

Engineering Systems Matrix: An Organizing Framework for Modeling Large-Scale Complex Systems

Jason E. Bartolomei, Daniel E. Hastings, Richard de Neufville, Donna H. Rhodes

Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA

Email: jason.bartolomei@mit.edu

Abstract. The scope and complexity of engineered systems are ever-increasing as burgeoning global markets, unprecedented technological capabilities, rising consumer expectations and ever-changing social requirements present difficult design challenges that often extend beyond the traditional engineering paradigm. These challenges require engineers and technical managers to treat the technological systems as a part of a larger whole. Existing system modeling frameworks are limited in scope for representing the information about engineering systems. This paper presents a conceptual framework and an improved modeling framework for engineering systems. Its value is that it allows engineers and managers an improved means to visually arrange information and structure discourse in a way that facilitates better systems engineering. It augments the existing literature by providing a clear and concise framework for an engineering system, and provides a methodology for engineers to tag and organize systems information in ways that allow for better collection, storage, processing, and analysis of systems engineering data.

Keywords: Engineering Systems, systems theory, social & technological systems, design structure matrix (DSM), coupled DSM, Engineering Systems Matrix (ESM), and knowledge basis

Manuscript received October 9, 2009.

J.E. Bartolomei is a research affiliate at the Massachusetts Institute of Technology, 77 Massachusetts Ave., Cambridge, MA 02139 USA; e-mail: Jason.Bartolomei@alum.mit.edu

D.E. Hastings is Dean of Undergraduate Education and Professor of Engineering Systems and Aeronautics and Astronautics, Massachusetts Institute of Technology, 77 Massachusetts Ave., Cambridge, MA 02139 USA; e-mail: hastings@mit.edu

R. de Neufville is Professor of Engineering Systems at the Massachusetts Institute of Technology, 77 Massachusetts Ave., Cambridge, MA 02139 USA; e-mail: ardent@mit.edu

D.H. Rhodes is a senior lecturer at the Massachusetts Institute of Technology, 77 Massachusetts Ave., Cambridge, MA 02139 USA; e-mail: rhodes@mit.edu

1. INTRODUCTION

Engineering systems (ES) is a field that seeks solutions to important, multifaceted socio-technical problems. (ESD 2007) ES has emerged in response to the growing complexity of human technological endeavors and the insufficiency of extant theoretical knowledge and understanding to guide engineers, managers, and policy makers responsible for the design and management of large-scale complex systems. It encompasses activities: (CESUN)

- System Engineering
- Technology and Policy
- Engineering Management, Innovation, Entrepreneurship
- System and Decision Analysis, Operations Research
- Manufacturing, Product Development, Industrial Engineering

The problems ES seeks to tackle are not new, as several disciplines have researched, developed and applied methods and tools for decades. However, Rouse (2007) explains large-scale complex systems require a broad “enterprise perspective” that integrates the behavioral, social, and life sciences, as well as management. ES seeks to integrate these disciplines to discover the fundamental principles and properties of systems aimed at fulfilling important functions in society (large-scale), characterized by a high degree of technical complexity, social intricacy, and elaborate processes (complex). Examples of these large-scale complex systems, or *engineering systems*, include critical infrastructure, healthcare delivery systems, and manufacturing systems. (ESD 2008)

Rouse (2007) presents several fundamental areas of research for large-scale complex systems in biological, engineering systems, and natural systems domains. These include:

- The full design objectives for such systems

- Approaches to architecting and modeling systems relative to these objectives
- A methods and tools for model development and use
- Means for evaluation and experimentation with models and real systems
- Approaches, technologies and tools to enable decision support for those who invest in, develop, operate, and use complex systems.

To advance scholarship in these fundamental areas of research, scholars must address underlying philosophical questions related to the ontology (the study of the nature) and epistemology (the study of how we learn and know) of engineering systems. This includes foundational research that explores questions like:

- What do we mean by Engineering Systems?
- What information is required to describe an Engineering System?
- Where does this information exist?
- How do we observe these systems?
- Do we have the tools to develop theories about them?
- What are the ontological or epistemological challenges created by integrating social sciences, engineering, and management?

This paper examines these questions by first reviewing the systems literature to identify existing theoretical gaps. Next, we present a first order ontological model of an engineering system. Using this conceptualization we present an organizing framework for modeling these systems, called the *engineering systems matrix*. We explore what information is required to describe an engineering system, where this information resides, a technique to elicit this information, and methods for putting this information to use.

For practitioners, this paper provides an improved means to construct a knowledge basis that facilitates better systems engineering. In addition, it also presents a methodology for engineers to tag and organize systems information in ways that allow for better collection, storage, processing, and analysis.

2. LITERATURE REVIEW

Systems engineering is one of several bodies of knowledge have emerged using a systems approach for organizing and interpreting the world. The systems view gives a distinct view of humans and nature. (Laszlo 1972) It contrasts with more traditional, reductionist view. Rather than disaggregate systems into simpler and simpler parts, the systems approach embraces a holistic view of the world. (Popper 1961 and 1972; M'Pherson 1974) A classic definition of a "system" is the integration of a set of elements into an orderly whole that functions as an organic unity. (Simon 1962; Marchal 1975; Rescher 1979) The organic unity displays holistic properties greater than the sum of the parts, which are defined as "emergent" properties. (Simon 1962; Bertalanffy 1968; M'Pherson 1974; Moses 2004)

A holistic view has been useful to understand various phenomena; including efforts to understand the social (Parsons 1964; Miller 1978), biological (Bertalanffy 1968; Miller 1978; Kitano 2002), economic (Boulding 1956; Forrester 1961), ecological (Pielou 1969; Graedel and Allenby 2003), historical (Callon 1990; Hughes 1983, 1987, 1990, 1998), political (Quade and Boucher 1968; Vickers 1983; Allison and Zelikow 1999), organizational (Beer 1967; Ackoff 1973; Senge 2006) and technical (Weiner 1948; Buede 2000; Sage 1992, Sage and Armstrong 2000). Consequently, several systems sub-fields have emerged spanning disciplinary boundaries as scholars have found that understanding different types of problems require diverse knowledge.

Systems engineering does not seem to fit nicely in any of the existing typologies. This is a theoretical gap in the systems literature. Take for instance a community hospital. Most would agree that a hospital is a complex system in that it consists of many interacting parts that exhibit well defined, albeit often poorly

understood, behaviors. However, if one considers the hospital's organizational components, infrastructure, technological devices, and the processes as constituent parts of the system, it becomes difficult to classify within the existing typologies. Outside the systems engineering literature, engineering systems have been described and theoretical roots for the study of these systems have been planted in the social-psychology and history of technology literature. These include socio-technical system theory (Trist and Bamforth 1951) and large technological systems (Hughes 1983).

2.1 Social-Technical Systems Theory

Trist and Bamforth (Trist and Bamforth 1951; Trist 1953) developed a socio-technical systems framework for understanding organizational behavior that moved beyond simply observing social interactions, to include explicit consideration of work-tasks and technical systems. (Badham, Clegg et al. 2000) Their work began a sub-discipline within industrial psychology and has since exported several concepts and theories to other disciplines including organization studies, human factors engineering, and management.

Emery (1993) provides an overview of the basic theoretical concepts underlying socio-technical systems (STS). He sees them as open systems that interact with an environment. The concept of purpose is particularly important as STS consist of social/organizational and technical components that interact to achieve the purpose of the system, usually the production of a good or delivery of a service.

Central to STS theory is the development of concepts of work relationships, human task and task interdependencies, and organizational structures that contribute to the production process. STS is devoted to understanding the social psychological factors associated with human-machine interactions (level of automation), team structures, and industrial organizational strategies.

2.2 Large Technological Systems

Hughes developed a comprehensive description for engineering systems he calls Large Technological Systems (LTS) that builds on many concepts from systems theory. (Hughes 1983; 1987; 1990; and 1998) He presents a description of the qualities of LTS, rather than a precise definition. Hughes describes LTS as a seamless web of diverse components that span several domains. LTS components include physical artifacts (technical), legislative artifacts (constraints), organizations (social), and natural resources. Hughes defines LTS as open systems that interact within an environment. For Hughes, the environment consists of two types of factors, those that are independent and those that depend on the system. An LTS is distinguished from its environment by the limits of control exercised by the system's components. These limits define the system boundary.

Hughes defines LTS as purposeful systems that exist to solve problems or fulfill goals, having mostly to do with "reordering the physical world in ways considered useful or desirable." (Hughes 1987) He adds that unlike the "other disciplines of art, architecture, medicine, and play"...LTS are "usually concerned with the reordering of the physical world to make it more productive of goods and services." He also describes a concept of traceability. Traceability implies that all functioning components of the system "contribute directly or through other components to the common system goal" (Hughes 1987) or purpose. He pays particular attention to large-scale systems, which he describes as those technological systems that are society-shaping and he offers several examples of this type of system, such as the U.S. Electric Power Grid.

Hughes discusses the concept of system structure as both the social and technical components of the system display a hierarchical structure. He explains that "the inventors, organizers, and managers of technological systems mostly prefer organizational hierarchy." (Hughes 1987) As a result, over time the technical artifact will tend to a hierarchic structure as well. His discussion of hierarchy aligns well with Simon's (1962) discussion of "The Architecture of Complexity."

In addition to defining and understanding the structure of LTS, Hughes places particular emphasis on the dynamic behavior of the system. Through his work, he successfully demonstrates that certain system behaviors can only be understood as socio-technical phenomena. He cautions researchers to be careful to select the level of analysis, particularly for hierarchic systems, as in the case of “large technological systems there are countless opportunities for isolating subsystems and calling them systems for purposes of comprehensibility and analysis.” Although he does not use the word, Hughes describes the concept of nested complexity, or the behavioral complexity exhibited between subsystems, at the various levels of a hierarchical systems. Hughes warns that the risk of not carefully understanding the appropriate levels of analysis can lead to “only a partial, or even distorted, analysis of system behavior.”

Hughes introduces several system behaviors in his discussion of the patterns of the evolution of LTS. He develops concepts and explains system behaviors that involve social and technological interactions, including invention, innovation, development, momentum, and reverse salients and several others. Hughes uses the LTS as a “less elegant but useful” systems approach for understanding the history of technology. (Hughes, 1987)

3. CONCEPTUALIZING ENGINEERING SYSTEMS

The accepted description of an engineering system is congruent with LTS and STS theory. Engineering Systems are characterized by a high degree of technical complexity, social intricacy, and elaborate processes, aimed at fulfilling important functions in society. Bartolomei (2007) unpacks this description and presents a first-order ontological model, or conceptualization, of an engineering system.

3.1 A Conceptualization for an Engineering System

A conceptualization is an abstract, simplified view of the world that is represented for some purpose. It provides a basis for a body of formally represented knowledge. That is, the objects, concepts, and other entities that are assumed to exist in some area of interest (Genesereth and Nilsson 1987). Explicitly or

implicitly, every engineering model stems from some conceptualization that describes its concepts, classes of system components, and relationships.

This paper's conceptualization for engineering systems seeks to address a theoretical gap that moves beyond existing descriptions towards an ontological model. It formalizes the identification and definition of the domains common to all engineering projects. Figure 1 represents our view of engineering systems domains, components, and relationships.

Figure 1: Engineering System Conceptual Model

These domains include the:

- **Environmental**, the exogenous components that affect or are affected by the engineering system.
- **Social**, the social network consisting of the human components and the relationships that holds among them.
- **Functional**, including the goals and purposes of the engineering system, as well as its functional architecture.
- **Technical**, the physical, non-human components of the system to include hardware, infrastructure, software, and information.
- **Process**, the processes, sub-processes, and tasks performed within or by the system.

3.2 Time

Engineering system components and their environment change over time. Components and their properties can change, be added or dropped. An engineering system exhibits emergent properties that result from interactions between the social and technical components. The system property "flexibility", defined as the measure of ease for which a system can change over time, is an emergent property of the system that can

only be understood by examining the social and technical domains. To measure flexibility, a system engineer must understand sources of change, how change will affect the system components, and who is responsible for authorizing and managing the change.

3.3 System Boundary

Engineering systems are bounded by their constituent components limits of control and alignment with the system's purpose. The components of an engineering system constitute a unified whole that interacts with exogenous components as an open system.

An engineering system may have components that no longer contribute to the goal of the system and remain passive. Some components may detract from the goals. For example, a human component may have localized goal(s) that detract from the global goals. (Merton 1957) Because engineering systems are cybernetic, the system should adjust, to address dysfunction and bring order to the system, but this does not always happen. (Easton 1965) Thus some employees may fight a policy, and the system will self-regulate and adjust to bring components into alignment with the goals of the system. It may replace the components, change policies, or make other adjustments. In some cases, if dysfunction is not addressed, the engineering system can cease. Mutations and adaptation are possible as well. Engineering systems are complex adaptive systems.

3.4 Other properties

Path dependence is another property of engineering systems. The current state of the system depends on previous states. Because engineering systems are socially constructed, one might think of an engineering system as the product of thousands of human decisions over time. Thus, changes of an engineering system can be observed and recorded. Thus, efforts to understand the consequences of human decisions, unexpected events, and other system behaviors require a deep understanding of each of the domains.

Engineering systems can exist at various scales of complexity. Whereas ES as a field focuses attention on “large-scale” complex systems, our conceptualization describes engineering systems at varying degrees of complexity, from the simplest of a single human interacting with an artifact for some purpose (fighter pilot and jet), to transnational supra-system that includes possibly millions of system components (NATO).

The structure of a technical system often reflects that of its organization. (Hughes 1987) We thus feel the social domain should be the basis for defining the levels of complexity of an engineering system. Our descriptions of its various levels are then:

- **Individual:** a person interacting with technical components for some purpose [a pilot operating an aircraft].
- **Group:** a collection of individuals interacting with technical components and each other, where their localized contributions support the global goal of the group [a fighter squadron with multiple pilots operating aircraft].
- **Organization:** interacting groups, whose local contributions support the organization’s goal [a fighter wing of multiple squadrons].
- **Enterprise:** interacting organizations, whose contributions support the enterprise [the United States Air Force consisting of multiple organizations].
- **Higher-Order:** enterprises, organizations, and groups, where each supports higher-level goals.

Higher order levels of complexity may exist as some engineering systems have multiple layers of hierarchy and it may be difficult to delineate between the different levels of complexity. The US Department of Defense (DoD) can be conceptualized as an engineering system as the localized goals of the constituent components (the services and others) exist to support its global goals. Each service can be understood as a system with well-defined system boundaries and each would probably fall under the enterprise level of

complexity. Therefore, the DoD is an example of an engineering system of a higher-order complexity. Other such examples may include NATO, OPEC, and the United Nations. ES is devoted to the study of this type of system.

Figure 2. Levels of Complexity

3.5 Existing conceptual models for engineering systems

Several conceptual models currently underlie well-established systems modeling frameworks. These include the Design Structure Matrix (Browning 2001, Eppinger 2003 and 2002, Guenov and Barker 2005, Steward 1981); Domain Mapping Matrix (Danilovic and Browning 2007); Axiomatic Design (Suh 1998, Guenov and Barker 2005); Unified Program Planning (Warfield and Hill 1972); Quality Functional Deployment/House of Quality (Guinta and Praizler 1993); CLIOS: Complex, Large-scale, Interconnected, Open, Socio-technical System (Dodder, McConnell et al. 2005); and DoDAF the DoD Architecture Framework (DoD 2003). Bartolomei (2007) examined these frameworks in detail and found they each were limited in scope. Some fail to represent important interactions across domains, capture concepts of time, or adequately include both the social and technological aspects of the system.

The DoDAF) is one of the most comprehensive systems modeling frameworks. In the mid-1990s the US DoD created the C4ISR architectural framework to integrate various military communication and surveillance weapons systems. Within this framework there were three main views of the architecture, for the Operational (OV), the Systems (SV), and the Technical (TV). Each represents a standardized collection of models using different modeling languages to represent a predefined aspect of the system. Each view provides details about the system valuable to the various stakeholders involved in developing or actively managing the system. The views were designed for different user groups surrounding a technological system. Military planners used the OV to define the concept of operations for a particular weapon system. Sub views within the OV ranged from simple graphical descriptions of battlespace to high-level network

diagrams easy for non-engineers to understand. On the other hand, the SV and TV included far more sophisticated engineering models to be used for technical communication.

The DoD based the DoDAF a mandatory work-product for every weapon system in the US inventory, on the C4ISR architectural framework. DoDAF Version 1.5 consisted of 26 products representing the operational, systems, and technical views of a system. (Cooper, Ewoldt et al. 2005; Richards, Shah et al. 2006) Each view is described through a set of “products” that includes diagrams, tables, graphics, and narratives. (See DoD 2003).

There were limitations in implementing the DoDAF. Little documentation was provided on how to construct them. Coupled with a focus on final view outputs in early user training, this led to a work product-centric approach to DoDAF development. As a result, many early DoDAF work products were pictures that were neither internally consistent nor complete in capturing relevant data. Furthermore, the DoDAF provided a discrete picture of each individual view, thus it is impossible to capture the dependencies and parallelisms among activities, processes, and supporting technologies (Richards, Shah et al. 2006). The lack of internal consistency was a problem both procedurally and structurally, in that the information presented in discrete views is not traceable to the other views. For example, OV-4 presented a diagram of the organization surrounding a system; however, there are no views that relate the organizational entities to elements in the technical and systems views or the nature of the relations.

In 2009, DoDAF 2.0 was released to address many of these shortcomings. An expanded set of viewpoints spanning the social, technological, and environmental domains replaced the Architecture views. A DoDAF Meta Model (DM2) integrated architecture viewpoints, to insure internal consistency and enhance the analytical and evaluative value of the framework. DoDAF 2.0 also captured information on how the systems interact with exogenous systems (Capability Viewpoints) and incorporated the idea of time for a

number of viewpoints (i.e. CV-3 Capability Phasing, OV-6b State Transition Description, PV-2 Project Timelines, SvcV-8 Services Evolution Description, SvcV-9 Services Technology & Skills Forecast, StdV-2 Standards Forecast).

Software applications were developed to streamline DoDAF construction. These promote internal consistency across views, improve data management, and some provide basic modeling and simulation capabilities. These advances provide procedural improvements in consistency and efficiency in model building.

DoDAF 2.0 is moving in the right direction. The scope of its underlying ontological model has expanded significantly. However, the means to represent important social, political, and economic considerations remain underrepresented. Future iterations of DoDAF should seek to better capture these considerations, and improve means to capture uncertainty/risk, dynamic complexity, and include better hooks for various quantitative analyses.

4. THE ENGINEERING SYSTEMS MATRIX

This paper presents the Engineering Systems Matrix (ESM) as an organizing framework for modeling engineering systems. The ESM operationalizes the conceptualization of an engineering system and addresses the limitations of existing frameworks by providing means to arrange information and structure discourse in a way that facilitates better systems engineering.

Figure 3: Engineering Systems Matrix

Visualize the ESM as an adjacency matrix with identical row and column headings. The diagonal represents the system components and the off-diagonal cells represent the relationship between components. The cells along the diagonal represent a graph of a particular class of node. Each node class

aligns with one of the engineering systems domains. The off-diagonal blocks of cells represent multi-partite relations that relate various classes of nodes. (Steward 1981, Bartolomei, Cokus et al. 2007; Danilovic and Browning 2007)

For those familiar with IEEE 1471, one can interpret the ESM as a collection of models addressing the whole system. Each row/column of the ESM represents a "view", or a particular aspect or a set of concerns. Moreover, the ESM establishes a basis for view-view interactions giving a more formalized, theoretically grounded perspective of an engineering system.

The ESM is designed as a rich data structure able to represent the pertinent information for designing and managing an engineering project. The ESM is both a hyper-graph and a multi-graph. A hyper-graph implies the graph contains different classes of nodes and there are interactions between nodes of different types (e.g. people and processes) A multi-graph implies multiple edge-types can exist between nodes. For example, two social actors might have both a financial and a communication relationship between them. (See Figure 4)

Figure 4: ESM sample edges and attributes

Each node and relation in the system can be described with **attributes**. An example of an object and corresponding attributes are shown in Figure 5. Attributes can be binary, string, numeric or a mathematical function. Lastly, the ESM represents how the system (nodes, relations, and attributes) changes over time.

Figure 5: ESM sample node and attributes

The ESM data structure consists of six node classes that correspond to the five engineering systems

domains. Below is a short description of each node class, with their interactions within each domain. We use a community hospital to highlight the concepts. Community hospitals display qualities of engineering systems by exhibiting a high degree of technical complexity, social intricacy, and elaborate processes. (ESD 2008)

4.1 System Drivers

Starting in the northwest corner, the Systems Drivers Matrix represents the environmental domain. It comprises the factors that act or are acted on by the system. (Bunge 1979) The system drivers include the economic, political, social, and technical influences that constrain, enable, or alter the characteristic of components in the system. Each system driver can have attributes that describe characteristics of each component. For a hospital, system drivers might include “government regulations”, “city utilities”, and “cost of electricity”.

System Drivers X System Drivers Interactions

System driver to system driver interactions may be important. For the hospital, an example of this type of interaction is the effect of “pharmaceutical pricing” (system driver) on “federal pharmaceutical subsidies” (system driver).

4.2 Stakeholders

The Stakeholder Matrix represents the social domain. Stakeholders are the human entities that contribute to the goals of the system and control components within the system. The extent of their control of the system defines the system boundary. To identify the stakeholders for a system, it is useful to ask the four questions suggested by Maier and Rechtin (2000): Who benefits? Who pays? Who provides? And who loses? Clarkson (1995) discusses stakeholders and stakeholder analysis.

Stakeholders X Stakeholders Interactions

Stakeholder-to-Stakeholder interactions form the social network for a system. These can be analyzed in several ways (Wasserman and Faust 1994). In a hospital, social network analysis can be used to understand interactions between staff (stakeholder to stakeholder) to improve organizational design, information flow and process efficiency.

4.3 Objectives

The Objectives Matrix represents the portion of the functional domain that defines the combined purposes of the system. Objectives include all articulated and unarticulated (implied) customers' needs for the system. A hospital objective might be "To provide responsive emergency medical care". Attributes of objectives include quantifiable requirements or performance parameters, such as "average response time to medical emergencies".

Objectives X Objectives Interactions

The Objectives Matrix allows modelers to represent the interactions between objectives. For example, in a hospital the objectives "minimize operating costs" and "maximize employee satisfaction" may be opposing goals to be considered in planning.

4.4 Functions

Functions describe what the system does to achieve objectives. They represent the second part of the functional domain. All functions must relate to at least one objective either directly or through another function. They can have measurable attributes. They may also relate to the form of the systems.. Each object and activity within the system maps to functions of the system. In a hospital a function that supports a system objective "to provide inpatient/outpatient health care" is the function "Provide Emergency Care".

Functions X Functions Interactions

Pahl and Beitz (1996) define the types of flows that exist between functions in products. These include signals (information), energy, material, and spatial relations. There can also be abstract relations between functions, such as hierarchic relations. In a hospital, the “Provide Emergency Care” function can be decomposed into sub-functions such as: “In-process Patients”, “Retrieve Information”, etc. See Blanchard and Fabrycky (2006) for a deeper discussion on functional analysis, decomposition, and allocation.

4.5 Objects

The Object Matrix represents the technical domain or the physical components of the system. These include infrastructure, objects needed to carry out the system functions, and physical entities used by the internal stakeholders. Other system objects include the architectural/physical entities required to carry out the functions or the physical “form” of the system. In the hospital, objects include software, hardware, and infrastructure.

Objects X Objects Interactions

Object-to-object interactions represent how the technical components interact. The Objects Matrix contains the information found in classic engineering models of technical systems. An example from a hospital is a computer network in which the “computer terminals” represent discrete objects and “signals flows” relate objects.

4.6 Activities

The Activities Matrix represents the process domain and includes the processes, procedures, tasks, and work units associated with an engineering system.

Activities X Activities Interactions

Activity-to-activity interactions have been well defined and studied. (Steward 1981; Warfield 1990; Eppinger, Whitney et al. 1994) In engineering systems, the understanding of task-to-task dependency is essential when trying to understand certain system behaviors. In a product development for example, the examination of task dependency can identify unnecessary iterations and feedback cycles that cause inefficiency. These dependencies can often be modeled with techniques ranging from process modeling to discrete event simulation. In a hospital, the activities associated with a particular medical procedure can be represented as parallel, sequential, and series tasks that represent all tasks from pre-operation to post-operation.

4.7 Cross-Domain Interactions

The section below examines and gives examples of important cross-domain interactions.

System Drivers X Stakeholders Interactions

System drivers often affect stakeholder components. For example, changing “demographics of a community” (system driver) requires that a hospital reorganize the mixture of specialties of “doctors” (stakeholders) employed by a hospital.

System Drivers X Objectives Interactions

System drivers often affect the system objectives. A housing development may increase the “community population” (system driver) that affects the “earning goals for the hospital” (objective).

System Drivers X Functions Interactions

A system driver affecting a function might include the hospital adding a new specialty of care (function), such as obstetrics, in response to “changing demographics” (system driver).

System Drivers X Objects Interactions

A system driver affecting an object might involve a “government regulation” (system drivers) that bans a “medical device” (object) used by the hospital.

System Drivers X Activities Interactions

System drivers affecting activities include “government regulation” (system driver) that requires special documentation or other actions (activities) after a medical procedure.

Stakeholders X System Drivers Interactions

Although stakeholders cannot control environmental factors, they can influence them. Thus “hospital doctors” (stakeholders) may volunteer in the community as hospital representatives to raise awareness for blood bank donations. The blood bank, an independent system that controls “community blood supply” (system driver) is an entity the hospital affects even though it does not control.

Stakeholders X Objectives Interactions

Stakeholder components define the objectives for a system. For example, if the hospital declares an initiative to reduce operating costs (objective), several stakeholders usually support, oppose, or are indifferent to it. An ESM can store information regarding various stakeholder positions to inform a game theoretic model to analyze courses of action to align interests. See Dixit and Skeath (2004) for a tutorial on game theory and real world applications.

Stakeholders X Functions Interactions

The stakeholder-to-function matrix maps the influence of stakeholders to the corresponding functions. For example, the hospital’s functions to “*Provide Emergency Care*”, relates to the stakeholders who influence the function.

Stakeholders X Objects Interactions

Stakeholders in the system generally control or supervise the operation of the technical components. There are thus interactions between stakeholders and objects. In the hospital, a “radiology technician” (stakeholder) may oversee an “X-ray machine” (object).

Stakeholders X Activities Interactions

Human components contribute and support many tasks within the system. In a hospital, a “pharmacist” (stakeholder) “dispenses medicine” (activity) for a particular medical procedure.

Objectives X System Drivers Interactions

System objectives often affect exogenous variables. In a hospital, the failure to “respond to medical emergencies” (objective) affects “public perception” (system driver) of the quality of medical care that the hospital provides.

Objectives X Stakeholders Interactions

System objectives can affect stakeholders. Thus a hospital’s goal “to maximize profit” (objective) can affect the doctor salaries (stakeholder attributes).

Objectives X Functions Interactions

The functional decomposition of a system begins with its objectives of the system,. For example, in the case of emergency response the goal of “providing responsive medical care” (objective) includes the functions: “receive message”, “transport patient”, and “treat patient”.

Functions X Objectives Interactions

Since all functions relate directly or through other functions to the system objectives, there will be several function-to-objective interactions. For example, in a resource allocation model, an analyst can measure the money spent per objective by summing the expenditures for the corresponding support functions.

Functions X Objects Interactions

Functional allocation maps function to the form of the system. Each function is likely to have a corresponding physical component. For example, the function “Store Information” corresponds to the hospital’s “medical records” (form).

Functions X Activities Interactions

Interactions exist between the systems’ activities performed and its functions. For example, the function “distribute medication” corresponds to the activities and procedures in the chain of tasks associated with delivering medicine to patients.

Objects X System Drivers Interactions

There are many interactions between the objects in an engineering system and the system drivers. In a hospital, there is a relationship between the its “water consumption” (object attribute) and the “city water supply” (system driver).

Objects X Functions Interactions

All objects in the system can be described with a functional description. In the case of the hospital, the cost attributes for the objects relate to those for the function. This relationship between objects and function costs has been well described in the value engineering literature (Fowler 1990). For example, the “costs” (object attribute) of a hospital’s information system relate to those of the hospital’s sub-function to “manage information”.

Objects X Activities Interactions

Most activities within an engineering system require objects in order to be accomplished. Medical procedures require medical devices. “Knee surgery” (activity) requires a “scalpel” (object).

Activities X System Drivers Interactions

Activities often interact with system drivers For example, analyses of procedures (activities) performed by a hospital are used to inform “community health trends” (system driver).

Activities X Stakeholders Interactions

Feedback from activities is often of interest to particular stakeholders in an engineering system. For example, the “success rate of a particular procedure” (activity attribute) is reported to a “hospital administrator” (stakeholder).

Activities X Objects Interactions

An engineering system includes various activity-to-object interactions. In a hospital, “hospital internal blood supply” (object) is affected by the various surgical procedures (activities) being performed.

Figure 6. Summary of ESM Domain Interactions

4.8 Time

The ESM is designed to represent a system as it changes over time. It must be able to represent changes both in the arrangement of nodes and relationships, and in systems attributes. Capturing time within the ESM is implemented through the system attributes for the nodes and relations. The ESM declares an existence attribute that defines the time intervals each node or relation exists within the system. Once these

are defined, the ESM allows the modeler to define how particular attributes change over time. Attributes can be defined as discrete or continuous, time dependent functions.

In the case of a hospital, the “costs for particular pharmaceutical products” (object attributes) could be treated as variables that change quarterly. These discrete price changes are stored in the ESM.

5. INSTRUCTIONS FOR BUILDING AN ESM

The ESM is useful tool for a variety of analysis including systems design, project management, and systems analysis. The construction of an ESM is an iterative, non-linear process as systems engineers have the flexibility to add, subtract, or change information about the system throughout the engineering process. Before building an ESM, the analyst must determine the goal of the project, define the system to be modeled, and determine the sources of information and data. Depending on the goal of the analysis the analyst may to model some or all of the engineering systems domains.

For a systems design project, a systems engineer using the ESM would begin by listing the stakeholders involved in the design to include customers, individuals involved in the design process, and suppliers. This information is used to construct the Stakeholders Matrix.

The next matrix to be constructed is the Objectives Matrix. Sources of information would include requirements documentation, marketing information from customer interviews and focus groups, and other sources. While constructing the objectives matrix, systems engineers can relate each objective to the stakeholders in the Stakeholders X Objectives Matrix. A systems engineer can derive number of analytical insights from these three matrices that compactly represent the key stakeholders and their objectives. This information is vitally important throughout the system lifecycle since systems engineers should understand which stakeholders affect and are affected by changes in objectives.

Next, systems engineers can construct the Functions Matrix by decomposing the systems objectives into constituent functions. Functional allocation relates the Functions Matrix to the Objects Matrix (Functions X Objects Matrix) and Activities Matrix (Functional X Activities Matrix) as systems designers ensure proposed designs support the system functions.

The Objects Matrix represents the technical design. It relates technical sub-systems and components to each other. The Activities Matrix relates the system processes and tasks to one another. The ESM also allows systems engineers to relate stakeholders to activities (Stakeholders X Activities Matrix), objects to activities (Objects X Activities Matrix) to capture important information about connectivity between domains.

Analysts can identify, store, and relate exogenous factors to be considered in the Systems Drivers Matrix. This allows them to capture important information that is often overlooked. As the system design matures over its life-cycle, the ESM can store system changes within and across domains.

For project management, a systems engineer might follow the same procedure in constructing an ESM. Depending on the maturity of the system being analyzed, the sources and quality of information required to build the ESM might vary.

Simple spreadsheet software (e.g. Microsoft Excel) can generally be used to construct a simple ESM and offers a variety of valuable mathematical options. Steward (1981), Browning (2002), and Sharman and Yassine (2003) show how simple binary adjacency matrices can be used to improve systems designs. However, more sophisticated tools are required to realize the full potential of an ESM to a systems engineering project.

6. COMPUTERIZED TOOL FOR BUILDING A FULLY ATTRIBUTED ESM

A fully attributed ESM is a highly complex representation of an engineering system. To simplify the model-building process, a team of MIT researchers developed software, the System Modeling and Representation Tool (SMaRT). Other knowledge-based engineering tools have been developed to store systems engineering data such as ICAD, Dassault's Knowledgeware, IBM's Rational System Architect, and Vitech's CORE.

What makes SMaRT unique is its ability to construct systems models, like the ESM, from systems documentation of text documents. It does this by using a new procedure for model building called Qualitative Knowledge Construction (QKC). SMaRT stores the information for the different classes of nodes, relations, and attributes over time and allows for the easy retrieval of information and its export for mathematical analysis. For a brief discussion of SMaRT see Bartolomei (2007).

Qualitative Knowledge Construction is a mode of incorporating qualitative information into systems models. Information can be collected through interviews, observation, and documentation. It is converted into text that is subsequently coded or "tagged" by the categories (nodes, relationships, attributes). Once coded, the data is automatically inserted into a rich data structure. The system is thus visually represented by the distribution of the values of the cells; the evidentiary data supporting that representation is contained within the cells. By developing a program for translating textual reports of observations and interviews into coded data capable of being systematically and quantitatively analyzed, the researcher can translate qualitative information into quantifiable data.

In practical terms, QKC offers a means for better bookkeeping when modeling an engineering system. It provides traceability of assumptions to data sources. Dong (2002) found that much of the information on an engineering system resides in the minds of the humans involved with it, not in technical documentation. QKC provides a means of capturing this knowledge and thus providing a transparency of assumptions and

traceability to sources that rarely exists in the study or practice of engineering. By explicitly establishing this transparency and traceability, QKC removes epistemological barriers leading to better model building, design, and management.

7. USING THE ESM

Building an ESM to model a system provides many advantages. The ESM provides a logical framework for organizing and representing information about a systems engineering project. Tools are available to take the information in the ESM to produce classic systems engineering products such as the DoDAF architecture viewpoints (i.e. the OV, SV and TV). Figure 7 represents where DoDAF information can be represented in the ESM.

Figure 7: DoDAF Views captured in ESM

The ESM can also be used for quantitative system analysis. Algorithms have been developed to analyze particular domains within the system. Sharman and Yassine (2004) present techniques for analyzing product architectures that can be applied to the Objects Matrix in the ESM. Browning (2002) presents algorithms to analyze systems processes using the DSM that can be applied to the Activities Matrix in the ESM. Bustnay and Ben-Asher (2005) present a system partitioning method that can be applied to an ESM using graph theory on N^2 charts to identify independent subsystems to simplify system definition, improve interface and module development, and improve system integration and verification. Guenov and Barker (2005) present the COPE design-decomposition model that relates information within and across functional, physical, and process domains. By applying COPE to an ESM, analysts can better identify potential conflicts in the system design and better understand the effect of design changes. Maurer and Lindemann (2008) present graph theoretic algorithms using multiple-domain matrices to analyze engineered systems. These algorithms can be applied to matrices within the ESM.

Bartolomei (2007) used such methods to improve a systems design for the United States Air Force Research Laboratory (AFRL). AFRL is the US Air Force's leading organizations dedicated to the discovery, development, and integration of war fighting technologies for air, space and cyberspace forces. AFRL was tasked to develop a miniaturized unmanned air vehicle (MAV) for US Special Forces. This project integrated several technologies in development within AFRL, industry and academia. Over a two-year period, Bartolomei observed this project and constructed an ESM that captured a time-series data set. He used it to analyze the dynamic complexity of the product development, in particular the impact of engineer turnover within the design organization, the effects of changing requirements on the design, and the evolution of the UAV system technical design. This analysis identified platform and modularity opportunities in the design by allowing system engineers to explore the sources and effects of design changes on the product development.

An ESM of the MAV product development system was constructed using the QKC methodology. Figure 8 is a network representation of the ESM early in the system development. The colors of the nodes represent the different domains. Treating the ESM data as network allows us to analytically examine system characteristics using a methods including network theory and graph theory.

Figure 8. A Network Representation of ESM

The analysis aimed at getting insight into important questions, including: Is our team well organized for the task? Are we allocating resources appropriately? Do we have the right processes in place to execute the project?

An important question integrating some of these concerns is “are we managing this project at the subsystem or component level?” Insights into this question drive organizational structure, resource decisions, and

process development. To glean insight into this question, we used a common network metric, betweenness centrality. This measures the number of times an element within the ESM occurs on the short path connecting two system elements. We used it to compare the top ten ranked objects in the Objects Matrix with the top ten ranked objects in the entire ESM.

Figure 9. Betweenness Centrality for Objects

The results shown in Figure 9 indicated subsystems have greater connectivity than components in the larger system. From a management perspective, this suggested that the MAV program was managed at the subsystem level rather than the component level.

Next, network analysis was used to identify and examine critical system nodes (people, technology, processes). We analyzed the connectedness of stakeholders within the MAV project to duplicate the previous findings. In social network analysis, betweenness centrality is associated with the control of information: stakeholders with higher betweenness have greater influence on a social network than those with lower betweenness.

Figure 10 shows the results of the analysis: the top ten stakeholders ranked by betweenness measure within the social network compared to the entire MAV ESM at time 3. Each acronym “XXXX” represents a particular individual.

Figure 10. Betweenness Centrality for Stakeholders

In the social domain, the highest ranked stakeholders were the chief engineer “PMWJ” and USAF customer “STCC”. PMWJ was the chief engineer; STCC the customer’s representative; PMBI the program manager;

SPOMD, SPOKE, and SPOGR were USAF staff responsible for managing the contract; and KTRDM the lead contractor responsible for the ground station and production contract. These results were as expected.

The rankings changed however when the analysis was expanded to include the functional, technical, and process domains, as shown on the right. For example, subcontractor KTRDM's ranking surpassed that of the program manager PMBI. The ESM thus revealed that KTRDM, a stakeholder seemingly less important when looking at the social network, was far more "connected" and had greater influence in the overall MAV engineering system. The ESM provided a means to better understand the roles of stakeholders involved in the system.

Figure 11. Centrality measures for MAV Stakeholders over Time

The ESM was used to examine how stakeholder characteristics changed over time. Figure 11 compares stakeholder's PMWJ and STCC's degree centrality (the count of connections for a node) and betweenness over a several periods. The analysis compared these centrality measures with those for their replacements at time 5. The metrics for both PMWJ and STCC grew over time and were always much larger compared to those of the average individual within the MAV system. At time 5, however, both PMWJ and STCC were removed and replaced with two new agents. The numbers show that the replacements had significantly smaller centrality measures, an indication that they were not as well connected in the system. In this case, the ESM and analysis can be used as a tool to mitigate downside risks associated with personnel changes in a system. For the MAV program, the changeover in personnel correlated with a decline in cost, schedule, and technical performance. A benefit of the ESM is that it gives new personnel a means to engage more deliberately in the system by revealing important connections within and across domains.

Figure 12. MAV Network Metrics

The next example shows how network measures provide quantitative insights to the entire MAV development system over time. As shown at the bottom five rows in Figure 12, the size of the MAV network and the density of the relationships changed over five different periods. These changes are not surprising, as frequent organizational and technology changes were well documented. It is interesting to note the difference in network metrics at time 5. The degradation of the number of relations far exceeds that of the number of nodes. The average degree $\langle k \rangle$ and clustering coefficient is lower compared to those in other periods and path length seems much longer. This observation prompted a reexamination of the system for insights as to what might have happened.

At time 5, the changes in the network metrics reflect the changes within the system. The loss of PMWJ and STCC was devastating. Their replacements from outside organizations had no experience with the MAV. This disrupted the cohesion of the MAV team as management changed from the flat structure created by PMWJ and STCC's social connections into what seemed to be a classic military stovepipe. The time 5 network shows a longer average path length and a smaller average clustering coefficient, which supports the qualitative data indicating significant structural changes when PMWJ and STCC left the project. Efforts to develop new MAV prototypes then rapidly diminished. Within 6 months, the project lost its capacity to develop MAV prototypes. The project objectives then turned to develop of a small subset of the original system.

The next analysis shows examples from the technical domain. A multi-disciplinary optimization (MDO) was done to analyze the trade-space of optimal design configurations of the MAV aircraft that would maximize the flight endurance and minimize the longest dimension of the aircraft. The analysis required information from the functional and technical domains, namely system requirements in the form of the objective function and a physics-based model of the system that described the MAV's technical performance. Using information from the objectives matrix, the objective functions were defined as Figure 13 shows.

Figure 13. Objective Function for MAV Design Optimization

A physics-based model described the performance of the MAV. Created at the United States Air Force Academy, it was validated by the AFRL MAV engineers. Built in Excel, it calculates aerodynamic performance characteristics. Figure 14 is a screenshot of the user interface.

Figure 14. Screenshot of Excel Based Model for MAV engineering performance

The result of the analysis was a design trade-space and approximated Pareto frontier (red circles), as Figure 15 shows. Each element in the diagram represents a different design configuration for the air vehicle. Designs lying on the upper left are Pareto optimal, those that cannot be simultaneously improved along both dimensions.

Figure 15: MAV Design Trade-space

The MAV development team wanted to identify where in the system to lay in flexibility to allow systems designers to easily change the system to maximize benefit and minimize cost. This flexibility is equivalent to having real options “in” the system. (de Neufville, 2002). These require deep knowledge about the structure and behavior of the technical system. The ESM is a means to store and represent the knowledge required to design flexibility a system.

Flexibility in engineering design enables engineers to better navigate system changes. These can be either emergent or directed. (Eckert and Clarkson, 2004). Emergent change is unexpected and occurs from within the boundary of the system. During the development of the MAV, one of the engineers on the team accidentally broke the wing of a prototype, while “playing” with system at his desk. This failure propagated

throughout the system, causing schedule delays, work disruptions, and other problems. In contrast, directed change is prescribed by stakeholders, usually as new requirements. In the previous example, while bending the wing the engineer conceived of a new “wing roll-up” design for the system. He called the contractor responsible for the wing and levied the new requirement. This directed change initiated a completely new wing design and affected various other aspects of the system.

The team used the ESM to brainstorm various scenarios and examine how the MAV would need to change due to new technologies, new threats, or other factors. One change scenario involved new stakeholders (i.e., the Army) who requested a MAV with longer endurance. Another scenario assumed suppliers would soon deliver a suite of miniaturized components (batteries) with better technical performance. The team used the Excel model for the MDO analysis to identify which system variables most significantly affected the objectives. A cost estimate was performed to determine the costs of different changes.

The team identified the wing and tail connectors and the batteries subsystems as opportunities for flexibility in the system. They found that they could design the fuselage and battery interfaces to enable the MAV to accommodate more batteries and larger wings if new requirements demanded these features.

In short, the ESM holds promise as a tool to gather, organize, and analyze information to better understand the structure and behavior of an engineering system. By using QKC to construct an ESM, researchers were able to gather information about a complex system with transparency. All data in the ESM was traceable to primary sources and open to discourse and scrutiny. The data structure allowed researchers to relate and organize data that included both the technological and the often underrepresented, social domain.

The ESM also holds promise as an industrial tool for systems engineers. The information used to build common systems level-models and products such as QFD, House of Quality, DoDAF, and DSM can be captured in a fully attributed ESM. Plus, the ESM captures information about the social and environmental

domains important to systems engineering projects. Lastly, the ESM has the added advantage of storing time-series information about project.

8. FUTURE WORK

The ESM has been applied to a number of engineering projects. While its conceptual model has proven sufficient to describe the systems of interest, we do not know if the five domains constitute a complete set. In particular, the environment domain needs to be more defined.

Ongoing research has been exploring interactions between two or more engineering systems. This uses multiple ESM's to represent distinct and interacting systems where each exists in the other's environmental domain. Its results show promise as means to explore the management and integration of disparate systems.

The underlying conceptual model and the ESM can improve existing systems engineering software tools (e.g. Vitech's CORE) as well as systems modeling frameworks (i.e. DoDAF 2.0). They provide a broader model for representing engineering systems. Commercially available systems-modeling tools could also integrate analytical methods.

Lastly, there is much to be done to develop guidelines for implementation. Better strategies for data abstraction (how much data to too much, too little?), and the development of criteria for when and how systems models should be applied are worthwhile subjects for research.

9. SUMMARY

The ESM allows modelers to represent each of the Engineering Systems domains. They can map interactions within and across domains and to parameterize these relations. They can also represent the system as it changes over time, both in the structure (changing nodes and relations) and attributes of the components.

Finally, the ESM allows us to treat social and technical interactions on an equivalent basis for the first time. This enables us to construct social and technical networks and to study their interactions with network analysis tools.

Acknowledgments

Jason Bartolomei would like to thank Thomas Hughes for his comments and feedback; David Broniatowski, Matt Richards, Adam Ross, Nirav Shah, and Jennifer Wilds at the MIT Systems Engineering Advancement Research Initiative (SEArI); Chris Glazner, Chris Lawson, Chris Magee, Joel Moses, Joe Sussman, and Dan Whitney in MIT's Engineering Systems Division (ESD); and Tyson Browning.

References

- R. L. Ackoff, (1973). "Science in the Systems Age: Beyond IE, OR, and MS." *Operations Research* **21**(3): 661-671.
- G. T. Allison and P. Zelikow, (1999). *Essence of decision : explaining the Cuban Missile Crisis*. New York, Longman.
- J.E. Bartolomei, *Qualitative Knowledge Construction for Engineering Systems: Extending the Design Structure Matrix Methodology in Scope and Procedure*. Engineering Systems Division. Cambridge MA, MIT, 2007. PhD.
- J.E. Bartolomei, M. Cokus, et al., "Analysis and Applications of Design Structure Matrix, Domain Mapping Matrix, and Engineering System Matrix Frameworks," MIT, 2007.
- L.P. Beckerman, *The Application of Complex Systems Science to Systems Engineering*. *Systems Engineering*, Volume 3, 2000, pp 96-102
- S. Beer, *Cybernetics and management*. London, English Universities Press, 1967.
- S. Beer, *The Brain of the Firm*. Chichester, Wiley, 1981.

- S. Beer, "The Viable System Model: Its Provenance, Development, Methodology and Pathology." *Journal of the Operational Research Society* 35 (1984), 7-35.
- L.v. Bertalanffy, *General system theory; foundations, development, applications*. G. Braziller. Inc., New York, 1968.
- B.S. Blanchard and W.J. Fabrycky, *Systems engineering and analysis* (4th ed.) Prentice Hall, New Jersey, 2006.
- K.E. Boulding, *General Systems Theory: The Skeleton of Science*. *Management Science* 2 (1956), 197-208.
- J.H. Brill, *Systems Engineering -- A Retrospective View*. *Systems Engineering* (1999), 258-266
- T.R Browning, *Applying the design structure matrix to system decomposition and integration problems: A review and new directions*, *IEEE Trans Eng Management* 48:3 (2001), 292–306.
- T. R. Browning, *Process integration using the design structure matrix*, *Syst Eng* 5(3) (2002), 180–193.
- T.R. Browning, *The Many Views of Process: Toward a process architecture framework for product development processes*. *Systems Engineering* 12 (2009), 69-90.
- D.M. Buede, *The engineering design of systems: Models and methods*, Wiley, New York, 2000.
- M. Bunge, *Ontology II: A World of Systems*, Dordrecht, D. Reidel, 1979.
- T. Bustnay and J.Z. Ben-Asher, *How many systems are there? Using the N² Method for Systems Partitioning*. *Syst Eng* 8 (2005): 109–118.
- M. Callon, *Society in the Making: The Study of Technology as a Tool for Sociological Analysis*. *The Social Construction of Technological Systems*. W. E. Bijker, T. P. Hughes and T. J. Pinch. Cambridge MA, The MIT Press, 1990, pp. 83-103.
- CESUN (20101), <http://www.cesun.org/>
- M.B.E. Clarkson, *A stakeholder framework for analyzing and evaluating corporate social performance*. *Academy of Management Review* 20:1 (1995): 92-117.

- C.A. Cooper, M. L. Ewoldt, et al. A Systems Architectural Model for Man-Packable/Operable Intelligence, Surveillance, and Reconnaissance Mini/Micro Aerial Vehicles. Department of Aeronautical Engineering. Wright-Patterson Air Force Base OH, AFIT, 2002.
- M. Danilovic and T. R. Browning, "Managing Complex Product Development Projects with Design Structure Matrices and Domain Mapping Matrices." *International Journal of Management* 25 (2002), 300-314.
- R. de Neufville, Architecting/Designing Engineering Systems Using Real Options. ESD Internal Symposium, Cambridge MA, MIT Engineering Systems Division, 2002.
- A. Dixit and S. Skeath, *Games of Strategy*. New York., NY. W.W. Norton and Company, 2003.
- DoD, DoD Architecture Framework: Version 1.0. D. A. F. W. Group, 2003.
- DoD, DoD Architecture Framework: Version 2.0. D. A. F. W. Group, 2009.
- R.J. Dodder, J. McConnell, et al. The Concept for Using the "CLIOS Process": Integrating the Study of Physical and Policy Systems Using Mexico City as an Example. MIT Engineering Systems Working Papers. Cambridge MA, MIT, 2005.
- Q. Dong, Predicting and managing system interactions at early phase of the product development process. Mechanical Engineering. Cambridge MA, MIT. 2002, PhD.
- C. Eckert and P. J. Clarkson, et al. "Change and customisation in complex engineering domains." *Research in Engineering Design* 15 (2004), 1-21.
- F. Emery, Characteristics of Socio-Technical Systems. *The Social Engagement of Social Science, Volume II: The Socio-Technical Perspective*. E. L. Trist and H. Murray. Philadelphia, University of Pennsylvania Press, 1993.
- S.D. Eppinger, Patterns of Product Development Interactions. ESD Internal Symposium, Cambridge MA, MIT Engineering Systems Division, 2003.
- S.D. Eppinger and D. E. Whitney, et al., "A Model-Based Method for Organizing Tasks in Product Development." *Research in Engineering Design* 6 (1994), 1-13.

- ESD (2008). Engineering Systems Strategic Plan. Cambridge MA,
http://esd.mit.edu/HeadLine/ESD_StrategicPlan2008.pdf
- J.W. Forrester, Industrial Dynamics. Cambridge MA. Productivity Press, 1961.
- J.W. Forrester, Urban Dynamics. Cambridge MA: Productivity Press, 1969.
- T.C. Fowler, Value Analysis in Design. New York NY, Van Nostrand, 1990.
- M.R. Genesereth and N. J. Nilsson, Logical foundations of artificial intelligence. Morgan Kaufmann, Los Altos, California, 1984.
- T.E. Graedel and B. R. Allenby, Industrial ecology. Upper Saddle River NJ, Prentice Hall, 2003.
- M.D. Guenov and S.G. Barker, Application of axiomatic design and design structure matrix to the decomposition of engineering systems. Systems Engineering 8:1 (2005), pp 29-40.
- L.R. Guinta and N. C. Praizler, The QFD book: the team approach to solving problems and satisfying customers through quality functional deployment. AMACOM, New York NY, 1993.
- T.P. Hughes, Networks of power : electrification in Western society, 1880-1930. Johns Hopkins University Press, Baltimore, 1983.
- T.P. Hughes, "The Evolution of Large Technological Systems" in The Social Construction of Technological Systems. W. E. Bijker, T. P. Hughes and T. J. Pinch. The MIT Press, Cambridge MA, 1987.
- T.P. Hughes, American genesis : a century of invention and technological enthusiasm, 1870-1970. Penguin Books, New York NY, 1990.
- T.P. Hughes, Rescuing Prometheus. Pantheon Books, New York, 1998.
- K. Keeney and H. Raiffa, Decisions with Multiple Objectives—Preferences and Value Tradeoffs, Cambridge University Press, Cambridge, UK, 1993.
- Kitano, Systems Biology: A Brief Overview. Science (2002), pp.1662-1664.
- E. Laszlo, The systems view of the world; the natural philosophy of the new developments in the sciences. G. Braziller, New York, 1972.

- P.K. M'Pherson, A Perspective on Systems Science and Systems Philosophy, *Futures* 6 (1974), pp. 219-239.
- M. Maurer and U. Lindeman, The Application of the Multiple Domain Matrix. *Proceedings IEEE International Conference on Systems, Man and Cybernetics (SMC 2008)*, 1998.
- R.K. Merton, *Social theory and social structure*. The Free Press. New York, NY, 1957.
- Marchal, J. H. (1975). "On the Concept of a System." *Philosophy of Science* 42: 448-468.
- J. Moses, *Foundational Issues in Engineering Systems: A Framing Paper*. Engineering Systems Monograph. Cambridge MA: <http://esd.mit.edu/resources/symposium2004.html>, 2004.
- J. Moses and T. Allen, *ESD Symposium Summary*. ESD Internal Symposium, Cambridge MA, 2002.
- E.T. Murman, T. J. Allen, et al. *Lean Enterprise Value*. New York, Palgrave, 2002.
- M. Newman, The Structure and Function of Complex Networks, *SIAM Review* 42:2 (1993), pp. 167-256.
- G. Pahl and W. Beitz, *Engineering design : a systematic approach*, Springer, New York, 1996.
- T. Parsons, Evolutionary Universals in Society, *American Sociological Review* 29 (1964), pp. 339-357.
- E.C. Pielou, *An introduction to mathematical ecology*. Wiley-Interscience, New York, 1969.
- K. Popper, *The Poverty of Historicism*. Routledge, London, 1961.
- K. Popper, *Objective Knowledge*, Oxford University Press, Oxford, 1972.
- E.S. Quade and W. I. Boucher, *Systems analysis and policy planning; applications in defense*. American Elsevier, New York, 1968.
- N. Rescher, *Cognitive Systematization: A Systems-Theoretic Approach to a Coherentist Theory of Knowledge*, Rowan and Littlefield, Totowa NJ, 1979.
- M.G. Richards, N. B. Shah, et al., *Managing Complexity with the Department of Defense Architecture Framework: Development of a System Architecture Model*. CSER 2006, Los Angeles CA.
- A.M. Ross, N.P. Diller, D.E. Hastings and J.M. Warmkessel, Multi-Attribute Tradespace Exploration with Concurrent Design as a Front-End for Effective Space System Design, *AIAA Journal of Spacecraft and Rockets* 41:1 (2004), 20-28.

- W. Rouse, Complex Engineered, Organizational and Natural Systems, *Syst Eng* 10:3 (2007), 260–271
- A.P. Sage and C.L. Lynch, *Systems integration and architecting: An overview of principles, practices, and perspectives*, 1998.
- A.P. Sage, *Systems engineering*, Wiley, New York, 1992.
- A.P. Sage and J. E. Armstrong, *Introduction to systems engineering*. Wiley, New York, 2000.
- P.M. Senge, *The fifth discipline: the art and practice of the learning organization*, Doubleday/Currency, New York, 2006.
- D.M. Sharman and A.A. Yassine, Characterizing Complex Product Architectures, *Systems Engineering* 7 (2003), 35-60.
- S.A. Sheard and A. Mostashari, Principles for Complex Systems for Systems Engineering, *Systems Engineering* 11 (2008) 295-311.
- H.A. Simon, "The Architecture of Complexity." *Proceedings of the American Philosophical Society* 106 (1962), 467-482.
- N. Sproles, The Difficult Problem of Establishing Measures of Effectiveness for Command and Control: A Systems Engineering Perspective. *Systems Engineering* 4 (2001). 145-155.
- D.V. Steward, The Design Structure System: A Method for Managing the Design of Complex Systems, *IEEE Transactions on Engineering Management* 28 (1981), 71-74.
- N.P. Suh, Axiomatic Design Theory for Systems, *Research in Engineering Design* 10 (1998), 189-209.
- J. Sussman, *Collected Views on Complexity in Systems*. ESD Symposium on Engineering Systems Research and Practice, Cambridge MA, <http://esd.mit.edu/resources/symposium2004.html>, 2002.
- A. Tomer and S.R. Schach, A Three-Dimensional Model for System Design Evolution. *Syst Eng* 5 (2002), 264–273.
- E.L. Trist, *Some Observations on the Machine Face as a Socio-Technical System*. Tavistock Institute Document 341. London, 1953.

E.L. Trist and K. W. Bamforth, "Some Social and Psychological Consequences of the Longwall Method of Coal Getting." *Human Relations* 4 (1951), 3-38.

G. Vickers, *The art of judgement: a study of policy making*, Harper & Row, London, 1983.

J.N. Warfield, *A science of generic design: managing complexity through systems design*, Intersystems Publications, Salinas, Calif., 1990.

J.N. Warfield and J. D. Hill, Unified Program Planning. *IEEE Transactions on Systems, Man, and Cybernetics--Part A SMC* 2:5 (1972),

S. Wasserman and K. Faust, *Social network analysis: methods and applications*. Cambridge University Press, New York NY, 1994.

N. Wiener, *Cybernetics*, *Scientific American* (1948), 14-19.