Software Engineering Institute

Profiling Systems Using the Defining Characteristics of Systems of Systems (SoS)

Donald Firesmith

February 2010

TECHNICAL NOTE
CMU/SEI-2010-TN-001

Acquisition Support Program

http://www.sei.cmu.edu

Carnegie Mellon

This report was prepared for the

SEI Administrative Agent
ESC/XPK
5 Eglin Street
Hanscom AFB, MA 01731-2100

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

This work is sponsored by the U.S. Department of Defense. The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.

For information about SEI publications, please visit the library on the SEI website (http://www.sei.cmu.edu/library).

Table of Contents

List of Figures

Acknowledgments

I would like to thank the following colleagues who reviewed drafts of this technical note and provided many useful comments and recommendations: Peter Capell, Paul Clements, Rick Kazman, Harry Levinson, Nancy Mead, Tom Merendino, Putcha V. Narasimham, Patricia Oberndorf, Patrick Place, and Carole Woody.

Executive Summary

All systems are not the same, and they vary greatly in terms of many important system characteristics. One subset of such characteristics is the quality characteristics (often informally referred to as the *ilities)* such as availability, capacity, extensibility, interoperability, maintainability, performance, portability, reliability, robustness, safety, security, testability, usability, and variability that, when decomposed into their more objective quality attributes, become the basis for system and software quality requirements. A second subset of important 'system' characteristics does not directly describe systems, but rather describes (1) the organizations involved with their acquisition, development, and operations, (2) the stakeholders of the systems, or (3) the types of projects involved (e.g., individual projects vs. programs of related projects and greenfield development vs. update of a legacy system).

This technical note primarily deals with a third subset of important system characteristics: those often involved in the various definitions of the concept *system of systems*. Individual systems vary in terms of system complexity, criticality, external coupling, distribution, emergence, evolution, governance, heterogeneity, intelligence, reuse, size, and variability. One can imagine using a meter for each of these characteristics that shows where the system lies along the associated scale. For example, Figure 1 shows such a meter[1] for the system characteristic complexity. Individual systems lie at different points along these scales based on the degree to which the systems exhibit the associated characteristics.

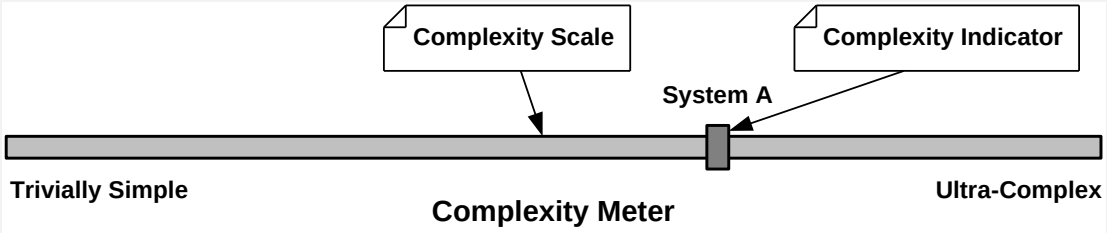

Figure 1: The Complexity Meter showing the Complexity of System A along the Complexity Scale

There are many good reasons to use such scales based on system characteristics to categorize systems:

Improved stakeholder understanding of their systems in terms of the characteristics used to various definitions of the term systems of systems

Improved stakeholder communication regarding these important system characteristics

Improved risk identification by using meters measuring these characteristics to point out those characteristics the values of which imply high risk

Improved system tracking by periodically relooking at the values of these meters to identify any characteristic, whose values are increasing to the point of indicating high risk

Improved decision making by using the values of these meters to help select the appropriate management and technical patterns to use

[1] Measurement scales for different characteristics will be discussed later in the report.

Improved processes by using method engineering to construct methods that are appropriate to the system being engineered

Improved ability to characterize any system as a whole, especially systems of systems

Based on the above benefits, this technical note is written for anyone who might want to use these characteristics as a basis for understanding and communicating information about different types of systems.[2] It is also written for anyone wishing to improve system risk management, tracking and decision making. This paper was also written for those who would understand some of the system characteristics that affect the appropriate system engineering methods to use. Finally, its audience includes anyone who is interested in gaining a better understanding of the defining characteristics of systems of systems.

Because they are typically found in definitions of systems of systems, the following are the defining characteristics of systems of systems:

- System-Level Characteristics
  - Complexity
  - Evolution
  - Negative Emergence
  - Size
  - Variability
- Subsystem-Level Characteristics
  - Autonomy
  - Governance
  - Heterogeneity
  - Physical Distribution
  - Reuse

This report analyzes four example systems in terms of the preceding system of systems characteristics. This technical note also discusses two other classes of system characteristics: the quality characteristics and programmatic characteristics[3] and discusses how the similar meters can be used to describe where systems lie along the scales associated with these two additional sets of system characteristics. Finally, the report discusses the various benefits of using these system of systems characteristics to profile systems.

[2] Because the defining characteristics of systems of systems are exhibited to some degree by every system, they can be used to understand and communicate information about all systems, not just systems of systems.

[3] Programmatic in the sense used here describes characteristics of the associated organizations, stakeholders, and endeavors (i.e., projects, programs consisting of multiple projects, and entire enterprises).

Abstract

The concept of a *system of systems (SoS)* has become very popular over the last decade, resulting in books, conferences, technical papers, and reports. However, there is no consensus as to exactly what the term means, and it has been given many different, though related, definitions. This technical note identifies and describes the characteristics that have been used in various definitions of the term system of systems. These SoS characteristics vary along corresponding scales and can form the basis of corresponding "meters" that serve as indicators of where a system lies along the associated scale. This report also discusses two other classes of system characteristics: quality characteristics and programmatic characteristics and how similar meters can be used to describe where systems lie along the scales associated with these two additional sets of system characteristics. Finally, the report discusses the various benefits of using these system of systems characteristics to profile systems.

1 Introduction

1.1 Background

The concept of a *system of systems (SoS)* has become very popular over the last decade, resulting in it being the focus of organizations (e.g., www.sosece.org) and the topic of books [Jamshidi 2009], conferences,[4] and technical papers and reports. However, there is no consensus as to exactly what the term means, and the term has been given many different, though related, definitions. Each of these definitions is based on a subset of the following characteristics: autonomous subsystems, complexity, criticality, evolution, external coupling, geographical distribution, governance, heterogeneity, internal coupling, negative emergence, reuse, size, and variability. These SoS characteristics that are parts of various reasonable definitions of system of systems form the foundation of this technical note.

1.2 Overview

This technical note begins by showing how a capture of relevant parameters can be represented by "meters" that display the degree to which systems exhibit certain characteristics. Diverse definitions of the term *system of systems* and the ramifications of these definitions in terms of the preceding list of system characteristics is described as is the close relationship between the concepts of a *system of systems* and an *ultra-large-scale system*. These system characteristics are then individually defined, described, and provided an associated metering approach illustrating some typical positions for some example types of systems along the associated scales. Two other subsets of important system characteristics are briefly described to illustrate the

- uniqueness of this subset of SoS characteristics
- wider context in which the concept of such meters used to display the degree to which systems exhibit various system characteristics can be extended

1.3 Intended Audiences

This report will benefit anyone who might want to use these system characteristics as a basis for understanding and communicating information about different types of systems in a succinct and meaningful way. It is also written for anyone wishing to improve system risk management, tracking and decision making. This report will also benefit those who would understand some of the system characteristics that affect the appropriate system engineering method to use. Finally, its audience includes anyone who is interested in gaining a better understanding of the defining characteristics of a system of systems.

[4] Two such conferences are the "IEEE International Conference on Systems of Systems Engineering" and the "SoSECE System of Systems Engineering Conference."

1.4 The Three Sets of System Characteristics

All systems are not the same. They vary greatly in terms of many important characteristics. These characteristics can be categorized as follows.

1.4.1 System of Systems Characteristics

Although the many published definitions of the term *system of systems* (SoS) vary significantly, there nevertheless exists a set of characteristics that are commonly cited as being indicative of such systems. Interestingly, these same characteristics also tend to be indicative of ultra-large-scale systems, a closely related but different concept. Although these defining characteristics of a system of systems actually apply to some degree to all systems, systems of systems and ultra-large-scale systems tend to exhibit these characteristics to a much higher degree than do more traditional systems, which are typically both smaller and simpler. The following SoS characteristics are the prime focus of this technical note:

- system characteristics including system complexity, evolution, negative emergence, size, and variability
- subsystem characteristics including subsystem autonomy, governance, heterogeneity, physical distribution, and reuse

1.4.2 Quality Characteristics

Systems also vary greatly in terms of their quality characteristics and their associated measurable quality attributes[5] that form the basis for engineering system and software quality requirements. A subset of these quality characteristics include:

- external characteristics such as availability, capacity, interoperability, performance, reliability, robustness, safety, security, usability, and variability
- internal characteristics such as extensibility, feasibility, maintainability, portability, reusability, and testability

1.4.3 Programmatic Characteristics

Systems also vary in terms of programmatic characteristics that do not directly describe systems, but rather describe (1) the organizations involved with their acquisition, development, and operations, (2) the stakeholders of the system, and (3) the types of endeavors used to develop or update, operate, and maintain the system:

- organizational characteristics including the type, number, and sizes of the organizations associated with the system; the type of governance;[6] the amount of authority, funding, scheduling, and regulation/policy; the management and engineering culture; geographic distribution,[7] and staff expertise and experience

---

[5] A quality characteristic is a type of quality, whereas a quality attribute is a part of a quality characteristic [ISO/IEC 9126-1 2001]. For example, performance (a quality characteristic) includes the quality attributes jitter, response time, throughput, etc.

[6] Note that although it was listed as a "system of systems" characteristic, governance is actually a programmatic characteristic of the subsystem organizations.

[7] Geographic distribution here refers to the distribution of the organizations (e.g., via outsourcing) rather than the distribution of the system's subsystems.

- stakeholder characteristics including the number and type of stakeholders, the stakeholders' authority, the accessibility of the developers to the stakeholders, the stakeholders' motivations and needs, and the degree of trust between the developers and the stakeholders
- endeavor characteristics including the type of endeavor, contracting, and development; life cycle and system scope; and the endeavor's duration, schedule, and funding

1.5 Meters to Display System Characteristics

Each of these characteristics can take on a range of values, and this range of values forms a measurement scale. For example, some systems are trivially small while others are ultra-large. As illustrated in Figure 2, one can also obtain a meter for each of these characteristics by adding an indicator to indicate the degree to which a system exhibits that characteristic [White 2008].

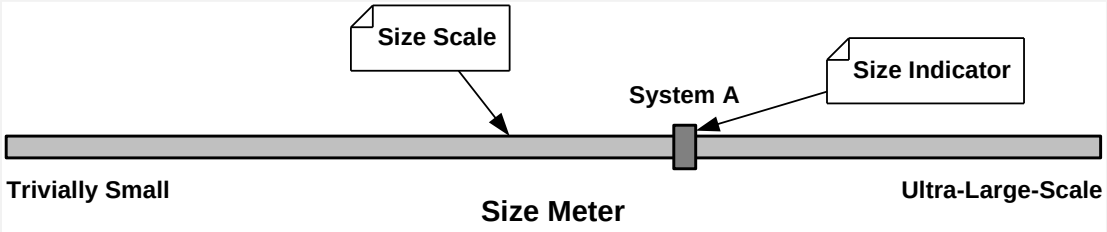

Figure 2: The Complexity Meter showing the Complexity of System A along the Complexity Scale

Figure 3 shows that there are multiple ways of measuring some characteristics, and the chosen measurement method will determine where a system lies along the scale. Size can be measured in terms of the quantity of software, area (also known as a *footprint*), number of subsystems, volume, and weight. Choosing different measurement methods may even determine the order of two different systems on a scale. System A may be smaller than System B when measured with one method, but System A may be larger than System B when measured using a different method.

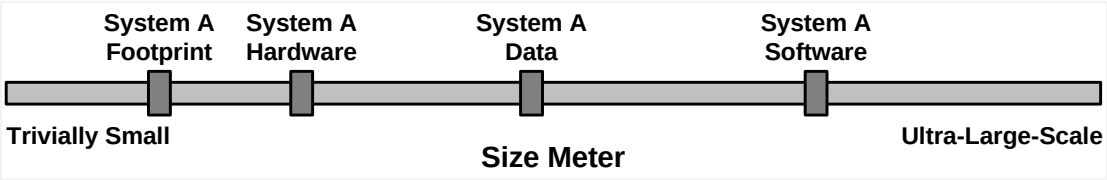

Figure 3: The Size Meter with Multiple Overlapping Scales and associated Measurement Methods

Some system characteristics are difficult or impractical to measure precisely. This is especially true when the values along the scales must be estimated early in the project because the data needed for accuracy is not yet available. Even though the meters are still useful, both accuracy and precision can be quite limited. One must be careful not to assume more accuracy and precision than can be justified by the data.

To overcome the limitations of these fuzzy scales, measurement scales are often divided into a relatively small number of disjoint categories so that the systems can be allocated with a reasonable degree of certainty to one of these buckets (see Figure 4). To achieve most of the benefits of using these meters, this level of accuracy and precision is adequate.

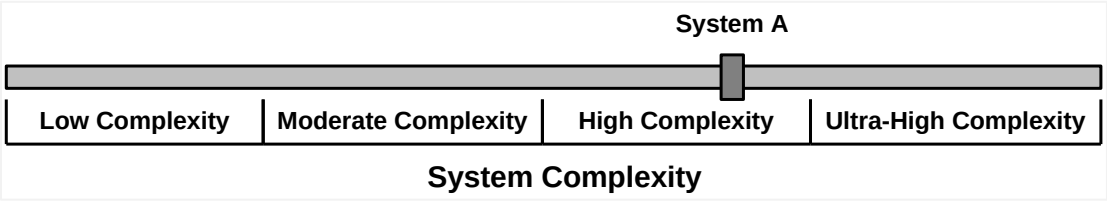

Figure 4: System Complexity Meter with Categorization

When the data are not yet available with which to determine an exact quantitative value for the system, the value will typically have to either be subjectively estimated based on the engineer's experience and expertise or estimated based on the values of existing similar systems. The fact that different stakeholders may disagree about where a system should lie along one of the measurement scales (that is, the value of the meter) should not be seen as a weakness of the approach, but rather as an indication of potential misunderstanding and as a trigger for discussion and consensus building.

Note that in this technical note, types of systems are typically used as examples to help clarify the general idea of these meters and convey how they might be used. Commonly understood types of systems are used instead of specific real-world individual systems to give a broadly understandable general sense of the location and ordering of various systems along the scales associated with different system characteristics. However, the following limitations can be raised:

- Variability within a single type of system. All systems of the same general type will not share the exact same values of these SoS characteristics. For example in Figure 9, high-end cars are more complex than low-end cars, and the same is true for individual aircraft. Thus, to be realistic when used to document a type of systems as opposed to an individual system, these meters should actually display a range of values rather than individual values. However, in this technical note, a single value on the measurement scale will typically be shown by the example meters because the purpose of these meters will usually be to show a single system instead of a range of systems of the same type.
- System definition. Because these example meters are notional and are merely included for illustrative purposes, the systems and types of systems on the example meters are purposely left undefined. On real projects, the individual system (or systems in the case of product lines) represented on the meters would be well defined and not a source of ambiguity.
- Scale is not shown. Because these example meters do not explicitly show specific measurement scales, it is difficult to determine whether the types of systems clump or are widely separated. On real projects where only an individual system is being represented on the meter, this would not be a problem. Because many of these scales are fuzzy, too much precision is not justified. This lack of precision can typically be addressed by breaking the measurement scale into a set of categories, the use of which is usually adequate for the purposes of the meters.

1.6 Benefits

There are many good reasons to use scales based on system characteristics as a means of categorizing systems. This is true for SoS characteristics as it is for quality characteristics and programmatic characteristics in almost any case. These benefits include:

- Improved Understanding. Although the concept of system of systems has become very popular over the last decade, there is no consensus as to exactly what it means, and the term has been given many different, though related, definitions. This diversity of meaning is likely due to the term itself, which is misleading because implies that virtually all systems are systems of systems.

  Using well-defined characteristics will improve the stakeholders' understanding of the important concepts (such as emergence and governance) underlying systems of systems and ultra-large-scale systems, both individually and how multiple characteristics tend to vary together (e.g., size and complexity). More importantly, it will help the stakeholders[8] to understand their system in terms of some of its most fundamental characteristics. This is especially important when dealing with the concepts associated with systems of systems and ultra-large systems, both of which typically refer to systems characteristics that lie at or near the high end of the scales displaying quantities of SoS characteristics.
- Improved Communication. People sometimes accidentally use the same terms with different meanings (unintentional homonyms), or use different terms with the same meanings (synonyms). By clearly defining the systems' important characteristics, developers, acquirers, and other stakeholders will be better able to communicate clearly the characteristics of their system and thus, the type of system (and subsystems) they are engineering or managing.
- Improved Risk Identification. Each scale for a system characteristic has an associated meter, the value of which can range from a minimum value to a maximum value for the characteristic. The scales have been organized in this technical note so that low values imply low project risk whereas high values imply high project risk. In other words, the value displayed by a meter measures the challenges and risks associated with the development or major update of the associated system. Thus, by looking at all of the system's meters, one can get a picture of the overall project risk; for example, extremely high if all of the meters show the system either on or near the right ends of its associated measurement scales.
- Improved System Tracking. Early in the project, the system's initial values along some of the scales for the system/subsystem characteristics may be largely *guesstimates* based on past *experience* and an initial vision of the system. Later on as understanding increases, the meters may move to better represent the system as stakeholder understanding of the system matures. Thus, one can use the values displayed by these meters to track the project's high-level understanding of the system as stakeholder understanding of the system matures.
- Improved Decision Making. Many decisions depend on the type of system being engineered. Acquirers, developers, and other stakeholders with authority will be better able to make appropriate decisions regarding the system and its development. For example, the location of the system along these scales may significantly affect which management and architectural patterns are appropriate to use. For example, management patterns can be affected by size and subsystem governance, whereas architecture patterns can be influenced by external and internal coupling, criticality, evolution, variability, and reuse.

---

[8] In this context, the word *stakeholders* means everyone with a significant legitimate interest in the system during any phase of the system's life cycle from initial acquisition through development, operation, and sustainment, to eventual retirement and disposal.

- Method Engineering. Because of the vast variability among different types of systems, all systems should not be treated the same way. Different systems exhibit different characteristics, and the values of some of these characteristics influence the number and attributes of the appropriate system engineering and management methods that should be used to develop, maintain, and operate them. The large variability in systems is the reason why no single system engineering method is sufficiently general and tailorable to meet the needs of all endeavors. These meters showing where a system lays along these scales support the use of situational method engineering by helping system engineers, technical leads, process engineers, and technical managers to determine the most appropriate system development methods.

2 Systems of Systems

The concept *system of systems* has become very popular over the last decade, bringing the idea to the attention of organizations (e.g., www.sosece.org) and the topic of many books [Jamshidi 2009], conferences,[9] and technical papers and reports. However, there is no consensus as to exactly what the term means, and it has been given many different, though related, definitions.

2.1 Almost Every System is Technically a System of Systems

Because the term *system of systems* includes the term *system*, it is important to first understand what a system is before one can understand what a system of systems is.

> **System**
> a cohesive integrated group of interdependent or interacting components that regularly collaborate to provide the behavior and characteristics that (1) are needed to meet valid stakeholder needs and desires and (2) cannot be provide separately by the individual components [Firesmith 2008]

As implied by the second half of the preceding definition, a system is more than the sum of its parts because a system provides beneficial emergent behavior or characteristics that are not present in its individual components working independently. For example, the car in your garage is a system, whereas all of the component parts of your car lying on the floor of your garage would not be a system.

Because the phrase *system of systems* is so short and simple, it should be easy to define. By definition, the largest components of any nontrivial system are themselves systems, typically called subsystems. The aggregation structure of most systems is potentially quite large consisting of subsystems containing subsystems containing subsystems down multiple layers until one finally reaches simple hardware, software, or other components that can be treated as blackboxes. Thus, a straightforward interpretation of the term *system of systems* would be any system consisting of smaller systems, and this definition also obviously applies to all nontrivial systems. In other words, all systems of systems are systems and almost all systems are systems of systems. Unless there is something more to the term system of systems than what is implied by its component words, then using the term *system of systems* instead of the simpler term *system* would add little or no additional meaning other than to emphasize that the component subsystems are important systems in their own right.

2.2 Characteristics of a Good Definition of System of Systems

Theoretically, it should be relatively straightforward to create good terms and definitions because simple and clear criteria exist for judging the utility of terms and their definitions. In practice, however, it is clear that properly defining technical terms is often difficult to do, partially because

[9] For example, the *IEEE International Conference on Systems of Systems Engineering* and the *SoSECE System of Systems Engineering Conference*.

the typical engineer tasked with creating glossaries defining technical terms and jargon has never been trained as a lexicographer. Standard guidelines for good terminology and definitions include:

1. The definition of a term should be unambiguous. There should only be one interpretation of the definition.
1. The technical terms in the definition of a term should also be defined. Only obvious, non-technical words in the definition need not be defined.
2. Definitions should be intuitive rather than misleading. A term should imply its definition, especially if the term is actually a descriptive phrase.
3. Definitions should have the right scope. As illustrated in Figure 5, the scope of the definition should match the scope of the term.
    a. Definitions should not be too general. The set of entities implied by the *term* should not be a proper *subset* of the set of entities specified by the *definition* of that term. Thus, everything matching the definition of system of systems should in fact be a system of systems, and there should not be anything that is not a system of systems that matches the definition of system of systems.
    b. Definitions should not be too specific. The set of entities implied by the *term* should not be a proper *superset* of the set of entities specified by the *definition* of that term. In other words, every system of systems should meet the definition of system of systems, and there should not be any systems of systems that do not match the definition of systems of systems.

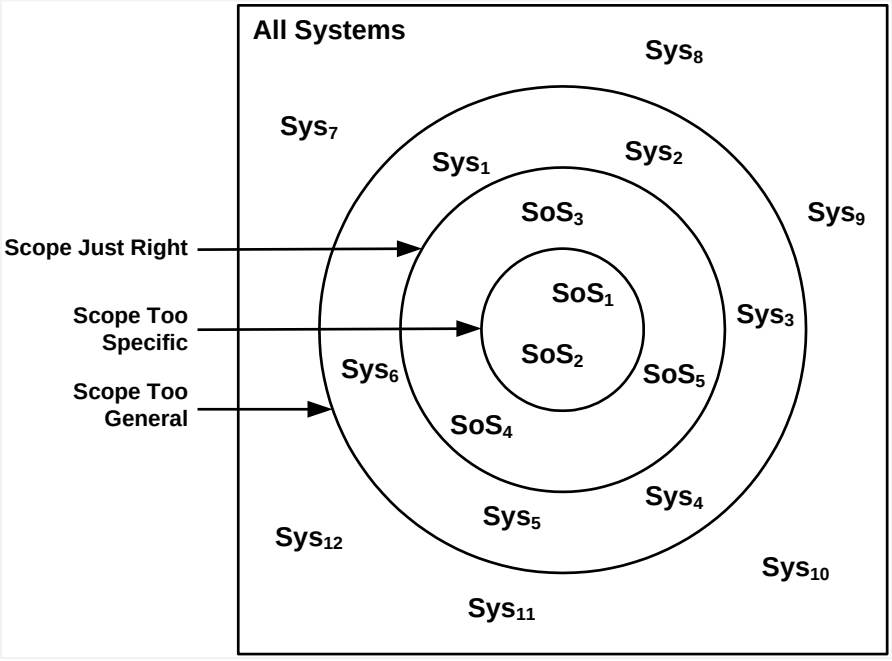

Figure 5: The Scope of the Definition of a System of Systems

The four preceding guidelines should definitely apply to the term *system of systems*. However, the way the term is typically defined violates guidelines 3 and 4b above:

3) A *system of systems* is not just any system of systems.

4b) A *system of systems* has specific attributes that only describe certain types of systems of systems rather than to all systems of systems.

2.3 Current Definitions and Descriptions

So how is the term *system of systems* actually defined in technical books, training materials, conference papers, and articles? The following are just a few of the definitions and descriptions of system of systems that one can find with very little effort:

- A "system of systems [is a] a collection of trans-domain networks of heterogeneous systems that are likely to exhibit operational and managerial independence, geographical distribution, and emergent and evolutionary behaviors that would not be apparent if the systems and their interactions are modeled separately." [DeLaurentis 2004]
- "An SoS is defined as a set or arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities." [DOD 2008]
- "A system of systems exists when a group of independently operating systems—comprised of people, technology, and organizations—are connected, enabling emergency responders to effectively support day-to-day operations, planned events, or major incidents." [Homeland Security 2009]
- Systems of systems are "metasystems that must function as an integrated complex system to produce desirable results. These metasystems are themselves comprised of multiple autonomous embedded complex systems that can be diverse in technology, context, operation, geography, and conceptual frame." [INCOSE 2009]
- "Systems of systems are large-scale concurrent and distributed systems the components of which are complex systems themselves." [Kotov 1997]
- "A collection of systems that is deliberately integrated to achieve a purpose not generally achievable by the individual systems functioning separately. The systems in a SoS are usually developed separately to accomplish their own specific purposes, and they could operate independently in the same environment associated with the SoS. … The component systems are physically distinct and could be geographically distributed. Typically their boundaries are crisp and stable, and the systems are bound together by well-defined interfaces. If any system is significantly changed or bypassed, the SoS generally continues to function, but its overall capability may be altered." [Kuras 2005]
- "Five principal characteristics are useful in distinguishing very large and complex but monolithic systems from true systems-of-systems.
  - Operational Independence of the Elements. If the system-of-systems is disassembled into its component systems the component systems must be able to usefully operate independently. The system-of-systems is composed of systems which are independent and useful in their own right.
  - Managerial Independence of the Elements. The component systems not only *can* operate independently, they *do* operate independently. The component systems are separately acquired and integrated but maintain a continuing operational existence independent of the system-of- systems.

  - Evolutionary Development. The system-of-systems does not appear fully formed. Its development and existence is evolutionary with functions and purposes added, removed, and modified with experience.
  - Emergent Behavior. The system performs functions and carries out purposes that do not reside in any component system. These behaviors are emergent properties of the entire system-of-systems and cannot be localized to any component system. The principal purposes of the systems-of-systems are fulfilled by these behaviors.
  - Geographic Distribution. The geographic extent of the component systems is large. Large is a nebulous and relative concept as communication capabilities increase, but at a minimum it means that the components can readily exchange only information and not substantial quantities of mass or energy." [Meier 1998]
- "This emerging system-of-systems concept describes the large-scale integration of many independent, self-contained systems in order to satisfy a global need." [Purdue University 2009]
- "Modern systems that comprise system of systems problems are not monolithic; rather they have five common characteristics: operational independence of the individual systems, managerial independence of the systems, geographical distribution, emergent behavior and evolutionary development." [Sage 2001]
- A system of systems is a "system comprising independent, self-contained systems that, taken as a whole, satisfy a specified need." [Northrop 2006]
- "A system of systems is a complex purposeful whole that:
  - Is composed of complex independent component parts whose high levels of interoperability enable them to be composed into different configurations and even different SoS,
  - Is characterized by contextual complexity that significantly affects its behavior and makes it difficult to understand,
  - Has ambiguous and/or changing boundaries, and
  - Exhibits emergent properties." (SOSECE definition) [Sheard 2006]
- "A configuration of systems in which component systems can be added / removed during use; each provides useful services in its own right; and each is managed for those services. Yet, together they exhibit a synergistic, transcendent capability" [USAF 2005]
- "A collection of *systems* that functions to achieve a purpose not generally achievable by the individual systems acting independently… Each system can operate independently and is managed primarily to accomplish its own separate purpose. A SoS can be geographically distributed, and can exhibit evolutionary development and/or emergent behavior." [White 2005]
- A system of systems is "a collection of task-oriented or dedicated systems that pool their resources and capabilities together to obtain a new, more complex, 'meta-system' which offers more functionality and performance than simply the sum of the constituent systems" [Wikipedia 2009]

Each of the preceding definitions and descriptions specifies a subset of the following "mandatory" characteristics of systems of systems. By parsing each of these definitions, we derive the following list of SoS characteristics and characteristics of their component systems:

- System of systems
  - exhibit (obscure) emergent behavior
  - are very large and complex
  - are (or need to be) highly flexible
  - are dynamically evolving
  - are geographically distributed
- Component systems (subsystems)
  - are heterogeneous (e.g., in terms of technology and operation)
  - were independent before being integrated into the system of systems
  - exhibit operational independence
  - exhibit managerial independence
  - exhibit schedule independence
  - are self-contained
  - are independently useful
  - are geographically distributed
  - are autonomous
  - are embedded
  - come from multiple domains
  - have different contexts
  - have different conceptual frames
  - are task-oriented
  - are dedicated
  - behave concurrently
  - are complex

As illustrated in Figure 6, the different definitions emphasize different characteristics.

There are two interesting attributes of these SoS characteristics:

- The characteristics can have a range of values. They are not binary or enumeration types.
- The basic characteristics apply to all systems, not just systems of systems. However, systems of systems tend to exhibit these characteristics more than other systems. Thus, being a system of systems is a matter of degree rather than of kind, and a meter measuring the degree to which a system exhibits these characteristics would be useful to have.

Different Definitions Emphasize Different Characteristics		Definition Sources													
		DeLaurentis et al.	Dept. of Defense	Homeland Security	INCOSE	Kotov	Kuras and White	Meier	Purdue University	Sage and Cuppan	SEI	SOSECE	US Air Force	White	Wikipedia
System Characteristics	Complexity				X	X						X			X
	Evolution	X						X		X		X	X	X	
	Neg. Emergence	X	X		X			X	X	X	X	X	X	X	X
	Size		X			X			X						
	Variability											X			
Subsystem Characteristics	Autonomy	X	X	X	X		X	X	X	X	X	X		X	
	Governance	X	X	X	X		X	X		X	X		X	X	
	Heterogeneity	X		X	X										
	Phy. Distribution	X			X	X	X	X		X				X	
	Reuse	X	X	X			X				X				

Figure 6: Different Definitions Emphasize Different Characteristics

2.4 There are two interesting attributes of the SoS characteristics that are listed in Figure 6.What is Usually Meant by the Term *System of Systems*?

Because almost all systems consist of subsystems that are themselves systems, the term *system of systems* is too general and misleading. The term has been given many different definitions, which are usually based on several of the system and subsystem characteristics documented in Section 2 of this technical note, whereby a system of systems is any system that lies towards the high risk ends of the meters for these system and subsystem characteristics.

> **System of Systems (SoS)**
> any system that is a relatively large and complex, dynamically evolving, and physically distributed system of pre-existing, heterogeneous, autonomous, and independently governed systems, whereby the system of systems exhibits significant amounts of unexpected emergent behavior and characteristics

The next section of this report addresses each of the individual system and subsystem characteristics that are important in the definition of the term system of systems. They will therefore be called SoS characteristics to differentiate them from other system characteristics such as quality characteristics and programmatic characteristics.

2.5 Systems of Systems and Ultra-Large-Scale Systems

Many systems of systems are constructed from large, pre-existing, independently useful and governed systems so that the size of the resulting system is as large as the union of these component

systems. In these cases, the resulting SoS is an ultra-large-scale (ULS) system [Northrop 2006], whereby the ULS is defined as any system of unprecedented scale in some of the following dimensions:

- lines of code
- amount of data stored, accessed, manipulated, and refined
- number of connections and interdependencies
- number of hardware elements
- number of computational elements
- number of system purposes and user perception of these purposes
- number of routine processes, interactions, and "emergent behaviors"
- number of (overlapping) policy domains and enforceable mechanisms
- number of people involved in some way

Conversely, the cost of ULS systems tends to mean that most such systems are composed of pre-existing, independently useful and governed systems of systems. Thus, systems of systems tend to be ultra-large-scale systems, and ultra-large-scale systems tend to be systems of systems. This is why the defining characteristics of systems of systems also tend to apply to ultra-large-scale systems.

3 Common System of Systems Characteristics

3.1 The System-Level Characteristics

Figure 7 illustrates the system-level characteristics and subsystem characteristics that are commonly incorporated into various definitions of the phrase *systems of systems*. The system-level characteristics describe the system as a whole, whereas the subsystem-level characteristics primarily describe the system in terms of the characteristics of its subsystems. Each characteristic has an associated scale, which is ordered left to right in terms of increasing risk. Any single given system will typically reside at different points along different scales, turning the scales into meters that measure the degree to which the system exhibits the associated characteristic [White 2008, Garcia-Miller 2009].

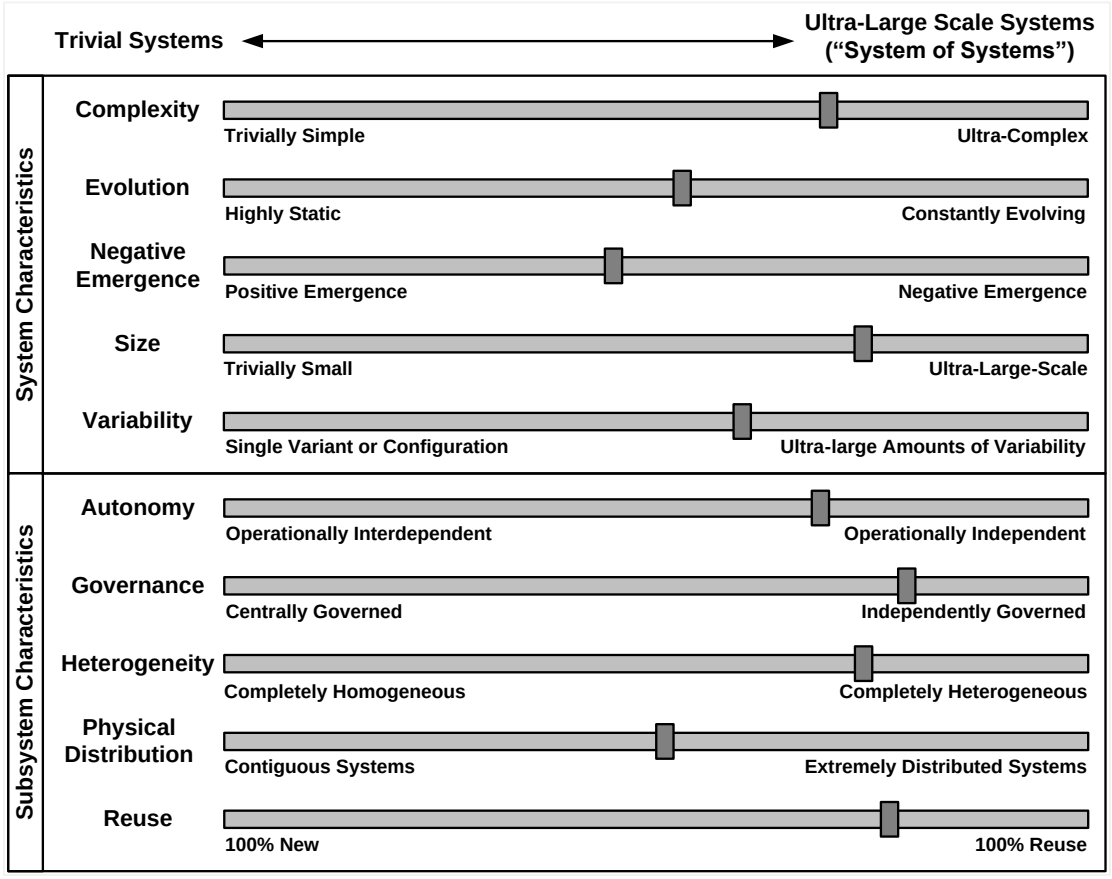

Figure 7: Meters Measuring Characteristics associated with the Definition of Systems of Systems

Most of the meters are based on a notional scale (i.e., a scale that is subjective, imprecise, and fuzzy rather than one with objective mathematical precision). For this reason, each scale is often decomposed into a small number of categories (often two-seven scales are used). This helps mitigate the problems caused by disagreements over the precise point where a specific system lies on any specific scale.

As illustrated in Figure 8, although a spider chart [Stevens 2008, White 2008, Simanta 2009] could theoretically have been used instead of a set of meters, meters were chosen over spider charts because:

- Spider charts do not scale well since the number of characteristics (radial threads) increases because their labels must get closer together, making the labels smaller and more difficult to read.
- Spider charts are harder to draw manually when the number of characteristics is odd or when it's a prime number such as 7, 11, 13, and 17 because it is hard to manually divide a circle into that number of wedges.
- The optimum number of categories into which the different scales are divided need not be the same for all scales.

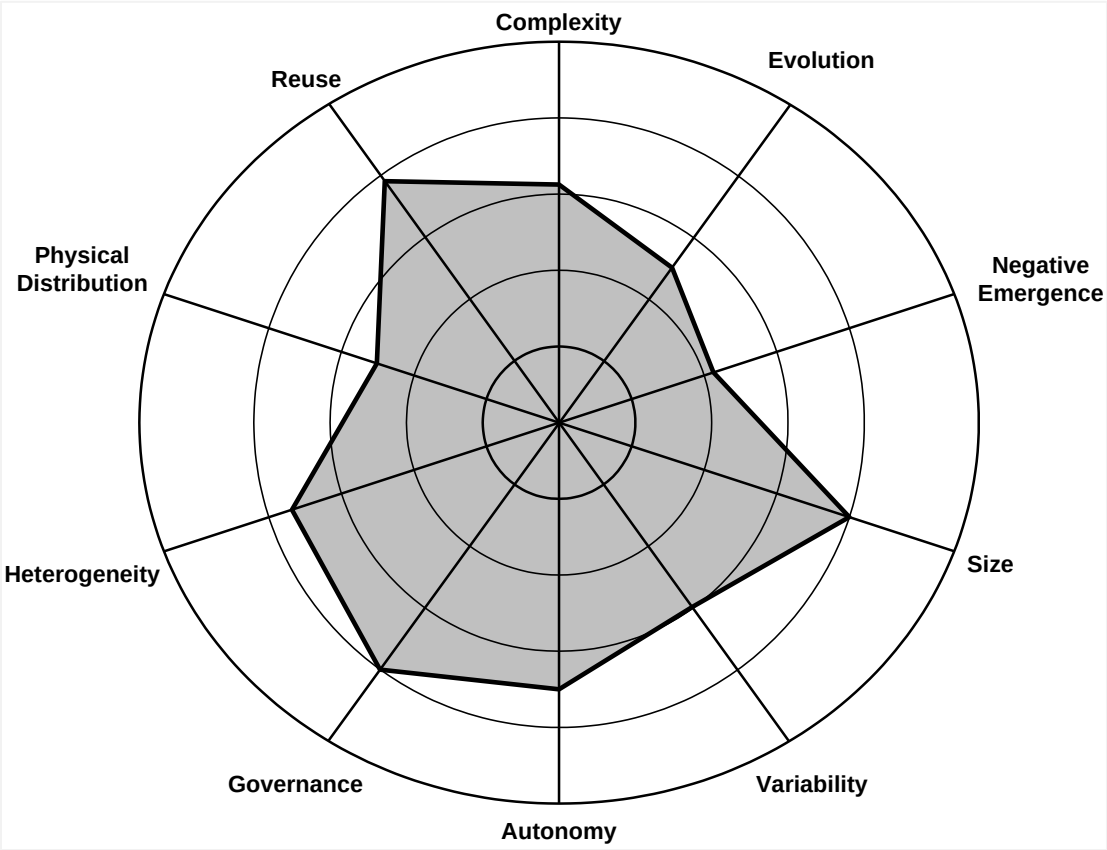

Figure 8: Spider Chart corresponding to Collection of Meters

Systems are composed of subsystems, and subsystems are almost always systems. Thus, the system concepts just described can be applied at the subsystem level and the system level. However, the values of the meters associated with these different system characteristic are typically not the same at the system and subsystem levels. For example, the trend from system to subsystem is towards smaller size and less complexity. Similarly, two subsystems of the same system need not have the same values on the same scale; the two associated meters may show different values for the two different subsystems.

Note that the recursive decomposition of systems (or subsystems) into lower level subsystems that are themselves systems does not go on forever because at some level, the component parts of a system are too simple to be worthy of being considered systems. Additionally, the real world is not infinitely decomposable, but eventually consists of elementary particles and fields that cannot be further decomposed.

As illustrated in Figure 7, systems can be characterized by the following important characteristics that describe individual systems as a whole:

- Complexity
- Evolution
- Negative Emergence
- Size
- Variability

These characteristics are described in the next sections.

3.1.1 System Complexity

Systems vary greatly in terms of complexity from the trivially simple to the ultra complex. In general, the complexity of a system increases with the:

- size of the system
- number of different missions supported or performed by the system
- number of requirements implemented by the system
- number and complexity of its individual subsystems
- number and complexity of the relationships, interfaces, and interactions between the system and:
  - its subsystems (internal interfaces)
  - external systems (external system interfaces)
  - human users and operators (human interfaces)
- heterogeneity of the system's subsystems (e.g., in terms of application domain and technologies incorporated)
- complexity of the system's technologies

Although systems tend to increase in complexity as they grow larger, this trend need not be simple nor is it guaranteed that larger systems are always more complex than smaller systems. For example, a system could consist of a *very* large number of identical simple subsystems and still be far less complex than a system consisting of a much smaller number of heterogeneous subsystems interacting in a highly complex manner. Another way that systems increase in complexity as the increase in size (in terms of number of subsystems) is because the subsystems must be integrated, possibly via multiple complex protocols subject to concurrency failures due to race conditions, priority inversions, starvation, dead-lock, and live-lock.

Some systems are naturally complex because they consist of a large number of highly interactive and tightly coupled subsystems. Such systems are far beyond the capabilities of any single or small number of stakeholders to understand. These complex systems are high risk because their

complexity (1) leads to large amounts of unexpected, unintended, and detrimental emergent behaviors and characteristics and (2) makes accidents practically inevitable. Such accidents are often referred to as "normal accidents" [Perrow 1984] because they should be expected to occur as a natural consequence of the systems' complexity combined with the inherent limitations of human understanding.

Ultra-large-scale systems are almost always extremely complex and thus lie at the high end of the *system complexity* scale. Systems of systems also tend to be complex and lie near the high end of the *system complexity* scale because they are typically composed of multiple pre-existing systems which themselves are complex.

Definition

System complexity is defined as

> the degree to which a system is difficult for its stakeholders to understand and analyze, especially due to having a large number of components connected by many complicated interfaces

Scale

Based on the previous definition of system complexity and ordered by increasing risk, system complexity forms a scale of systems that ranges from:

- Trivially Simple Systems

  Trivially simple systems are systems that are very easy for all of their stakeholders to understand because of their simple characteristics, structure, and behavior.
- Ultra Complex Systems

  Ultra complex systems are systems that are extremely difficult (or impossible) for any one of its stakeholders to completely and correctly understand.

Examples

Ordered by increasing risk due to complexity, examples of systems categorized in terms of their degree of complexity include the following.

- Simple Systems with Low Levels of Complexity
  - automated teller machines (ATMs)
  - vending machines, which consist of only a few different types of very simple subsystems
- Systems with Moderate Levels of Complexity
  - low-end cars
  - small aircraft
  - television sets
- Systems with High Levels of Complexity
  - high-end cars, which consist of many tightly coupled and complex subsystems
  - large commercial aircraft and military aircraft
- Systems with Extreme Levels of Complexity

- aircraft fleets, which include not only the individual aircraft but also the associated ground-based flight planning, logistics, maintenance, and training systems
- oil refineries
- smart grid (electric national power grid)
- World Wide Web

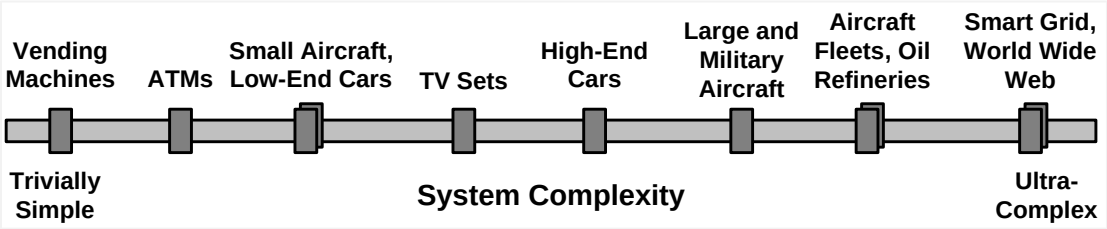

Figure 9: The System Complexity Meter with Example Systems

3.1.2 System Evolution

Systems are rarely if ever static, and they change over time for any of the following reasons.

- The system's requirements change due to associate changes in
  - the goals and needs of the system's stakeholders, which sometimes change unexpectedly
  - the operational profiles describing how the system is used
  - the unintended or unanticipated ways users and operators may begin to use the system
  - the physical environment in which the system operates
  - relevant laws and regulations
  - the threat environment in which the system operates. Such changes might happen when the number and type of attackers change and the attackers' means, motives, and opportunities change.
- The system's defects and vulnerabilities are identified and fixed.
- The system's technologies are updated (e.g., during technology refreshes).
- The system depends on external systems that change (e.g., functionality or interfaces) so that the system itself must change to remain consistent.
- The system consists of independently governed subsystems that change (see Subsystem Governance on page 29).
- Subsystems can be added or removed from the overall system, possibly in a dynamic fashion (i.e., dynamic [re]configurability).

Both ultra-large-scale systems and systems of systems typically lie near the constantly evolving end of the *system evolution* scale because such systems tend to

- interact with their environment in more ways than small simple systems
- have more stakeholders whose needs can change

- incorporate more types of technology that are themselves constantly improving
- depend on more external systems that are themselves changing
- incorporate so many independently governed systems as subsystems, some of which are typically in the process of being upgraded

Definition

System evolution is defined as

> the degree to which (in terms of rate and impact) the goals and requirements for a system (and its subsystems) change over time

System evolution includes

- Major Updates
  - updates to match new releases
  - major technology refreshes
- Maintenance
  - adaptive maintenance to adapt the system to current changes in requirements, environment, procedures, or legislation
  - corrective (a.k.a., reactive) maintenance to fix system after failures or defects have been found
  - perfective maintenance to make minor improvements in the system (e.g., improve efficiency, performance, reliability, or usability)
  - predictive maintenance to change the system in preparation of future changes
  - preventative maintenance to prevent future failures

The rate of change can be measured in terms of the number of changes per unit time and the size (e.g., impact or radical nature) of the changes.

Scale

Based on the previous definition of evolution and ordered by increasing risk, system evolution forms a scale of systems that range from:

- Highly Static Systems

  Systems that change slowly or not at all, that have highly stable requirements, use highly stable technology, and exist in highly stable environments.
- Constantly Evolving Systems

  Systems that are constantly evolving to meet new requirements, use the latest technology, and adapt to a constantly changing environment.

Examples

Ordered by increasing risk due to evolution, examples of systems categorized in terms of their degree of evolution include the following.

- Static Systems that Do Not Evolve
  - consumer electronics

  - televisions
- Systems with Very Low Rates of Evolution
  - satellites and space probes, which can have new software uploaded
- Systems with Low Rates of Evolution
  - nuclear power plants
  - vending machines
- Systems with Moderate Rates of Evolution
  - aircraft
  - production Unmanned Aerial Vehicles (UAVs)
  - weapons systems
- Systems with High Rates of Evolution
  - cars that change via replacement of tires, oil, windshield wipers, worn out parts, and defective parts (e.g., due to recalls)
  - software information assurance security products
- Systems with Extreme Rates of Evolution
  - national electric power grid[10]
  - research robotic cars
  - research unmanned aerial vehicles (UAVs)
  - World Wide Web

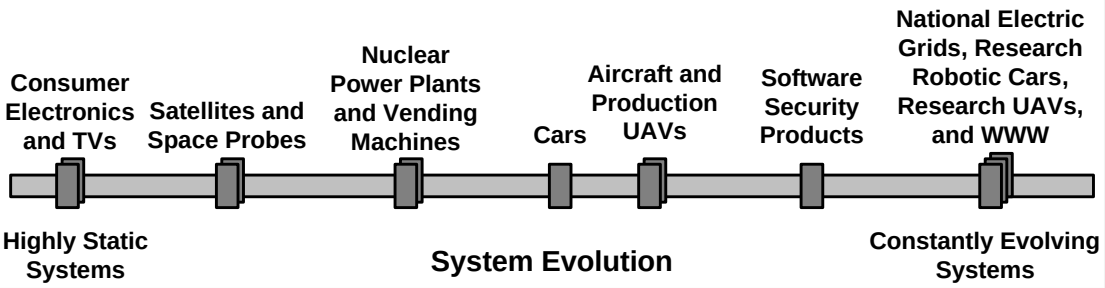

Figure 10: The System Evolution Meter with Example Systems

3.1.3 System Negative Emergence

All systems are more than the sum of their parts. By definition, a system is a cohesive integrated set of interdependent or interacting components that collaborate to provide new behaviors and characteristics that the individual components do not provide separately. In other words, all systems exhibit new emergent behaviors and characteristics due to the relationships, interfaces, and interactions between their subsystems. In fact, beneficial emergent behaviors and characteristics are the reason why systems are developed. Systems are engineered to ensure that they have the beneficial emergent behaviors and characteristics they need to meet their system-level require-

[10] Although the technology of the current national electric power grid has changed little in the last 50 years, the composition of the grid in terms of distribution systems and meters changes rapidly. Even the stability of the technology is about to change as countries move to the use of smart grids.

ments. However in practice, systems sometimes also exhibit emergent behavior and characteristics that are unplanned, unexpected, and undesirable. For example, integration testing may uncover failures and defects that are due to such negative emergent behavior.

Systems vary greatly in the amount and types of emergent behaviors and characteristics that they exhibit. Small and simple systems containing only a small number of subsystems integrated by simple interfaces tend to have only a small number of emergent behaviors and characteristics, and these are typically beneficial, intended, and easily foreseeable and predictable from the behavior and characteristics of their subsystems.

However, the larger and more complex that systems become, the greater the likelihood that some emergent behaviors and characteristics will be detrimental, unintended, and difficult to predict before the system is built and operational. Thus, negative emergence tends to become more important as systems increase in size and complexity. This is why [negative] emergence[11] is often identified as a property of large and complex "systems of systems." However, it is important to remember that small and simple systems also have emergent behavior and characteristics, and the benefits of emergence are the fundamental reason why systems are engineered. A key goal of system engineering is to maximize the amount of positive emergence while minimizing the amount of negative emergence, with the knowledge that it is extremely unlikely that the amount of negative emergence will go to zero for any system that is not trivially small and simple.

It is interesting to note that software defects are typically major sources of system negative emergence. Some of the reasons for this are listed below.

- Software is used to implement the majority of the functionality of large and complex systems.
- Software itself can be very complex in terms of its architecture and logic, the large number of software components, and the many interfaces between these components.
- Systems are typically delivered with subtle software defects, which are often a result of missing requirements or due to rare and exceptional cases.
- Software is commonly used as the "glue" to integrate many subsystems, and integration defects often cause unexpected negative emergence.

As mentioned when discussing complexity, some systems are naturally complex because they consist of a large number of highly interactive and tightly coupled subsystems. These complex systems are high risk because their complexity often leads to unexpected and unwanted emergent behavior that makes accidents practically inevitable. Such accidents are often referred to as "normal accidents" [Perrow 1984].

Because they typically exhibit large amounts of unexpected, unintended, and detrimental emergence, both ultra-large-scale systems and systems of systems tend to lie at the high end of the *system negative emergence* scale.

[11] When emergent behavior is listed as a part of a definition of the term system of system, only negative emergence is usually intended.

Definition

System negative emergence is defined as

> the degree to which the new behaviors and characteristics of a system that result (i.e., emerge) from the interaction of the system's subsystems are detrimental, unintended, and difficult to predict from the behaviors and characteristics of these individual subsystems

The term *system of systems* has typically emphasized the unexpected and difficult-to-predict nature of negative emergent behaviors and characteristics because from the standpoint of system success, these have been the most important. Although these three different concepts can vary independently, definitions of negative emergence combine because they tend to occur together:

- Detrimental emergence—rarely intended or easy to predict because it is often avoidable with the proper controls.
- Unintended emergence—often unintended because it is difficult to predict from the behavior and characteristics of the subsystems
- Difficult to predict emergence—tends to be detrimental and thus unintended because requirements engineering, architecture engineering, and design analyses concentrate on easy-to-predict desired behavior. Difficult-to-predict emergence is typically associated with missing requirements and failures.

The degree to which emergent behaviors and characteristics are negative can be measured both in terms of the number of such behaviors and characteristics and the severity of the harm that results.

Scale

Based on the previous definition of negative emergence and ordered by increasing risk, system emergence forms a scale of systems that range from:

- Systems with only Positive Emergence

  Such systems only exhibit emergent behavior and characteristics that are intended, easily predicted, and beneficial.
- Systems with Unacceptable Negative Emergence

  Such systems exhibit unacceptable levels unintended and unpredicted detrimental emergent behaviors and characteristics.

Examples

Ordered by increasing risk due to negative emergence, examples of systems categorized in terms of their degree of negative emergence include the following.

- Systems with Low Levels of Negative Emergence
  - military aircraft
  - televisions[12]

[12] Note that the negative emergence associated with vending machines (e.g., overeating of junk food) and televisions (e.g., health problems due to inactivity) are behaviors of the system users, not the behavior of the systems.

  - vending machines
- Systems with Moderate Levels of Negative Emergence
  - commercial aircraft, from which emerged the rapid international spread of diseases, hi-jacks, terrorist attacks, and the need for invasive expensive security screening
  - national electric grids
- Systems with High Levels of Negative Emergence
  - cars, from which emerged traffic jams, urban sprawl, excessive commute times, tens of thousands of fatal motor vehicle accidents, dependence on foreign oil, trade imbalances, air pollution, etc.
  - Internet/World Wide Web, from which emerged computer viruses, spam, adware, cyber-crime, cyber-stalking, pornographic websites, etc.
  - nuclear power plants, from which emerged nuclear accidents, nuclear proliferation,
- Systems with Unacceptable Levels of Negative Emergence
  - none

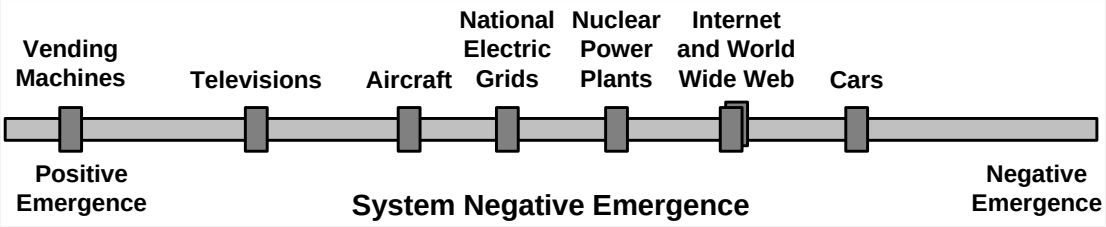

Figure 11: The System Negative Emergence Meter with Example Systems

3.1.4 System Size

Systems vary greatly in size from the quite small through mid-range and large to ultra-large. Size can also be measured in numerous ways. If the system is physical (as opposed to being purely software), then size can be measured in terms of mass or physical dimensions (e.g., footprint or volume). Because they are aggregations of subsystems, system size can also be indirectly measured in terms of the number of subsystems and the size of these subsystems.

By definition, ultra-large-scale systems lie at the ultra-large end of the *system size* scale. Systems of systems also tend to be large or ultra-large and therefore lie near the ultra-large end of the system size scale.

Definition

System size is defined as

> the amount or magnitude of the system with regard to a suitable dimension

Depending on the type of system component to be emphasized by the definition, the size of a system can be defined and measured in many of the ways, several of which are described below.[13]

- the sum of the sizes of the system's subsystems (e.g., calculated recursively until the following definitions apply
- the amount of data stored in the system (e.g., measured in terms of megabytes)
- the amount of software in the system (e.g., measured in terms of thousands of source lines of code)
- the mass or weight of the physical parts of the system
- the number of functions performed by the system
- the number of requirements the system implements
- the physical dimensions of the physical components of system (e.g., measured in lengths, widths, heights, areas,[14] and volumes)

Scale

Based on the previous definition of size and ordered by increasing risk, system size forms a scale of systems that ranges from:

- Trivially Small Systems

  Trivially small systems are physically tiny and composed of only a very small number of components.
- Ultra-Large-Scale Systems

  Ultra-large-scale systems are systems of unprecedented scale in any of the following dimensions [Northrop 2006].
  - number of subsystems (or component systems)
  - number of computational elements (e.g., computers, mass storage, and network connectivity devices)
  - amount of software (e.g., measured in terms of lines of code)
  - amount of data stored, accessed, manipulated, and refined
  - number of hardware elements
  - number of connections and interdependencies between subsystems and other components
  - number of system purposes and user perceptions of these purposes
  - number of routine processes and interactions
  - amount of emergent behaviors and properties
  - number of stakeholders that are involved in some way

---

[13] Because a system can be composed of many different types of subsystems, defining, calculating, and estimating the overall system size can involve adding, apples, oranges, and other fruits and vegetables. Thus, it may be more useful to speak of a system's sizes rather than a system's size unless the system is composed of only one type of component (e.g., software or physical hardware). Thus, one could talk about the size of a vehicle in terms of its physical dimensions, weight, and amount of software it contains.

[14] The area a system takes up is often called the system's footprint. Length, area, and volume are especially important when the system has to be stored or located in a place where space is a premium such as within a spacecraft and aircraft or onboard a ship.

Examples

Ordered by increasing risk due to size, examples of systems categorized in terms of their degree of size include the following.

- Small Systems
  - televisions
  - vending machines
- Moderately Sized Systems
  - aircraft
  - cars
- Large Systems
  - aircraft fleets including ground-based flight planning, maintenance, support, and training systems
  - global positioning systems, including satellites
  - nuclear power plants
  - petroleum refineries
- Ultra-Large-Scale Systems
  - national air traffic control (ATC) systems
  - national electric power grids
  - national telecommunications systems

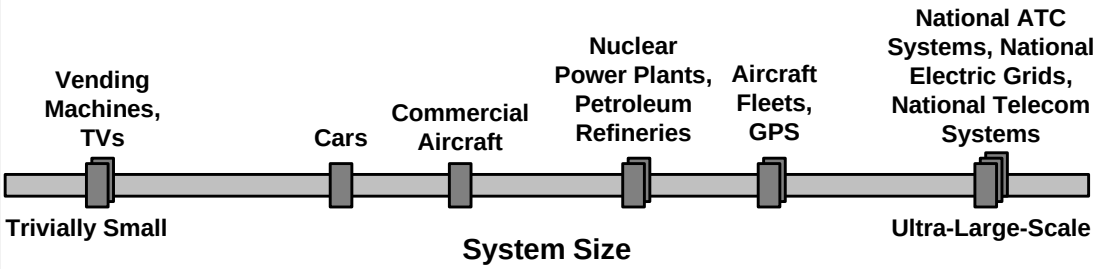

Figure 12: The System Size Meter with Example Systems

3.1.5 System Variability

Sometimes, either only one system of a given type exists (e.g., a space probe) or all of the systems of a given type are identical (e.g., a single type of automated teller machine). On the other hand, systems often come in multiple versions, variants, or configurations. For example, the systems may be instances of different models within a product line of systems (e.g., automobiles) or the systems may come in individualized versions (i.e., personalization) or country-specific versions (i.e., internationalization).

Similarly, it is often important to be able to easily, quickly, and inexpensively reconfigure a system by adding, modifying, or removing some of its subsystems. Although this reconfiguration usually requires some system maintenance to take place, it can happen automatically in certain cases if this level of reconfigurability is engineered into the system. For example, a system may

have a high level of variability if it has a service oriented architecture (SOA) that supports the automatic discovery of the subsystem that provides a newly needed service.

System evolution and system variability are closely related. System evolution discusses the existence of different versions over time, whereas system variability discusses the existence of multiple variants at essentially the same time.

Because they are so expensive to develop and operate, the number of any one type of ultra-large-scale system tends to be small. For this reason, ultra-large-scale systems tend to lie near or at the low end of the system variability scale. Similarly, systems of systems also tend to be fairly large and expensive and therefore often lie near the low end of the system variability scale.

Definition

System variability is defined as

> the degree to which a single type of system simultaneously exists in multiple variants, versions, or configurations

Scale

Based on the previous definition of variability and ordered by increasing risk, system variability forms a scale of systems that ranges from:

- Single Variant or Configuration Systems

  Such systems come in only a single variant or configuration.
- Systems with Very High Levels of Variability

  Such systems come in thousands of variants and configurations.

Examples

Ordered by increasing risk due to variability, examples of systems categorized in terms of their degree of variability include:

- Single Variant or Configuration Systems (No Variability)
  - Internet
  - World Wide Web
- Systems with Low Levels of Variability
  - nuclear power plants
  - robotic cars
  - unmanned aerial vehicles
- Systems with Moderate Levels of Variability
  - aircraft
  - vending machines
- Systems with High Levels of Variability
  - cars
  - national and regional air traffic control (ATC) systems

- national and regional electric power grids
- national and regional telecommunications systems
- petroleum refineries
- televisions

- Systems with Very High Levels of Variability
  - local area networks (LANs)
  - municipal electric power grids
  - wide area networks (WANs)

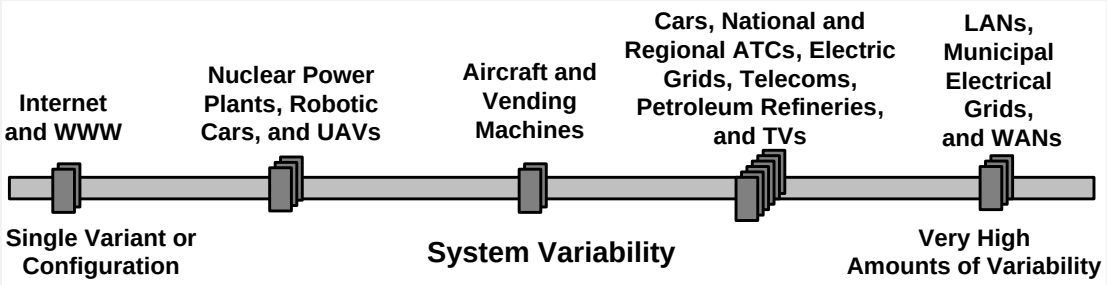

Figure 13: The System Variability Meter with Example Systems

3.2 Subsystem-Level Characteristics

In addition to variability due to system characteristics, systems also vary greatly due to the characteristics of their subsystems. Unlike the system characteristics that characterize the system as a black box (i.e., without referring to the system's subsystems), subsystem characteristics exist primarily at the subsystem level and therefore characterize the system in terms of its subsystems. These subsystem characteristics include:

- Autonomy
- Governance
- Heterogeneity
- Physical Distribution
- Reuse

3.2.1 Subsystem Autonomy

Systems vary greatly in terms of the degree to which their subsystems depend on and intraoperate with each other:

- **Interdependent Subsystems**

  Subsystems that are interdependent and closely collaborate to form a synergistic symbiosis

- **Independent Subsystems**

  Subsystems that are independent, stand alone and are individually useful, self-contained, and not controlled by other subsystems. Such autonomous subsystems are often only held together by a common user interface that provides access to the individual subsystems. Such collections of uncoupled subsystems are often referred to as virtual systems.

Although this variability may be due to the fact that some of the subsystems are truly independent of each other, it may also be due to the way the related subsystems are integrated together so as to eliminate or minimize the direct dependencies between them.

Subsystem coupling can be measured in terms of the number of interfaces between subsystems and the degree to which these interfaces do not violate the encapsulation of the subsystems (i.e., the degree to which *service providing* subsystems hide their implementation details from the *client* subsystems that depend on them). One way of producing systems composed of highly decoupled subsystems is to use a service oriented architecture (SOA).

Because their subsystems are typically independently developed legacy systems, both ultra-large-scale systems and systems of systems tend to lie at the low end of the *subsystem autonomy* scale.

Definition

Subsystem autonomy is defined as

> the degree to which the subsystems within a system are independent, stand alone and are individually useful, self-contained, and operationally independent (i.e., neither controlled by nor controlling other subsystems)

Scale

Based on the previous definition of subsystem autonomy and ordered by increasing risk, subsystem autonomy forms a scale of systems that ranges from:

- Systems Composed of Operationally Interdependent Subsystems

  Systems consisting of highly integrated subsystems whereby each subsystem depends on and has interfaces to the internals of many other subsystems
- Systems Composed of Operationally Independent Subsystems

  Systems consisting of a single common interface to collections of totally independent subsystems that are self-contained, useful by themselves, and operate independently of (i.e., neither control nor are controlled by) each other

Examples

Examples of systems categorized in terms of their degree of subsystem autonomy include:

- Systems with Low Levels of Subsystem Autonomy
  - Internet and World Wide Web
  - national air traffic control system
  - national electric power grid
  - robot swarm, which consists of multiple small robots that communicate via broadcast
- Systems with Moderate Levels of Subsystem Autonomy
  - aircraft including unmanned aerial vehicles (UAVs)
  - cars including robotic cars
  - televisions
  - vending machines

- Systems with High Levels of Subsystem Autonomy
  - insurance services (SOA)
  - logistics systems (SOA)
  - reservation systems (SOA)

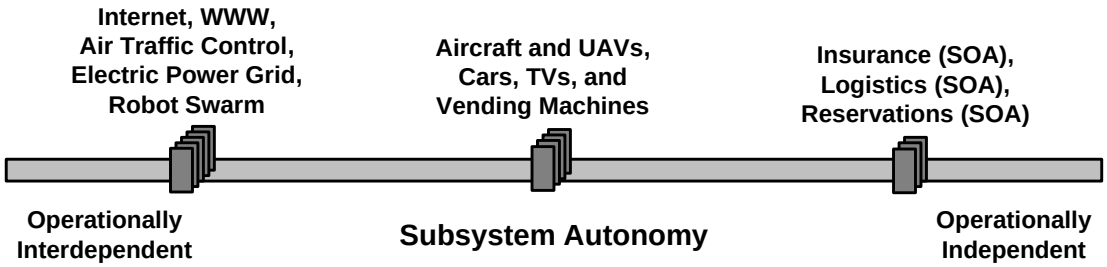

Figure 14: The Subsystem Autonomy Meter with Example Systems

3.2.2 Subsystem Governance[15]

Systems vary greatly in terms of the degree of independent governance of their subsystems. Subsystem governance is based on the degree to which the subsystems are specified, managed, funded, developed, owned, operated, maintained, and sustained independently of each other and of the overall system. The system with independently governed subsystems may have many unrelated stakeholders with various levels of authority over various parts (subsystems) of the system.

Although it is often quoted as the single most important characteristic of *systems of systems*, independent governance is in fact *neither* new *nor* specific to systems of systems. Independent governance occurs to some degree any time there is significant reuse. And reuse does not have to be at the level of a complete, independently useful subsystem for independent governance to be important. The reused components can be of any size from major subsystems and configuration items all the way down to individual small hardware (e.g., screws and fasteners) and software (e.g., single procedures or classes) units. The reason why governance is often cited as a SoS characteristic is that its ramifications, risks, and importance grow with the size of the component being reused. For example, reusable components tend to be developed and maintained independently of the overall system in which they are going to be reused, and this can lead to major scheduling and compatibility problems.

Both ultra-large-scale systems tend to be made from pre-existing subsystems with independent governance because such systems are too large and expensive to be developed from scratch and therefore tend to incorporate pre-existing legacy systems as subsystems, whereby these subsystems have been developed and are governed independently. Similarly, many systems of systems tend to incorporate a lot of reuse of relatively large components and this reuse often implies independent governance. Thus, ultra-large-scale systems and systems of systems tend not to be at the low end of the system governance scale.

---

[15] Note that governance is a subsystem rather than system characteristic because it is defined in terms of the governance of the system's subsystem more than the governance of the system. Also unlike the other system and subsystem characteristics, system governance has more to do with the organizations governing the system's subsystems than the systems characteristics themselves. As such, it could just as easily been grouped with the programmatic system characteristics based on the organizations, stakeholders, and endeavors associated with the system and its subsystems.

Definitions

Subsystem governance is defined as

> the degree to which the subsystems of a system are governed (e.g., specified, managed, funded, developed, owned, operated, maintained, and sustained) in a independent, decentralized, and uncoordinated manner

According to reports written in 1998 and 2008 [Meier 1998, DOD 2008], the following are ways to categorize systems of systems in terms of how their subsystems are governed.[16]

> **Directed Systems (of [Sub]systems)**
> centrally governed systems, the subsystems of which are governed in a centralized and coordinated manner as part of the system
>
> **Acknowledged Systems (of [Sub]systems)**
> systems that have their own objectives, management, funding, and authority, but the subsystems of which retain their own independent management, funding, and authority in parallel with the overall system
>
> **Collaborative Systems (of [Sub]systems)**
> systems without centralized objectives, management, authority, responsibility, and funding, the subsystems of which are voluntarily governed to support the overall system to address shared or common interests
>
> **Virtual Systems (of [Sub]systems)**
> systems, the subsystems of which are independently governed in a completely distributed and uncoordinated manner as stand-alone systems

The preceding four categories are not as distinct as their definitions imply. Few systems are purely directed, acknowledged, collaborative, or virtual. For example, if a *single* large system is composed of new or pre-existing subsystems of two or more of any of the following types of subsystems with regard to subsystem governance, then what kind of system is it?

- Subsystems (as in a pure *Directed* System or System of Systems) that are
  - created specifically to be part of the system
  - directly governed as part of the overall system
  - governed by a single acquirer, single developer, etc.
- Subsystems (as in a pure *Acknowledged* System or System of Systems) that are
  - not created specifically to part of the system
  - separately governed in accordance with *common* policies, directives, and/or contracts
  - governed by subcontractors, different business units of a single prime contractor or system integrator, or sister organizations
- Subsystems (as in a pure *Collaborative* System or System of Systems) that are
  - not created specifically to part of the system
  - separately and voluntarily governed in accordance due to enlightened self-interest

[16] Note that this way of categorizing systems of systems applies equally to all systems although how a system's subsystems are governed is a major characteristic used in many definitions of systems of systems.

  - governed by different business units of a single prime contractor or system integrator, sister organizations, or vendors
- Subsystems (as in a pure *Virtual* System or System of Systems) that are
  - not created specifically to part of the system
  - governed in a completely independent manner
  - governed by independent vendors or competitors (e.g., COTS)

Although how a system's subsystems is governed is a major characteristic used in many definitions of systems of systems, the preceding four ways to categorize systems including systems of systems apply to all systems.

Because systems of independently governed subsystems are typically the largest, most difficult, and highest risk systems of systems to engineer, governance is typically a key aspect of various definitions of systems of systems. Note that because their subsystems are typically separately developed and governed systems, ultra-large-scale systems of systems also tend to lie at the high end of the *subsystem governance* scale.

Scale

Based on the previous definition of governance and ordered by increasing risk, system governance forms a scale of systems that ranges from:

- Directed Systems with Centrally Governed Subsystems

  Systems, the subsystems of which are governed in a centralized and coordinated manner as part of the system
- Virtual Systems with Independently Governed Subsystems

  Systems, the subsystems of which are governed as independent systems in a distributed and uncoordinated manner

Examples

Ordered by increasing risk due to governance, examples of systems categorized in terms of their degree of subsystem governance include the following.

- Directed Systems of [Sub]Systems (i.e., Systems of Centrally Governed Subsystems)
  - Directed Systems with Single Development/Maintenance Organization

    Single developer and maintainer for the system and all of its subsystems with one or more system owners and operators:

    Digital Watches
  - Directed Systems with Single Developer Organization and multiple Subsystem Vendors

    Single developer for the system and most subsystems with vendors for the other subsystems

    small unmanned aerial vehicles (UAVs)

    vending machines
  - Directed Systems with Prime, Subcontractors, and Vendors

Prime Contractor or system integrator for system plus subcontractors and vendors for many of the subsystems:

aircraft
cars
nuclear power plants
petroleum refineries
television sets

- Acknowledged Systems of [Sub]Systems (i.e., Systems of Independently Governed Subsystems)

  Centrally governed overall system with independently governed subsystems

  - global positioning systems (GPS)
  - many individual military systems of systems

- Collaborative Systems of [Sub]Systems (i.e., Systems of Collaboratively Governed Subsystems)

  Policy organization for system with independent but collaborating contractors, subcontractors, and vendors for the subsystems:

  - Internet
  - national air traffic control system
  - national electric power grid

- Virtual Systems of [Sub]Systems (i.e., Systems of Independently Governed Subsystems)

  Ad hoc system with independent competing prime contractors, subcontractors, and vendors for the subsystems:

  - World Wide Web

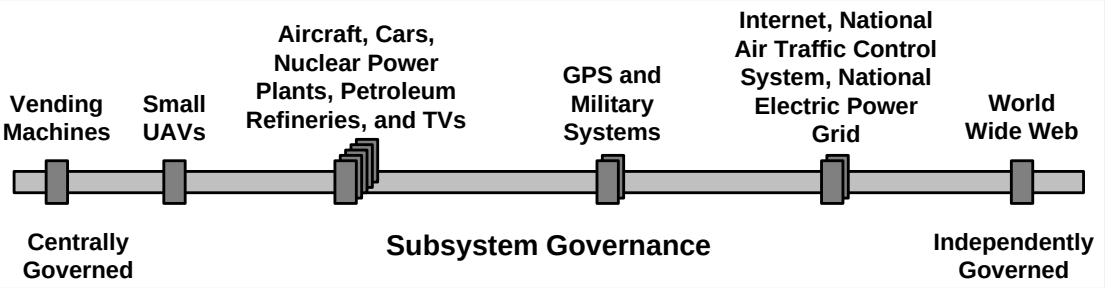

Figure 15: The Subsystem Governance Meter with Example Systems

3.2.3 Subsystem Heterogeneity

Systems vary greatly with regard to the degree to which their subsystems differ from each other. There are many ways in which the subsystems of a system may be homogeneous or heterogeneous including, but not limited to, their:

- Application Domain. Subsystems may vary by their application domain such as aviation, banking, finance, government, insurance, pharmacology, telecommunications, transportation, or weapons.

- Subsystem Type. Subsystems may vary by type such as whether the subsystems include (or consist solely of) data, hardware, software, equipment, people or organizations, facilities, and manual procedures.
- Subsystem Technology. Subsystems may vary in terms of the technologies with which they are developed such as materials, nanotechnology, and software technologies (e.g., CORBA, relational or object databases, Java, .NET, Service Oriented Architecture).

Because their size tends to make their subsystems relatively heterogeneous, both ultra-large-scale systems and systems of systems tend to lie at the high end of the *subsystem homogeneity* scale.

Definitions

Subsystem heterogeneity is defined as

> the degree to which the subsystems of a system differ from each other in that they (1) have different goals, objectives, and requirements, (2) have different behavior and characteristics, (3) provide unrelated functionality, (4) belong to different application domains, and (5) are implemented using different technologies

Scale

Based on the previous definition of heterogeneity and ordered by increasing risk, system heterogeneity forms a scale of systems that ranges from:

- Systems with Completely Homogeneous Subsystems

  Systems, the subsystems of which are of the same type, have similar goals and objectives, belong to the same application domain, or are implemented using the same technology
- Systems with Completely Heterogeneous Subsystems

  Systems, the subsystems of which are not of the same type, have dissimilar goals and objectives, do not belong to the same application domain, or are implemented using different technologies

Examples

Ordered by increasing risk due to heterogeneity, examples of systems categorized in terms of their degree of subsystem heterogeneity include the following.

- Systems with Highly Homogeneous Subsystems
  - fleets of aircraft
- Systems with Moderately Heterogeneous Subsystems
  - global positioning systems (GPS)
  - national air traffic control systems
  - vending machines
- Systems with Highly Heterogeneous Subsystems
  - aircraft
  - cars
  - Internet

- national electric power grids (especially a smart grid)
- televisions
- World Wide Web

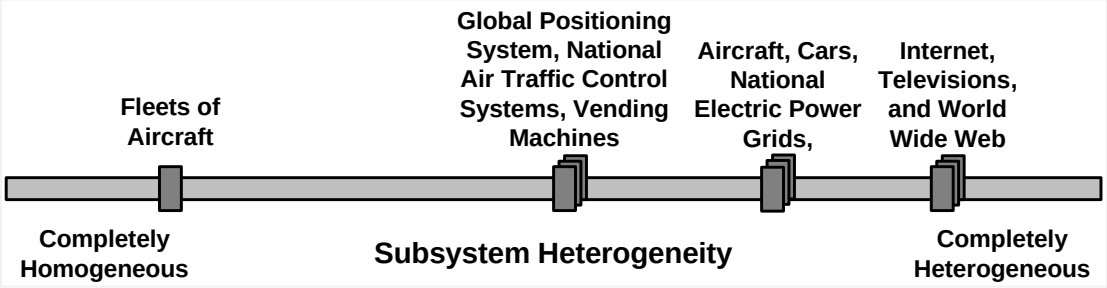

Figure 16: The Subsystem Heterogeneity Meter with Example Systems

3.2.4 Subsystem Physical Distribution

Many systems are contiguous with all of their subsystems physically touching. Such systems of collocated subsystems may be contrasted with distributed systems, the subsystems of which are physically located in different places. The subsystems of software-intensive systems can be integrated with local area networks (LANs), metropolitan area networks (MANs), wide area networks (WANs), national area networks (NANs), global area networks (GANs), and even interplanetary networks.

Ultra-large-scale systems tend to lie at the higher ends of the *subsystem physical distribution* scale because they are so large and consist of so many subsystems, at least some of which are typically distributed physically.

Systems of systems also tend to lie at the higher end of the *subsystem physical distribution* scale because it is likely that most of their previously existing, individually useful, independently governed systems exist in different physical locations.

Definition

Subsystem physical distribution is defined as

> the degree to which the subsystems of a system exist in different physical locations

Scale

Based on the previous definition of physical distribution and ordered by increasing risk, system geographical distribution forms a scale of systems that ranges from:

- Contiguous Systems

  Contiguous systems are systems, the subsystems of which are all collocated in the same physical place.
- Extremely -Distributed Systems

Extremely distributed systems are systems, the subsystems of which are physically distributed to vastly separate locations both on and off the planet.

Examples

Ordered by increasing risk due to physical distribution, examples of systems categorized in terms of their degree of subsystem physical distribution include:

- Systems with Contiguous Subsystems
  - aircraft
  - production cars
  - robotic cars
  - television sets
  - unmanned aerial vehicles (UAVs)
  - vending machines
- Systems with Subsystems Distributed across one or more Buildings
  - local area networks (LANs)
  - manufacturing lines
- Systems with Subsystems Distributed across Cities
  - metropolitan area networks (MANs)
  - municipal smart grids
- Regionally Distributed Systems (across one or more states)
  - petroleum pipelines
  - regional electrical power grids
- Nationally Distributed Systems
  - national air traffic control systems
  - national electrical power grids
  - wide area networks (WANs)
- Globally Distributed Systems
  - fleets of aircraft (plus ground-based training and support subsystems)
  - Internet
  - World Wide Web
- Systems with Extremely Distributed Subsystems
  - global positioning systems (GPS) including satellites
  - space exploration systems (with space probes, communications, and ground control)

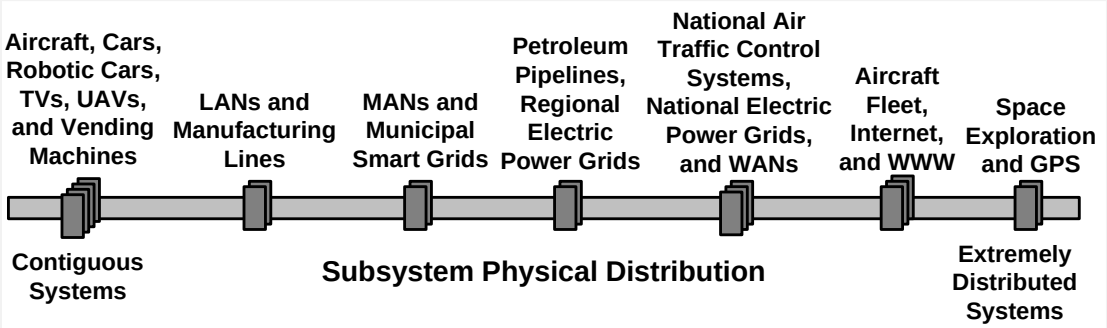

Figure 17: The Subsystem Physical Distribution Meter with Example Systems

3.2.5 Subsystem Reuse

Although essentially all systems exhibit some degree of reuse in terms of the architecture, design, and implementation of their subsystems, the amount of reuse can vary greatly from system to system. Note that the size and scope of the reused components can also vary from the quite small hardware or software part to entire subsystems or groups of subsystems.[17] Finally, the layer in a layered architecture in which the reused components exist can vary (e.g., underlying hardware, operating system, middleware, data, application, and user interface). The maximum size and level of reuse can be used to categorize systems as follows.

- Systems Having Pre-Existing Units as Subsystems

  Systems, the primary (largest) reused subsystems of which are individual units:

  - [computer] software units (CSUs) such as procedures or classes of objects
  - Hardware parts
- Systems Having Pre-Existing Components as Subsystems

  Systems, the primary (largest) reused subsystems of which are components:

  - [computer] software components (CSCs) consisting of multiple CSUs
  - Hardware assemblies consisting of multiple hardware units
- Systems Having Pre-Existing Configuration Items as Subsystems

  Systems, the primary (largest) reused subsystems of which are configuration items:

  - [computer software] components (CSCIs) consisting of multiple CSCs
  - hardware configuration items (HWCIs) consisting of multiple hardware components
- Systems Having Pre-Existing Systems as Subsystems

  Systems, all of the reused subsystems of which are major pre-existing systems in their own right

Many of the issues raised by the development of systems of systems built from previously existing, independently governed systems are the very same issues that are involved in the reuse of previously existing components at any level. For example, coordinating diverse develop-

[17] For example, the maximum size and abstraction level of the software components being reused has increased (more or less) steadily over the years from procedures, classes of objects, operating systems and databases, middleware, frameworks, services to entire software applications and systems. Similar trends have occurred with hardware and subsystems containing hardware, software, and data etc.

ment/maintenance schedules and managing independent and often competing stakeholders and their associated requirements.

Because they typically incorporate large numbers of legacy systems as subsystems and therefore are more driven by bottom-up availability of these subsystems than top-down by requirements, both ultra-large-scale systems and systems of systems tend to lie at the high end of the *subsystem reuse* scale. Such systems tend to be driven by the availability of pre-existing subsystems rather than by well engineered requirements.

Definition

Subsystem reuse is defined as

> the degree to which the subsystems of the system have been reused regardless as to whether they are commercial-off-the-shelf (COTS), government-off-the-shelf (GOTS), military-off-the-shelf (MOTS), organizational-internal reuse, open source, and freeware

Scales

Based on the definition above of reuse and ordered by increasing risk, system amount reuse forms a scale of systems that ranges from:

- Systems with Essentially 0% Reuse (Requirements-Driven "Greenfield" Development). Systems, all of the subsystems of which were (or are being) developed specifically for the associated system
- Systems having 100% Pre-existing Subsystems (Availability- and Reuse-Driven Development). Systems, all of the subsystems of which are pre-existing and reused

Examples

Ordered by increasing risk due to reuse, examples of systems categorized in terms of their degree of subsystem reuse include:

- Systems with Essentially 0% New Subsystems
  - [vanishingly rare for any nontrivial system]
- Systems with Low Levels of Reuse
  - advanced space probes
  - hospital pharmacy systems
- Systems with Moderate Levels of Reuse
  - aircraft
  - air traffic control systems
  - nuclear power plants
- Systems with High Levels of Reuse
  - automated teller machines
  - cars
  - elevators

  - national smart power grid
  - televisions
  - vending machines
- Systems with Essentially 100% Reuse
  - Internet
  - simple websites generated using a website development tool designed for use by non-technical developers

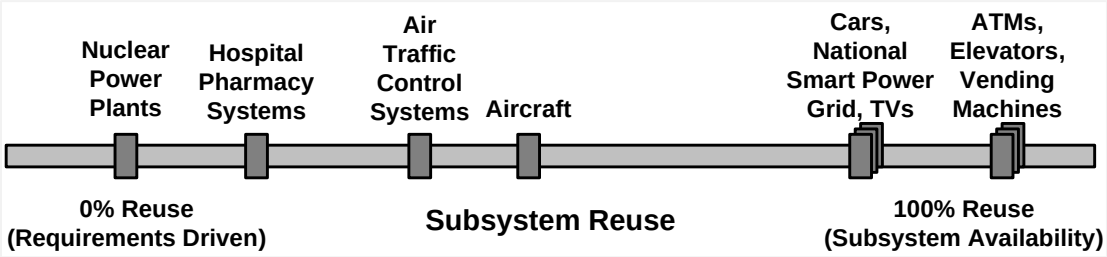

Figure 18: The Subsystem Reuse Meter with Example Systems

3.3 Correlations Between Characteristics

Although the preceding system and subsystem characteristics describe different system properties, these characteristics are far from independent of each other. Instead many of these different characteristics tend to be positively correlated with each other so that increases in one characteristic tend to imply increases in many of the other characteristics. This is one reason why both ultra-large-scale systems and systems of systems both tend to lie near the high end of the scales associated with the system of systems characteristics.

Similar correlations occur between the different quality characteristics. For example, increases in security tend to be inversely correlated with increases in performance, while increases in availability tend to be positively correlated with increases in reliability and maintainability.

For example, as System Complexity increases:

- Evolution tends to increase because there are more components and relationships between components that are subject to change. Actually, this increase is in the need for evolution rather than evolution itself because as the system becomes more complex, it becomes more difficult, expensive, and time consuming to change the system without introducing defects.
- Negative Emergence tends to increase because complexity makes it more difficult to predict what emergence will occur and the emergence is more likely to be negative because it becomes more difficult to avoid defects and unwanted side effects.
- Size tends to increase because larger systems tend to have more components and more relationships between these components. This is actually correlation rather than cause, because it is the increase in size that usually causes the increase in complexity.
- Subsystem Heterogeneity tends to increase because differences between subsystems is part of the definition of system complexity

In Figure 19, the large preponderance of plus signs over minus signs makes it clear that the ten SoS characteristics tend to be positively correlated, that is, they all tend to occur together. This is a major reason why they all have shown up in different definitions of the same term: system of systems. Some of these characteristics may be more important than others. Similarly, some may be more foundational and others may be more derived.

3.4 Foundation vs. Derived System of Systems Characteristics

Given that there are so many SoS characteristics and that they are correlated, then it may be that some of these characteristics are foundational and are therefore essential aspects in the definition of the concept of system of systems. On the other hand, some of the SoS characteristics are derived in the sense of being implied, either directly or indirectly, by the other characteristics.

Figure 19 shows the causal relationships between the SoS characteristics and also shows that the most foundational SoS characteristic appears to be subsystem reuse. If a system is essentially composed of a set of pre-existing subsystems (i.e., the system lies at the high end of the *subsystem reuse* scale), then the system tends[18] to have high levels of almost all of the other system of systems characteristics, listed below.

- **High Level of Subsystem Reuse**

  If a system is primarily or completely composed of pre-existing subsystems that are individually useful and reusable, then the system *tends* to have:

  – at least a moderately high level of subsystem heterogeneity

    The system tends to have at least a moderately high level of subsystem heterogeneity because the different subsystems reused will tend to be different from each other.

  – a large system size

    The system's size tends to be considerably larger than any of its component reused subsystems.

  – a high level of subsystem governance

    The system tends to have a high level of subsystem governance because its subsystems tend to come from multiple, possibly competing, sources and thus tend to be governed independently of the overall system and each other.

  high level of system evolution

    If the system is large and its subsystems are independently governed (especially in terms of maintenance and major upgrades), then a relatively large number of subsystems will be evolving independently of each other and the probability increases that at least one of them will be evolving at any given time.

  higher levels of subsystem variability

    The system has at least a moderately high level of system variability because it is likely that if the subsystems were independently developed to be reusable, then they are likely to have standard interfaces which increases the likelihood that they can be replaced by other subsystems, possibly from competing suppliers.

[18] Note that all of the relationships in Figure 20 represent tendencies and thus are probable rather than certain. It is possible though unlikely that some of the arrows on Figure 20 do not exist for certain systems because the levels of some SoS characteristics are independent of each other.

somewhat higher levels of subsystem physical distribution

If the system has high levels of subsystem governance and autonomy, then there is a small tendency for some of these autonomous and independently governed subsystems to be at different physical locations.

- at least a moderately high level of subsystem autonomy

  The system tends to have at least a moderately high level of subsystem autonomy because the reusable subsystems were probably not developed to be directly interoperable with each other, leading to the selection of an architecture that decouples them such as a Service-Oriented Architecture (SOA).

- **High Level of System Complexity**

  A system tends to have a high level of *system complexity* if it is large, rapidly evolving, exists in multiple variants, and consists of reusable subsystems that are heterogeneous, independently governed, and physically distributed. It is difficult to understand systems that exhibit these properties.

- **High Level of Negative Emergence**

  A system tends to have a high level of *negative emergence* if it is large, complex, complex, rapidly evolving, exists in multiple variants, and consists of reusable subsystems that are autonomous, heterogeneous, independently governed, and physically distributed. Under such circumstances, it is difficult to predict and avoid unexpected and detrimental system behaviors and characteristics.

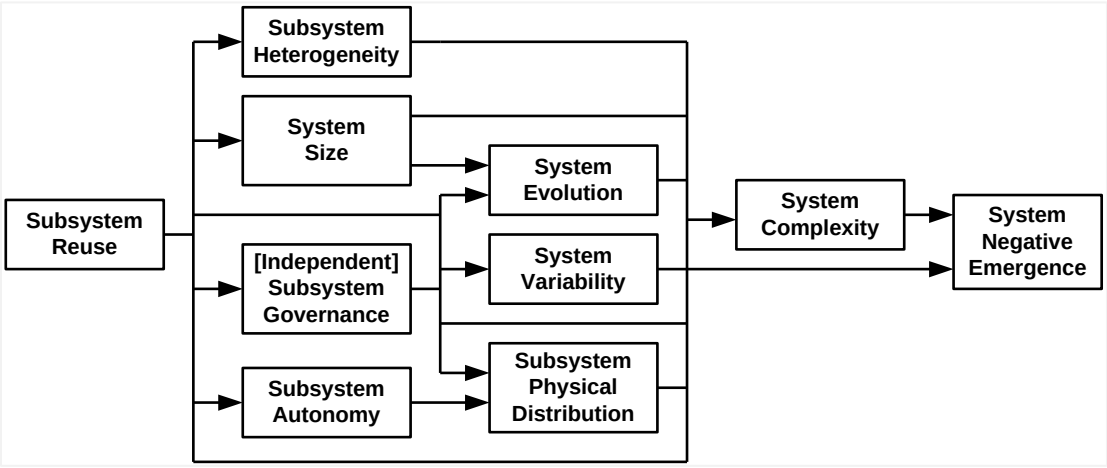

Figure 19: Foundational (Left) vs. Derived (Right) SoS Characteristics

To summarize, *system reuse* seems to be the most fundamental of the SoS characteristics because it tends to cause the system to exhibit all of the other characteristics. Thus, reuse is actually more foundational than governance when it comes to identifying systems of systems because reuse of [sub]systems is the primary cause of independent *subsystem governance*. Interestingly, *system complexity* is the primary cause of *system negative emergence* and complexity is increased by all of the other SoS characteristics. This explains perhaps why emergence is such a common component of the definitions of systems of systems.

4 Example Systems and Their SoS Characteristics

This section uses meters based on SoS characteristics to profile four systems that vary from the relatively trivial to the ultra-large-scale systems:

- refrigerated vending machines
- luxury automobiles
- strike fighter aircraft fleet
- national smart electric grid

4.1 Refrigerated Vending Machines

A refrigerated vending machine is a simple system that sells (i.e., vends) cold drinks or food. A typical refrigerated vending machine

- Dispenses items on payment.
- Keeps perishable foods and drinks at an appropriately cold temperature,
- Makes and dispense change.
- Records sales and system status.
- Prevents theft of products and cash.

A typical refrigerated vending machine primarily consists of the following components.

- a cabinet consisting of:
  - an outer casing typically made of galvanized steel that is colored using acrylic powered coatings (which hold up better than paint)
  - a door including a locking mechanism, dispensing bin, and front panel that is typically made of a clear tough polycarbonate plastic
  - an inner steel liner called a tank
  - insulation between the casing and tank that is typically made of polyurethane foam
- a controller consisting of microprocessor, software, keypad, digital display, and power supply for providing power to vend solenoids and sensors
- a control panel for item selection including keypad and LCD panel for cost and system status information
- a door assembly for loading products into the machine and servicing the vending machine; the door assembly includes a lock
- an electric power distribution box that provides power to the motors, lights, and refrigeration unit consisting of a transformer, fuses, main power switch, and three-wire electric power supply cord
- multiple feeder tray assemblies that hold the products (e.g., perishable food and drink cans or bottles) to be sold, each of which consist of wire spirals to move products, an electric motor to turn the wires, a vend solenoid to control the electric motor, and a sensor to determine if a tray is empty

- a payment assembly including card, coin, and bill validators, change maker and coin return, and coin and bill holders
- a refrigeration unit for keeping the products cold; the unit consists of a thermostat, compressor, condenser fan, evaporator fan, start capacitor, start relay, etc.

A vending machine is obviously not an ultra-large-scale system and meets very few of the criteria for a system of systems.

System of Systems Characteristics

Refrigerated vending machines exhibit the following important SoS characteristics.

- **System Complexity**

  A refrigerated vending machine lies at the low end of the *System Complexity* scale and falls into the *Simple Systems with Low Levels of Complexity category.* Although the previous (incomplete) component list shows that vending machines are not trivial, they are nevertheless relatively simple, consisting of a relatively small number of components that individually are relatively simple and that are connected in a simple and straightforward manner.

- **System Evolution**

  An individual refrigerated vending machine lies near the low end of the *System Evolution scale* and falls into the *Low Rates of Evolution category.* It is typically subject to a relatively low level of corrective maintenance (e.g., replacement of a failed component). Significant changes typically occur with new models of vending machines rather than major upgrades to existing vending machines.

- **System Negative Emergence**

  A refrigerated vending machine lies at the low end of the *System Negative Emergence scale* and falls into the *Systems with Low Levels of Negative Emergence category.* Vending machines have one well-known negative emergent behavior. Because most bill validators will not return a valid bill, a vending machine can be misused (from the viewpoint of the owner) as a change-making machine if the customer cancels the purchase instead of selecting an item.

- **System Size**

  A refrigerated vending machine lies at the low end of the *System Size scale* and falls into the *Small Sized Systems category*. Vending machines are quite small by all measurement methods including numbers of subsystems, amount of data stored, amount of software, weight of physical parts, number of functions performed, number of requirements, and physical dimensions.

- **System Variability**

  Refrigerated vending machines lie at the high end of the *System Variability scale* and fall into the *Systems with High Levels of Variability category*. This is because there are many different types of vending machines and many different models of vending machines, which are available from different vendors. New versions of these models also come out every few years.

- **Subsystem Autonomy**

  A refrigerated vending machine lies near the middle of the *Subsystem Autonomy scale* and falls into the *Systems with Moderately Autonomous Subsystems category*. The components of a refrigerated vending machine are moderately coupled in terms of physical, electric, and digital interfaces.

- **Subsystem Governance**

  A refrigerated vending machine lies near the low end of the *Subsystem Governance scale* and falls into the *Directed Systems (with Single Developer Organization and Multiple Subsystem Vendors) category*. Although individual competing businesses build vending machines, some of the electronic components are purchased from specialized vendors.

- **Subsystem Heterogeneity**

  A refrigerated vending machine lies near the middle of the *Subsystem Heterogeneity scale* and falls into the *Systems with Moderately Heterogeneous Subsystems category*. On the one hand, it contains eight quite different major subsystems (types) including some that are systems in their own right (the refrigeration unit and payment assembly), structural elements, sensors, motors, and a microprocessor with software. On the other hand, it contains a large number of identical feeder tray assemblies.

- **Subsystem Physical Distribution**

  A refrigerated vending machine lies at the low end of the *Subsystem Physical Distribution scale* and falls into the *Systems with Contiguous Subsystems category*. All of the subsystems of a refrigerated vending machine are physically touching each other.

- **Subsystem Reuse**

  A typical refrigerated vending machine lies at the high end of the *Reuse scale* and falls into the *Systems with Essentially 100% Reuse category*. Except for the manufacturing of essentially standard cabinets, the profit margin is so low and the competition between vending machine producers is so strong that it is not cost effective to develop new internal components. Vending machines essentially form a product line of systems. Note that this massive reuse is a major reason why it is typically not necessary to engineer rigorous requirements for new vending machines.

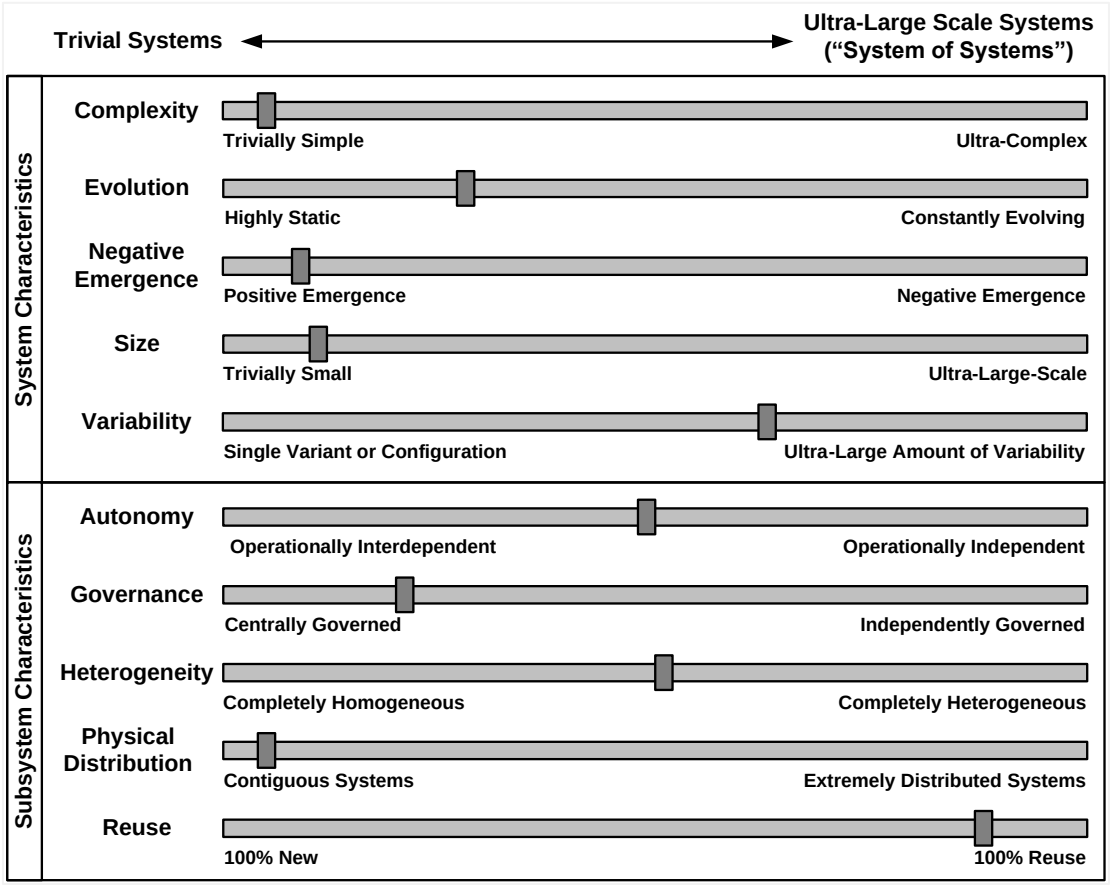

Figure 20: Meters for the System Characteristics of Refrigerated Vending Machines

4.2 Hybrid Electric Cars

A car is a small, self-propelled, four-wheeled passenger land vehicle that is designed primarily[19] for the private as opposed to public transportation of up to eight seated passengers. A hybrid electric car is a car that is primarily powered by electric motors and storage batteries, but also has a small gas engine for recharging the batteries. Examples of hybrid electric cars are the Toyota Prius, Honda Civic Hybrid, Lexus HS, Mercedes S400, BMW S6 Hybrid, and Porsche Cayenne S.

A hybrid electric car performs the following functions:

- transports a small number of passengers and their possessions
- provides high gas mileage via the use of electricity and regenerative breaking
- provides the typical features (e.g., air conditioning, heating, radio) of a car powered by an internal combustion engine
- provides adequate levels of safety (e.g., via air bags, seat belts, traction control, antilock braking, etc.)

[19] Unlike busses which are designed for public transportation, only a relatively small percentage of cars are used as taxis.

A hybrid electric car primarily includes, but is not limited, to the following subsystems:

- auto body
- battery system
- electrical system
- electric motors
- electronics system (approximately 75 electronic control units with approximately 1GB of software not including GPS navigation data)
- engine cooling system
- exhaust and emissions system
- gas engine
- heating and air conditioning system
- fuel system
- regenerative braking system
- safety system
- seat control system
- sound system
- steering system
- wheels and tires

System of Systems Characteristics

Although a hybrid electric car is not an ultra-large system, it is by some definitions a system of systems that exhibits the following SoS characteristics:

- **System Complexity**

  A hybrid electric car lies in the middle of the *System Complexity scale* and falls into the *Systems with Moderate Levels of Complexity category* or *Systems with High Levels of Complexity category* depending on whether it is a low-end, medium, or high-end car.

- **System Evolution**

  A hybrid electric car lies in the middle of the *System evolution scale* and falls into the *Systems with Moderate Rates of Evolution category.* An individual car typically changes via replacement of tires, oil, windshield wipers, batteries, worn out parts, and defective parts (e.g., due to recalls).

- **System Negative Emergence**

  A hybrid electric car lies in the middle of the *System Negative Emergence scale* and falls into the *Systems with Moderate Levels of Negative Emergence category.* If they come into widespread use, hybrid electric cars can place such a great demand on electricity so as to necessitate the development of many new electric power plants (with its associated pollution) and significant upgrades to the national, regional, and municipal electric grids. Like traditional cars with internal combustion engines, hybrid electric cars can contribute to traffic jams, urban sprawl, excessive commute times, and tens of thousands of fatal motor vehicle accidents.

- **System Size**

A hybrid electric car lies just below the middle of the *System Size scale* and falls into the *Moderately Sized Systems category*. Physically, cars are moderately sized in terms of length, width, height, and weight. Cars are somewhat above moderate size in terms of numbers of components. Luxury cars (including hybrids) incorporate approximately 75 electronic control units (ECUs) with approximately 1GB of software (not including GPS navigation data) incorporating more than 100 million object code instructions [Ebert 2009].

- **System Variability**

  A hybrid electric car lies above the middle of the *System Variability scale* and falls into the *Systems with a Large Amount of Variability category.* There are several makes of hybrid vehicles, each of which has multiple models of hybrid vehicles, and the different models can have different colors and accessories.

- **Subsystem Autonomy**

  A hybrid electric car lies near the middle of the Subsystem Autonomy scale and falls into the Systems with Moderate Levels of Subsystem Autonomy category. The system components are coupled via physical, electric, mechanical, and software interfaces.

- **Subsystem Governance**

  A hybrid electric car lies near the low end of the *Subsystem Governance scale* and falls into the *Directed Systems (with Prime, Subcontractors, and Vendors) category*. The levels of subcontractors can be quite deep. For example, one Japanese auto manufacturer had 170 primary subcontractors that consigned parts manufacturing to 4,700 secondary subcontractors that enlisted 31,600 tertiary subcontractors [Okimoto 1988].

- **Subsystem Heterogeneity**

  A hybrid electric car lies near the high end of the *Subsystem Heterogeneity scale* and falls into the *Systems with Highly Heterogeneous Subsystems category*. A hybrid electric car contains many subsystems that have different application domains (e.g., electricity, electronics, ergonomics, propulsion, safety, and structures) and technologies (e.g., batteries, electric motors, and real-time software).

- **Subsystem Physical Distribution**

  A hybrid electric car is at the low end of the *Subsystem Physical Distribution scale* and falls into the *Systems with Contiguous Subsystems category*. All of a system's subsystems are in physical contact with one another.

- **Subsystem Reuse**

  A hybrid electric car lies just below the middle of the *Subsystem Reuse scale* and falls into the *Systems with Moderate Levels of Reuse category.*[20] Although a hybrid electric car involves the development of a significant amount of new components and software, it nevertheless reuses a moderate amount of components from other car models. For example, the Honda Civic Hybrid shares many components with the other traditionally powered Civics. This can be contrasted with traditional cars that exhibit major reuse.

[20] Unlike hybrid electric cars, traditional cars with internal combustion engines fall into the systems with major reuse category because there are fewer new components to develop.

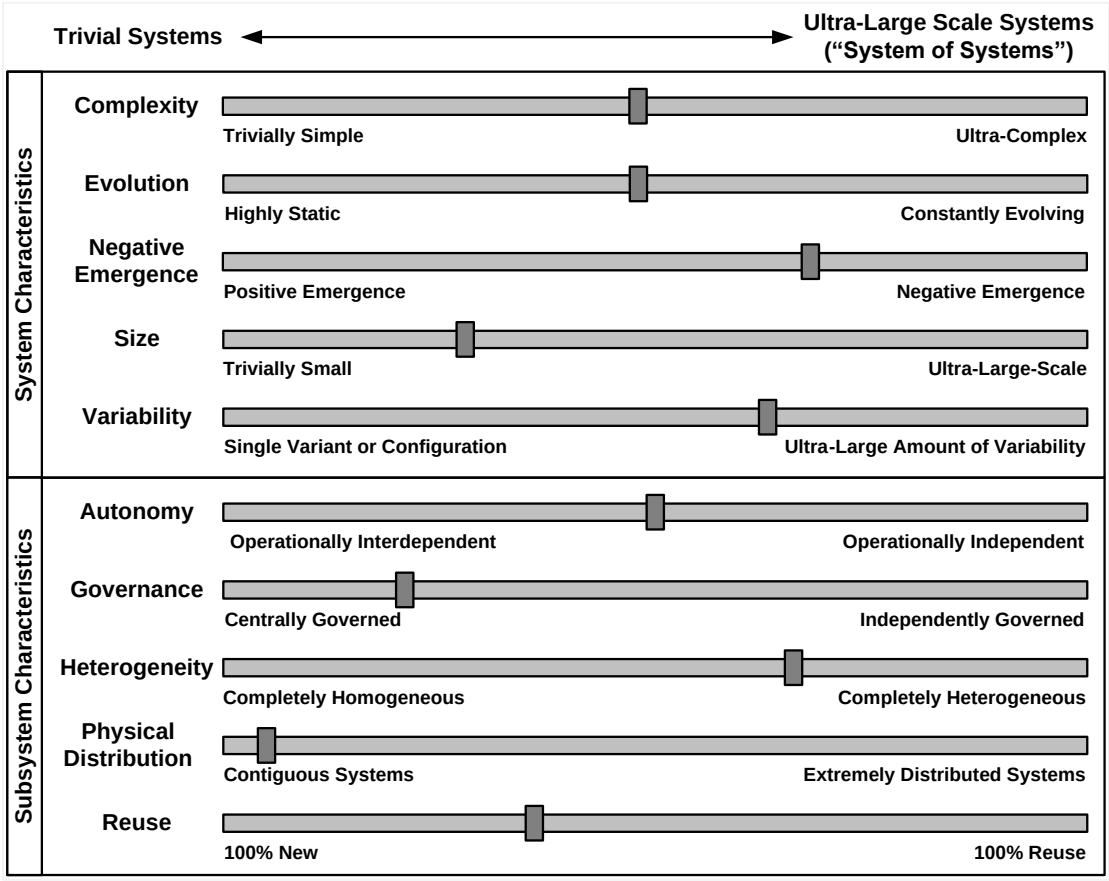

Figure 21: Meters for the System Characteristics of Hybrid Electric Cars

4.3 Strike Fighter Aircraft

A strike fighter is a military aircraft that is capable of precision attacks (a.k.a., "surgical strikes") on surface targets including ships while remaining sufficiently maneuverable and well armed with both air-to-air weapons and countermeasures to defend itself in air combat. Examples of strike fighters include the F-35 (Lightning II, formerly the Joint Strike Fighter), F-15E (Strike Eagle), the European Panavia Tornado IDS (Interdictor/Strike), the Russian Sukhoi Su-30 (Flanker-C), and the Chinese Xian JH-7 (Flounder or Flying Leopard).

A typical strike fighter aircraft can be viewed either of two ways:

- A strike fighter is only the physical aircraft itself, which is part of (and separate from) a larger system of systems that also includes ground-based assets such as:
  - a training system consisting of a training management system, training materials, one or more training facilities, and various pilot and maintainer simulators
  - a mission planning and debriefing system
  - a maintenance management system
  - a resupply system
- A strike fighter is the larger system of systems, only one part of which is the physical aircraft.

System of Systems Characteristics

A state of the art military strike aircraft is a large-scale system and is by some definitions a system of systems that exhibits the following SoS characteristics:

- **System Complexity**

  A military strike aircraft lies near of the *System Complexity scale* and falls into the S*ystems with High Levels of Complexity category.* A military strike aircraft is very complex, consisting of a great many complex components, especially its extensive software.

- **System Evolution**

  A military strike aircraft lies somewhat above the middle of the *System Evolution scale* and falls into the *Systems with Moderate Rates of Evolution category.* A military strike aircraft will evolve over time as new releases (primarily software) provide new functionality and technology refresh updates occur. Due to the very high cost of updates (including testing), these iterations are carefully planned and coordinated at appropriate time intervals. Maintenance updates to replace parts nearing the end of useful life occur significantly more often.

- **System Negative Emergence**

  It is very difficult to predict the level of negative behaviors and characteristics that will emerge from a system as complex as a military strike aircraft. Such systems cannot by their very nature be either exhaustively specified or exhaustively tested in spite of the great amount of requirements engineering and testing (e.g., unit, integration, ground and airborne laboratory testing, specialty engineering testing, flight testing, and operational testing) that is performed. On the one hand, the size and complexity of strike aircraft would imply a relatively high-level of negative emergence. On the other hand, the great deal of time, effort, and funds invested in the development of a strike aircraft would imply a relatively low level of negative emergence. Experience does reveal a non-trivial number of defects causing failures of one degree or another during flight testing, operational testing, and operations. A reasonable compromise might be to estimate that a new military strike aircraft would lie between the low end and middle of the *System Negative Emergence scale* and thus fall into the *Systems with Moderate Levels of Negative Emergence category.*

- **System Size**

  A military strike aircraft lies in the middle of the *System Size scale* and falls into the *Moderately Sized category.* Although a military strike aircraft itself is relatively small, it becomes significantly larger when you add training facilities, mission support, maintenance facilities, and sustainment (e.g., parts supply).

- **System Variability**

  A military strike aircraft lies near the middle of the *System Variability scale* and falls into the *Systems with Moderate Amounts of Variability category.* A military strike aircraft can come in multiple variants for different services and international partner/customer nations (e.g., the F-35 Joint Strike Fighter) and different individual aircraft (identified by tail numbers) can hold different versions of software due to the impossibility of performing simultaneous upgrades on entire squadrons of aircraft.

- **Subsystem Autonomy**

A military strike aircraft lies in the middle of the *Subsystem Autonomy scale* and falls into the *Systems with Moderate Levels of Subsystem Autonomy category.* Although the different parts of a military strike aircraft are physically integrated and multiple components (e.g., computers, sensors, and actuators) are integrated by software and data links, a great deal of effort goes into decoupling these subsystems where ever practical. Thus, there is no direct connection between the control of flight surfaces and external communication (e.g., to other aircraft, air force bases, and satellites). Although software is used to integrate and control practically all else on the aircraft, architects typically use techniques such as a modular open software architecture including open standard interface protocols and the proper decomposition and allocation of the software (e.g., to different computers or different virtual machines within the same computer).

- **Subsystem Governance**

  A military strike aircraft lies near the low end of the *Subsystem Governance scale* and falls into the *Directed Systems of Centrally Governed Subsystems category.* A military strike aircraft is so large that it is typically developed by a prime contractor (system integrator) and subcontractors, and uses COTS parts from various vendors.

- **Subsystem Heterogeneity**

  A military strike aircraft lies near the high end of the *Subsystem Heterogeneity scale* and falls into the *Systems with Highly Heterogeneous Subsystems category.* A military strike aircraft contains a great number of different types of subsystems including structural, propulsion, avionics, etc.

- **Subsystem Physical Distribution**

  An individual[21] military strike aircraft lies at the low end of the *Subsystem Physical Distribution scale* and falls into the *Systems with Contiguous Subsystems category.* Although an individual military strike aircraft itself is contiguous, a squadron of such aircraft is distributed regionally or globally when training facilities, mission support, maintenance facilities, and sustainment (e.g., parts supply) are added to the aircraft.

- **Subsystem Reuse**

  A military strike aircraft lies near the middle of the *Subsystem Reuse scale* and falls into the *Systems with Moderate Reuse category.* On the one hand, a state-of-the-art military strike aircraft is meant to be a major improvement over existing aircraft, and as such, calls for a great deal of development of new hardware and software. On the other hand, there is significant opportunity for reuse if one is developing a product line of aircraft (e.g., the three F-35 variants). Much of the flight software can also be reused in the pilot simulators.

[21] Naturally, there is a big difference between individual aircraft and a fleet of aircraft and their ground-based supporting training, maintenance, and logistics subsystems.

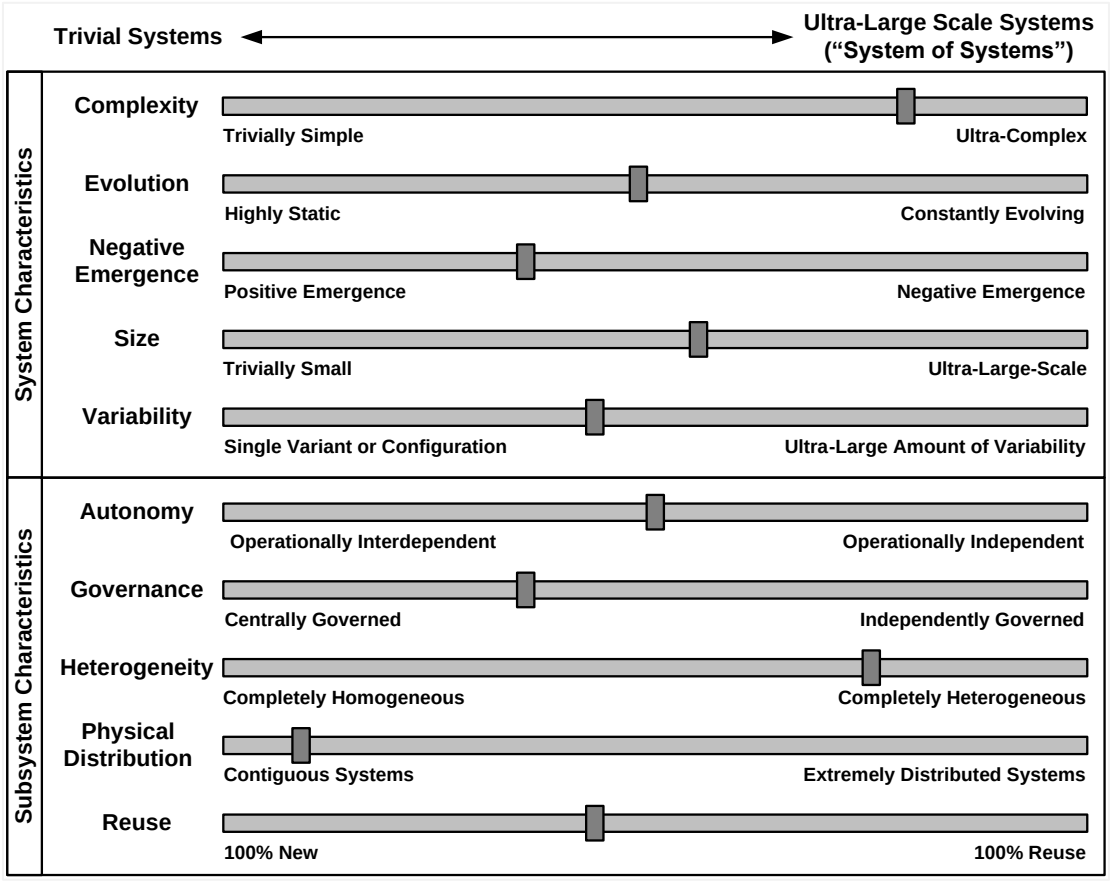

Figure 22: Meters for the System Characteristics of a Military Aircraft Fleet

4.4 National Smart Grid

A smart grid (a.k.a., smart electric grid, smart power grid, and intelligent grid) is a modernized international, national, regional, or municipal electric supply network based on the heavy use of modern digital technology [Ambrosio 2009, DOE 2008].

The functions of a national smart grid are to

- provide higher quality electricity free of sags, spikes, disturbances, and outages with increased availability, reliability, safety, and security
- self heal in the face of emergency situations such as extreme weather, solar storms, electromagnetic pulse, and terrorist attacks
- support decentralized power generation so that homes and businesses can be both electricity consumers and suppliers
- be flexible enough to support all major generation and storage technologies
- be flexible so that consumers can participate in grid operations by:
  - selecting electricity providers based on cost or type (e.g., green suppliers such as solar, wind, and biomass)
  - scheduling electricity usage (e.g., during off peak hours)
- offer increased energy efficiency to support energy independence and reduce global warming. For example, introduction of a smart grid could reduce U.S. electric needs by 6%,

reduce peak power demands by 27%, and increase renewable energy sources by 20% [Brecht 2009].

A national smart grid includes

- transmission networks for moving electricity large distances
- distribution networks for moving electricity from transmission networks to consumers
- integrated communication to support real-time control and data interchange
- advanced sensors (e.g., phasor measurement units) for monitoring usage, power quality, equipment health, transmission line temperature, etc.
- smart meters for measuring real-time energy consumption
- smart energy panels for intelligently distributing electrical power
- software intensive control and information systems for operator and manager control and network monitoring, demand management, real-time sensor fusion, decision support, storage of usage data, recording of anomalies, etc.

A national smart grid interoperates with, but does *not* include

- power plants and other electricity generation equipment that supplies electricity to the grid
- power storage systems that temporarily store electricity from the grid when it is not needed
- power consumer equipment including motors, lighting, heaters and air conditioners, smart appliances, etc.

System of Systems Characteristics

By practically all definitions, a national smart grid is both an ultra-large-scale system and a system of systems that exhibits the following SoS characteristics:

- **System Complexity**

  A national smart grid lies at the high end of the *System Complexity scale* and falls into the *Systems with Extreme Levels of Complexity category.* A smart grid varies from high to ultra-high complexity due to its large number of heterogeneous components that are tightly connected by numerous power, data, and control interfaces.

- **System Evolution**

  A national smart grid lies at the high end of the *System Evolution scale* and falls into the *Systems with Extreme Rates of Evolution category.* A smart grid is undergoing constant evolution as it evolving from a traditional power grid to a modern intelligently controlled power grid. It is highly likely to change as new requirements are implemented using new and rapidly evolving technologies (e.g., superconductivity, new storage batteries, and advances in green technologies).

- **System Negative Emergence**

  A national smart grid lies in the middle of the System Negative Emergence scale and falls into the Systems with Moderate Levels of Negative Emergence category. Approximately 20% of current power outages in the current electric grid are caused by negative emergent behaviors and attributes. Although it is impossible or impractical to know what the negative emergent behavior of the smart grid will be before it is built, there are several papers that mention the risk of negative emergent behavior in context with the smart grid. Due to the

complexity of the smart grid and past experience with previous simpler grids, one would predict the occurrence of a significant number of negative emergent behaviors over time.

- **System Size**

  A national smart grid lies at the high end of the *System Size scale* and falls into the *Ultra-Large-Scale Systems category*. Regardless of the way size is measured, a smart grid, especially an international or national smart grid, would fall into the ultra-large-scale of system sizes.

- **System Variability**

  National smart grids lie somewhat higher than the middle of the *System Variability scale* and fall into the *Systems with High Levels of Variability category.* Although only a few national electric grids currently qualify as [semi]smart, every major developed country or grouping of countries (e.g., the European Union) is either developing a smart national grid or thinking about developing one.

- **Subsystem Autonomy**

  A national smart grid lies near the low end of the *Subsystem Autonomy scale* and falls into the *Systems with Low Levels of Subsystem Autonomy category*. A smart grid is a network of networks (e.g., transmission networks and distribution networks) with a great deal of control and monitoring via sensors, smart meters, etc. As such, its subsystems exhibit a relatively large amount of interdependence.

- **Subsystem Governance**

  A national smart grid lies near the high end of the *Subsystem Governance scale* and falls into the *Collaborative Systems of Subsystems category.* Due to its physical size, an international or national smart grid will be developed by many utilities under partial direction (policy) and partial funding from international (e.g., EU), national (e.g., USA, Denmark) governments and regulatory agencies (e.g., the US Department of Energy). Vendors are also developing much of the technology with the intent to sell it to the developers of multiple smart grids. Thus, smart grids tend to have characteristics of both acknowledged and collaborative systems of systems.

- **Subsystem Heterogeneity**

  A national smart grid lies near the high end of the *Subsystem Heterogeneity scale* and falls into the *Systems with Highly Heterogeneous Subsystems category.* A smart grid consists of transmission and distribution networks, advanced sensors, smart meters and energy panels, integrated communications systems, and many different software systems performing quite different functions.

- **Subsystem Physical Distribution**

  A national smart grid lies somewhat above the middle of the *Subsystem Physical Distribution scale* and falls into the *Nationally Distributed Systems category.* The geographical distribution of a smart grid will naturally depend on whether it is a municipal, regional, national, or international grid. Regardless, all smart grids will be relatively widely distributed systems.

- **Subsystem Reuse**

  A national smart grid lies in the middle of the *Subsystem Reuse scale* and falls into the *Systems with Moderate Levels of Reuse category.* Because of its reliance on new technology, a

smart grid will be replacing many legacy parts of the current grid with new components. Although this would indicate a low level of reuse, it will be financially and practically impossible to replace all of an international, national, or even regional electric grid at once because of its sheer size and cost and the need to continue supplying electricity. While new components (smart meters) will be added, many major parts of the grid will also be upgraded rather than replaced. Therefore, early versions of the smart grid will reuse much of the existing grid. Thus, the smart grid will transition from high to moderate levels of subsystem reuse over time.

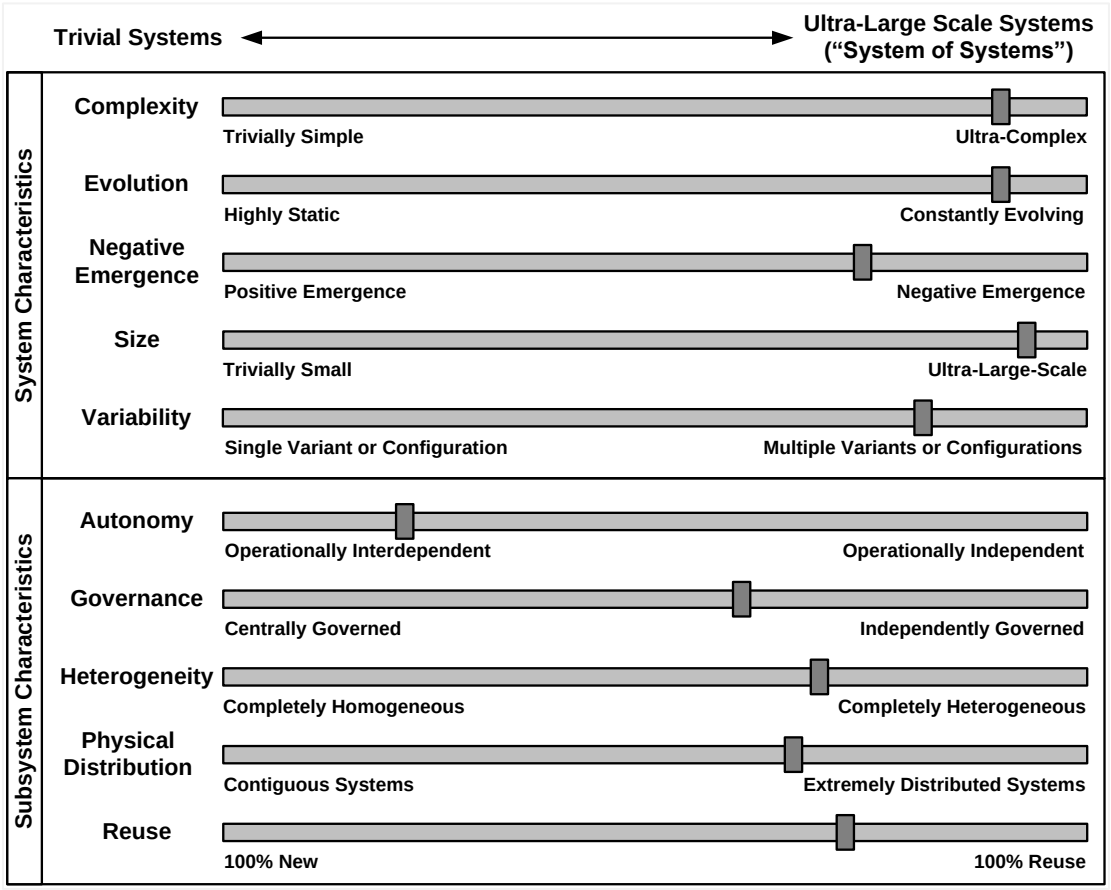

Figure 23: Meters for the System Characteristics of a National Smart Grid

5 Two Other Sets of System Characteristics

5.1 Quality Characteristics

The previous section identified and discussed the large number of characteristics that often appear in the definitions of systems of systems. Other important system characteristics are those that define the different types of quality of the system. These characteristics are documented in quality models such as "Software Engineering – Product Quality – Part 1: Quality Model" [ISO/IEC 2001] and "The Method Framework for Engineering System Architectures (MFESA)" [Firesmith 2008].

5.1.1 Definitions

To clearly gain a consensus understanding of what quality means, it is important to understand quality models and their components. Figure 24 illustrates the relationship between the following concepts:

Quality Model
a hierarchical model for defining, specifying, and measuring the different types of quality of a system or subsystem in terms of the model's component quality characteristics, quality attributes, and associated quality measurement scales and methods

Quality Characteristic
a high-level characteristic or property of a system or subsystem that characterizes an aspect of its quality

Quality Attribute
a major measurable component (aggregation) of a quality characteristic

Quality Measurement Scale
a measurement scale that defines numerical values used to measure the degree to which a system or subsystem exhibits a specific quality attribute

Quality Measurement Method
a method, function, or tool for making a measurement along a quality measurement scale

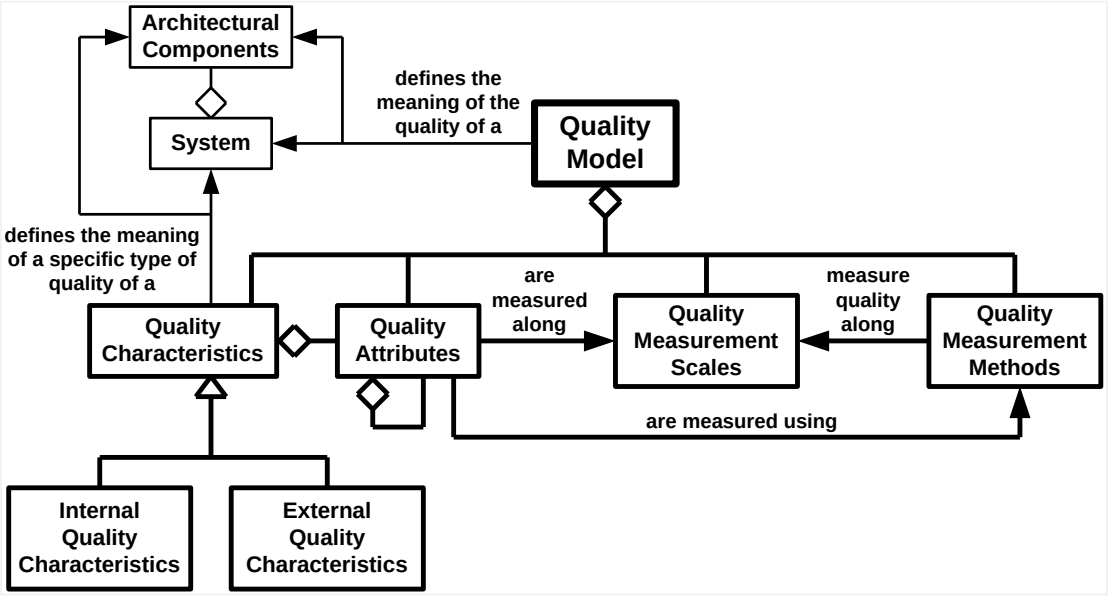

Figure 24: The Four Components of a Quality Model

Figure 25 illustrates the meters that show the values of an example system along the scales associated with several internal and external quality characteristics such as availability, capacity, extensibility, interoperability, maintainability, performance, portability, reliability, robustness, safety, security, testability, usability, and variability. These quality characteristics in turn are decomposed into their more objective quality attributes that become the basis for system and software quality requirements. Quality characteristics are also generalizations (i.e., informal averages) because the scales really belong at the level of quality attributes of the quality characteristics. For example, the quality attributes jitter, response time, and throughput of the quality characteristic performance can be measured along objective quality measurement scales (e.g., time or count per time), whereas performance itself can only have a rough and fuzzy subjective scale.

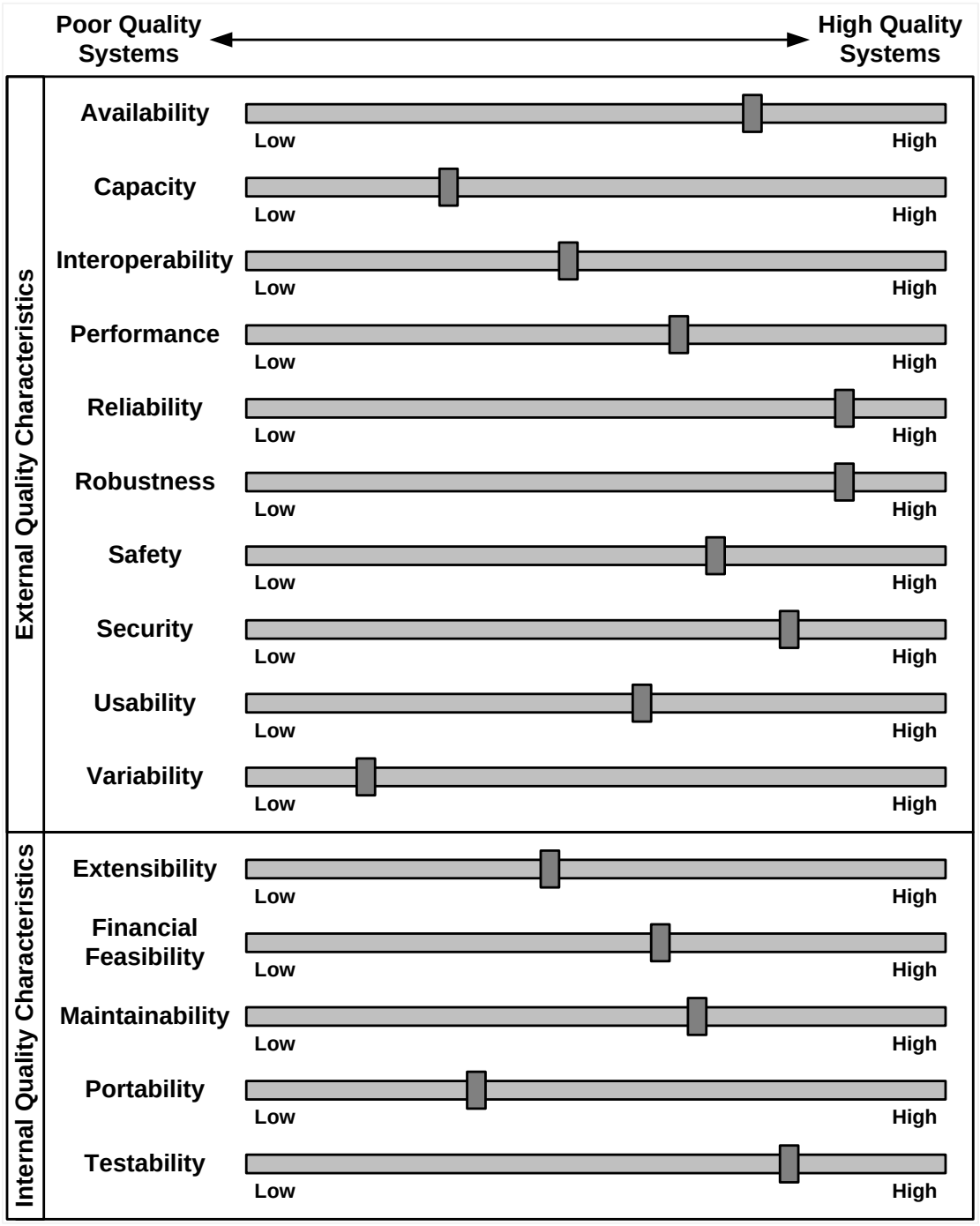

Figure 25: Example Meters for System Quality Characteristics

5.2 Programmatic Characteristics

As illustrated in Figure 26, systems also vary greatly in terms of their programmatic characteristics including:

- **Organizations Associated with Systems**
  - Types of Organizations. Systems vary greatly in terms of the types of their associated organizations such as acquisition, funding, management, development (including system integrator or prime contractor, subcontractor, and vendor), certification and accreditation, regulating, operation, maintenance, and sustainment organizations.
  - Number of Organizations. Systems vary greatly in terms of the number of their associated organizations, whereby each type of organization may have one or more of its own organizations or a single organization may play the role of multiple organization types.
  - Size of Organizations. Systems vary greatly in terms of the size of their associated organizations.
  - Type of [Subsystem] Governance. System organizations vary greatly in terms of the type of their governance (e.g., directed, acknowledged, collaborative, and virtual systems). Note that this is commonly viewed as a subsystem characteristic rather than an organizational characteristic, which is why it was included in the previous section on subsystem characteristics.
  - Amount of Authority, Funding, Scheduling, and Regulation/Policy. System organizations vary greatly in the amount of their authority, funding, and scheduling. They also vary in terms of the laws, regulations, and policies that constrain their operation.
  - Management and Engineering Culture. System organizations vary greatly in terms of their management and engineering culture. Some organizations are early adopters of new paradigms, methods, and technologies, whereas others are conservative and late adopters.
  - Geographical Distribution. System organizations vary greatly in their geographical distribution, both in terms of the locations of different parts of individual organizations and related organizations (e.g., subcontractors and vendors), especially in a time of increased globalization and outsourcing.
  - Staff Expertise and Experience. The personnel belonging to the organizations associated with the system can vary greatly in terms of expertise, training, and experience.
- **System Stakeholders**
  - Number of Stakeholders. Different systems with their associated organizations and endeavors also vary widely in terms of the number of stakeholders. Some systems developed for local use inside a small organization may have only a handful of stakeholders, whereas most systems have hundreds or thousands of stakeholders. At the high end of the scale, it is not uncommon for some of the largest and most critical systems (e.g., the Internet) to have literally hundreds of millions of stakeholders.
  - Type of Stakeholders. Different systems with their associated organizations and endeavors vary widely with regard to the types of stakeholders they have. This can include various types of acquirers, developers, maintainers, operators, and users. The stakehold-

ers can also be regulators, safety or security accreditors/certifiers, subject matter experts, and members of the public.

- Stakeholder Authority. The different stakeholders of a system have varying amounts of authority to: (1) drive the system requirements, architecture, design, implementation, integration, and deployment of the system, (2) set policy and process requirements, (3) affect funding, and (4) determine acceptability or accreditation of the system (e.g., that it is sufficiently safe or secure to be placed in operation).
- Stakeholder Accessibility. On Agile projects, the developers need to be able to collaborate closely with collocated stakeholders to generate the user stories that form their requirements. On other projects with multiple levels of formal contracts separating organizations, it is often impossible for subcontractor developers (e.g., requirements engineers and architects) to gain direct access to actual system users. This variability in accessibility greatly influences requirements engineering and management issues such as scope control.
- Stakeholder Volatility

  A small percentage of projects have little if any stakeholder volatility, at least with respect to stakeholders with authority. On large projects with long durations (especially those that last multiple years), it is almost certain that many of the important stakeholders will come and go. This volatility affects the appropriate system engineering methods to use (for example, by emphasizing the use of baselined documents over verbal understandings).
- Stakeholder Motivation and Needs

  The amount and distribution of stakeholder needs (e.g., requirements) and motivations (e.g., nice to haves) varies from system to system, and over time as conditions change and stakeholders come and go. The motivations and needs are also often inconsistent from one stakeholder to another. This volatility and the inconsistencies influence how best to perform requirements engineering and scope control.
- Degree of Trust

  The amount of trust among important stakeholders varies from system to system.

  Where an adversarial situation with little trust exists, there tends to be a need for more formality in requirements engineering, management, and verification and validation (e.g., testing). On other projects where the stakeholders can establish a close collaboration built upon trust, less formality is sometimes more appropriate.

- **Endeavors Involving Systems**
  - Type of Endeavor. The endeavor associated with the acquisition, development, operations, or sustainment of a system can vary in scope from an individual project, a program of related projects (e.g., as is typical when developing a product line of systems), or an entire enterprise.

- Type of Contracting. The type of contracting associated with the endeavor can vary in terms of formality from a totally informal verbal agreement to a formally documented and legally binding contract. It can also vary in terms of the type of contract (e.g., fixed-price or cost plus fixed fee).
- Type of Development. The type of development associated with the endeavor can vary from the development of a totally new system through making a relatively small and simple enhancement to a legacy system.
- Life-Cycle Scope. The scope of the system life cycle associated with the endeavor can include part or all of acquisition, development, manufacturing, operations, sustainment, and/or retirement.
- System Scope. The scope of the system associated with the endeavor can vary from an individual subsystem through the development/integration of an entire system of subsystems to the development or maintenance of an ultra-large system of pre-existing systems.
- Endeavor Duration. An endeavor to produce or update a small, simple system may be completed within a small number of weeks. However, as the system grows in size and complexity, it is common for endeavors to last months, years, or even decades.
- Endeavor Schedule. Schedules are rarely adequate for the development of many systems, especially large and complex ones. Endeavors also vary in terms of the criticality of meeting deadlines and in the difficulty of coordinating the schedules of numerous organizations and teams within organizations.
- Endeavor Funding

  The adequacy of funding can vary significantly, although there is a strong tendency to underfund system development, especially for larger and more complex systems. An over abundance of funds rarely occurs.

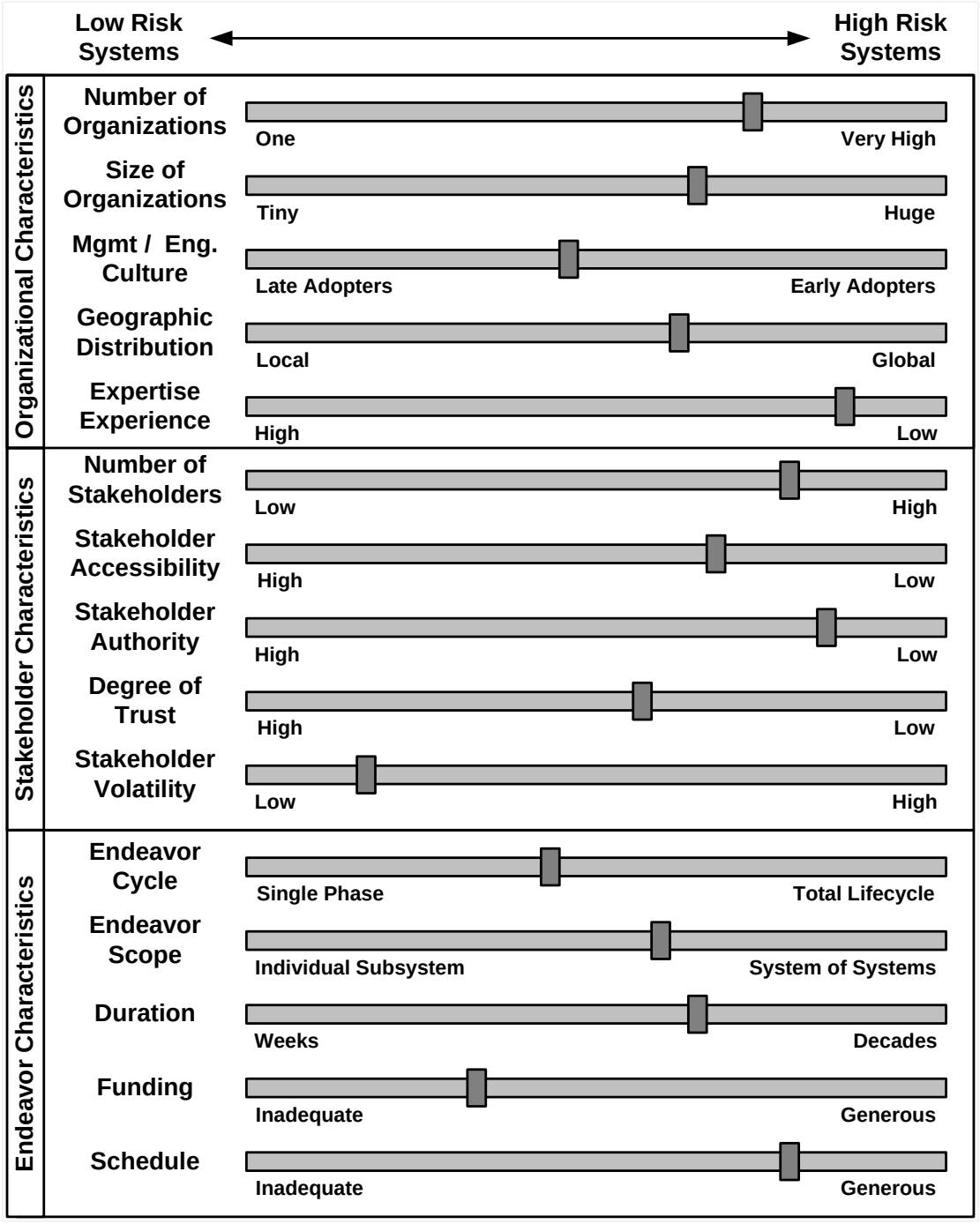

Figure 26: Example Meters for System Programmatic Characteristics

The appropriate number and attributes of the appropriate system engineering methods used to develop, maintain, and operate systems is clearly influenced by the degree to which all of these characteristics exist (that is, those characteristics addressed in Section 2 of this report, the system's quality characteristics and quality attributes, and those just listed above). Clearly, no single system engineering method is sufficiently general and tailorable to meet the needs of all systems, regardless of their characteristics.

6 Usage

There are many reasons to use meters and scales based on system characteristics to categorize and profile systems. This is true for SoS characteristics as it is for quality characteristics and programmatic characteristics. These uses are described in this section.

6.1 Improved Understanding

Although the concept of system of systems has become very popular over the last decade, there is no consensus as to exactly what it means, and the term has been given many different, though related, definitions. This diversity of meaning is likely due to the term itself, which is misleading because it implies that virtually all systems are systems of systems.

Using well-defined characteristics will improve the stakeholders' understanding of the important concepts (such as emergence and governance) underlying systems of systems and ultra-large-scale systems, both individually and how multiple characteristics tend to vary together (e.g., size and complexity). More importantly, it will help the stakeholders[22] understand the system in terms of some of its most fundamental characteristics. This is especially important when dealing with the concepts of systems of systems and ultra-large systems, both of which typically refer to systems with characteristics that mostly lie at or near the high end of most of the scales of the defining SoS characteristics.

6.2 Improved Communication

People sometimes accidentally use the same terms with different meanings (unintentional homonyms), or use different terms with the same meanings (synonyms). By clearly defining the system's important characteristics, developers, acquirers, and other stakeholders will be better able to communicate clearly the characteristics of their system and thus, the type of system (and subsystems) they are engineering or updating.

6.3 Improved Risk Identification

Each scale for a system characteristic has an associated meter, the value of which can range from the minimum to maximum valid value. The scales have been organized in this report so that low values imply low project risk and high values imply high project risk. In other words, the value displayed by the meters measures the challenges and risks associated with the development or major update of the associated system. Thus, by looking at all of the system's meters, one can get a good idea of the overall project risk. For example, a value of extremely high is appropriate if all of the meters show the system either on or near the right ends of its associated measurement scales.

[22] In this context, the word *stakeholders* means everyone with a significant legitimate interest in the system during any phase of the system's life cycle from initial acquisition through development, operation, and sustainment, to eventual retirement and disposal.

Like a steam pressure meter in the red, any SoS meters reading near the high end of their associated scales indicate high (and possibly excessive) risk. This warns management and technical leadership to take steps to:

- Manage the risk by making the associated characteristic an official risk item in the program repository, monitor its impact and status, and report it on a regular basis.
- Lower the risk by eliminating or changing the requirements that cause the excessive risk.
- Mitigate the risk by modifying the system architecture or management and engineering methods to use more rigorous methods and extensive testing for those parts of the system that are business, safety, or security critical.
- Transfer the risk by acquiring insurance or negotiating associated clauses in subcontractor contracts or vendor licensing.

6.4 Improved System Tracking

Early in the project, the system's initial values along some of the scales for the system/subsystem characteristics may be largely guesstimates based on past experience and an initial vision of the system. Later, as understanding increases, the meters may move to better represent the system as stakeholder understanding of the system matures. Thus, one can use the values displayed by these meters to track the project's high-level understanding of the system as stakeholder understanding of the system matures.

6.5 Improved Decision Making

Many decisions depend on the type of system being engineered. Acquirers, developers, and other stakeholders with authority will be better able to make appropriate decisions regarding the system and its development. For example, the location of the system along these scales may significantly affect which management and architectural patterns are appropriate to use. For example, management patterns can be affected by size and subsystem governance, whereas architecture patterns can be influenced by complexity, evolution, reuse, physical distribution, and variability.

6.6 Method Engineering

Sections 4 and 5 cataloged three sets of system characteristics that can vary greatly from system to system. In turn, this great variability in system characteristics can strongly influence the proper characteristics and contents of appropriate situation-specific methods for engineering systems.

Because of the vast variability among different types of systems, all systems should not be treated the same way [Stevens 2008]. Different systems exhibit different characteristics, and the values of some of these characteristics influence the number and attributes of the appropriate system engineering and management methods that should be used to develop, maintain, and operate them. The large variability in systems is the reason why no single system engineering method is sufficiently general and tailorable to meet the needs of all endeavors.

The solution to the problem of dealing with such great variability is not to mandate the use of a single system or software engineering or management method, no matter how well it may be based on industry best practices. Rather, a more effective approach is to use method engineering (ME) to engineer one or more appropriate method for the engineering effort [Welke 1992, Brink-

kemper 1996, Rolland 1997, Brinkkemper 1998, Firesmith 2002, Henderson-Sellers 2003, Firesmith 2008, Henderson-Sellers 2008, Rolland 2008]. Often called situational method engineering (SME) because it seeks to engineer methods appropriate to the situation at hand, ME enables the production of appropriate system engineering methods that are specific to and appropriate for the given system, organizations, endeavors, and stakeholders.

Situational method engineering involves creating or obtaining reusable method components (such as work products, work units, and performers of work units) and storing them in a method repository. SME also involves creating situation-specific methods from these components by

- selecting the appropriate system engineering method components for a repository of reusable method components
- tailoring these selected method components to meet the specific situation
- integrating the selected and tailored method components to form an appropriate cohesive and consistent system engineering method

When multiple organizations such as the prime contractor or system integrator, subcontractors, and vendors are developing a system, a single system engineering method may not be appropriate. When developing a large and complex system, the differences between its subsystems may require the use of different system engineering methods by the associated integrated product teams (IPTs). For example, a safety-critical subsystem will typically need a more formal and powerful method based on the subsystem's safety evidence assurance level (SEAL) than will a subsystem with no business-, safety-, or security-criticality.

With such a large number of system, organization, stakeholder, and endeavor characteristics pulling in potentially different directions, it is non-trivial to determine the exact properties of the most appropriate system engineering method to generate using situational method engineering. Fortunately, it is much easier to determine the level of method completeness and formality that is more-or-less optimal. For example, certain parts of certain systems justify the use of formal methods and model-driven development (MDD). Some systems require a document-driven approach, whereas other parts can benefit from more of an Agile approach. Some systems or subsystems can be developed using a relatively sequential waterfall approach, whereas other systems benefit greatly from the use of an iterative, incremental, parallel, and time-boxed development cycle.

A key concept to remember about method engineering is that there is no silver bullet, no single best way to develop, operate, maintain, or sustain all systems. System acquirers, managers, and engineers should be wary of the claims of any person or organization attempting to market their specific engineering method as "the best." Instead, it would be wise to think in terms of what is sufficiently appropriate for the kinds of systems or subsystems in terms of system characteristics, required system qualities, and the characteristics of the system's associated organizations, stakeholders, and endeavors.

7 Conclusion

Systems vary greatly in terms of their inherent characteristics. These characteristics can be categorized as follows.

- **System of Systems Characteristics**

  Although these characteristics are often included in definitions of the phrase "systems of systems", they actually apply to some degree to all systems. These are the following system and subsystem characteristics that are the prime focus of this technical note:

  - System characteristics including system complexity, negative emergence, evolution, size, and variability
  - Subsystem characteristics including subsystem autonomy, governance, heterogeneity, physical distribution, and reuse

- **Quality Characteristics**

  Systems vary greatly in terms of their quality characteristics (a.k.a., the "ilities") and associated component quality attributes. The quality attributes of these quality characteristics are the foundation for engineering quality requirements:

  - External characteristics such as availability, interoperability, reliability, safety, security, and usability
  - Internal characteristics such as feasibility, maintainability, portability, and reusability

- **Programmatic Characteristics**

  Systems also vary greatly in terms of the following programmatic characteristics:

  - organizational characteristics including the type, number, and sizes of the organizations associated with the system; the type of governance;[23] the amount of authority, funding, scheduling, and regulation/policy; the management and engineering culture; geographic distribution,[24] and staff expertise and experience
  - stakeholder characteristics including the number and type of stakeholders, the stakeholders' authority, the accessibility of the developers to the stakeholders, the stakeholders' motivations and needs, and the degree of trust between the developers and the stakeholders
  - endeavor characteristics including the type of endeavor, contracting, and development; life cycle and system scope; and the endeavor's duration, schedule, and funding

Each of the system of systems characteristics has an associated meter that measures roughly where a system lies along the associated scale. Although we have not demonstrated it in this technical note, similar meters and scales can also be developed for the measurable quality attributes of the more general quality characteristics, and for the programmatic characteristics. These meters provide a high-level description of the system having the following benefits: improved under-

[23] Note that although it was listed as a "system of systems" characteristic, governance is actually a programmatic characteristic of the subsystem organizations.

[24] Geographic distribution here refers to the distribution of the organizations (e.g., via outsourcing) rather than the distribution of the system's subsystems.

standing, improved communication, improved risk management, improved system tracking, and improved decision making.

Given this wide variability among systems, no single system or software engineering method is sufficiently general and tailorable to be optimal for engineering all systems. In fact, different methods are often needed for different subsystems within the same overall system. This is the reason for the increasing recognition of the need to use situational method engineering to develop one or more appropriate system/software engineering methods that meet the specific foundational, quality, and programmatic needs of the system.

Finally, because almost all systems consist of subsystems that are themselves systems, the phrase *system of systems* is too general and misleading. The phrase has been given many different definitions, which are usually based on several of the system and subsystem characteristics documented in Section 2 of this report, whereby a system of systems is any system that lies towards the high risk ends of the meters for these system of systems characteristics:

> **System of Systems (SoS)**
> any system that is a relatively large and complex, dynamically evolving, and physically distributed system of pre-existing, heterogeneous, autonomous, and independently governed systems, whereby the system of systems exhibits significant amounts of unexpected emergent behavior and characteristics

Because the phrase *system of systems* strongly implies any system and different "systems of systems" exhibit different levels of these system and subsystem characteristics, misunderstandings can be avoided if the phrase is replaced by the meters of the relevant characteristics to provide a clearer and more specific description of the actual *system of systems*.

Appendix: Glossary

Acknowledged Systems (of [Sub]systems)
systems that have their own objectives, management, funding, and authority, but the subsystems of which retain their own independent management, funding, and authority in parallel with the overall system

Collaborative Systems (of [Sub]systems)
systems without centralized objectives, management, authority, responsibility, and funding, the subsystems of which are voluntarily governed to support the overall system to address shared or common interests

Directed Systems (of [Sub]systems)
centrally governed systems, the subsystems of which are governed in a centralized and coordinated manner as part of the system

Emergent behavior
a system's behavior that is not explicit in the behaviors of the system's individual subsystems but rather provided by the interaction of two or more of the system's subsystems

Heterogeneous system
the degree to which the subsystems of a system differ from each other in that they (1) have different goals, objectives, and requirements, (2) have different behavior and characteristics, (3) provide unrelated functionality, (4) belong to different application domains, and (5) are implemented using different technologies

Managerial independence
the degree to which the subsystems of a system are owned and operated by different organizations (i.e., are managed independently of each other)

Meter
a device that measures or displays a specific value by means of the position of an indicator along a scale

Quality Attribute
a major measurable component (aggregation) of a quality characteristic

Quality Characteristic
a high-level characteristic or property of a system or subsystem that characterizes an aspect of its quality

Quality Measurement Method
a method, function, or tool for making a measurement along a quality measurement scale

Quality Measurement Scale
a measurement scale that defines numerical values used to measure the degree to which a system or subsystem exhibits a specific quality attribute

Quality Model
a hierarchical model for defining, specifying, and measuring the different types of quality of a system or subsystem in terms of the model's component quality characteristics, quality attributes, and associated quality measurement scales and methods

Scale
a line representing an increasing range of values whereby position on the line represents a specific value in the range

Schedule independence
the degree to which the development and maintenance schedules of the subsystems of a system are uncoordinated (i.e., are scheduled independently of each other)

Subsystem
a component of a system that is itself a system

Subsystem autonomy
the degree to which the subsystems within a system are independent, stand alone and are individually useful, self-contained, and operationally independent (i.e., neither controlled by nor controlling other subsystems)

Subsystem characteristic
any characteristic of a system that describes its individual subsystems

Subsystem governance
the degree to which the subsystems of a system are governed (e.g., specified, managed, funded, developed, owned, operated, maintained, and sustained) in an independent, decentralized, and uncoordinated manner

Subsystem heterogeneity
the degree to which the subsystems of a system differ from each other in that they (1) have different goals, objectives, and requirements, (2) have different behavior and characteristics, (3) provide unrelated functionality, (4) belong to different application domains, and (5) are implemented using different technologies

Subsystem physical distribution
the degree to which the subsystems of a system are distributed in different physical locations

Subsystem reuse
the degree to which the subsystems of the system have been reused regardless as to whether they are commercial-off-the-shelf (COTS), government-off-the-shelf (GOTS), military-off-the-shelf (MOTS), organizational-internal reuse, open source, and freeware

Subsystem synergy[25]
the collaboration of subsystems to achieve the system's emergent system behavior or characteristics

System
a cohesive integrated group of interdependent or interacting components that regularly collaborate to provide the behavior and characteristics that (1) are needed to meet valid stakeholder needs and desires and (2) cannot be provide separately by the individual components

System characteristic
any characteristic of a system that describes an individual system as a whole rather than its individual subsystems

System complexity
the degree to which a system is difficult for its stakeholders to understand and analyze, especially due to having a large number of components connected by many complicated interfaces

System evolution[26]
the degree to which (in terms of rate and impact) the goals and requirements for a system (and its subsystems) change over time

System negative emergence
the degree to which the new behaviors and characteristics of a system that result (i.e., emerge) from the interaction of the system's subsystems are detrimental, unintended, and difficult to predict from the behaviors and characteristics of these individual subsystems

System requirements risk
the degree of risk associated with poorly engineered requirements that are incomplete, immature, and volatile

System size
the amount or magnitude of the system in terms of some suitable scale (for example, in terms of the system's mass, physical dimensions, and total number of subsystems at all levels of the aggregation structure)

System of systems[27]
(1) a system, the largest components of which are themselves systems (i.e., subsystems) (2) any system that is a relatively large and complex, dynamically evolving, and physically distributed system of pre-existing, heterogeneous, autonomous, and independently governed systems, whereby the system of systems exhibits significant amounts of unexpected emergent behavior and characteristics

---

[25] Synergy is the result of system-internal cooperative interactions between subsystems and interactions of the subsystems with the system-external environment.

[26] Note that this is not to be confused with system evolvability (maintainability) which is the ease with which a system can be changed as its goals and requirements change over time.

[27] The first definition is explicit based on the components of term "system of systems", whereas the second definition is often what is meant in practice, especially in the system of systems community.

System variability
the degree to which a single type of system simultaneously exists in multiple variants, versions, or configurations

Ultra-large-scale system
any system, the size of which is unprecedented when measured along multiple scales

Virtual Systems (of [Sub]systems)
systems, the subsystems of which are independently governed in a completely distributed and uncoordinated manner as stand-alone systems

References/Bibliography

URLs are valid as of the publication date of this document.

[Ambrosio 2009]
Ron Ambrosio. "Research for a Cleaner, Smarter Energy Future." Southern California Smart Grid Research Symposium, Irvine, CA, April 2009.

[Boxer 2008]
P. Boxer, E. Morris, W. Anderson, & B. Cohen, "System-of-Systems Engineering and the Pragmatics of Demand." IEEE Systems Conference, Montreal, Canada, April 2008.

[Brecht 2009]
Lyle A. Brecht. "Implementing the National Smart Grid Initiative." *Capital Markets Research*, May 2009.

[Brinkkemper 1996]
Sjaak Brinkkemper, Kalle Lyytinen, & Richard J. Welke. "Method Engineering: Principles of Method Construction and Tool Support." *Proceedings of the IFIP TC8, WG8.1/8.2 Working Conference on Method Engineering*. Atlanta, GA: Springer-Verlag, August 1996.

[Brinkkemper 1998]
Sjaak Brinkkemper, Motoshi Saeki, & Frank Harmsen. "Assembly Techniques for Method Engineering," Advanced Information Systems Engineering" 381 - 400. *Proceedings of the 10th International Conference on Advanced Information Systems Engineering*, London, UK: Springer Verlag, 1998.

[DeLaurentis 2004]
Daniel A. DeLaurentis & Robert K. Callaway. "A System of Systems Perspective for Future Public Policy," *Review of Policy Research, 21,* 6 (2004): 829-837.

[DOD 2008]
Office of the Deputy Under Secretary of Defense for Acquisition and Technology, Systems and Software Engineering. *System Engineering Guide for Systems of Systems, Version 1.0,* ODUSD(A&T)SSE, August 2008.

[DOE 2008]
Department of Energy. *The Smart Grid: An Introduction*, 2008.

[Ebert 2009]
Christof Ebert & Capers Jones. "Embedded Software—Facts, Figures and Future," *IEEE Computer*, *42*, 4 (April 2009): 42-52.

[Firesmith 2002]
Donald Firesmith & Brian Henderson-Sellers. The OPEN Process Framework: An Introduction. London, UK: Addison-Wesley, 2002 (ISBN: 0201675102).

[Firesmith 2008]
Donald Firesmith, Peter Capell, Dietrich Falkenthal, Bud Hammons, DeWitt Latimer, & Tom Merendino. *The Method Framework for Engineering System Architectures (MFESA)*, Taylor and Francis, November 2008.

[Garcia-Miller 2009]
Suzanne Garcia-Miller, Lisa Brownsword, & Patrick Kirwan. "Organizational Implications of Systems of Systems." 12th *NDIA Systems Engineering Conference*, San Diego, CA, October 2009.

[Henderson-Sellers 2003]
Brian Henderson-Sellers. "Method Engineering for OO System Development." *Communications of the ACM*, *46*, 10 (October 2003): 73-78.

[Henderson-Sellers 2008]
Brian Henderson-Sellers, Jolita Raylte, & Sjaak Brinkkemper eds. *Situational Method Engineering: Fundamentals and Experiences*. New York, NY: Springer-Verlag, 2008 (ISBN: 9780387739465) .

[Homeland Security 2009]
Department of Homeland Security. "The System of Systems Approach for Interoperable Communications." 2009. http://www.safecomprogram.gov/NR/rdonlyres/FD22B528-18B7-4CB1-AF49-F9626C608290/0/SOSApproachforInteroperableCommunications_02.pdf

[INCOSE 2009]
National Center for Systems of Systems Engineering. "What is Systems of Systems Engineering?" Old Dominion University. http://www.eng.odu.edu/ncsose/What_is_SOSE.shtml

[ISO/IEC 2001]
International Standards Organization / International Electrotechnical Commission. *Software Engineering – Product Quality – Part 1: Quality Model* (ISO/IEC 9126-1) June 2001.

[Jamshidi 2009]
Mo Jamshidi ed. *System of Systems Engineering: Innovations for the 21st Century*, Hoboken, N.J.: Wiley, 2009 (ISBN 9780470195901).

[Kotov 1997]
Vadim Kotov. "Systems-of-Systems as Communicating Structures" (HPL-97-124). Hewlett Packard Computer Systems Laboratory, October 1997.

[Kuras 1995]
M. L. Kuras & Brian E. White, "Engineering Enterprises Using Complex-System Engineering," 1-15. *1st Annual System of Systems Engineering Center of Excellence (SOSECE) Conference Proceedings*, Johnstown, PA, June 2005.
http://www.sosece.org/pdfs/1stSOSeceConf/presentations/white.pdf

[Meier 1998]
M.W. Maier. "Architecting Principles for System of Systems." *Systems Engineering*, *1*, 4 (1998): 267-284.

[Northrop 2006]
Linda Northrop, et al. *Ultra-Large-Scale Systems: The Software Challenge of the Future*. Software Engineering Institute, June 2006 (ISBN: 0-9786956-0-7). http://www.sei.cmu.edu/uls/

[Okimoto 1988]
Daniel I. Okimoto & Thomas P. Rohlen. *Inside the Japanese System: Readings on Contemporary Society and Political Economy*. Stanford, CA: Stanford University Press, 1988 (ISBN: 0804714258).

[Perrow 1984]
Charles Perrow. *Normal Accidents: Living with High-Risk Technologies*. Princeton, NJ: Princeton University Press, 1984 (ISBN: 0691004129).

[Popper 2004]
Steven W. Popper, Steven C. Bankes, Robert K. Callaway, & Daniel DeLaurentis. *System-of-Systems Symposium: Report on a Summer Conversation*. Potomac Institute for Policy Studies, Arlington, VA, July 2004.

[Purdue University 2009]
Purdue University, College of Engineering. "System of System (SoS)." Last accessed January 2010. https://engineering.purdue.edu/Engr/Research/Initiatives/SoS/

[Rolland 1997]
Colette Rolland. "A Primer for Method Engineering." *Proceedings of the INFORSID Conference (INFormatique des ORganisations et Systemes d'Information et de Decision)*, Toulouse, France, June 1997.

[Rolland 2008]
Colette Rolland. "Method Engineering: Towards Methods as Services," 10-11. International Conference on Software Process (ICSP) 2008, Leipzig, Germany, May 2008. http://www.springerlink.com/content/978-3-540-79587-2/

[Sage 2001]
Andrew P. Sage & Christopher D. Cuppan. "On the Systems Engineering and Management of Systems of Systems and Federations of Systems." *Information, Knowledge, Systems Management*, *2*, 4 (2001): 325-345.

[Sheard 2006]
Sarah A. Sheard. "Systems of Systems Necessitates Bridging Systems Engineering and Complex Systems Science," 1-24. *2nd Annual System of Systems Engineering Center of Excellence (SOSECE) Conference Proceedings*, July 2006.

[Simanta 2009]
Soumya Simanta. "Requirements Engineering for Systems of Systems," 30. 12th *NDIA Systems Engineering Conference*, San Diego, CA, October 2009.

[Stevens 2008]
Renee Stevens. "Profiling Complex Systems." *IEEE International Systems Engineering Conference (SysCon 2008)*, Montreal, Canada, April 2008.

[USAF 2005]
United States Air Force. *Scientific Advisory Board Report on System-of-Systems Engineering for Air Force Capability Development* (SAB-TR-05-04). United States Air Force, United States Air Force Scientific Advisory Board, July 2005.
https://acc.dau.mil/GetAttachment.aspx?id=150224&pname=file&aid=28788

[Valerdi 2007]
Ricardo Valerdi, Adam M. Ross, & Donna H. Rhodes. "A Framework for Evolving System of Systems Engineering." *CrossTalk, 20*, 10 (October 2007): 28-30.

[Welke 1992]
Richard J. Welke & Kuldeep Kumar. "Method Engineering: A Proposal for Situation-Specific Methodology Construction," 257-268. Systems Analysis and Design: A Research Agenda, Wiley, 1992.

[White 2005]
Brian E. White. "Engineering Enterprises Using Complex-System Engineering (CSE)." *1st Annual System of Systems (SoS) Engineering Conference*, Johnstown, PA, June 2005.

[White 2008]
Brian E. White, "Complex Adaptive Systems Engineering (CASE)." *3rd System of Systems Conference National Institute of Standards and Technology*, Gaithersburg, Maryland, December 2008.

[Wikipedia 2009]
Wikipedia. "System of Systems (SoS)." Last accessed January 2010.
http://en.wikipedia.org/wiki/System_of_systems

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
(Leave Blank)	February 2010	Final

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
Profiling Systems Using the Defining Characteristics of Systems of Systems (SoS)	FA8721-05-C-0003

6. AUTHOR(S)

Donald Firesmith

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213	CMU/SEI-2010-TN-001

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
HQ ESC/XPK 5 Eglin Street Hanscom AFB, MA 01731-2116	

11. SUPPLEMENTARY NOTES

12A DISTRIBUTION/AVAILABILITY STATEMENT	12B DISTRIBUTION CODE
Unclassified/Unlimited, DTIC, NTIS	

13. ABSTRACT (MAXIMUM 200 WORDS)

The concept of a *system of systems (SoS)* has become very popular over the last decade, resulting in books, conferences, technical papers, and reports. However, there is no consensus as to exactly what the term means, and it has been given many different, though related, definitions. This technical note identifies and describes the characteristics that have been used in various definitions of the term system of systems. These SoS characteristics vary along corresponding scales and can form the basis of corresponding "meters" that serve as indicators of where a system lies along the associated scale. This technical note also discusses two other classes of system characteristics: quality characteristics and programmatic characteristics and how similar meters can be used to describe where systems lie along the scales associated with these two additional sets of system characteristics. Finally, this technical note discusses the various benefits of using these system of systems characteristics to profile systems.

14. SUBJECT TERMS	15. NUMBER OF PAGES
system of systems, SoS, system characteristics, programmable characteristics	87

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL