], particularly when compared with the cost of fielding a mistake.

The following collection of questions can be useful in reviewing the systematization of the problem and identifying potential opportunities for improving results. Many of these questions will be asked by reviewers, clients, and people who might be impacted by the engineering activity.

System representation:

- What is the general definition of the system we are interested in? Is the general definition adequate?
- Have we missed important aspirations in our process?
- How did we devise the set of possible strategies, and why have we down-selected to this set?
- Have we thought carefully about unintended consequences? Have we taken advantage of possibilities for leverage?
- What will be our measures of system response and why?
- What will be the controls used to implement intervention/changes and why?
- What will be our choice of system and environment and why?
- How will we structure the system (e.g., interaction amongst entities) and why?

Conceptual model implementation within a computational environment:

- Why do we use the particular modeling approach rather than other approaches?
- How will we translate the conceptual model into conceptual code?
- How will we translate the conceptual code to operational code?
- How do we confirm that the answers the model produces are reasonable?

[20] In more classical engineering, movement from an early phase to a later phase in the process is accompanied by significant increases in budget and risk, often ten- to one-hundred-fold. Don't rush the early processes, as they provide the best opportunity for choosing wisely.

4.2 Designing and Testing Solutions

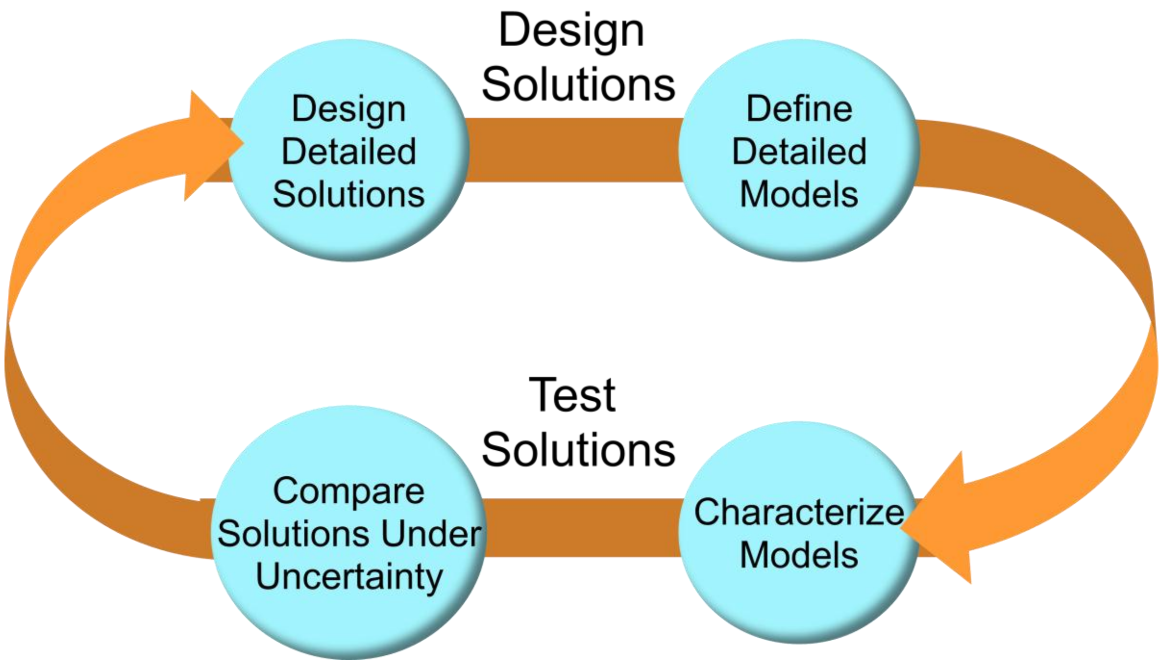

Figure 4: Diagram of Design and Test portion of CASoS Engineering Process

Error! Reference source not found. illustrates the Design and Test portion of the CASoS ngineering Process. At this point we have the following:

1. A conceptual model of what we have, the CASoS, in terms of entities and interrelationships
2. A conceptual design of what we want to do, consisting of one or more aspirations and strategies for implementing aspirations
3. Understanding gleaned from the conceptual design phase (and possibly other iterations of this phase)

We desire to actualize solutions – to field them. At this point we need to perform detailed design of strategies, increase detail in models where necessary, characterize models, and compare solutions in detail. We need to consider the behavior of the system and strategies sufficiently broadly and deeply to determine further what is necessary in order to reach fieldable, actualizable solutions.

4.2.1 Design Detailed Solutions

Designing solutions involves getting down to brass tacks – defining in detail exactly how things will be done, and how success of their implementation can be measured. This is merely a more detailed treatment of strategy design as executed in the defining phase above.

We adopt the word "solutions" here to refer to detailed "strategies". We now design whatever tactics are required to implement the solution.

4.2.2 Define Detailed Models

Defining detailed models involves including whatever detail is necessary to address the new details embodied in the detailed solutions, along with whatever model behavior is necessary to confirm that the change to the CASoS can be trusted to produce desired outcomes.

At this point detailed models are most assuredly computational, as they likely include systemic effects that are impossible to compute manually, even if their “physics” is quite simple (e.g. prediction over a sufficiently large social network, even of a simple behavior such as contagion, requires a computational model).

4.2.3 Characterize Models

Model characterization is necessary to confirm that detailed models are appropriate to use for making decisions. With increase in model complexity, we see increased likelihood of error. It is important to confirm that model behavior is appropriate to support the decisions that will be made, before expending effort to make decisions (based on potentially flawed models).

Depth of coverage in model characterization can vary, being shallow for early design iterations, more thorough as the process converges on final designs. Effectively balancing depth of coverage against funding and time pressures can be difficult, particularly if staff have a predisposition to avoid characterization activities. Automation could be very effective in this area.

Note that model characterization and validation does not necessarily require the model to be compared to real world data. It is necessary to confirm the model behaviors that support the evaluation of strategies/solutions (perhaps confirming trends, or confirming that behavior matches expert expectation).

This shift of responsibility does not mean that model behavior will not be examined, or that the examination will not be thorough. Examination of model behavior should be sufficiently thorough to defend the engineering decisions made based on it; parameter sweeps should be sufficiently dense that they uncover potential unexpected behavior over a sufficiently wide range of parameter values.

Fundamental behavior change over small parameter change is suspect, complexity notwithstanding, for the reason that such changes undermine confidence in our ability to make engineering changes that will work.

Finally, note that mathematical theory is a means of reducing effort in this area. Use of phase portraits and bifurcation theory is appropriate for continuum models, and can profoundly decrease the density of parameter sweeps required to characterize behavior of dynamical models.

A number of behavior characterizations have proven to be important in characterizing and validating CASoS models. We describe them here.

Intrinsic Behavior. The behavior of the model, absent external drivers and application of strategies/solutions, should behave appropriately for any change in parameter, over a reasonably wide range of potential behaviors. Perform parameter sweeps and confirm the smoothness and direction of behavior changes. Direction of behavior changes should be appropriate for individual parameter changes. Perform sensitivity analysis and confirm the direction of changes. Minimally, analysis should cover those parameters used by engineered strategies, and behaviors measured by those metrics used in decision processes. Any parameters and behaviors not validated by those processes must be held constant for any engineering design process; it might be better if they weren't present in the model. Also, the presence of external drivers (e.g. random number generators) is suspect – they represent behavior that is not intrinsic to the model (and thus not addressable by engineering change) and might introduce undesired effects in the decision process (e.g. changing model topology might also change the behavior of a random number generator, biasing behavior in an undesired way).

Strategies/solutions. The behavior of the model under influence of various strategy/solution models must be confirmed, both in terms of fundamental behavior change (e.g. change from open-loop to goal seeking behavior) and in terms of trends over parameter changes, for a sufficiently wide range of parameters to serve the engineering decision function.

Complex Behavior. The magnitude and sources of complex behavior must be confirmed, to the degree that complex behavior has been included in the model (i.e. it is possible to construct a non-complex model of a complex system if the decisions to be made with it are insensitive to the presence of complexity). If strategies will attempt to affect the complex behavior exhibited by the system, the structures that generate complex behavior must be confirmed (it is not clear this is possible). Recurrence plots can be used to examine signals for presence of complex behavior[21]; metrics exist for measuring the degree of complex behavior present in a signal[22]. If complexity is not included in the model, insensitivity of the engineered strategy to complex behavior must be confirmed (e.g. by injecting complex signals and observing the ability of the engineered solution to cope).

Adaptive Behavior. The magnitude and sources of adaptation must be confirmed, to the degree that adaptive behavior has been included in the model, to the extent they affect engineering decisions. If adaptivity is not included in the model, insensitivity of the engineered strategy to adaptive behavior must be confirmed.

System-of-System Behavior. The magnitude, sources, and types of system-of-systems behavior must be confirmed, to the extent they affect engineering decisions. If system-of-systems behaviors are not included, insensitivity of the engineered strategy to system-of-system behavior must be confirmed.

Dynamical Behavior. The potential behavior modes present within dynamical systems should be explored. For continuum systems this means enumerating and characterizing attractors and plotting their bifurcation behavior relative to parameter change. For discrete and/or hybrid systems, similar suitable measures need to be identified and executed. (Identification of stationary orbits in discrete systems is similar to characterizing attractors.). In any event, characterization should include understanding of the behavior relative to a variety (ideally, all) of starting locations in the state space, as well as characterization of changes in behavior relative to parameter space. Characterizing complex dynamical behavior

[21] c.f. http://en.wikipedia.org/wiki/Recurrence_plot

[22] c.f. http://en.wikipedia.org/wiki/Recurrence_quantification_analysis

can be profoundly more expensive than characterizing simple behavior. This is a case where fidelity can come with a very high price, as well as significantly increasing the difficulty of making design decisions.

Parametric Uncertainty. Uncertainty in behavior due to uncertain model parameters must be characterized. This can be realized as a local sensitivity analysis about an interesting point(s) in model behavior space.

Structural Uncertainty. For cases where model topology varies, it is necessary to characterize changes in model behavior as a function of changes in topology. Typical CASoS models capture behavior as a function of social network topology; such models frequently exhibit complex behavior that likely arises from a combination of network topology and discrete behaviors in the local agent physics model (e.g. contagion model). Characterization should include investigation of ordinary simple topologies (e.g. star, fully connected, random, small world) as well as community topologies, to determine when complex behavior emerges and when/whether mean field behavior emerges. Such characterization might be compared with real world measures (at least in terms of number of entities) to determine whether complex behavior emerges appropriately relative to model topology, and whether model connectivity seems appropriately structured and sufficiently dense.

Time Step. Time step (for systems involving dynamic behavior) should be examined to determine whether model behavior is convergent. Note that for chaotic systems, correctness cannot be guaranteed for any discrete time stepping algorithm [Yao].

Burn In. In cases where adaptive entities are implemented, a training period might be required for the model. Determining the length of training required as well as the possibility that the training time might vary with model parameters is important to ensure that model evaluation is successful.

Model characterization constitutes verification and validation exercises, as long as the behavior of the model is compared to some expectation (not necessarily real-world behavior) the results are documented and, in some sense, repeatable (to permit independent confirmation of results).

As part of the characterization activity, consider characterization of the size of the model relative to the CASoS being modeled. A systematic determination of which model structures don't contribute significantly to overall system behavior can prove highly instructive. For engineering purposes, it is important to understand the essential behavior in order to minimize effort and focus on the most important levers. This does not necessarily mean that the model must be simplified, but rather that model structures should be segregated into those that cause fundamental behavior versus those that don't. Focusing on fundamental model behavior tells us what we need to know about how the system behaves and how we might change that behavior. The other structures tell us where we might implement changes to the system. Separating the two significantly decreases engineering effort, by eliminating redundancy associated with multiple engineering implementations that really cause the same effect to the system.

4.2.4 Compare Solutions under Uncertainty

Each solution is evaluated for its behavior relative to a metric or set of metrics. The values produced will vary as a function of parameter values and as a function of uncertainty, both parametric and structural (e.g. social network topology). Choice of solution (or combination of solutions) is a function of these sources of uncertainty, and the space of possibilities should be traversed thoroughly if results are to be trusted.

Comparison of solutions and making a decision is the realm of decision theory. We have discussed the principal salient issues above.

The metrics chosen have a fundamental effect on decision. It is as instructive to determine how much the metrics have to change (e.g. in terms of weighting parameters) in order to change design decisions. Ideally, design decisions should be robust to uncertainty, location in state space, and metrics used to choose them. In reality, some compromises will have to be made. Consideration of the effects of metrics does not necessarily require additional model execution – it merely requires preserving the values used to compute the metrics for each model execution, then performing parameter sweeps over the parameters used in computing metrics. An example of this would be a disaster scenario, where damages due to the disaster involve loss of lives and loss of property. It might be necessary to determine a weighting factor that places a monetary value on human life. While deciding on such a value might be difficult, it is less instructive to determine what form decisions take for specific values of that parameter than it is to understand how much a life has to be valued before it changes the decision about what to do.

4.3 Actualizing

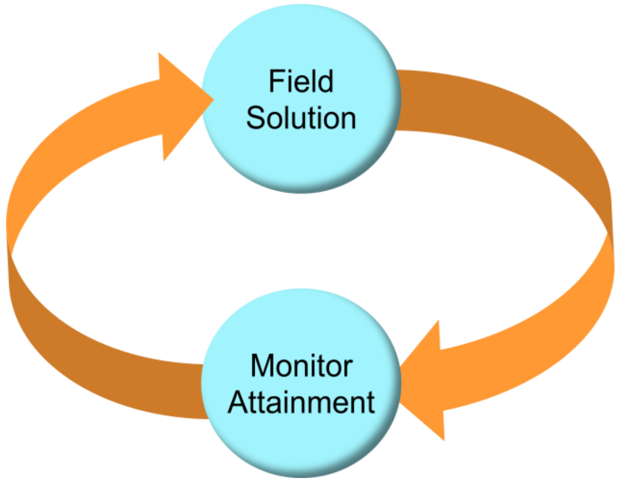

Figure 5: CASoS Actualization Process

Actualization is the process of fielding implementations of the chosen strategy. Preparation for this has included defining the system, exploring a range of aspirations and potential implementations, down-selecting appropriately, modeling to test potential implementations, estimate parameter values, building confidence, managing uncertainty. **Figure 5** is a diagram showing the process: field the solution, monitor the attainment of results, adjust the solution until results are as hoped.

If adjustment doesn't work, there are feedback loops to return to previous steps in the process (back to the drawing board). Be prepared to quickly go back if necessary, because the fielded solution is in the field, and events may unfold more quickly than long design loops can respond.

Previous steps have constructed conceptual and detailed models of various attributes of the CASoS and how they are affected by various strategies (and combinations of strategies). Such models have been evaluated to compare the quality of various solutions, to measure robustness and resilience, etc. There may be significant uncertainty in the parameters and in model structure,

In classical engineering fields, fielding a new product is always a time of excitement and worry. While due diligence has been applied to attempt to ensure the product will work, there is always the possibility of failure, because some aspect of the product has not been considered, the product is used in unexpected ways, implementation (e.g. manufacturing) defects occur, etc. The real world is difficult and risky.

A common practice in classical engineering is to try a new solution in a limited market. The analogous situation for CASoS would be to test the solution in some limited region, if possible. By doing so, bugs in the solution can be ironed out, defects exposed, and confidence in the solution can be gained before fielding the solution across the entire CASoS. There are difficulties and risks associated with fielding to a limited region:

- A limited region may not exist. CASoS are interconnected, and there may simply be no way to define a limited region.

- Fielding to a limited region may not scale well to the entire CASoS.
- Fielding to a limited region may provide the larger CASoS a means of learning about the solution and devising more effective barriers to implementation than would occur were the limited fielding not performed.

Scalable solutions are a variant on this theme: devise a solution whose implementation can be scaled up and monitored over time, so that the impact of the solution can be proven with the real system before the full impact of the solution is felt. Noting that CASoS are complex, they exhibit nonlinear response to inputs, and this must be taken into account in attempting to scale a solution:

- Response may be stepwise, so significant increases in solution scaling may be required before any response is measured.
- Response may be delayed, so there may be significant waits between possible scaling of the solution.
- Adaptivity means that gradual scaling may be accompanied by gradual adaptations that defeat the solution.
- System-of-system behavior means that much of the behavior may be unobservable, so the full impact of scaling a solution may not be knowable.
- System-of-system behavior means that portions of the system may be modified according to unexpected schedules, so a regimen of scaling changes might end up being applied to substantially different systems.

Actualization involves making a change to the CASoS. That change is likely to be irreversible: the possibility of adaptation means the CASoS changes due to fielding of the solution, and those changes may not be undone (and might spawn further changes) if the solution is removed.

Actualization involves applying changes to important systems that affect peoples' lives. There is a significant moral obligation associated with such fielding; implementers might need to observe and tune the solution for a long time before being able to leave it.

The question arises of knowing when the solution is ready for actualization. It is hoped that through testing that the proposed solution has been thoroughly tested, checked for undesirable side effects, compared against and possibly combined with a wide variety of other possible solutions. Hopefully, uncertainty has been quantified, but it might be large. Trial implementations might have been made. There may not be enough time to perform any more tests, and the tests may not be conclusive. When is it ready?

Historically, many attempts to change CASoS systems have been made; some attempts have been successful, while others have failed. Rest assured that any attempts to think broadly and to test the model provide more confidence than many past attempts have involved. At some point, the decision to field must be made. You've done your best, and you'll be there as it unfolds. Good luck.

It is important to realize that sometimes, despite our best efforts in developing a framework that follows sound, principled processes, a CASoS may be so unpredictable that every effort to understand it through modeling merely confirms that the system cannot be predicted. In such cases, where all our best principled efforts seem to be for nothing and we still have to do something, what do we do?

4.4 Summary

Just like processes for other engineering activities, the CASoS engineering process permits iteration and concurrent activity. The process is less of a formal sequence of events than it is a collection of activities that should be actively engaged in with focus on those actions that provide greatest leverage to the overall problem. It is likely that activities will not be complete; focus on adequate definition to answer questions, adequate testing to make decisions and confirm behavior.

Because we are dealing with CASoS, we know we cannot capture all interconnections, all possibilities for adaptivity, all system-of-system interactions. And we know we'll be learning along the way. As we learn about the system, our model of the system can improve, leading to increased possibilities for additional strategies. As we define new strategies, our understanding of what we need from the system improves, and our models should improve.

Focus efforts on understanding the interplay between what is, what can be and what should be, on expanding possibilities, on improving ideas and increasing the likelihood that something will work.

5 ISSUES TO KEEP IN MIND

Beyond the general process outlined above, there are a number of other aspects of CASoS engineering processes that fundamentally affect the effort. Here we identify and discuss a number of issues that have arisen through our work over the past few years in CASoS Engineering.

5.1 Hallmarks of Good Engineering Solutions

An examination of engineering disciplines reveals that the hallmarks of good engineering solutions vary across different engineering disciplines, as a function of the kinds of problems the engineering discipline addresses. As CASoS is currently in its infancy, it is difficult to predict what good engineering might mean. We review a number of engineering domains and the characteristics of good solutions to suggest possibilities. We follow with our current speculation on what might represent good CASoS engineering solutions.

5.1.1 Mechanical Engineering

Mechanical engineering addresses a range of physics effects, including kinematics, dynamics, magnetics, hydraulics, pneumatics, aerodynamics. Many mechanical systems embody multiple physics effects, so transformations between these effects are important. Many of the effects present in mechanical systems are nonlinear. Mechanical systems range in parts count from single digits to 100 thousands or millions (e.g. airplanes, space shuttles).

A hallmark of good mechanical engineering is minimal parts count, as this tends to result in greater reliability. Mechanical assemblies tend to perform multiple functions (e.g. an engine block positions moving parts relative to each other, provides a clean environment within which they interact, enables lubrication, enables cooling).

5.1.2 Electrical Engineering

Electrical engineering focuses on electrical effects, including continuum and discrete operations, linear and nonlinear effects, in assemblies ranging in parts count from single digits to multiple millions of components (e.g. VLSI).

Hallmarks of good electrical engineering include single function components, and physical separation of components to minimize crosstalk. The computer industry has benefitted greatly from scalability of designs, which is greatly facilitated by biasing components to operate in linear domains.

5.1.3 Systems Engineering

Systems engineering focuses on development of large systems, with parts counts in thousands to millions. At large scales, the particular physics captured within elements is less important than the interactions between components.

A hallmark of good systems engineering is the management of interactions between components. The only interactions permitted should be those designed into the system.

5.1.4 Software Engineering

Software engineering focuses on assemblies of algorithms and data structures. The engineered structures within software engineering are expected to be malleable, and to be continually modified throughout their lifetimes.

Hallmarks of good software engineering are clear, easily understood implementations, limited interaction among components, efficient implementations.

5.1.5 CASoS Engineering

CASoS engineering focuses on complex systems that can adapt, and themselves might be complex systems. While the count of individual elements may be high, this is not the predominant characteristic. Complexity means that the interactions among system elements are not managed or manageable. Adaptivity means the system changes, so solutions engineered for such a system need to accommodate that adaptivity. System of systems means that the system hides details, and that the component systems can change due to unanticipated causes.

Hallmarks of good CASoS solutions are solutions that can evolve and grow with the systems, accommodate complex behavior, deal with incomplete information. Solutions will become intrinsic parts of the CASoS, so it is ideal if they intrinsically interconnect the elements of the CASoS in ways that improve the robustness and resilience of the overall system. Further, since unexpected interactions are likely to be the norm, it is good for such solutions to be clearly favored by elements of the system, so that the system adapts to protect the solution (rather than adapting to, misdirecting or rejecting the solution).

5.2 Uncertainty

The sources of uncertainty in a CASoS are unprecedented. Each aspect of a CASoS (complex, adaptive, system-of-systems) produces a number of sources of uncertainty in model parameters, structure and measurement, some of which are outlined below.

Complex behavior emerges due to interactions between otherwise simple behaviors; such behavior is counter-intuitive, and can defy comprehension. It is impossible to correctly model all of the behavior present within any element of a CASoS, or to completely capture all of the potential interactions within the CASoS (consider, for example, simply the people within a CASoS: the potential behavior space and social networks involved are staggeringly large). Capturing data about the internal state of the CASoS is also hard, due both to uncertainty about what to measure as well as the difficulty of actually instrumenting such a system.

Since a CASoS is adaptive, its structure may change continuously, including changes that are a response to any attempt to measure it. This leads to both uncertainty about the model structure as well as to significant measurement uncertainty, if adaptation isn’t specifically planned for.

The interacting sub-systems that compose the CASoS can each contain hidden structure that contributes to the model uncertainty. Where portions of the system cannot be directly examined, the structure of any model of such will most certainly be uncertain.

Pragmatically, we need methods and tools that permit engineering of our aspirations in the face of such uncertainty.

Methods that focus on robust choice provide a reasonable approach to dealing with uncertainty. To the extent that the effects of structural, parameter, and data uncertainty can be quantified, it is possible to compare approaches and seek those that are maximally insensitive to uncertainty. To this end, it is wise to maintain flexibility in choice of solution, so that the robustness of solutions can be considered (along with other considerations such as cost and ease of implementation).

Tools for ensuring robustness of choice include the following:

1. Uncertainty quantification. If uncertainty of model behavior is quantified, model behavior comparisons can be informed by that uncertainty, ensuring that narrowness of uncertainty is also a criteria in comparisons (and that comparisons don't favor a mean with wide uncertainty over a slightly narrower mean with much narrower uncertainty).
2. Model exploration tools, including sensitivity analyses, parameter sweeps, phase portraits, bifurcation analyses, reachability analyses. This family of tools can be used to explore the behavior space around a given solution (with the size of the explored region dictated by the amount of uncertainty present), to determine whether system behavior might experience some undesirable change in the neighborhood of a given solution.
3. As yet unidentified tools that would search for possible structure changes (insertion or removal of interrelationships) that would fundamentally affect the robustness of a given approach. The model exploration tools previously identified assume a constant model structure, and there is a clear need for exploration that focuses on structure change (both to deal with structural uncertainty in modeling, and to address the possibility that some change to the system, due to the adaptive nature of the system or because the component subsystems can be changed independently,) might affect the robustness of a given choice.

Models provide a means of evaluating potential solutions without risking change to the real CASoS. They are useful cartoons of reality, and should not be mistaken for reality. Models need to capture the behaviors of the system that are important to the aspiration being addressed. It is important to capture those behaviors we care about, include a means of making changes to the model that reflect what we can do to affect the real system, and include necessary structure that captures what the system will do, including side-effects that might be important.

There is a tradeoff between increased model detail and cost, chance of error, ability to understand the model, and likelihood of complete coverage of the model parameter space (such coverage being important in testing, searches to ensure robust solution behavior). This tradeoff is illustrated in **Figure 6**. Higher levels of detail, while frequently thought of as improving model fidelity, can significantly decrease the likelihood of project success. The art of modeling includes finesse in choosing what to model and how, in order to maximize value when the model is used to engineer changes to CASoS.

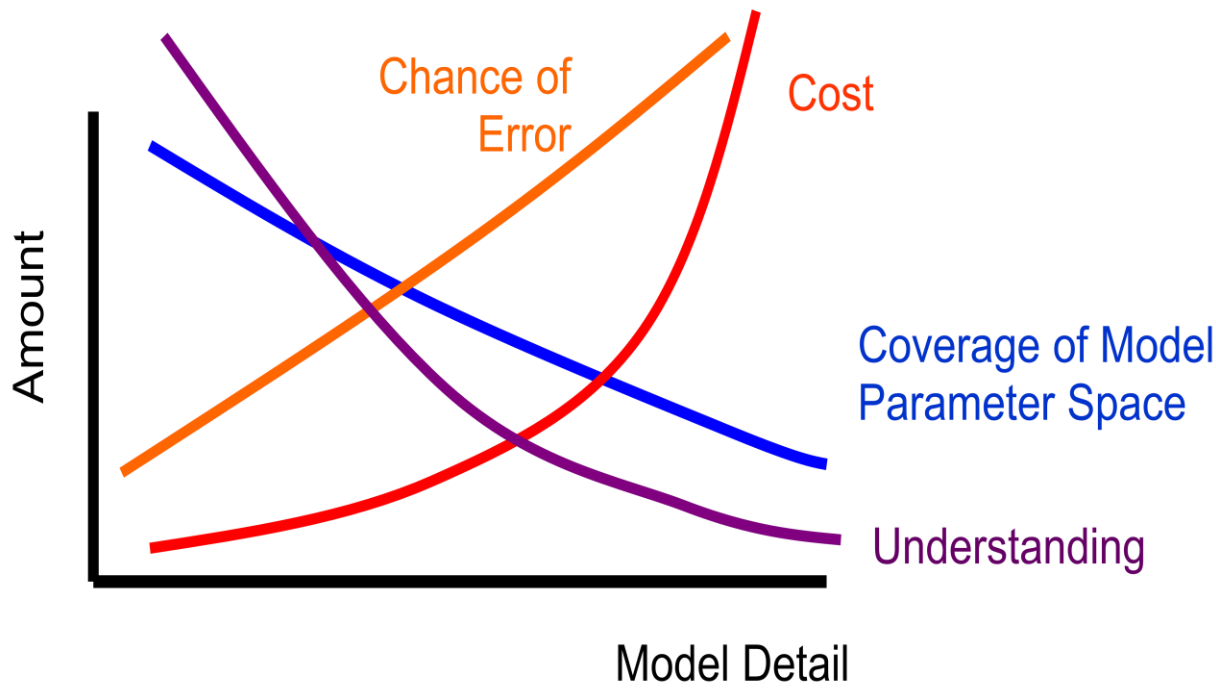

Figure 6: Relationship between various aspects of model behavior and level of model detail

Since model detail tends to be added iteratively, it makes sense to develop and test the model iteratively. **Figure 7** illustrates this iterative process. During each iteration, various potential choices are evaluated against a version of the model to measure the degree of their effects and to measure their uncertainty. The process is complete when the model possesses sufficient fidelity to provide clear distinctions among choices[23]. It is possible at each iteration to revise aspirations, the conceptual model, or the analysis, in order to adequately explore the potential solution space. It is possible to focus an iteration entirely on reducing uncertainty, possibly by more careful data acquisition, or increase / decrease in model detail.

Further discussion of model construction can be found in section 5.3.

[23] Note that, although rather unlikely, it is possible for two choices to have identical costs, benefits, and side-effects, even though they are implemented differently. The model must be sufficient to represent the differences between the choices, and capable of evaluating costs, benefits, and side effects sufficient to support decision making (not necessarily sufficient to make the decision).

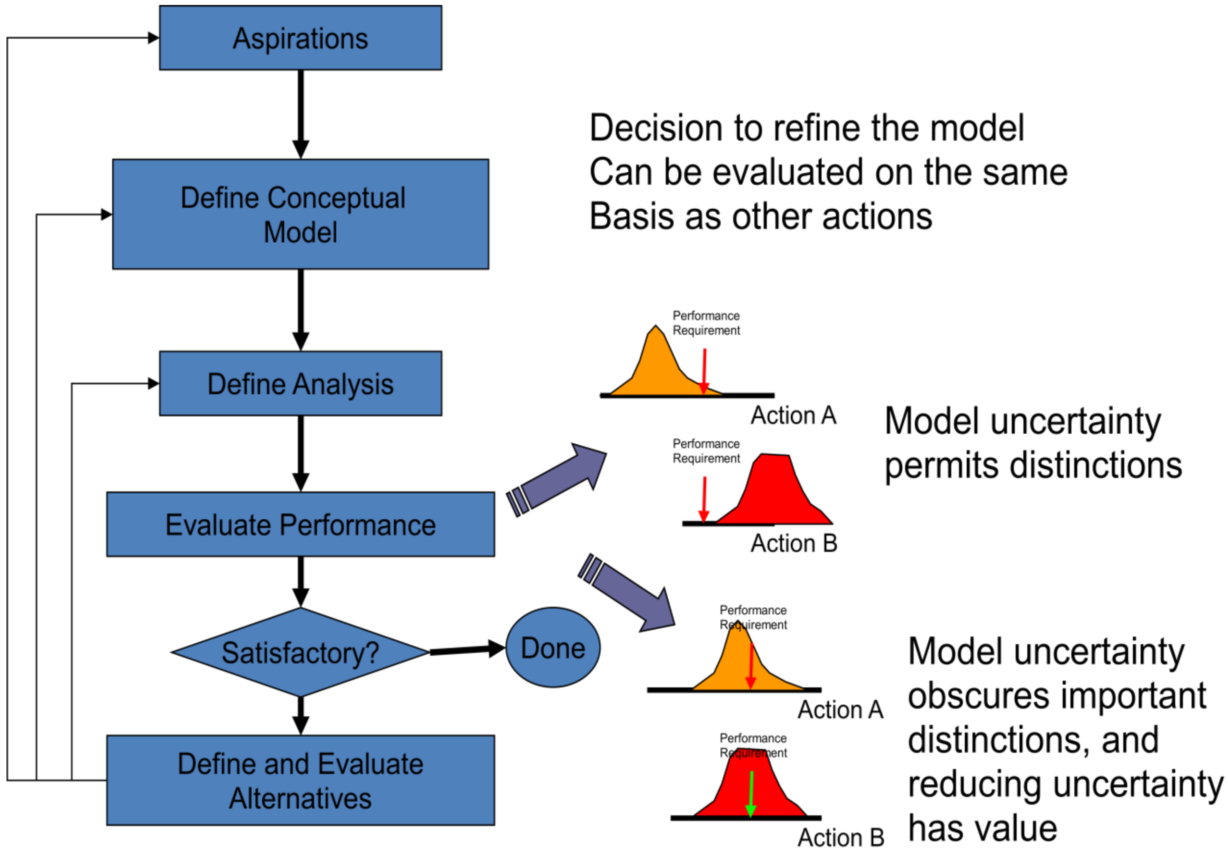

Figure 7: Process of model refinement that addresses uncertainty

5.3 Validating Solutions[24]

Before actualization will occur, models and engineered solutions will be subjected to significant scrutiny. This is particularly the case where the proposition is high risk and/or affects critical systems involving large numbers of people (the typical case with CASoS). This section addresses approaches to building trust in the engineering solutions proposed. Note that the units under scrutiny are the engineering solutions, not necessarily the models that are employed (the sensitivity of the engineering solution to model formulation might actually be quite small). Processes that treat the engineering solutions as the artifacts that need to be trusted include proving that underlying models can be trusted, but we must also find means of considering the engineering solutions, including consideration of side-effects, alternatives considered, and risks of failure.

Consideration of trust-building early in the engineering process can make the testing process easier (because artifacts can be engineered to be easy to test if planned for), and can also

[24] We avoid using the word “confidence” here, because the terms “confidence building” and “verification and validation” have loaded meanings from the modeling community.

result in better results (because more alternatives are considered if it is necessary to show that the results are better than the alternatives).

The large uncertainties associated with CASoS (see section 5.1) can certainly hinder confidence. Following are a collection of approaches that can be taken to the process of increasing trust; while the list is by no means exhaustive, these techniques have proven useful in past experience.

Reputation and political leverage are frequently used within political and commercial domains to obtain support for fielding changes to CASoS. While such an approach is clearly not good science, it is certainly a factor in building or undermining confidence for engineered products (reputation certainly affects who we trust to sell us automobiles and who we choose to govern our countries). While processes for developing and exploiting these approaches are outside of the scope of this document, we include this possibility for completeness. It is strongly recommended that other, more formal processes are undertaken – exclusive reliance on reputation and political leverage is bad practice, particularly if the solution later fails and litigation is involved.

Risk management can significantly help in improving trust. For example, piloting a CASoS change in a limited environment, or devising means of containing or minimizing bad outcomes can significantly help boost confidence before the change is fielded broadly.

Transparency of models and processes can clarify intent. It is easy to distrust complexity and obfuscation, while it is much easier to trust simplicity and openness.

Verification and Validation are formal means of illuminating the logical structure of a model, for testing it, and for clearly determining and communicating strengths and weaknesses. While some in the modeling community claim that validation is impossible (c.f. Sterman), validation is a standard engineering process that people engage in daily. The important distinction that ensures validation is possible and meaningful lies in the definition of the term: Validation means determining whether an artifact is suitable for the intended use. In engineering practice it is not necessarily required to show that a given artifact accurately reflects the real world or to prove that a solution always works, but to show the degree to which the behavior of the artifact is sufficient to support the engineering purpose proposed. Validation is thus a search for evidence that the artifact meets the need. A good choice of aspirations and a clear understanding of what the aspiration requires can significantly impact the kinds of model required, and the proof required of their adequacy. Just as focusing on engineering application helps avoid the kind of endless modeling that might be involved in "understanding" as a possible aspiration, an early focus on V&V can provide insights into how modeling efforts can be focused to address the fundamental requirements. Ensuring that the model is testable, and that those tests can be easily executed, provides significant focus, and helps to avoid superfluous model details.

Less formal notions of testing are also possible (some of this is referred to as "confidence building"). Such tests can be helpful in increasing confidence, but are less effective than defining a formal framework that links the tests to performance requirements.

5.4 Data

Obtaining true representative data for CASoS can be a general, continuing problem. A variety of concerns make data acquisition and use difficult:

1. Since the system is complex, data will appear to be noisy. Since interconnections drive the complex behavior, the overall system has to be fully understood and carefully measured in order that important sources aren't left out. Correct characterization of complex and chaotic behavior can require vastly larger datasets than similar non-complex systems (think millions of samples as a start).
2. Since the system is adaptive, its behavior changes, so data can have a very short shelf life. Further, adaptivity means that the system might change its behavior in response to being measured, making measurement a questionable endeavor.
3. The system-of-systems aspect means that significant portions of the CASoS may refuse to be measured (they are essentially black boxes, whose internals cannot necessarily be inspected). This convolves with the complex systems issue – the system refuses to allow access to measurement of the interconnected systems that would permit correct characterization of the complex behavior.
4. CASoS systems frequently are composed of subsystems that include human behavior, whose behavior can be notoriously difficult to measure. Also, many humans don't necessarily *want* their behavior to be measured.
5. Much of the data regarding CASoS systems tends to be qualitative. Attributes such as happiness are very difficult to quantify, frequently are expressed in relative terms. The social sciences can provide some help in this area, but it continues to be difficult.
6. Much of the data we measure is highly uncertain. Due to many of the concerns mentioned above (qualitative nature, uncertainty of system structure, measurement difficulty, "noise") uncertainty can be very large.
7. Many CASoS systems involve high consequence components, so experiments with such systems are not possible. As a result, the only data available might be in a very narrow region in the typical operating regime, with complete ignorance of what happens in stress circumstances.

Given the wealth of difficulties regarding data, it might seem impossible to produce reasonable models of systems. There are, however, reasonable approaches to dealing with data problems:

1. Models that are somewhat data independent (insensitive, focus on structure rather than details)
2. Processes that accommodate lack of data (e.g. sensitivity studies, bifurcation studies, etc)
3. Engineering solutions that are insensitive to lack of data (e.g. favor control over prediction).
4. Engineering solutions that permit calibration in the field
5. Analysis of trends – trends in right direction may be sufficient for engineering purpose, even if magnitudes cannot be trusted.
6. Look harder for data. Ordinary operating conditions and zero conditions (how does the system behave with *everything* disabled) generally exist.

7. Don't model with higher degrees of nonlinearity than the data supports. Don't engineer solutions that depend to heavily on the particulars of the nonlinearities of the solutions.
8. Procedural considerations – log date of collection, source, uncertainty, etc.

5.5 How to Model and for What Purpose

The most important requisite for models is that they must tell us what we *need* to hear, not what we *want* to hear.

Hearing what we want to hear produces disasters like the loss of the Space Shuttle Challenger, which exploded due to failure of an o-ring seal. NASA officials were implicated in the disaster, as they allowed the launch to continue despite concerns voiced by engineers. We have a fundamental duty, in engineering changes to CASoS, to understand the risks of changing these systems, to hear and act upon what the systems and the models tell us, so that we do the right things, not just what we want to hear.

"Essentially all models are wrong, but some are useful" [Box]. CASoS models are wrong – complexity is nearly impossible to correctly capture, adaptive behaviors will cause reality to diverge from the model, systems-of-systems hide important details and change on their own schedules. Even without these concerns, modeling is hard.

Be wary of biases: biases in the model, and biases toward particular choices of solution. If the model is being constructed entirely to convince people of decisions that have already been made, be concerned.

Our purpose is to produce useful engineering change, and any models we build must serve the purpose of comparing strategies and clarifying the behavior of the CASoS being changed. Keep models simple, to the point, and sufficiently rich to serve the engineering need. Complex models are hard to understand, hard to V&V, and can cost a huge fraction of the engineering budget.

[Miller] provides a good description of practices for computational modeling.

5.6 Spiral Model Implementation Process

It's important to remember that even a "first cut" application might be the "solution" to a problem. (e.g. Politicians, parents, and others have been addressing CASoS-like issues for a long time, often on short fuse with no computational model and very little time to make a decision.) Remember that even though a CASoS is always impossible to model completely, a crisp, insightful solution might be quite simple.

We propose the following process:

1. Think about aspirations. Decide on some very simple model that addresses a meaningfully broad set of possibilities.
2. Build a first cut model of the system, with minimal structure so we see somewhat realistic behavior. V&V this to an appropriate degree
3. Construct a first cut solution to one or more aspirations. V&V this appropriately.
4. Ask whether the model/solution has converged to solutions we believe in. Convergence might be based on:

    a. We’re out of time, and we have to go with what we have.
    b. Further loops through the process haven’t changed the answer
    c. An analysis of the model/solution shows that any fidelity-increasing changes would not change the answer for the likely magnitudes of any proposed changes.
5. Ask what model/solution fidelity changes can be made. Ask whether changes to aspirations need to be made. Ask which of these would be most important to improving the answer.
6. Judiciously increase fidelity, choose a different aspiration to address, rethink aspirations. Iterate from 1, 2 or 3 as appropriate.
7. Document what spirals were executed in the model implementation process. Consideration of which spirals might be executed next, or what was not done, can provide a wealth of information regarding practicality of the process, patterns in thought processes, patterns in the things we routinely miss.

In thinking about solutions to the world’s problems, it’s interesting to note that policy makers tend to have a library of typical solutions they use (e.g. raise/lower taxes, increase government spending, send the kid to the corner). Many of our “solutions” look, in model terms, like disruptions; e.g. change a parameter to some new value. Even so, maintaining a library of solutions (perhaps there’s a general mathematical form for them?) might be a good idea. A worry is that focusing *too much* on the *standard solutions* might stifle creativity, but most of the progress made by most engineering disciplines involves at least some codifying of typical solutions (note, e.g., standardization of fasteners, standard integrated circuits, standard linearizations used in electrical engineering, standard formulations (e.g. Newtonian physics) used across engineering).

Model fidelity is an interesting trap. Increased fidelity does not necessarily improve answers. Increased fidelity certainly increases model run time (decreasing ability to fully understand the model), decreases the likelihood that model development addresses issues at a consistent level of detail across the board, increases numerical difficulties in producing model solutions (increased fidelity can decrease numerical accuracy). It is possible that, even with profoundly increased fidelity, the solution doesn’t significantly change. Beginning with Aspirations in mind provides a means of checking this – if increased fidelity won’t change the answer, don’t pursue it.

5.7 Challenge the Process, Challenge the Results

The point of defining the CASoS Engineering Framework is to devise means of obtaining important solutions to some of the world’s greatest problems, while avoiding behaviors that dilute progress, add little real value. CASoS are big, difficult, exhibit counter-intuitive behavior, defy efforts to understand them.

While our best efforts to date have achieved interesting successes, it is important to keep in mind that our best efforts, while perhaps more successful than other efforts, may still be wrong and inadequate in the face of the world’s big problems. It is healthy to occasionally play devil’s advocate, to challenge the processes and the results, to determine whether something better could be done. We devise this framework as scaffolding upon which work can be performed more efficiently, in an attempt to capture our best; it is necessary to remember that the scaffolding might constitute a kind of intellectual prison, that we might limit our successes in an attempt to improve them.

We propose that careful critical study of processes followed, and results obtained, be performed on every CASoS activity. This is not a typical “peer review”, which runs the risk of focusing on approval with a few critical points noted. It requires honest, almost skeptical review, possibly involving parallel competing efforts, with thorough, objective comparisons, in order to be successful.

5.8 Seek Patterns

From the inception of CASoS work, we have recognized significant patterns across very different CASoS application domains. Models tend to have very comparable structures; aspirations tend to fall into very consistent categories; processes for actualization tend to be very consistent. Any solution attempted in any CASoS application area may adversely affect other related CASoS. Every CASoS may be related to all the rest (it may just be one big CASoS after all). People who have worked on a number of CASoS applications tend to be better at working on new ones than those who haven’t.

The CASoS Engineering Framework means nothing if there are not patterns in different applications. The Framework will be maximally effective if those patterns are recognized and exploited (thus, we aren’t reinventing everything each time we address a new application). Maintaining a library of CASoS applications is thus critical. Solutions spanning the space from simple first cut problems to detailed large-scale activities will all be useful in seeking the patterns in CASoS applications.

Some patterns that are emerging in current CASoS work include:

- Contagion (spread of ideas, disease, fire)
- Exchange (trading one thing for another)
- Evolution (striving for advantage in order to survive or thrive)
- Multi-level selection (evolution at one level affects evolutionary forces at other levels)

5.9 Avoid Rushing the Process

The outline of the CASoS Engineering process includes a variety of possibilities; at any step, it is possible to enumerate a number of possibilities or to keep many possibilities open rather than downselecting. Keeping possibilities open provides the promise of improved decision-making when more options are understood, but also provides for potential combinatorial explosion of possibilities that must be explored.

In the case where time and money are plentiful, many promising possibilities should be examined. CASoS embody some of the world’s biggest problems, and care should be exercised in deciding how to address them. When time and money are short, the process can be shortened, with the caution that the results are likely to be suboptimal.

Consider also, that these are complex, adaptive systems of systems. Being complex, we may be wrong in our understanding. Being adaptive, the system can change itself in response to the application of a potential solution, either rendering it ineffective or possibly misusing it, causing great harm and likely adversely impacting the credibility of people attempting to apply a solution. Being systems of systems, changes to the system may well occur in ways that are not visible from the outside, so solutions may need to be adaptable to respond to the possible behavior of the system. It is therefore wise to have a number of solutions in hand, and to understand their likely interrelationships, before applying solutions.

We currently focus much of our effort on producing models, far less effort on choice of solutions, and almost none on alternative solutions. A change of focus is necessary if we are to move to the place where we architect solutions that reflect the true attributes of a CASoS. Further, we possess no theory to quantify the importance of a broad view when defining solutions in the face of CASoS challenges. Currently, decisions about engineering solutions are likely to be made based on budget and time criticality considerations. Such an approach is guaranteed to rush the solution, and may prove suboptimal.

It would be useful to have tools that enable quick, easy exploration of a wide range of possible aspirations and solutions (possibly explore the entire solution space?). If appropriate tools exist and can be applied easily/automatically, early and with high confidence, CASoS solutions would certainly improve.

Note that the total cost of all design work is generally a very small fraction of overall implementation cost. Consider, for example, the cost of designing a bridge, which involves paper, pencil and a few expensive people thinking, vs. the cost of building the bridge, which involves steel, concrete, a large amount of labor (performed by less expensive people working). A typical figure in product design is that 10% of the cost is design, 90% implementation. Mistakes during design make the downstream figures much higher (consider the financial and political costs of being sued for being wrong, or the price of scrapping parts in order to put a hole where it *should have* gone). Most engineering processes (requirements, testing, design reviews, etc) are targeted at reducing the costs of downstream mistakes.

5.10 Shortcutting the Process

While this section seems to contradict Section 5.10, there are dire emergencies when there is no time to follow a detailed process. Here we attempt to provide the shortest process with any hope of success.

If the behavior of a CASoS must be changed, it might possible to “whack it with a stick” and obtain a behavior change. While such an approach may be nothing more than wishful thinking, there are circumstances under which such an approach actually works. The following conditions are possibilities under which the “whack it with a stick” might work:

1. The “whack” is applied to system variables; that is, variables that represent the current state of the system. Such “whacks” don’t fundamentally change the game, but change our position within the state space of possibilities. An example might be infusing cash into an economic system; how much money everybody has is a continuously changing variable, and doesn’t necessarily define the fundamental relationships within the system.
    a. The system possesses multiple “equilibria” (equilibria, stationary orbits, etc); the “whack” must be large enough to move the system from its current equilibria to a new equilibria.

   b. The system possesses a single "equilibrium" and slow time constants: the "whack" moves it far enough to achieve an objective, and the system responds slowly enough that the "whack" achieves the desired goal before the system returns to its previous equilibria.
   c. The system is sufficiently sensitive to inputs that it moves easily relative to the "whack"; repeated small "whacks" can keep it in a desired pseudo-equilibrium.
2. The "whack" is applied to system parameters; that is, variables that define the nature of the game itself. Ordinarily, these parameters are constants within the system. An example might be changing the cash rates within an economic system, or implementing a fundamentally new technology that changes how we relate to the world.
   a. For systems of single "equilibria", the sensitivity of the location of the "equilibrium" must be sufficiently large relative to the "whack" that it moves where you want it. Further, the time constant of the system must be fast enough that the system moves to the new "equilibrium" fast enough to serve the need. Also, change in the equilibrium may create some new equilibrium, and it may be in an undesired location, and easier to converge to than the desired equilibrium.
   b. For systems of multiple "equilibria", the same rules apply as for single equilibrium. Treat the multiple equilibria the same as potentially new equilibria in the single equilibrium case.

In either of these cases, it is also necessary that some aspect of the system be "whackable"; there must be variables that can be affected, in order that the system be changed.

CASoS systems are likely to be hysteretic; some difference in behavior might be produced even if the system is essentially returning to roughly the same "equilibrium".

Evidence that such an approach might be failing is that the system continues to revert to its original behavior after being "whacked". Multiple attempts to "whack" the system in the same way are unlikely to produce different results. Absent evidence that the system contains multiple equilibria and a willingness to "whack" the system harder, multiple attempts to "whack" the system in the same way are a waste of effort. Try something different.

Consider seeking aspects of the system that haven't been previously "whacked"; adaptable systems tend to adapt to being "whacked", so repeated "whacking" is not guaranteed to consistently produce desired results.

5.11 Creativity in Defining and Selecting Aspirations and Implementations

A classical exercise in learning about creativity in the engineering process involves the action of creating a shelf to hold some object at some distance above the floor. Each team is provided with a small box; within the box are an odd collection of string, thumbtacks, sticks and such. Typically, teams will look at the contents of the box, brainstorm solutions that would use all the contents of the box, and rush to build an odd contraption using sticks for a shelf, string to hold it together and to provide guy wires to attach it to light fixtures, etc. One clever solution involves dumping everything out of the box, grabbing two thumbtacks, and tacking the box to the wall as a shelf.

Overall, since building the contraption takes a lot of time while “thinking outside of the box” takes very little building time, the thinkers can take much more time in deciding what to do. They can avoid red herrings (e.g. the strings and sticks) by thinking about what the essential requirements are, looking around to see what’s really available (including the box, which most teams miss), thinking in possibly irreverent ways (e.g. putting tacks into the wall).

In our mad rush to “get it done” we have to be careful to ensure that we’re really thinking, that we’re thinking along many different lines, so that our preconceptions and biases don’t take us to profoundly suboptimal results. An engineer’s job is more than to “get it done”; it’s to think carefully about the possibilities for implementation, the possible unexpected ramifications, the effects on implementation on other implementations and on the environment.

A variety of approaches are possible, spanning the range of brainstorming, structured dialogs, comparison with previous solution processes, vetting with more experienced personnel. Time permitting, a variety of approaches should be considered, to ensure that both revolutionary and evolutionary thinking has occurred, that thinking has covered inside and outside of the box.

The entire cradle-to-grave lifecycle of a solution can have impacts that should be explored. This exploration requires significant extrapolation, much of which is, at best speculative. The world can change a lot during the lifetime of any product (e.g. global warming and peak oil were hard to imagine in the days of inventing the first automobile, when the world population was a small fraction of what it is today). Although it’s hard to imagine, ultimately the solution may well be discarded; retirement and disposal/recycling of the artifacts that implemented the solution will ultimately be somebody’s problem.

5.12 Optimization, Robustness, Resilience

A common thread in engineering design is that of optimization, the process of making a product better, in terms of some objective function, frequently expressed in terms of cost, efficiency, etc. Frequently, optimization tends to be cast in terms of constrained optimization, that is, optimization where there are bounds that limit feasible space. A common result of constrained optimization is for the resulting optimal solution to occur on a constraint boundary; such a solution is not robust, because small perturbation to the system, either due to uncertainty, manufacturing variability, etc, produces an infeasible solution. Robust design algorithms seek nearly-optimal solutions that are robust in the sense of avoiding being too-near to constraints, so that a solution is both nearly optimal and reasonably robust.

Robustness tends to have another meaning, that of being strong. A robust design might be one that is, in some sense, rigid or hardened, so as to be insensitive to external influences. For our purposes, we will treat robustness as avoiding being too-near to constraints, while using rigidity to denote strength.

Another notion we discuss is that of resilience, or ability to recover or adjust easily to change.

For purposes of designing CASoS solutions, we have yet to fully understand what the concept of optimal, robust solutions might mean, or what optimal solutions are. Note that CASoS systems are, in general, incompletely modeled, and are frequently described in terms of parameters whose values are very uncertain. Any use of optimization must be sensitive to these concerns.

To date, our thinking on the subject has produced the following notions:

1. Systemic solutions. Solutions that address the system as a whole, rather than focusing locally on portions of a CASoS, are more likely to produce systematically good solutions than those that aren't. Much of our work in devising engineering framework processes focuses on means of encouraging systemic thinking and analysis, to encourage good systemic solutions, as an attempt to achieve a measure of optimality, robustness and resilience.

   Systemic solutions need to address the following:

   A. The system is modeled as a system, so there is opportunity to measure effects relative to the system's interactions.
   B. Many aspects of the system are explored, so that improvement in some aspect of CASoS behavior isn't achieved at the expense of some other aspect (thereby shifting a problem rather than solving it).
   C. Interactions between various solutions are explored, so that interactions between solutions don't render them ineffective.
   D. Interactions between the solution and the environment are explored, so that negative impacts beyond the CASoS being influence are contained.
2. Adaptability. Since the CASoS is an adaptable entity, it may well adapt in some unexpected way to any solution that is applied to it. Solutions must encompass the adaptability of the CASoS, themselves be adaptable, or there must be someone in-the-loop as the solution is applied, in order to adjust the solution during its application, to ensure the solution is actually working as intended. Inadequacies in adaptability are most obvious during Actualization, when it might be extremely costly to retrofit adaptability into a solution. A possible fruitful area for investigating optimality would be to determine an optimal amount of adaptability in the solution (relative to cost, the degree of adaptability of the CASoS).
3. Resilience. The notion of resilience is important in dealing with CASoS solutions. Rigid, inflexible solutions, even if optimal, have no place when dealing with a system that can exhibit counterintuitive behavior and can adapt to confound solutions.
4. Optimal control. Optimal control seeks to devise paths between some system state and a desired state, via a path that is, in some sense, optimal. Such tools might well be useful in devising CASoS solutions; while it is not clear what the effect of optimizing a solution might be (perhaps resulting in lack of robustness, etc), it may well be reasonable to optimize the path of fielding such solutions. Again, presently this is unclear, but is worth consideration.

It is not clear what the effect of classical engineering optimization (including robust design) might have on CASoS solutions. It may well be too early to devise optimal solutions; indeed, for time-critical problems, it is likely that satisfying solutions (solutions that meet the need, independent of any concern for optimality) might be the only possible solutions.

5.13 Addressing Hot Topics

CASoS topics frequently address issues that people have strong opinions about. Some current items include global warming, peak oil, terrorism, choices in economic theory. Such topics can cause strong, possibly violent, reactions, especially regarding various assumptions about behaviors that are embedded in models.

You have a choice in how to approach this, and choices can escalate or reduce tension, depending on the reaction you want to elicit. It is easy to increase tension, more difficult to reduce it.

Approaches that increase tension include:

1. Describing the model in terms of only one side of an issue.
2. Skewing model content towards one side of an issue.
3. Giving the appearance of having something to hide. Hiding model structure, failing to explain the modeling choices that result in one side or the other being favored.

Approaches that can decrease tension include:

1. Describing a model in more general terms, suggesting that it might be useful for more than one application. Show what the model does and how, then show its application to specific hot topics. This can encourage fair-minded evaluation of a model's structure independent of prejudices that can be elicited when the model is labeled with hot topic issues.
2. Creating model structures that capture multiple aspects of the problem, showing how the model results can be skewed towards any particular view, and then showing what results occur when a balance is considered. Also possible is to show how far modeling assumptions have to be pushed to produce a different answer.
3. Seek the high ground and objectivity in modeling, and avoid embedding political preferences wherever possible, permitting balancing between extremes if avoidance is not practical. This is difficult, requires discipline, and can be helped by creating a modeling team that captures a variety of disparate preferences.

It is possible for outside influences to attempt to exercise inappropriate influence over a model, to attempt to ensure that the model produces their specific point of view. Transparency in modeling constructs can help avoid some of this, as can wide, free dissemination of models, so that attempts to adjust the model are obvious to many, can be helpful. We don't possess sufficient approaches at this time to prevent such problems, however.

5.14 Varying Scales

Attention to variation in scales across a system can fundamentally affect the form that solutions take. For instance, events that happen in millisecond time scales can be separated from events that require years to occur – the slow system can be treated as a constant for purposes of understanding the fast system. Such separation not only affects modeling, but affects engineering solutions – the fundamental differences in scale mean that engineered solutions can also address the different scales separately, effectively treating one component or another as being outside of scope relative to development of the solution.

Models of systems can span a variety of scales, spatial, temporal, organizational, and otherwise. Strong variations in scale can result in models that are difficult to solve numerically and exhibit very long run times. Separation of different scales can be a useful approach, where applicable.

5.15 Concurrent Development

Concurrent engineering [c.f. Hartley] is an engineering discipline with potential applicability to CASoS Engineering. Concurrent engineering is the discipline of involving various disciplines early in a product lifecycle. For example, for an engineered product such as a cellular phone, concurrent engineering would involve users, analysts, manufacturers, field engineers, and any other appropriate stakeholders from the earliest conceptual design phases. Seeking the input of a broad spectrum of stakeholders provides an opportunity to make changes to the design early, and ensuring that the design focuses on the needs of a broad set of individuals, long before expensive commitments are made. The discipline parallelizes an otherwise serial design process, securing early feedback when it can be most effective.

Concurrent Engineering notions are present in the current framework, in the form of encouragements to seek input from a broad range of sources early in the process of understanding the CASoS and defining aspirations. As we learn more about the CASoS engineering process, we may find significantly more opportunities to include input from various sources, including people who will actualize the change, people potentially impacted by changes, and other vested interests.

In a real sense, the current framework represents a kind of concurrent engineering process. We develop goals and subgoals in parallel with the development of our understanding of the CASoS itself. Even through actualization there is the possibility that further lessons are learned about the structure of the CASoS, and design changes are made to accommodate these lessons. Design, modeling and actualization all occur simultaneously, in parallel.

Further study of Concurrent Engineering and its applicability to the CASoS Engineering Framework is warranted.

5.16 Product Lifecycle Considerations

Product Lifecycle Management [c.f. Saaksvuori] is a discipline for managing products through a series of stages, including introduction, growth, maturity, and decline. Specific planning is performed for each stage in the product’s lifecycle, to ensure that appropriate actions are taken as a function of changes in how the product is used.

During introduction, focus is on product branding, pricing to achieve market penetration or high skim to recover development costs, distribution is selective, and promotion is aimed at innovators and early adopters. During growth, product quality is maintained and features may be added to increase market; pricing is maintained while demand increases with little contribution, distribution channels are grown, and promotion is aimed at a broader audience. During maturity, competition grows sales slow. Maturity focuses on different activities: features may be enhanced to differentiate the product, pricing may be lower, distribution is intensive, and promotion focuses on product differentiation. During decline, the product may be maintained, harvested (continued to be offered to the core market), discontinued or divested to another firm.

Relative to CASoS, a PLM view might be useful with respect to two important lifecycles: the lifecycle of the CASoS, and the lifecycle of an engineered change to a CASoS.

The lifecycle of a CASoS is frequently regarded as unbounded, but there are certainly exceptions. Certain infrastructures, such as the petroleum infrastructure, might have a bounded lifetime. The lifetime of the infrastructure might be limited by the availability of resources that enable use of the infrastructure (e.g. fossil fuels) or by the availability of preferred alternatives (e.g. fusion energy). Certainly a lifecycle model might be helpful in considering aspirations for the CASoS. During growth, the CASoS would require nurturing and guidance; during decline, the CASoS might require graceful degradation or graceful transition to alternatives.

The lifecycle of an engineered change might also benefit from a PLM perspective. Issues related to development of the engineered change and acceptance during its growth are significantly different than issues related to maturity (e.g. maintenance) and decline (lack of need, failure to be continually effective). Further, PLM addresses product mix -- how to simultaneously manage multiple products, each at a potentially different point in its lifecycle, and the interactions among such products in the marketplace. Such a perspective might prove useful as we seek to understand strategies involving multiple interacting aspirations, subgoals, and actions.

Consideration of PLM issues might certainly produce a useful suite of tools, methods, perspectives that would help in addressing lifecycle issues. Specifically, since others have developed a clear understanding of effective ways of addressing changes as an engineered device ages, we should learn to take advantage of such results. Even simple processes, such as recycling, are much easier to implement if the needs for such processes (e.g. easily recycled materials and assemblies) have been considered early in the engineering process.

5.17 Thinking Broadly

We have attempted, in devising the process above, to encourage broad thinking about the CASoS. First, think about aspects of the CASoS that make it a CASoS. Then think about possible changes to the CASoS. Then think about interactions among the various aspects. Build a simple model regarding interactions, and think about those interactions, before moving on to any detailed modeling, or detailed consideration of the various possibilities.

It is not clear that we have been successful, yet, in encouraging broad thinking. If engineering is truly omnidisciplinary and CASoS are some of the broadest kinds of systems, we need to find ways to ensure that the many disciplines have been considered. Devices such as Albright’s systems framework [Albright], which address many kinds of aspects of a system, are a means of addressing the system broadly. Whether such approaches should be used in initial problem framework, or left to subsequent detailed phases, is unclear; what is clear is that use of some such approach can provide a simple catechism-like approach to ensuring that many kinds of input have been solicited, and are present in the representation.

It is also clear that activities such as HOQ can grow rapidly as the representation grows. Encouraging wide-ranging consideration of aspects of the problem will tend to grow the number of issues quickly, possibly resulting in very large HOQ representations (with consequent potential for quadratic growth of issues to be addressed). Providing a means for obtaining the benefits from such activities, without excessive effort is a subject for further R&D.

5.18 Simplicity

Einstein is quoted as saying “make everything as simple as possible, but not simpler.”

A fundamental hallmark of good science and good engineering is simplicity. Simple solutions tend to be robust, easily understood and implemented, easily generalized to larger contexts.

Just because the system is CASoS doesn’t mean that the solution is complex, adaptive, or a system of systems. It might be something much simpler. Advertisers, the media, the military and others use a variety of simple techniques to elicit desired behaviors. For example, protecting an asset might require imagining every potential threat, creating specific approaches to deal with each threat, and implementing them all. A possibly simpler approach involves devising a general solution for protection that creates a barrier with very limited access, limiting the directions from which threats can come and the forms they can take. Controlling access redefines the problem from one in which any threat can be any size and take any form, and changes it to something more easily managed. Of course, there may some threat that is so large that no barrier works. We simply devise the barrier to be proof against manageable threats; unmanageably large threats are so bad that there’s no point in worrying about protecting the asset any more.

5.19 Fun

While it may seem curious, perhaps flippant, to include such a subject in a serious work, we are quite serious about the importance of fun to the process. Engineering is a creative pursuit. It should involve a significant component of joyful, playful, intense manipulation of possibilities. Frequently the most important successes can be found in the subtleties of the problem space, which are not likely to be explored if the process is dull and dreary. If fun is missing from your engineering processes, stop and fix your process.

6 REFLEXIVELY APPLYING THE FRAMEWORK TO ITSELF

In the CASoS Roadmap we identified the opportunity to reflexively apply the CASoS engineering to the CASoS engineering process. Here we interpret the engineering framework a CASoS and define aspirations.

6.1 The Engineering Framework as a CASoS

We present the CASoS catechism, noting how each aspect applies to the CASoS Engineering Framework.

6.1.1 System

The engineering framework consists of a notional theory and practice of how to engineer changes to a CASoS, which is derived from theory and practice in complex systems, adaptive systems, and engineering (particularly systems and system-of-systems engineering). The theory is mapped to a series of processes. Each process increases understanding of the CASoS or creates or increases fidelity in the design of processes intended to improve the behavior of the CASoS. The system also includes the people and processes who add to the theory and practice base, and those who map theory to CASoS practice.

6.1.2 Environment

The environment for the framework includes the Engineering Environment (which embodies tools to implement processes in the framework), the CASoS Engineers who will use the theory and the Engineering Environment to devise change to CASoS, the CASoS itself, entities affected by the CASoS or changes to the CASoS.

6.1.3 System of Systems

The components of the engineering framework can be viewed as a system of systems. The underlying theories that support CASoS are all being developed independently, based on available researchers, funding, and need. The drivers for improving engineering practice are much more application-oriented, while the drivers for improving complex systems practice are more research-oriented, simply due to the relative maturity of the disciplines and the needs of their practitioners.

The people developing various theories and practice generally work for a very diverse set of organizations, across academia, industry, and government. While significant sharing of information occurs across the organizations, some of the advancements in the field are held proprietary, or are slowly shared simply because of the organizational distances between practitioners.

6.1.4 Complex

Additions to the theory and practice base tend to happen locally, within specific organizations, and are then shared to others in rather discrete packets. Adoption of new theory and practice requires propagation through social networks; that propagation is very similar to spread of fads or diseases through social networks. Since adoption is a rather discrete operation and occurs across significant numbers of interconnections, the spread of theory is expected to exhibit complex behavior.

6.1.5 Adaptive

All changes to theory are adaptations due to external stimuli – conditions ranging from economic competitiveness, the need to address larger problem spaces, desires to be more focused on customer needs. The process of devising the framework will have to adapt both to variability in funding, varying rates of development of underlying theory, and variations in ability to field the framework in the real world. The framework itself and the process of developing the framework will have to adapt.

6.2 Aspirations

1. We aspire to be able to control CASoS problems.
    a. We aspire to develop an engineering framework that provides an effective structure for engineering changes to CASoS. That framework will blend theory and practice from the following areas:
        i. complex adaptive systems
        ii. systems and system of systems engineering
2. We aspire to influence (CASoS control) a sufficiently large community that CASoS engineering becomes standard practice, and that it provides capability for a critical mass of practitioners to solve real world problems.
    a. Develop and field first to Phoenix working group
    b. Field engineering framework to larger community as it develops

7 NEXT STEPS

We will apply the Framework to problems of interest. We will iterate with the writing above (general Framework, process of applying the Framework, etc.) as we learn, especially in the Designing and Testing and the Actualization phases (as we use our active problems as test beds for how to do this), add new "things to keep in mind". The definition of the CASoS Engineering Framework is a living process that itself is a CASoS.

We have multiple purposes for our applications:

- Test the Framework
- Identify modifications that we need to make to Framework (and make them).
- Identify requirements to be met by the cyber workbench
- Train others in its use and build team
- Critically review the process, to ensure that the framework is a scaffolding rather than a prison.
- Seed a library of past CASoS activities to enable pattern identification across various applications.

We are in the process of documenting the process of learning and implementing first cut applications. We expect that every new person who is added to the team would be responsible for doing one of these and each new project will do one of these first cut applications as part of the process of learning/growing CASoS Engineering.

REFERENCES

AECPD, *Canons of ethics for engineers*, American Engineers' Council for Professional Development, 1947.

Albright, David, private communications regarding a systems framework, April 2008.

Beyeler, W., et al, Developing Conceptual Models for Complex Adaptive Systems of Systems (CASoS), SAND report under development.

Bloch, A., *Murphy's Law, and Other Reasons Why Things Go WRONG*, Price Stern Sloan, 1977.

Box, George E. P., Norman R. Draper, *Empirical Model-Building and Response Surfaces*, Wiley, 1987.

CASK05, http://en.wikipedia.org/wiki/File:A1_House_of_Quality.png, accessed October 1, 2009.

Colbaugh, Rich, private communication, 2011.

Ficalora, J.P., *Quality Function Deployment and Six Sigma, Second Edition: A QFD Handbook (2nd Edition)*, Prentice Hall, 2009.

Glass, R., Ames, A., et al, A General Engineering Framework for the Definition, Design, Testing and Actualization of Solutions within Complex Adaptive Systems of Systems (CASoS) with Application to the Global Energy System (GES), SAND2008-7952, December 2008

Hartley, J. and S. Okomoto, *Concurrent Engineering: Shortening Lead Times, Raising Quality, and Lowering Costs*, Productivity Press, 1998.

Hazelrigg, G. A., *Systems Engineering: An Approach to Information-Based Design*, Prentice Hall, 1996.

Ioannou, *Modeling and Control of Complex Systems*, CRC Press, 2007.

Kelly, T., *The Art of Innovation: Lessons in Creativity from IDEO, America's Leading Design Firm*, Doubleday, 2001.

Kiss, I., et al, "Engineering Complex Dynamical Structures: Sequential Patterns and Desynchronization", *Science*, 2007.

Jamshidi, M., *System of Systems Engineering*, Wiley, 2008.

Kossiakoff, A. and W.N. Sweet, *Systems Engineering Principles and Practice*, 2002.

Miller, J.H. and S.E. Page, *Complex Adaptive Systems*, Princeton University Press, 2007, pp. 245-254.

Pahl, G. and W. Beitz, *Engineering Design: A Systematic Approach*, Springer Verlag, 1996.

Peters, R., *A Critique for Ecology*, Cambridge University Press, 1991.

Saaksvuori, A., and A. Immonen, *Product Lifecycle Management*, Springer, 2008.

Scholl, E., ed, Schuster, H., *Handbook of Chaos Control*, Wiley, 2007.

Simon, H.A., *The Sciences of the Artificial*, MIT Press, 1996.

Tabuada, P., *Verification and Control of Hybrid Systems: A Symbolic Approach*, Springer, 2009.

Yao, L.-S., "Computed Chaos or Numerical Errors", arXiv:nlin/0506045v7 [nlin.CD], 2010.

APPENDIX A: METHODS, THEORIES, AND FIELDS OF CONTRIBUTION FOR APPLICATION WITHIN THE CASOS ENGINEERING FRAMEWORK

Methods, theories, and fields of contribution can be organized into a series of categories. The following, while far from complete, exemplifies such categorization:

System Definition and broad conceptual modeling:

Components and influence charting

Analogy

Dimensional analysis and similitude (similarity theory)

Measurement definition

Definition of Environment boundaries

Aggregate methods and theories from which basic results may be taken and applied:

Thermodynamics and Statistical Mechanics

Non-equilibrium thermodynamics

Percolation

Networks

Game theory

Complex adaptive systems

Trading and markets

Detailed modeling and simulation approaches:

System dynamics

Agent-based modeling

Discrete-event system modeling

Markov modeling

Cellular automata

State machines

System optimization and control

Discrete and continuum linear and nonlinear optimization

Hybrid, adaptive and optimal control

Engineered Collectives (e.g. decentralized control)

Agent-based optimization

Extended and ensemble Kalman filters

Evaluation and Discovery:

Uncertainty quantification

Sensitivity analysis

Experimental design

Data mining

APPENDIX B: QUESTIONS AND SHORT ANSWERS FOR CASOS QUIZZES

We have compiled a list of questions people tend to ask or should be asking about CASoS along with some answers.

What is a CASoS?

> A complex, adaptive system of systems. They exhibit unexpected behaviors due to the complexity of their interconnections, they adapt, and they're constructed of components sufficiently complex and difficult that each component can be considered a system in its own right. Systems involving multiple humans are generally CASoS.

Why are you working on CASoS rather than something else?

> All the world's biggest issues (e.g. wars, energy, poverty, hunger, nukes, …) are CASoS problems. We as a lab owe it to the world to take a shot at this.
>
> The most challenging problems of the coming decades will involve CASoS. In many ways this shift from more traditional engineering problems is due to increasingly pervasive and global scale of production and distribution networks in the areas of food, energy, water, labor, finance, etc. Expansion of these networks to the global scale along with increasing resource consumption has removed excess capacity from many interconnected distribution networks.

Why do you believe that CASoS systems are hard?

> They are big, complicated, counterintuitive, hard to measure, hard to modify because they adapt. They are generally very important, so they're hard to experiment with (expense, difficulty in making changes to them, difficulty in measuring results, ethics). System and environment definition are often not clear-cut. The Adaptive nature of CASoS includes emergent and transient definition of systems and relationships between them and their environment.

Why do you believe that CASoS systems are sufficiently easy that you can do something with them? Why do you believe your process for dealing with CASoS systems is sufficient?

> We've been working with some (e.g. Pandemic Influenza). We've seen some interesting successes, frequently requiring a fresh look at the problem. It seems to take more than a first-order naive model of them. A systems view is generally necessary, and it requires thinking about multiple potential questions, multiple potential answers, and examining the interrelationships between them. Curiously, when we've constructed the "right" model, it often requires simpler models with easier-to-verify parameters than the typical models being constructed for them.

What is your CASoS process?

> We're still working on it, but so far we've done well with broad, high-level thinking, combined with simulation that emphasizes the network nature of the system. We're leveraging traditional systems engineering processes (requirements, conceptual design, detailed design, test, manufacture). We're leveraging complexity theory, so

we're looking at ways of separating forests from trees, and ways of measuring when the complexity can be large. We understand that systems adapt, so we're looking at solutions that change the system in ways that ensure that the solutions grow with the adaptations.

How is CASoS Engineering different than SoSE (system of systems engineering)?

SoSE focuses on the structure of the system, and so SoSE doesn't have to be complex or adaptive. Generally the concerns in SoSE are that the systems are evolving concurrently, are hard to manage, cannot be changed (unlike components which can be altered to make the system work). CASoS adds to this the notions that the system is complex (counterintuitive behavior a function of the interactions), and adaptive (the systems and the connections between systems change to adapt to changes in the environment, so solutions don't keep working by themselves).

How do you deal with the fact that CASoS systems adapt? Doesn't this mean that no solution will ever be adequate?

It means that solutions have to also be adaptive, to change with the system as the system changes. It means that point solutions are probably not adequate, and that it's a good idea to have multiple solutions in your pocket before you go trying to change a CASoS. Same thing as walking into a class of 3-year-old children. They adapt too. Don't turn your back on them, have a wide ranging game plan, be prepared to adapt your solutions to the way the system works, and it can be a lot of fun.

Are you just making all this up?

Well, yes and no. There isn't a lot of theory out there, so we're collecting what we can, trying it, adapting it to fit, and continuing forward. Such has been done before in any number of fields - when you're trying something new, you sometimes need new tools.

But, we have been working on some of these kinds of problems for some time. We know the feel of effective solutions, and know something about how to obtain them. We're not going in empty-handed.

What are your tools for working on this?

Our tools and approaches include complexity theory, systems engineering, system-of-system engineering; modeling and simulation; conceptual design; systems thinking; combinatorial consideration of interactions among system elements (random juxtaposition?); structured ways of thinking about the problem; broad unstructured ways of thinking about the problem; questioning the wisdom of previous attempts; paying attention to the underlying assumptions of previous solutions that might have at least partially worked.

Why do you believe various CASoS systems are comparable and deeply similar to each other?

1. We've seen the same thing in classical systems engineering (compare electrical, hydraulic, spring-mass-dashpot, etc systems and you find identical underlying mathematics).

2. Fractal behaviors are an important part of complex systems. Fractals are self-similar. We expect to see similarities across domains, as well as similarities between various levels of the same problem.
3. In writing the CASoS roadmap, we discovered significant similarities across domains, both in their structure, in the kinds of models one might build, in the kinds of issues one might address in trying to solve something. Glass saw similarity between pandemic, wildfire, and the need to spread ideas. Both the technical solution and the political solution to the problem had structural similarities to putting out a fire (or starting one).

Isn't it all just one big CASoS?

Sure, constructed of systems which themselves are CASoS, and so forth. Great fleas have lesser fleas, upon their backs to bite 'em, and the little fleas have lesser fleas, and so ad infinitum.

If it is just all one big CASoS, where do you start?

Almost anywhere that's convenient. Initial thinking about any problem is cheap; it's the details where things get expensive. So think hard about the issues, think hard about the potential solutions, think hard about how to change the world and achieve big wins -- it's a means of seeing many avenues to work on, and is relatively cheap compared to going down wrong, narrowly considered paths.

What are the ethics of changing the question you're asking to match the answers you think you can find, rather than just solving the problems?

The issue is in seeing the problems for what they really are. You might find that some of the biggest problems aren't what you thought they were, but you have to see the big picture. Take any big problem, and you might conclude that it's bigger than it really is (mountains out of mole hills), or that it's only a problem because people insist that the solution benefit them in selfish ways.

How is this different from CAS work going on at places like Santa Fe Institute?

Places like SFI are studying complex adaptive systems. We're interested in engineering solutions for such systems. We are *very* interested in SFI's results, but we're also interested in solving problems. Recall that a stronglink/weaklink concept permits nuclear weapons to address a wide variety of potential threats in a very simple, robust way. Maybe we can find similar levers to address CASoS issues -- it's possible that the solutions can be very simple, but we have to think well about the problem to make sure we're really solving real issues.

So why can't we apply typical engineering approaches to these problems? What makes them so special/hard? Why do we need a big project in this area?

Problems that arise from CASoS are, by definition, complex and are not well predicted or controlled by traditional engineering techniques. Here is a simple example problem to help make the case. Water security work at Sandia has resulted in algorithms and software to predict the next measurement of water quality based on several hundred recent measurements of that water quality. Water quality is a function of the characteristics of the water source and the treatment, amount of mixing with other waters and the flowpath it takes to reach the sensor. None of these things change drastically over time, treatment, mixing and aging are diffusive processes and the algorithms work quite well. Given that the stock market is also a time series, why do we still work at Sandia? Changes in the value of the stock market are one output of a globally connected CASoS and are considerably different than those that change water quality. Today, social-political-technical information that impacts the stock market and other complex systems moves at the speed of light (literally, across fiber optic cables) and combines with other information at multiple time scales in a non-diffusive manner and impacts the stock market (system of interest) in drastically different ways than we could possibly predict with traditional engineering tools.

DISTRIBUTION

1	MS1209	Larson, J.A.	90
1	MS0131	Drayer, D.D.	163
1	MS0166	Brown, L.E.	165
1	MS0145	Wall, F.D.	210
1	MS0431	Vinson, J.B.	230
1	MS0421	Skocypec, R.D.	240
1	MS0116	Hingorani, S.L.	250
1	MS0421	Cox, A.B.	280
1	MS0127	Woodall, T.	430
1	MS0351	Rottler, J.S.	1000
1	MS0123	Chavez, D. LDRD Office	1011
1	MS1427	Barbour, C	1100
1	MS1421	Tsao, J.Y.	1120
1	MS1315	Shinn, N.D.	1130
1	MS1315	Nelson, J.S.	1131
1	MS0321	Leland, R.W.	1400
1	MS1322	Dosahjh, S.S.	1420
1	MS1319	Ang, J.A.	1422
1	MS1319	Brightwell, R.B.	1423
1	MS1322	Aidun, J.B.	1425
1	MS1318	Hoekstra. R.J.	1426
1	MS1318	Hendrickson, B.A.	1440
1	MS1318	Stewart, J.R.	1441
1	MS1320	Collis, S.S.	1442
1	MS0378	Strack, O.E.	1443
1	MS1321	Summers, R.M.	1444
1	MS0316	Hutchinson, S.A.	1445
1	MS0321	Mitchiner, J.L.	1460
1	MS1323	Rogers, D.H.	1461
1	MS1188	Glickman, M.R.	1462
1	MS1188	Lakkaraju, K.	1462
1	MS1327	Wagner, J.S.	1462
1	MS1327	Bennett, P.C.	1463
1	MS1327	Raybourn, E.M.	1463
1	MS1327	Speed, A.E.	1463
1	MS1327	Hart, W.E.	1464
1	MS1326	Backus, G.A.	1465
1	MS1326	Boslough, M.B.	1465
1	MS1316	Rintoul, M.D.	1465
1	MS0110	Johannes, J.E.	1520

1	MS1078	Hermina, W.L.	1710
1	MS1411	Sun, A.C.	1814
1	MS0351	Croessmann, C.D.	1910
1	MS0359	Westrich, H.R.	1911
1	MS1137	Jones, K.A.	1932
1	MS0457	Sleefe, G.E.	2200
1	MS1454	Pfeifle, T.W.	2556
1	MS0968	Linebarger, J.M.	2661
1	MS0640	Carson, T.L.	2994
1	MS0725	Gutierrez, S.M.	4100
1	MS1231	Thornton, A.L.	5220
1	MS1221	Langheim, G.R.	5230
1	MS1164	Keese, D.L.	5400
1	MS1157	Yourick, P.D.	5430
1	MS1221	Peery, J. S.	5600
1	MS1328	Cook, B.K.	5624
10	MS1027	Ames, A.L.	5635
1	MS1027	Colbaugh, R.	5635
1	MS1027	Johnson, C.M.	5635
1	MS0620	Horton, R.D.	5640
1	MS0936	Scholand, A.J.	5741
1	MS0724	Hruby, J.M.	6000
1	MS0721	Tatro, M.L.	6100
1	MS1104	Zayas, J.R.	6102
1	MS1104	Robinett, R.D.	6110
1	MS1108	Carlson, J.J.	6111
1	MS1108	Finley, R. E.	6111
1	MS1033	Hanley, C.J.	6112
1	MS1140	Guttromson, R.	6113
1	MS1440	Schoenwald, D.A.	6113
1	MS1188	Nanco, A.S.	6114
1	MS1104	Torres, J.J.	6120
1	MS1124	Minster, D.G.	6121
1	MS1124	Laird, D.L.	6122
1	MS1124	Wilson, D.G.	6122
1	MS0734	Martino, A.	6124
1	MS1188	Griffith, R.O.	6130
1	MS1137	Detry, R.	6131
1	MS1138	Finley, P.D.	6131
1	MS1138	Jones, B.S.	6131

1	MS1138	Jones, D.A.	6131
20	MS1138	Glass, R.J.	6132
1	MS1138	Kleban, S.D.	6132
1	MS1137	Mitchell, M.	6132
1	MS1138	Mitchell, R.	6132
1	MS1138	Moore, T.W.	6132
1	MS1137	Starks, S	6132
1	MS1138	Verzi, S.J.	6132
1	MS1188	Thompson, B.M.	6133
1	MS1004	Skroch, M.J.	6134
1	MS0771	Orrell, S.A.	6200
1	MS1002	Corbett, D.W.	6500
1	MS0781	Nelson, J.E.	6520
1	MS0781	Richardson, C.B.	6525
1	MS0769	Moya, R.W.	6600
1	MS0757	Jordan, S.E.	6612
1	MS0734	Kelley, J.B.	6632
1	MS1375	Wilson, R.K.	6800
1	MS1377	Webb, E.K.	6814
1	MS0701	Walck, M.C.	6900
1	MS0735	Merson, J.A.	6910
1	MS0706	Borns, D.J.	6912
1	MS0751	McKenna, S.A.	6912
1	MS0750	Halloran, A.R.	6913
1	MS0734	Ivey, M.D.	6913
1	MS0754	Rigali, M.J.	6915
1	MS1033	Blankenship, D.A.	6916
1	MS1138	Ammerlahn, H.R.	6920
1	MS1138	Garcia, P.	6920
1	MS1138	Brodsky, N.S.	6921
1	MS1138	McCornack, M.T.	6923
1	MS1138	Beyeler, W.E.	6924
1	MS1138	Brown, T.J.	6924
1	MS1138	Conrad, S.H.	6924
1	MS1138	Corbet, T.F.	6924
1	MS1137	Ehlen, M.A.	6924
1	MS1138	Kaplan, P.G.	6924
1	MS1137	Kelic, A.	6924
1	MS1138	Parrott, L.K.	6924
1	MS1138	Pless, D.J.	6924
1	MS1138	Horschel, D.S.	6925

1	MS1138	Kuzio, S.P.	6926
1	MS1137	Malczynski, L.A.	6926
1	MS1137	Passell, H.D.	6926
1	MS1137	Tidwell, V.C.	6926
1	MS9001	Stulen, R.H.	8000
1	MS0124	Hwang, R.Q.	8004
1	MS9004	Davies, P.B.	8100
1	MS9003	Wu, J.	8112
1	MS9054	Carling, R.W.	8300
1	MS0734	Downes, P.S.	8360
1	MS9054	Pontau, A.E.	8360
1	MS9151	Napolitano, L.M.	8900
1	MS9159	Mayo, J.	8953
1	MS9159	Ray, J.	8954
1	MA9151	Hutchinson, R.L.	8960
1	MS9158	Armstrong, R.C.	8961
1	MS9158	Vanderveen, K.B.	8961
1	MS0805	Cook, W.R.	9530
1	MS0899	RIM-Reports Management	9532 (electronic copy)

Sandia National Laboratories