Volume 3 - Number 2 June 2005
SEPO
COLLABORATIONS
Inside Knowledge for MITRE's Systems Engineering Community mitre.org/work/sepo/

Perspectives on Complex-System Engineering

What is a complex system, and what makes it behave the way it does? As systems become increasingly large and must seamlessly interoperate with other systems in ways that were never envisioned, system engineers are bumping into the limits of the tenets, principles, and practices traditionally used in systems engineering.

Scale is an important consideration in the topic of complex systems. An entity on one scale (e.g., a honeybee) is not the same thing on another scale (e.g., a beehive). A hive, although made up of thousands of bees, behaves in ways different from just a collection of thousands of individual bees. The hive will produce honey and wax, whereas a bag of honeybees will simply be a bag of angry honeybees.

What Is Complexity?

The only consensus on what makes something complex is that the definition of complexity is evolving.

Many terms—complex systems, system of systems, megasystems, and enterprise systems—suggest complex systems. But how people apply such terms can vary widely, making it difficult for the rest of us to zero in on the essence of complex systems. It should come as a relief to know that the experts don't always agree either.

A fuzzy concept. "Complexity is a fuzzy concept," says Lashon Booker, principal engineer for the Center for Washington Command, Control, and Communications (WC3). "There are prototypical examples of complex systems, such as economies and social systems; however, trying to draw boundaries and determining that everything inside is complex and everything out isn't, I think, misses the point. Complex systems typically have some characteristic properties, but the extent to which a particular system exhibits any given property can vary."

A matter of number. "What makes a thing complex is the number of decisions that have to be made regarding its design, the number of people or organizations that have to be involved in those decisions, and the fact that they're probably inconsistent," says Judith Clapp, Center of Air Force Command and Control Systems. "Complex systems have inconsistent objectives, so decisions have to be negotiated. People have tried to make the terms complex systems, system of systems, megasystems, and enterprise systems refer to different things; it's not clear that everybody agrees about that yet."

Degree of control. An important consideration is whether a system, such as an enterprise or a system of systems, is planned for by some unified process or independently developed and later merged. Another factor is a matter of control. "If the system is a set of independent systems, and you have no control over those decisions, then, maybe it's a different kind of system. But again, I don't see a clear-cut distinction between whether an enterprise system or a system of systems involves more or less of this independent decision making. I think we're still trying to come up with useful distinctions," Clapp continues.

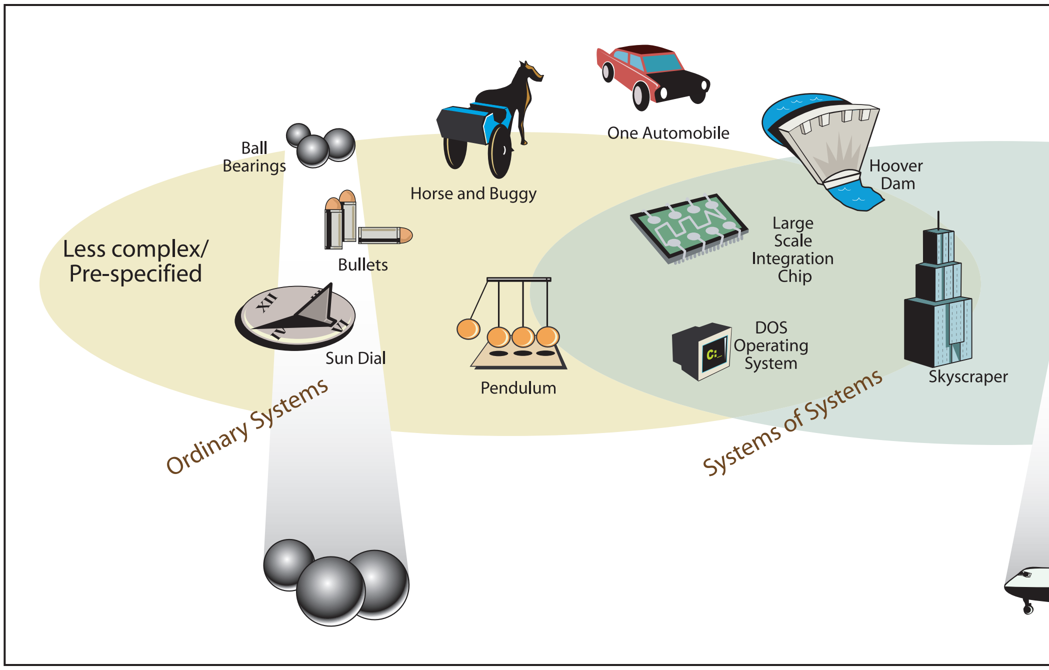
Interaction of people. Complexity increases when the number of systems (developed as standalone entities or with interoperability in mind) and disparate stakeholders increase. As Vanessa Fong, director of Software Systems Engineering for the Center for Advanced Aviation System Development, puts it, “The complex part is the interaction of people. The social systems, markets, and the overall operating environment of a system very much impact complexity.”

The Nature of Complex-System Engineering

According to SEPO’s Michael Kuras, complex-system engineering (cSE) is characterized as a deliberate mimicry of the processes that drive natural evolution in order to focus and accelerate the development of complex systems.

“Nature doesn’t have centralized control. Its behavior is shaped by competing and cooperative objectives, and adaptation,” says Donna Rhodes, principal research engineer, Center for Technology, Policy, & Industrial Development at MIT. “I agree with Renee Stevens’s observation that ‘complex systems can exhibit self-organization with decentralized control, emergent behavior, and adaptation to their environment over time.’ General systems theory emphasizes looking at the fundamental practices and behaviors of natural systems in order to apply them to complex systems. Complex-system engineering can benefit from such natural systems lessons regarding emergent behavior and self-organization, and understanding these in context of complexity and chaos theory.”

> “We are encountering systems that are so complex and have such long descriptions that the conventional approach of specification by decomposition no longer works.”



Yaneer Bar-Yam, president of the New England Complex Systems Institute (NECSI) and a pioneer in the study of complexity as a discipline, has been teaching the subject for over 25 years. “The basic issue, from a systems engineering perspective,” Bar-Yam says, “is that we are encountering systems that are so complex and have such long descriptions that the conventional approach of specification by decomposition no longer works. And in that context, people are looking for a paradigm that will help deal with these high complexity systems.”

Both Engineering Methods Coexist

Understanding all this complexity is something that traditional systems engineering cannot help us with. For sufficiently complex systems, traditional systems engineering will not work.

The traditional systems engineering method, according to Bar-Yam, is basically design by decomposition. The method abstracts a high-level description that is then separated into components and the components are designed at least quasi-separately.

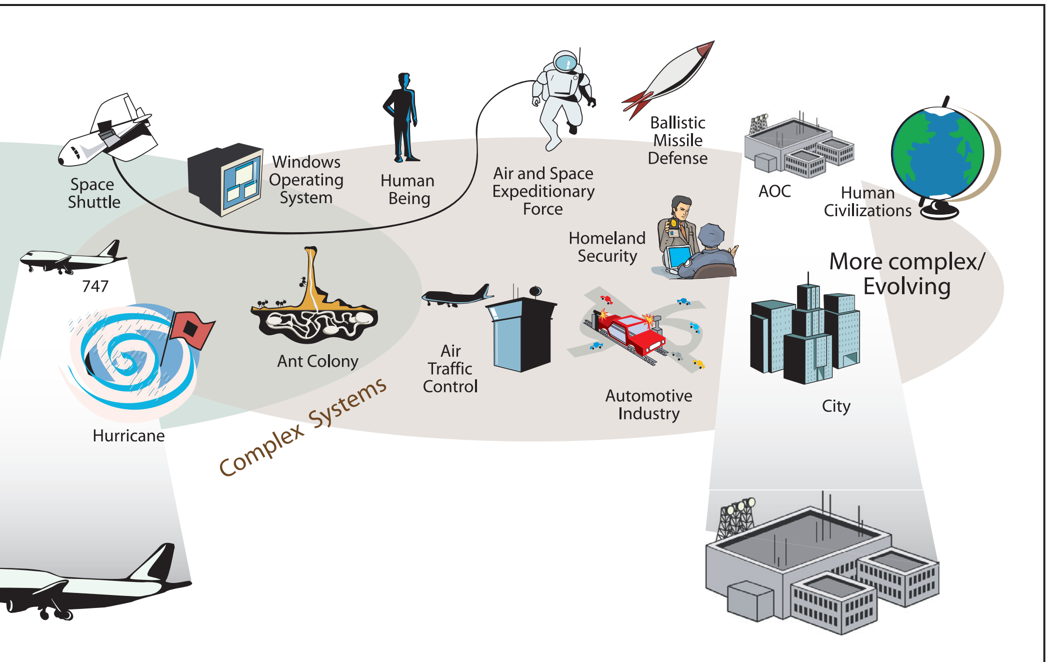
Traditional systems engineering can serve two key roles in designing:

- Systems that are not overly complex
- Interactions among components of complex systems

The complex-system engineering role, on the other hand, is designing the

- Environment and processes by which the system is going to be created, which is separate from designing the system itself
- Components of the system for the system as a whole

These two methods are nested, and their applicability is determined by the scope, scale, and detail of the system under investigation. The idea that you throw out traditional systems engineering because it is broken just isn’t correct. Mostly it fails to work because it is applied to systems that are beyond a threshold of complexity.



business engineering, which entails organizing people's roles and responsibilities, defining policies and procedures, managing complex business models and programs, and introducing change without which you cannot develop systems to support a complex environment.

The Internet allows communication to happen. The Internet illustrates the need for at least a minimal set of standards that give people the ability to interact.

Steve Wagner, principal engineer for the Center for Enterprise Modernization, agrees. "Complex-system engineering is less dependent on technology than traditional systems engineering. Traditional systems engineering becomes an enabling capability to address the more complex "business" relationships you are trying to create. Our work now encompasses the business component to create a greater capability."

Complex-system engineering brings nontraditional tools to bear from different domains to find solutions.

"In the traditional system engineering process, the objective of the design is the system," Bar-Yam says. "You start with the idea of the system, and when you're finished, the system is designed. In complex-system engineering, there is no product. The thing that you're creating is an environment or process by which the system is going to appear and continue to improve over time.

"For example, take the manufacturing process. When you're designing the manufacturing process, you're not designing the system. In complex-system engineering, the key thing is the design of the process *without* and *before* the design of the system. It's very different from making the design."

In complex-system engineering, an intentionally formed environment serves as a framework in which systems exist and/or are created. "People create applications that use the Internet," says Bar-Yam. "The people who designed the Internet created an environment in which all these other things are being created. The Internet designers created a context in which those other things have been created and work.

"In complex-system engineering, the nature of the environment you want to create is not just a piece of software or a piece of hardware. It has to do with a set of rules about how people engage with each other and the process of change."

To put it another way, Rina Levy, principal engineer for the Center for Enterprise Modernization, says complex-system engineering is an integral part of complex

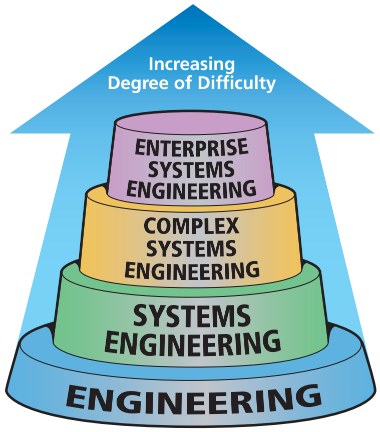
Highlights of the next issue of Collaborations …

WHAT'S NEXT?

In our next issue, we will look at systems engineering tools used at MITRE. What are systems engineering tools? How are they applied to the work done at MITRE and by whom? We will highlight tools used by various Centers and offer insights to their ease of use and effectiveness as well as the value the tools add to the programs in which they are employed.

WHAT DO YOU THINK?

We welcome feedback from our readers. Do you agree or disagree with what is in the newsletter? Do you have information you would like to add? We want to publish your responses to share with our other readers. Send email to sepo@mitre.org with your suggestions.



Engineering Discipline Sets

"We're finding new ways to bring to bear these nontraditional processes over into different domains," says Wagner.

MITRE is well positioned to advance cSE because we are in so many dissimilar domains, but we speak the same fundamental engineering language. When we speak architecture or we speak business process, we each have our own domain and our own ways to apply it, but we can agree that there's a need for a process, an underlying architecture, and test and validation. So when your customer is, say, a health care organization and you listen to what they need, it maps immediately to a manufacturing supply chain problem.

Systems engineering and complex-system engineering live together. Treating them separately doesn't make any sense. cSE builds on the capabilities of traditional systems engineering but has its own unique perspective of focusing on the system environment.

System Engineering Process Toolkits Are Available on SEPO's Public Web Site

SEPO has developed a set of flexible, yet robust, systems engineering processes, and the training, guidance, and tools that our users need when they implement these processes. We call these packages "process toolkits," and they define the core systems engineering processes applicable to the development, fielding, and sustainment of complex systems. We currently offer three toolkits on-line and on disk:

- Risk Management
- Partnering
- Program Assessment

For a CD version of a toolkit, please contact us at sepo@mitre.org.

Who We Are

The MITRE Systems Engineering Process Office (SEPO) is a nexus for systems engineering information and activity at MITRE. Our team brings together useful systems engineering resources, provides guidance on systems engineering processes, and participates in systems engineering activities throughout The MITRE Corporation.

Systems engineering resources are available through the SEPO Library, which contains a broad spectrum of information and knowledge to help you on such topics as acquisition, systems engineering, software engineering, decision support, and process management.

Systems engineering expertise is available through the SEPO Technical HOTline. Emails sent to the HOTline reach multiple subject matter experts, who can provide answers to your questions, connections to other experts on the subject here at MITRE, or contact with other people who are working on the same problem.

We offer systems engineering guidance through our SEPO Toolkits. Toolkits are available online or on CD and include many topics on the system engineering process. For guidance in another area, such as Software Engineering, Acquisition, the Capability Maturity Model - Integrated (CMMI) Process, or sponsor-specific systems engineering areas, please contact our team.

Collaborations is a publication of SEPO. For additional information, please contact:

Brian E. White, Ph.D., SEPO Director
781-271-8218; Email; SEPO@mitre.org

This newsletter is a publication of the Systems Engineering Process Office (SEPO). Additional copies are available.

The MITRE Corporation, 202 Burlington Road, Bedford, MA 01730-1420, and 7515 Colshire Drive, McLean, VA 22102-7508

www.mitre.org/work/sepo/

David Cleary, editor
Margaret Hill, copy editor
Lori Fermano-Pass, designer