Engineering Elegant Systems:

Postulates, Principles, and Hypothesis of Systems Engineering

A Whitepaper

Michael D. Watson

NASA Marshall Space Flight Center

8 November 2017

ACKNOWLEDGEMENTS

The progress made in defining elegant, product focused systems engineering is due to the work by the past and present members of the NASA Systems Engineering Research Consortium. These researchers represent excellence in their areas of expertise and consistently demonstrate great knowledge, intuition, and research. It is a pleasure working with and discussing with them the challenging questions raised in the discipline of systems engineering. Bryan Mesmer, Phil Farrington, and Paul Collopy at the University of Alabama in Huntsville have been outstanding leaders for the university consortium. Present and past members are:

- Air Force Research Laboratory – Wright Patterson, Multidisciplinary Science and Technology Center: Jose A. Camberos, Ph.D., Kirk L. Yerkes, Ph.D.
- George Mason University: John S. Gero, Ph.D.
- George Washington University: Zoe Szajnfarber, Ph.D.
- Iowa State University: Christina L. Bloebaum, Ph.D., Michael C. Dorneich, Ph.D.
- JBS Solutions: Sherrie Nash, Joey Shelton, Ph.D., Kenny Mitchell, Jeff Downs
- Massachusetts Institute of Technology: Maria C. Yang, Ph.D.
- Missouri University of Science & Technology: David Riggins, Ph.D.
- NASA Langley Research Center: Anna R. McGowan, Ph.D., Peter A. Parker, Ph.D., William Cirillo, Maria Ivanco, Kevin D. Earle
- Oregon State University: Irem Tumer, Ph.D., Christopher Hoyle, Ph.D.
- SpaceWorks Enterprises, Inc.: John Olds, Ph.D.
- Stevens Institute of Technology: Khaldoun Khashanah
- Texas A&M University: Richard Malak, Ph.D.
- The University of Alabama in Huntsville: Phillip A. Farrington, Ph.D., Dawn R. Utley, Ph.D., Laird Burns, Ph.D., Paul Collopy, Ph.D., Bryan Mesmer, Ph.D., P. J. Benfield, Ph.D., Wes Colley, Ph.D.. Jeff Little, Ph.D., George Norris, Ph.D.
- The University of Arkansas: David C. Jensen, Ph.D.
- The University of Bergen: Erika Palmer
- The University of Colorado – Colorado Springs: Stephen B. Johnson, Ph.D.
- Doty Consulting Services: John Doty, Ph.D.
- The University of Michigan: Melissa Greene, Panos Y. Papalambros, Arianne Collopy
- The University of Texas, Arlington: Paul J. Componation, Ph.D., Susan Ferreira, Ph.D.

TABLE OF CONTENTS

1. SYSTEMS ENGINEERING FRAMEWORK

Systems engineering as a discipline is comprised of two main thrusts: System Integration, and Discipline Integration. In this framework, these two thrusts encompass four components: Mission Context, System Integrating Physics, Organizational Structure and Information Flow, and Policy and Law. Figure 2-1 illustrates this systems engineering framework.

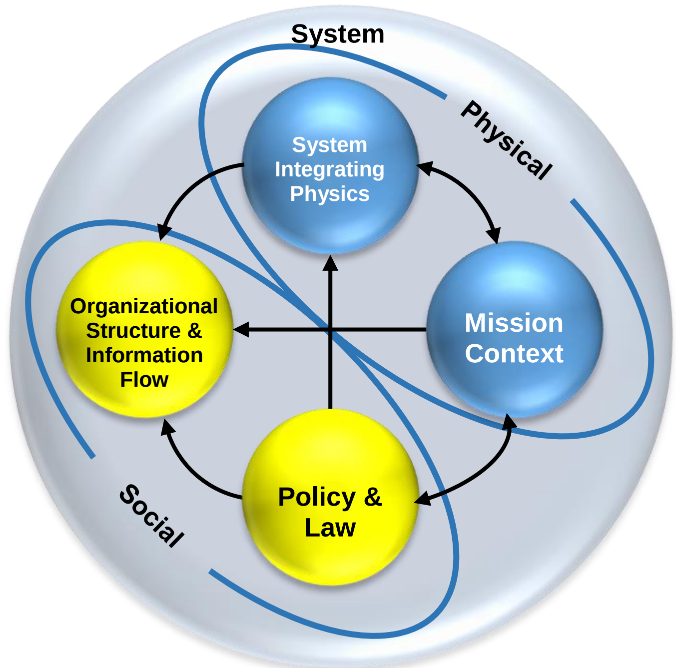

Figure 1-1. Systems Engineering Framework Relationships

System Integration consists of the physical and logical aspects of the system. System Integrating Physics includes the system integrating logic (for logical systems) as the control of many systems is based on logic (i.e., software). The software must have input on the system state to affect the intended system control, and is coupled with the physical system. Environmental interactions such as thermal or radiation where hardware bit errors create logical anomalies in the operation of the system affect software. Also, included as part of System Integrating Physics are the human system integration aspects where the physical and logical functional design must consider human physiology and psychology. This provides a coupling of the user, operator, maintainer, and manufacturer to the system structure, and forms a bridge with the social systems that build, operate, and use the system. Mission context affects both the physical/logical system aspects as well as the social aspects. Mission context is part of System Integration and mainly focuses on the definition of these aspects of the system. The social aspects of mission context are important and the physical/logical choices made for the system can emphasize or amplify these. For example, when a planetary satellite is intended for Neptune the social perturbations are small. When the physics determines that a nuclear-powered satellite is necessary for this distance from the sun, much greater social concern is generated due to potential interaction of the nuclear device with the Earth’s environment in the unlikely occurrence of an accident during launch. In this example mission context influence of the physical system on the social response can be seen.

The social aspects are a major thrust defined by the Organizational Structure and Information Flow, and in the Policy and Law. Organizational Structure and Information flow deal with the maintenance and flow of system information within the organization. This brings in the aspects of sociology in the functioning of the organization. Information flow is a key element in designing and operating an elegant system. Systems engineering assures that the organizational structure supports the necessary flow of information among the system disciplines and assures the design captures this information flow. Gaps, barriers, and organizational reservoirs of information in the flow of information through the organization particularly concern systems engineers. The system design and operations represent the knowledge of the system residing in the organizational structure.

Policy and Law are generally social influences on the system. Policy and Law certainly influence the physical/logical aspects of the system (e.g., requiring a crash-proof casing for the nuclear power cell for launch for the Neptune mission) but are included with the social aspects of the system due to their social considerations. Figure 2-2 shows the mapping of the theory to the systems engineering framework and the characteristics of an elegant system.

2. SYSTEMS ENGINEERING POSTULATES, PRINCIPLES, AND HYPOTHESES

The Systems Engineering Consortium has identified a set of postulates, principles, and hypotheses to articulate the basic concepts that guide systems engineering. These postulates and hypotheses emerged looking at the work of Ludwig Boltzmann and his postulates on gas distributions as an early example of how to characterize the interactions of complex systems. This led us to articulate a set of underlying postulates and hypotheses underlying systems engineering, leading to the 7 postulates and 3 hypotheses stated in this section. These postulates define the domain of systems engineering as well as the system aspects and system influences that are of concern to the systems engineer. The hypothesis contains the seeds of a holistic mathematical basis for systems engineering. In addition, the system postulates define a set of systems engineering principles. The principles serve as an extension of the postulates and are listed after them.

2.1 SYSTEMS ENGINEERING POSTULATES

A postulate is something assumed without proof to be true, real, or necessary.[1] The postulates of systems engineering identify the basis for the discipline. These are further expanded by a set of principles in Section 2.2 below.

Postulate 1: Systems Engineering is system and environment specific, and context dependent.

Description: This is the first and foundational statement on systems engineering. The product (i.e., the system) and its operational environment drives systems engineering and the system's integrating physics, logic, social and cognitive relationships (i.e., context) that are foundational to the specific product or system. Essential to this is the understanding of the mission or use of the product as formulated by the product goals. This includes the aspects of the system needed to operate in an elegant manner and thus considers the entire product lifecycle.

Evidence: The ubiquitous tailoring of systems engineering approaches provides strong support for this postulate. Systems engineering must be consistent with the system being developed or operated. Our research surveying the "NASA 17 Systems Engineering Processes" provides support for this postulate indicating 72% of companies interviewed have systems engineering processes unique to their product. More than 7% of the respondents[2] do not follow a standard process.

Implications: This postulate states that any application of systems engineering should be organized based on consideration of the system being developed or operated. The systems engineering methods applied to a product will and should vary in emphasis and application based on the nature of that product, its environment, and its context.

Postulate 2: The Systems Engineering domain consists of subsystems, their interactions among themselves, and their interactions with the system environment

Description: From a physical, logical, and structural sense, a system is not a single mechanical, or electrical, or chemical entity; it encompasses a set of interacting subsystems. Systems engineering is concerned with combining multiple subsystems, of various physical and logical types, into a *best-balanced* functional whole to accomplish the mission goals. This postulate addresses the system integration aspects of systems engineering. Postulate 3 addresses the discipline integration aspects below.

Evidence: The Individual engineering disciplines deal with the development of their specific functions extremely well. When these functions are integrated with each other and with the environment, the inter-relationships drive the final system performance including emergent properties not evident from the individual system functions. Thus, the engineering of the individual functions is well addressed while the integration of the engineering functions is what makes these functions a system. The domain of systems engineering is the set of these integrated relationships.

Implications: The systems engineer focuses on the interaction of these subsystems, not as a design engineer focused on the details, but as a well-versed integrator. These system interactions, including interactions with the system environment, can drive the design as strongly as the subsystem functions themselves and, when coupled, can potentially create unexpected system responses. The systems engineer must predict and manage these responses.

Postulate 3: The function of Systems Engineering is to integrate engineering disciplines in an elegant manner

Description: The systems engineering discipline is its own engineering discipline, but it is not independent from other engineering and social disciplines. Systems engineering seeks to integrate and incorporate the other engineering and social disciplines in an elegant manner to produce an elegant system throughout the system lifecycle. This postulate addresses the discipline integration aspects of systems engineering. Postulate 2 above addresses the system integration aspects.

Evidence: Any complex system is developed by multiple engineering disciplines with many social aspects influencing the integration. These engineering disciplines with social influences work in an integrated fashion, formerly and informally, to produce these systems.

Implications: The interaction of the disciplines is the focus of the systems engineering domain. The objective is a basic understanding of each discipline with a detailed understanding of their interactions. This incorporates various organizational integration aspects. The systems engineer must be cognizant of the organizational and sociological influences on the system development and operations. The systems engineer must also "engineer" these relationships.

Postulate 4: Systems engineering influences and is influenced by organizational structure and culture

Description: The technical aspects of the system are not the only focus of systems engineering. The system under development drives the development process which has a corresponding influence on the structure of the system's developmental and operational organizations. Similarly, the structure of the organization has an influence on the engineering of the system. These factors also impact the culture of the organization.

Evidence: Organizational mirroring provides examples where the organization maps to system functions. Our current research in "Biased Information Sharing" also shows that system margin is maintained by the organization and not always clearly identifiable in the system design.

Implications: The systems engineer must be cognizant of the culture, the organizational interactions, and their potential impact on the design of the system. The systems engineer must understand how information flows through the organization, is filtered and interpreted by the organization, and is captured by the system design or operational procedures. The systems engineer should work with project management and line management to address issues in organizational information flow and culture to improve the elegance of the system.

Postulate 5: Systems engineering influences and is influenced by budget, schedule, policy, and law

Description: Every project has overarching constraints that extend beyond the physical and environmental. Specifically, most (if not all) projects have a limited budget and schedule. In addition, all systems must conform to established organizational and government policy and laws. These policies and laws put additional constraints on budgets, schedules, and technical solutions. These factors provide a context in which the system is developed and operated. In addition, the system design choices also influence these factors. Government policy and law is based on the understanding of legislators on what systems can actually achieve their intents. Similarly corporate/company policy is influenced by the types of systems the corporation or company chooses to develop.

Evidence: Every project has these constraints. Infinite budgets or schedule do not exist. Policy and law application pervade our systems. Government policy and law are based on the legislators understanding of solutions needed to accomplish their intents. Similarly, corporate/company budgets and schedules are based on the executives understanding of the budget and timeframe necessary to develop a system. This understanding can be seen in budget and schedule allocations, which encompass both a total funding and a timeframe understanding, that are provided by the government or corporate/company executives.

Implications: Social choices drive the establishment of these constraints. People make choices to define budget limits, schedule limits, policies, and laws, whether at the national or organizational level. Thus, physical and logical solutions through these constraints link social choice theory. These choices are based on an understanding of system's abilities to achieve the government and corporate/companyexecutives intents. This understanding drives the budget and schedule allocations and the policies put in place. Similarly, the available budget, available expected duration, existing policy and law can influence choices in the development of a system.

Postulate 6: Systems engineering spans the entire system life-cycle

Description: Systems engineering is not just a development phase activity but continues throughout system operation, decommissioning, and disposal. The organizational relationships and goals change as the system progresses through these phases, but systems engineering continues to integrate the system functions and the system disciplines throughout all phases of the system life-cycle. Operations engineering is responsible for the operation of the system. Systems Engineering is responsible for the various changes/upgrades to the system capabilities.

Evidence: Systems engineering during the development phases is well understood. During the operational phases, systems engineering is still essential as the system goes through maintenance upgrades, new application adaptations, obsolescence driven re-designs, etc. In addition, during decommissioning and disposal, systems engineering is essential to deal with the proper decoupling of the system and ensuring conformance with policy and laws affecting the system disposal.

Implications: As the system progresses through its life cycle, the need for systems engineering changes. A shift takes place from development to operations in terms of the scope of changes and organizational responsibility. Operations engineering is responsible for operating the system while Systems Engineering is responsible for the system changes/upgrades. The baseline operational system, then, becomes the medium in which operational phase system changes take place. The organization changes significantly as the system transitions from development to operations. Organizational relationships and needs are different. Culture can be very different. All of this affects the system and must be dealt with in systems engineering. Another organizational change and culture shift occurs during decommissioning and disposal.

Postulate 7: Understanding of the system evolves as the system development or operation progresses

Postulate 7 Corollary: Understanding of the system degrades during operations if system understanding is not maintained.

Description: A deeper understanding of the system as a whole is gained as the system progresses through development and operations. As the system progresses through development, more detailed decisions are needed and as understanding deepens these detailed decisions can be made. Understanding of the system could also regress, if organizational changes occur due to inactivity of an organizational element (loss of experience), retirement of key experienced individuals, or closure of suppliers.

Evidence: This deepening of understanding is seen in any system development. The technical assessment process shows this as systems progress from concept review to requirements review to design review to acceptance review. Lessons learned from the operations phase are abundant for any system. This deepening of understanding of the system and its application drives commercial product upgrades or new models. Regression of system understanding can be seen in some life cycle extension activities. When system understanding is not maintained, the basis of systems specification becomes unclear and some systems have been found not to perform (either underperform or over perform) to their system specifications. In addition, operational procedures can lose their basis and be difficult to determine when they should be retired or maintained as the system ages.

Implications: Requirements are derived as the system design progresses. Thus, while mission requirements (i.e., part of understanding the mission context) are defined at the beginning of development, the system requirements cannot be established up front. They are a function of the design choices made and are understood progressively throughout the development phase. This also applies to cost and schedules, particularly for new systems where the development or operations result in unexpected changes. Similarly, systems engineers develop models to predict system capabilities, and then refine these models as testing and operational experience is achieved. System models gain fidelity as the design progresses and the interaction between subsystem design maturity and system model maturity must be managed by the systems engineer. These system models become the basis of system operations, as discussed in “Engineering Elegant Systems: Theory of Systems Engineering”, Draft 3, Section 4.9.2. If the system basis is not maintained, then the understanding of why certain procedures or specifications where defined can be lost. This becomes problematic for aging systems, particularly as they reach the generational gap for the workforce after 20 years of service.

2.2 PRINCIPLES OF SYSTEMS ENGINEERING

Systems engineering postulates form the basis of the principles of systems engineering. Accepted truths which apply throughout the discipline define principles. These truths serve as a guide to the application of systems engineering.

Principle 1: Systems engineering integrates the system and the disciplines considering the budget and schedule constraints

This is the application of **Postulate 5**. Budget and schedule constrains the integration of the system and the integration of the disciplines developing or operating the system. Note that budget is the amount allocated to execute the system development or operation and is not the actual cost. The focus of systems engineering is to keep the cost within the budget or recommend when the solution space defined by budget and schedule does not meet the intended system application.

Principle 2: Complex Systems build Complex Systems

This principle is fundamental to the execution of systems engineering. The systems engineer must deal with both the complex system (the organization) that develops the system and the complex system itself. This dual focus forms the basis of the systems engineering framework [i.e., 1) mission context and systems integrating physics and 2) organization structure and information flow]. **Postulates 4 and 5** also capture this duality when the systems engineer is responsible for both integration of the systems discipline functions defined in **Postulate 2** and the development organization disciplines defined in **Postulate 3**.

Principle 3: The focus of systems engineering during the development phase is a progressively deeper understanding of the interactions, sensitivities, and behaviors of the system

This principle is the application of **Postulate 7**. What you do up front does not confine systems engineering and it does not fade as one progresses through the system development. Instead, the knowledge captured, maintained, and improved by systems engineering deepens as the discipline organizations complete their development work and the system functions are integrated. This deepening of understanding enables the systems engineering decisions necessary to produce an elegant system. The focus of systems engineering is on understanding the interactions of the system, many of which are not apparent until system integration (e.g., physical integration, logical integration), as current systems engineering tools do not allow sufficiently deep understanding of system interactions (which we are addressing with new tools discussed in “Engineering Elegant Systems: Theory of Systems Engineering”, Draft 3, Section 5). This leads to a continuous reduction in system uncertainties and identification of system sensitivities. The systems engineer should understand the behavior of the system, including the emergent behaviors, prior to the operational phase. As the development progresses the systems engineer seek the *best balance* of performance, cost, schedule, and risk.

There are several sub-principles to this progressively deeper understanding of the system interactions, sensitivities, and behaviors.

Sub-Principle 3(a): Requirements and models reflect the understanding of the system

The accuracy and completeness of system requirements and system models reflect the understanding of the system. A system that is not well understood lead to poorly stated requirements, requirement gaps, and inaccurate system models and representations. The objective of system engineering is to understand the system (Principle 4(a)) which then produces the proper specification of requirements and proper representation of the system in the system models.

Sub-Principle 3(b): Requirements are specific, agreed to preferences by the developing organization

Preferences are an individual attribute. The organization as a whole, however, must at some point consolidate these individual preferences and agree on specific values (i.e., performance, cost, schedule) that the system will achieve. These agreed-to preferences along with some agreement on the uncertainty in their measure are the system requirements. These are specific to the system being developed and the requirements (agreements) that are necessary for the successful completion of the system should be carefully defined as part of systems engineering. Integration of the disciplines is dependent on these requirements (agreements) between the different disciplines developing or operating the system. Configuration management is an important systems engineering function in maintaining these requirements (agreements) and managing their change in a consistent and coherent manner.

Sub-Principle 3(c): Requirements and design are progressively defined as the development progresses

Mission requirements are defined early in the understanding of the system as a part of Mission Context. The remaining technical requirements are derived based on system design decisions that progress throughout the development phase. Subsystem requirements are not defined completely until PDR and component requirements may not be fully defined until CDR.

Sub-Principle 3(d): Hierarchical structures are not sufficient to fully model system interactions and couplings

System interactions and couplings are varied, involving serial, parallel, nested, and looping relationships. Often there are multiple peer relationships that provide connections among system functions and the environment. Looping, nested and peer relationships support interactions and couplings not seen in hierarchical structures which generally only indicate parent/child relationships. In addition, hierarchical structures do not distinguish subtle interaction effects from strong interaction effects.

Sub-Principle 3(e): A Product Breakdown Structure (PBS) provides a structure to integrate cost and schedule with system functions

The PBS ties cost and schedule to the system functions and components. Cost and schedule are defining constraints (**Postulate 5**) on the system and must be clearly tied to the system functions and operations. The project manager is concerned with labor allocations through the Work Breakdown Structure (WBS). The systems engineer is concerned with the system unit cost and driving cost components seen through the PBS.

Principle 4: Systems engineering spans the entire system life-cycle

This is the application of **Postulate 6** through a set of sub principles that are important throughout the system life cycle. Some of the roles of systems engineers are highlighted in the following sub-principles.

Sub-Principle 4(a): Systems engineering obtains an understanding of the system

Understanding the system is essential to the successful development of any system. The level of understanding of the system possessed by the systems engineer underpins everything they do in terms of engineering the system.

Sub-Principle 4(b): Systems engineering models the system

Systems engineering develops and maintains system-level models to aid in the design and analysis of the system. "Engineering Elegant Systems: Theory of Systems Engineering", Draft 3, Section 4 describes the specific system-level modeling approaches.

Sub-Principle 4(c): Systems engineering designs and analyzes the system

Systems engineering performs design and analysis at the system level. Ideally, this is not merely a cognitive integration of the results of various discipline models, but rather uses system-level models to perform design at the system level. This then informs the system-level guidance to the discipline design to ensure the design closes at the system level as design analysis cycles are conducted. System analysis of the integrated results from the discipline analysis is then performed in a coherent level based on the system-level physics/logic.

Sub-Principle 4(d): Systems engineering tests the system

System engineering is a critical aspect of system testing. The system engineer should define test objectives at the system level to ensure testing not only accomplishes specific discipline test objectives but also at the system level. This can involve separate system tests, modification of discipline tests for system level objectives, or system-level analysis of test data to obtain a system level understanding.

Sub-Principle 4(e): Systems engineering has an essential role in the assembly and manufacturing of the system

The manufacturing of the system is an integrated activity between the system components and the tooling. In addition, changes during manufacturing often have system level implications and can unexpectedly change system interactions. While this sub-phase is the purview of the manufacturing engineer, the systems engineer must stay involved to understand changes, update models, and perform analysis to ensure manufacturing changes are understood at the system level.

Sub-Principle 4(f): Systems engineering has an essential role during operations and decommissioning

Systems engineering has a key role in system operations which are defined by system interactions. We obtain further understanding of the system interactions as the system operational experiences mature. These lead to updates of system models used for operations, and potential system maintenance upgrades or fixes. Similarly, systems engineering provides the understanding during decommissioning in how to de-integrate the system.

Principle 5: Systems engineering is based on a middle range set of theories

Systems Engineering is comprised as a set of middle range theories as discussed in "Engineering Elegant Systems: Theory of Systems Engineering", Draft 3, Section 1.2. Since there is not a unified theory of physics, nor a unified theory of logic, nor a unified theory of sociology, then there is not a unified theory of systems engineering. Three possible theoretical bases are represented in the sub-principles below. These categories are broad systems engineering theoretical basis, system specific physics/logic systems engineering theoretical basis, and sociological systems engineering theoretical basis.

Sub-Principle 5(a): Systems engineering has a physical/logical basis specific to the system

Systems engineering incorporates the fundamental physical and logical mathematical concepts specific to the system. Thus, the mathematical basis of systems engineering incorporates the mathematical basis of the system physics/logic. The systems engineer must fully understand that this is different for different types of systems (**Postulate 1**).

Sub-Principle 5(b): Systems engineering has a mathematical basis

There are several theories that are important to systems engineering, which enable a mathematical basis for the discipline. Systems engineers, in engineering the system, manage information about the system and its interactions as defined in **Postulate 2**, using this information to make development and operational decisions. The laws and relationships defined in Information Theory govern the information on the system. This also applies to the management of system information through the organization as contained in **Postulate 3**. Systems engineers use this information to control the system design or system operations which bring in control theory in a broad scope of controlling the information flow about the system and in defining the control methods to be used to control system states within relevant acceptable ranges over time. Statistical engineering is also a significant mathematical tool which allows for systems understanding and accounts for *uncertainties and sensitivities* as indicated by **Postulate 2**. Below are 7 broad theoretical bases for systems engineering:

1. Systems Theory Basis: **Postulate 2** derives this basis. Systems Engineering uses key concepts such as the division between system and the environment, and the recursive nature of systems engineering concepts as they apply to different "levels" of the system.

2. Decision & Value Theory Basis: Rational decision-making about the design of a system requires mapping of stakeholder preferences into a single scale of value. **Hypothesis 3**, below, states this is a feasible approach.

3. Model Basis: System information is represented and maintained in models, and exported to documents when needed. "Engineering Elegant Systems: Theory of Systems Engineering", Draft 3, Sections 4 and 5 discuss specific system-level models.

4. State Basis: Systems representations maximize use of state variables, and functions are defined as mappings from input states to output states. "Engineering Elegant Systems: Theory of Systems Engineering", Draft 3, Section 4.4 addresses this explicitly.

5. Goal Basis: Systems exist to achieve goals, which are represented as constraints on the output state variables of functions. "Engineering Elegant Systems: Theory of Systems Engineering", Draft 3, Section 4.4 addresses this explicitly.

6. Control Basis: Constraints on function output state variables are achieved by using the physical laws to control those state variables within their ranges.

7. Knowledge Basis: Individuals and organizations construct and maintain knowledge of the system. Systems engineering takes advantage of existing knowledge structures and improve formation of new knowledge across them. Information Theory is an important part of this basis. This knowledge basis is a key aspect of Discipline Integration discussed in "Engineering Elegant Systems: Theory of Systems Engineering", Draft 3, Section 5.

8. Predictive Basis: Knowledge of the system is inherently uncertain. Uncertainties must be modeled probabilistically to understand the level of confidence in system knowledge so as to enable proper decision-making.

Sub-Principle 5(c): Systems engineering has a sociological basis specific to the organization

Systems engineering incorporates the fundamental sociological concepts specific to the development and operations organization. This is a result of **Postulates 3** and **4**.

Principle 6: Systems engineering maps and manages the discipline interactions within the organization

This is an application of **Postulates 3 and 4**. Organizational mirroring, or the correspondence of the organization to the system, is an essential mapping activity in managing the information flow and engineering of the system. The maturity of the engineering organization establishes the need for organizational structure formality. Organizations inexperienced in a specific system will require more formal structure to successfully develop the system. Seasoned organizations with a specific system can operate successfully with little formal organization. Note that project management and organizational line management are concerned with organizational unit responsibilities and personnel matters. A concern of the systems engineer is how these units interact as part of system knowledge and understanding (system information) flows through the organization. The systems engineer works with project management and line management to resolve identified system information gaps or barriers in the organizational structure as these gaps and barriers will lead to flaws in system design, manufacturing, and operation. System dynamics models provide an approach to this principle as discussed in "Engineering Elegant Systems: Theory of Systems Engineering", Draft 3, Section 5.6.

Principle 7: Decision quality depends on the coverage of the system knowledge present in the decision-making process

This principle derives from **Postulate 2**. Engineering organizations often create trade study or task teams to investigate and resolve specific problems, which is a process of organizational flattening. . Decision effectiveness depends on involving the right decision-makers with a sufficiently complete understanding of the decision context and the decision to be made. Decisions are process dependent. Decision methods are directly driven by the information needed by the decision makers.

Principle 8: Both Policy and Law must be properly understood to not overly constrain or under constrain the system implementation

This is the application of **Postulate 5**. Policy and Law act as important constraints on the system. Requirements should not always contain Policy and Law though they are often written in a requirement-like format. The context for the policies and laws is much different, often being much looser than requirements and more likely reflecting high-level system expectations than specific system functional or operational choices. Often, most interpret Policy as having more flexibility than Law. The systems engineer should understand how much flexibility is acceptable by those who set the policy (whether government or organizational) and those who pass the laws.

Principle 9: Systems engineering decisions are made under uncertainty accounting for risk

This principle derives from **Postulates 2, 3, 4, and 7**. Information about the system is progressively understood through the development process and through the operations process. There are several sources of uncertainty in the development and operations. Some of this is natural based on the progressive understanding of the system (**Postulate 7**). Uncertainty exists due to the inability to predict the future with certainty. Uncertainty arises from many aspects of systems engineering, including limited knowledge on system environments and social aspects of the organization which affects information maintenance, creation and flow. Sensitivities must also be understood to ensure the proper focus is given to the different uncertainties. Uncertainty and sensitivities then should be modeled throughout the process. Systems engineering decisions need to be made with sufficient understanding of the system context and the knowledge that uncertainty does exist even as understanding is gained.

Principle 10: Verification is a demonstrated understanding of all the system functions and interactions in the operational environment

Ideally requirements are level (i.e., at the same level of detail in the design) and balanced in their representation of system functions and interactions. In practice requirements are not level and balanced in their representation of system functions and interactions. Verification seeks to prove that the system will perform as the designers expect based on their requirements, models, and designs. This leads to the principle that the proper performance of the system functions (i.e., outputs are within required ranges for a given input state) is the focus of system verification. If requirements are truly level and balanced, then verification of the system functions will result although some redundancy of effort may be expended. If the requirements are not truly level and balanced, then the focus of system verification should be on the system functions. By focusing on the proper system functions, a verification approach can be defined for the system which focuses on its successful application.

Principle 11: Validation is a demonstrated understanding of the system's value to the system stakeholders

System validation is based on the stakeholder's expectations, not on the system requirements, models, and design information. It melds the system as designed and as built with the system as expected by the stakeholders. It is often assumed that the requirements reflect the stakeholder expectations. This is difficult to accomplish in practice due to the melding of external stakeholder expectations with developer expectations. Thus, requirements do not clearly reflect the stakeholder (internal or external) expectations in many system developments. System value models appear to provide a mathematical basis to define and guide the system development with the stakeholder's expectations. "Engineering Elegant Systems: Theory of Systems Engineering", Draft 3, Section 5 discusses this more.

Principle 12: Systems engineering solutions are constrained based on the decision timeframe for the system need.

This principle deals with the time changing nature of systems based on when the decisions for the system are made. The systems engineering solution for a system is formed by the context of the current state of the art and emerging available technologies. For example, what formed the context for air passenger travel in 1935 was very different from the context found in 1965. With the pace of technological advancements, the available solution sets for a given system can change noticeable over as a little as 5 – 10 years such as seen in the electronics industry over the last 5 decades. Thus the decision timeframe is an important aspect of the solution set available to the systems engineer.

Over time, the degree of consistency in stakeholder and user preferences tends to diminish due to environmental changes, emerging technologies, or changes in the makeup of stakeholder and user communities. For systems with long life cycle phases these communities and their preferences can change significantly. This is seen primarily in the operations phase and can also occur in the development phase of long developments. This variation becomes more pronounced as the system life time increases. And with more variation in stakeholders and stakeholder preferences, changes can be introduced to the system which can impact the system's ability to adapt to these preferences or stretch out system long duration developments. A key to managing these social driven changes, is to recognize when these shifts indicate the need for a different system and the time for the current system to move into decommissioning.

2.3 SYSTEMS ENGINEERING STRATEGIES

Based on the current postulates and principles discussed above, there are several strategies of systems engineering. These strategies are approaches to systems engineering that flow out from the mathematical basis defined in *sub-principle 5(b)*. These strategies provide the basic approach to engineer a system at the system level.

Strategy 1: System Theory Strategies

Description: There are two aspects to this strategy dealing with the system as a whole.

1. Systems engineering divides its space of representation into the system, the system's environment, and the system's internal and external contexts (**Postulate 2**).

    The system is the item being designed, assessed, built, and operated. It is the entity engineered to achieve one or more purposes. The environment is the physical, logical, and human environment in which the system is operated. The context constitutes the institutional, legal, political, and economic elements that do not directly interact with the operational system, but which define the system's purpose(s), create the system, and otherwise influence the system. The "internal context" includes the organizations that design, assess, build, verify and validate the system, over which the systems engineer and project manager have some control. The "external context" includes organizations that provide guidance and resources to these organizations, and other factors often beyond direct control of any organization, such as economic and political influences and constraints. Over the life of a system, there can be changes to the system itself, to its operational environment, and to its context. All of them influence a system's purposes, and to the judgment of how well or poorly those purposes are being achieved.
2. In hierarchical representations, systems engineering concepts are typically applied recursively to each level of the hierarchy.

    The recursive strategy is typical of systems. One frequently finds the same idea, such as what "the system" is or what constitutes cause or effect, being applied in different ways to the same physical components or behaviors. This is often due to people having control of, or being interested in different parts of the system. As an example, for an organization that builds a system component, that component is "the system" of most relevance for them. They can and should apply systems engineering strategies and concepts to their component in a manner equally valid as those in charge of the entire larger system. Systems engineering theory, concepts, practices, and terminology must allow for these differences in point of view and should enable accurate communication of information across them. Note, that as stated in sub-principle 3(d), hierarchical representations do not sufficiently represent the system interactions.

Strategy 2: Value Theory Strategies

Description: These strategies deal with the value that the system provides to the stakeholders of the system. System users and operators are an important group of stakeholders when examining system value. "Engineering Elegant Systems: Theory of Systems Engineering", Draft 3, Section 4.6 describes an approach to system value modeling which is based on these strategies.

1. System value is derived from von Neumann-Morgenstern (vN-M) utility functions.

    Von Neumann-Morgenstern utility functions were the starting point for the development of game theory, and are now the basis for an active ongoing program of engineering research in what is often called value theory. This research is based on the idea that to make rational decisions from human preferences, one must create a mathematical representation that is based on a single axis of scalar numbers. For example, money measured in dollars, euros, yen, or some other comparable scale is a very common way in which humans use a single scale of value across a variety of human preferences. Von Neumann and Morgenstern showed that if value can be measured with a single scalar metric, then a variety of mathematical operations can be performed and be used as the basis for a "rational decision." Of course, this is a very strict interpretation of what "rational" means, clearly fitting the needs of mathematical and economic research. However, much effort is now going into applying this approach to engineering as a means to rigorously specify the purpose(s) of a system, and then be able to assess designs against those purposes. Ideally, one desires to create and select the optimal design among all possible design options.

2. When it is not possible to construct vN-M utility, other goals, constraints, or uses of the system can be used to define system goals and preferences.

   For systems, whose purposes can be clearly stated monetarily in terms of making profits, for example, the application of vN-M utility is relatively straightforward. However, for any system in which profit is not the primary purpose, then some other scalar metric could be selected and used as the single measure of value for that system.

3. Specification of requirements should be delayed if practicable during system design and development, in favor of mathematical representation of preferences.

   This is derived from Principle 3 where requirements are progressively defined as the design matures. Specifying requirements too early leads to unnecessary constraints on the system design and can lead to the failure or violation of system constraints during development.

Strategy 3: Model Strategies

Description: System models are an essential systems engineering tool as stated in *sub-principle 4(b)*. System models provide integrated knowledge about the system and the system environment as a whole. Models may be formal or informal (in the minds of individuals). Improving systems engineering requires increasing use of appropriate formal models that have specific uses. Building formal models for their own sake is worse than useless, as it diverts time and resources from useful purposes. All formal models must have specific, known uses to be worthwhile to create and maintain.

1. Systems engineering maximizes the use of models to represent, maintain, and generate knowledge.

   System models provide integrated knowledge bases of the system. Among other things, these models provide a transport medium to communicate system-level information across the system life cycle. The knowledge developed about the system in the development phase is transferred to the operations phase through the system models, and then transferred to the decommissioning phase in a similar manner.

2. System-level representations shall at a minimum include those for value, intention, design, failure, performance, behavior, and agency.

   These types of system models provide valuable information on various aspects at the system level. Of course, the system itself is the full representation of the system. System models, while covering the full scope of the system, only provide a partial view of the systems. The model types identified here provide a set of system models that provide useful system-level, integrated views of the system.

Value models represent stakeholder preferences, ideally using a single scalar metric.

Intention models translate the preferences of the value models into more specific statements of intention for the system, specified ideally as constraints on state variables over time. Models of intention specify what the system "should do" or "ought to do", as opposed to what the designed system actually does. Two types of intention models have been identified to date, a formal Concept of Operations and the Goal-Function Tree.

Design models represent the designed system, as opposed intentions for the system. Information from intention models can be mapped to design models, by mapping the common state variables and constraints between the two types of models. Since the designed system aims to achieve the goals specified by intentions for the system, by definition there must be at least one output of functions in design models that correspond to a stated intention in the intention models. The mapping from intention to design can be "many to many", as opposed to merely "one to one", "many to one", or "one to many". Design models include "physical containment models", which represent components existing inside of other components, such as subsystems existing inside the physical mold-line of the system as a whole. Directed graphs can represent abstract system-level interactions as integrated design models.

Failure models represent mechanisms by which design model components fail and their effects propagate through the system, or by which intentions are violated. Since many failure effects propagate along the same paths as exist in the nominal design, nominal design models are a starting point to create design failure models. However, failures often create new paths that are not represented in the nominal design models, such as electrical short-circuits, or an explosion releasing debris that impacts other components that are not physically connected to each other nominally. Thus failure models are more complete representations of the system than nominal models. Other failure models are based on intention, by assessing ways by which intention is violated using a top-down hierarchy of failure to meet goals. While today these are usually based on natural language, these can be transformed into state-based models that are the logical complements of the GFT.

Performance models come in a variety of types. The main types described here are non-simulation performance models, such as root-locus analyses in linear control theory or Fourier techniques used in radio frequency system analysis. Any non-behavioral methods of assessing performance are included here.

Behavior models are representations that simulate system behavior. These can include abstract models such as executable state machines, but can also include time-domain simulations that range from purely software simulations with no "real" hardware or software, to simulations that include mixes of the system's actual hardware, software, and humans, to full system tests in which the entire actual system is being tested using a simulated environment. The data generated using these models mimic to greater or lesser degree the actual behavior of the system.

Agency models are representations of the "agents" that manage, design, build, test, and analyze the system. These include representations of the organizations and individuals involved with these activities (e.g., Agent Based Models (ABM)), and include critical management representations such as cost, schedule, and organization hierarchy models. System Dynamics models provide a modeling framework to capture the organizational interaction with the system or system design. Agency models are essential to describe and assess critical attributes and performance of the organizations that create the system.

3. Systems engineering shall provide abstract, system-level-compatible representations of discipline models.
   System models provide a medium to integrate the various discipline model results, providing the integrated system view to inform engineering decisions at the system level. To do this, there must be representations of disciplinary knowledge that can integrate with system-level models. This is related to **Postulate 3** and **Principles 7 and 9**.

Strategy 4: State Strategies

Description: Systems engineering is concerned with sufficient knowledge and understanding of system state over time. The state representations of the system then are an essential strategy for elegant systems engineering.

1. Systems engineering makes use of system state variables in system representations.
   System state variables provide a complete and concise representation of the actual system conditions in any set of circumstances (e.g., environmental conditions, performance conditions, operational uses). As such, this is a key tool for the systems engineer as discussed in "Engineering Elegant Systems: Theory of Systems Engineering", Draft 3, Section 4.4.
2. System functions are defined as mappings of input states of state variables to output states of state variables.
   Defining functions as mapping of input and output state variables ($y = f(x)$) provides an unambiguous definition of system functions separate from the specifics of the design that perform those functions (transformations-mappings). This is invaluable to systems engineering and provides the basis for structuring system requirements, system level design, and guidance for discipline-level design and analysis.

Strategy 5: Goal Strategies

Description: The goals of the system define the intended uses of the system. Understanding these goals is critical to an elegant system design. The system development and operation must be tracked to these goals to ensure that the design and operations are meeting the stakeholder's intents. Modeling system goals is discussed in "Engineering Elegant Systems: Theory of Systems Engineering", Draft 3, Section 4.4.

1. System goals are represented both in terms of operational description and of hierarchy.
   Goals define the intentions for the system. Operational description of the goals is necessary to ensure the system application is properly understood. Goals are typically

hierarchical (i.e., goals and sub-goals). This hierarchy can have many forms (e.g., needs, goals, and objectives (NGO)) and must be understood and managed by systems engineering.

2. To the maximum extent practicable, systems engineering shall define goals as constraints on the ranges of output state variables of a function over a specified period of time.
   Mathematically, a goal forms a constraint on the system operation, defining when the system is successful in achieving the goal and when it is not. This is represented as: Goal = rl < y < rh , where y = f(x) between times t0 and t1.

Strategy 6: Control Strategies

Description: Because engineered systems are mechanisms that use and control physical laws to achieve goals, systems engineering relies heavily on control theory concepts. That is, achieving a goal means constraining state variables within relevant ranges, which is what is meant by "controlling" the state variable. Systems engineering takes it as axiomatic that engineering is by its very nature about control. Given this point of view, control theory concepts and strategies are fundamental. This does not mean that systems engineering is limited by current control theory. Rather, systems engineering assumes that current control theory applies, but also that its ideas must be extended beyond the classical domains of linear and robust control.

1. Systems engineering shall provide design and performance representations of the system.
   This is related to system modeling as discussed under **Strategy 3** and *sub-principle 4(b)*.
2. Systems engineering shall simultaneously, with nominal system design, also address design of the system to mitigate the failure to achieve goals.
   Systems engineering is not only concerned with the success of the system but in addressing and responding to system failures (*minimizing unintended consequences* and providing for *system robustness*). This is a fundamental part of the system design and must be addressed in concert with the nominal system design.
3. Systems engineering shall deploy passive and active means to control state variables within appropriate constraints so as to achieve the corresponding goal.
   Systems engineering should consider all means available to achieve system goals. Control of state variables can be achieved by providing passive control of system physics, such as with structures, or active control through open or closed-loop control systems.
4. Systems engineering shall use and extend classical control theory concepts of state estimation and state control to assess the system's ability to achieve goals.
   Control theory application is an essential part of the system design and analysis as discussed in *sub-principle 5(b).* Using concepts of state estimation and control provides a basis for defining system performance metrics for those parts of the system under active control.

Strategy 7: Knowledge Strategies

Description: Knowledge strategies aim to address the human cognitive and social factors at play in the engineering of complex systems (**Postulate 4** and **Principle 6**). These include the fact that organizations and institutions are the centers of knowledge generation and maintenance, but that "knowledge" as such refers to what individual humans understand about a system. In some sense organizations "know" more than any of the individuals in the organization, in other equally important sense only individuals in an organization "know" anything at all. There is no collective mind, only individual minds in a collective enterprise. Working together through social mechanisms and organizations, these individual minds can create a device that uses and encapsulates their knowledge.

1. Systems engineering shall use existing sources of knowledge about the system.
   There are many sources of knowledge about the system within the development or operations organization. Systems engineering should know and make use of these sources of information.
2. Systems engineering shall accept the variability of human interpretation of acceptable and expected system behaviors.
   System organizations are social structures and sociological principles are important for systems engineering to understand how information (i.e., knowledge) about the system flows through the organization. Information gaps, barriers, and reservoirs all exist within the social structure of the organization.
3. Systems engineering shall model the interaction of the system and the organization to identify information gaps, barriers, and reservoirs.
   Systems engineering must be cognizant of how the social structure of the organization affects the understanding of the system and the transferal of system knowledge from the organization to the design and operation. "Engineering Elegant Systems: Theory of Systems Engineering", Draft 3, Section 5.7 addresses this through the application of systems dynamics modeling and in general by creating new system-level knowledge capture and maintenance mechanisms.

Strategy 8: Predictive Strategies

Description: Predictive strategies aim to forecast a variety of future events and their ramifications for the project building the system, and of the system itself. (**Postulate 7** and **Principles 1, 3, 4, 5, 7, and 9**). These include prediction of cost and schedule information for managing the project that creates the system, but also similar information for operations. Important predictive methods are deployed to assess various characteristics of the future system, such as performance, mass margins, computer resource margins, availability and reliability. All of these techniques use probabilistic techniques to address uncertainties of prediction, making probabilistic methods a central aspect of systems engineering.

1. Systems engineering uses predictive models of performance, dependability, cost, and schedule described above.
2. Systems engineering predictive models includes assessments of uncertainty.
   All predictions are uncertain, and hence require estimates of these uncertainties.

2.4 SYSTEMS ENGINEERING HYPOTHESES

The hypotheses are statements that the consortium members are debating and believe can be proven (or perhaps disproven) through research. These statements challenge some of the heuristic notions found in complexity theory and are set in a practical application context (i.e., with real boundaries and constraints) rather than in a theoretical infinite context.

Each of the hypotheses are time dependent as discussed by **Principle 12** above.

Hypothesis 1: If a solution exists for a specific context, then there exists at least one ideal Systems Engineering solution for that specific context

Description: For a given system context that has a system solution, there exists an ideal (optimal or best-balanced) design for the system to accomplish the mission. Budget, schedule, decision timeframe, policy, law, and organizational culture define the context.

Evidence: This hypothesis is stated to drive objective research into the question of an optimal system configuration (i.e., a *best-balanced* system). Hamilton's Principle directly proves this through the relation:

$$\int_{t_1}^{t_2} (\delta T - \delta V + \delta W) dt = 0. \tag{1}$$

Exergy is an expansion of this principle and our research on exergy efficiency of a rocket indicates that an optimal system with an objective of efficiency can be defined across multiple configurations. This is a result that has not previously been achievable in a quantifiable manner. In addition, the value model seems to offer the ability to define an objective function to optimize the system in each context.

Implications: This hypothesis makes no statement about a global optimum. Rather, this hypothesis states there is a local optimum within the confines of the specific developmental and operational context. Note, this means that if this context changes, the local optimum may also change. In the absence of the knowledge of a best balance, the system's development appears as a sociological balance of organizational preferences.

Hypothesis 2: System complexity is greater than or equal to the ideal system complexity necessary to fulfill all system outputs

Description: In each operational context and decision timeframe, the minimum system complexity required to fulfill all the system outputs is the optimal system complexity and the complexity of alternative system designs are equal to or greater than the ideal (i.e., optimal). Note that this is not a simpler is better hypothesis. Minimal complexity involves all aspects of the system as defined by context in Hypothesis 1 description. Being simple in only one context is not necessarily the system with the minimal complexity. The minimal complexity solution involves a best balance of the system and may lead to some aspects being more complex than alternatives and other aspects being less complex. Systems engineers define the minimal complexity holistically and not based on a subset of system aspects. The definition of system complexity is a much-debated topic. Refer to Appendix B for a more detailed review of complexity.

Evidence: This is similar to the statement of Occam's razor. As Albert Einstein is reputed to have said, "everything should be made as simple as possible, but not simpler" (Einstein, n.d.), which underlines a powerful truth of system modeling and systems engineering.

Implications: This hypothesis asserts that less complexity is preferable for a given context. This also states that a more complex system solution than the optimum can fulfill the system application, but not as elegantly. One must realize that the system complexity necessary to complete all intended outcomes of the system satisfies all its operational needs.

Hypothesis 3: Key Stakeholders preferences can be represented mathematically

Description: Systems engineers must understand and mathematically represent the preferences of key stakeholders to make decisions that are consistent with the stakeholder's preferences and to accomplish system goals. This also provides a basis for the validation of the system performance. Making such representations provides a basis for understanding decisions made at any point in the system development.

Evidence: Several approaches have represented preferences in mathematical form including Game Theory and Decision Theory.

Implications: A system value model should be constructible for a given system and stakeholders.

Hypothesis 4: The real physical system is the perfect model of the system

Description: This hypothesis provides a statement of the idea that has long been espoused among statistical modelers. The physical system is the only complete, full, or perfect model of the system.

Proof: Kullback-Liebler Information provides a definition for "ideal" information. This information measure indicates how close a particular model matches the real physical system and is defined as:

$$I(f,g) = \int f(x)\log(f(x))\,dx - \int f(x)\log(g(x|\theta))\,dx \tag{2}$$

Setting this relationship to zero provides a relationship to define the differences in a given model to the real system. This provides a proof that the perfect model of the system is the system itself.

$$\int f(x)\log(f(x))\,dx - \int f(x)\log\big(g(x|\theta)\big)\,dx = 0 \tag{3}$$

$$\int f(x)\log(f(x))\,dx = \int f(x)\log\big(g(x|\theta)\big)\,dx \tag{4}$$

Note, also that copies of systems are not physically identical.

$$f_1(x) \neq f_2(x) \neq \cdots \neq f_n(x) \tag{5}$$

Thus, the physical system only represents itself identically and not other physical copies of the system.

Implications: This provides a mathematical proof of the idea that has long been espoused among statistical modelers. A perfect model, being the system itself, means all other models have limitations which must be recognized. There are various system models that can show various aspects of the system, but no system model can show the complete system. In addition, one copy of the physical system is not identical with another copy of the system. Thus, variation in copies of the same physical system is to be expected at various tolerance levels depending on the design and fabrication approaches.

REFERENCES

[1] Webster's New World Dictionary, Third College Edition, ed. Neufeldt, V., Guralnik, D. B., Simon & Schuster, NY, 1988, pg. 1055.

[2] Componation, P. J.,Dorneich, M., Guiping, H., Farrington, P. A., Hanson, J. L., "Systems Engineering and Project Success in Government and Commercial Organizations", Proceeding of the American Society for Engineering Management, 2013 International Annual Conference, 2013.