NASA/TP–20205003644

Engineering Elegant Systems: Theory of Systems Engineering

M.D. Watson
Marshall Space Flight Center, Huntsville, Alabama

B.L. Mesmer and P.A. Farrington
The University of Alabama in Huntsville, Huntsville, Alabama

June 2020

NASA/TP–20205003644

Engineering Elegant Systems: Theory of Systems Engineering

M.D. Watson
Marshall Space Flight Center, Huntsville, Alabama

B.L. Mesmer and P.A. Farrington
The University of Alabama in Huntsville, Huntsville, Alabama

National Aeronautics and
Space Administration

Marshall Space Flight Center • Huntsville, Alabama 35812

June 2020

Acknowledgments

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

Michael D. Griffin, Ph.D., former NASA Administrator

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., Samantha Marquart Brainard, Ph.D.

Iowa State University: Christina L. Bloebaum, Ph.D., Michael C. Dorneich, Ph.D.

JBS Solutions: Sherrie Nash, Ph.D., Robert S. Ryan, Kenny Mitchell, Jeff Downs

Multidisciplinary Software Systems Research Corporation (MSSRC): Larry Lambe

Massachusetts Institute of Technology: Maria C. Yang, Ph.D.

Missouri University of Science & Technology: David Riggins, Ph.D.

NASA Johnson Space Center: Gordon A. Voss, Ph.D.

NASA Langley Research Center: Peter A. Parker, Ph.D., Ken Toro, Anna R. McGowan, Ph.D., William Cirillo, Marie Ivanco, Kevin D. Earle

NASA Marshall Space Flight Center: James G. Andrews, Jeri G. Eckley, Jennifer S. Stevens

NASA Marshall Space Flight Center, Aetos Corporation: Michael L. Culver

North Carolina State University: Jon Stallings, Ph.D

Alabama A&M University: Emeka Dunu, Ph.D.

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., Robert Price

Tri-Vector Corporation: Joey Shelton, 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 J. Nelson, Ph.D., Dale Thomas, Ph.D., Sean Owens, Casey Eaton

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: Panos Y. Papalambros, Ph.D., Melissa Greene, Ph.D., Arianne Collopy, Ph.D.

The University of Texas, Arlington: Paul J. Componation, Ph.D., Susan Ferreira, Ph.D.

TRADEMARKS

Trade names and trademarks are used in this report for identification only. This usage does not constitute an official endorsement, either expressed or implied, by the National Aeronautics and Space Administration.

Available from:

NASA STI Information Desk
Mail Stop 148
NASA Langley Research Center
Hampton, VA 23681–2199, USA
757–864–9658

This report is also available in electronic form at
<http://www.sti.nasa.gov>

PREFACE

The NASA Systems Engineering Research Consortium was founded to investigate the engineering and mathematical basis of systems engineering. The Consortium brought together some tremendous systems engineering researchers from across the country to contribute their investigative work to an integrated body of knowledge. I have had the great privilege of working with the researchers, discussing their research, and bringing together their tremendous intellectual understandings to define the basis of systems engineering.

In the summer of 2010, as NASA was transitioning from the cancellation of the Constellation program to the Exploration Systems Framework, NASA Marshall Space Flight Center (MSFC) Engineering Directorate Associate Director, Garry Lyles, asked my thoughts on systems engineering. He was looking for a way to advance the discipline. After discussing some of the characteristics of a systems engineer stated by the Jet Propulsion Laboratory/Gentry Lee, Garry asked that I speak with former NASA Administrator Mike Griffin, who was serving as The University of Alabama in Huntsville (UAH) Eminent Scholar and Professor of Mechanical and Aerospace Engineering. The conversation with Mike revealed several common ideas on the challenges for systems engineering and the path systems engineering needed to take to advance as a discipline. We agreed to establish a research effort to consider the advancement of systems engineering and provide an engineering and mathematical basis for the emerging discipline. The MSFC Space Launch System (SLS) program supported the establishment of the Consortium, and a list of researchers who had a strong focus on the engineering basis of systems engineering were asked to participate in early research efforts.

Initial efforts in the consortium were an exploration of different engineering approaches for systems engineering. Phillip Farrington became the principle investigator (PI) when Mike left UAH and became Chief Executive Officer of Schaffer Corporation. The four characteristics of an elegant system defined in Mike's paper, 'How do we fix System Engineering?' guided the effort. These characteristics provided some focus, but a framework needed to bring all of the different engineering aspects together and show their relationships to these characteristics of systems engineering. We derived the framework, beginning in the spring of 2013, looking at the four different aspects of systems engineering identified in the early research: Mission context, system integrating physics, organizational structure and culture, and policy and law. These four areas provided two focuses to systems engineering: system design and integration, and discipline integration.

The systems engineering framework helped to focus the research and identify areas not studied by the Consortium. The Consortium adjusted the research portfolio at this point to address these understudied areas. Mike and I had lunch or breakfast about every 3 months to discuss the progress and direction of the research. Mike challenged the Consortium to find a set of postulates that provided the basis for systems engineering. He used Maxwell Boltzmann's work on the gas distribution laws as an example. After looking at this, the Consortium drafted the first set of systems engineering postulates in the fall of 2013. These postulates have changed and improved over the last 5 years as the Consortium membership reviewed them and contributed new understanding.

The development of the framework, postulates, and hypotheses integrated the research results progressing at each of the Consortium member organizations. This progress led to several advances in system design and integration, and discipline integration, forming the engineering and sociological basis of systems engineering. System design and integration advances include the understanding of the application of systems engineering processes, identification of system integrating physics (system exergy and optical transfer function), information theoretic statistics, state variable approaches (goal function tree and state analysis model), application of Multidiscipline Design Optimization (MDO), system value modeling, and various system modeling approaches in these areas. Section 4 of this Technical Publication describes these system design and integration approaches. Discipline integration approaches include understanding organizational social behavior influences on systems engineering, cognitive science, sociological principles, understanding the impact of government oversight, and various system modeling approaches in these areas. Section 5 describes these discipline integration approaches.

As our research went on, Phillip Farrington retired from UAH and Bryan Mesmer became the Consortium PI. The research led to a deeper understanding of the postulates and their expansion into a set of principles. These principles provided more indepth understanding of systems engineering, providing guidance on systems engineering that leads to the realization of elegant systems. In addition, a set of hypothesis emerged to address some limitations defined in some areas of system complexity and system value modeling. Proofs are in development for these and one hypothesis has been promoted to a principle once a proof (following information theoretic statistical approaches) was constructed.

A special collaboration developed with the Air Force Research Laboratory at Wright Patterson (AFRL-WP) Air Force Base as we investigated the system integrating physics of rockets in 2014. Over a series of lunches, Mike and I discussed mass, energy, and entropy as defining the integration of a rocket. As I went to investigate this, I discovered the work done on thermodynamic exergy at AFRL-WP. This work opened up the realization that physics (thermodynamics in this case) already contained system integration formulas that systems engineering could make use. This has been a very fruitful area of advancement in systems engineering. In reviewing some work on life extension programs at AFRL-WP, we realized that the system information from development was not maintained by the specification, leading us to define the corollary to postulate 7.

One of the last and most difficult areas to define in research is the mathematical basis of systems engineering. Our collaboration with AFRL-WP contributed significantly again. They had done some work with mathematical category theory and, upon looking at this, we realized that this provides the mathematical structure to define a system and therefore, the mathematical basis of systems engineering. This is still new with on-going research. Category Theory provides systems engineering with its unique mathematical constructs, specific to the discipline of systems engineering, and opening up the basis for understanding systems. The development of this will provide a strong foundation for the advancement of systems engineering in system understanding, definition, system model basis, test and verification, and operation.

It has truly been a pleasure to work with all of the researchers participating in the Consortium. Their intellectual might and enthusiasm for the advancement of systems engineering is energizing. The contributions made by these researchers are available on the Consortium Web site (https://www.nasa.gov/consortium). There are over 100 papers on the site documenting the significant contributions by the researchers. Their contributions have been the foundation to advancing the engineering basis of systems engineering!

Michael D. Watson, Ph.D.
NASA Marshall Space Flight Center

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF TABLES

LIST OF ACRONYMS, ABBREVIATIONS, AND SYMBOLS

AAA	avionics air assembly
ABM	agent-based model
ACS	atmospheric control and supply
AI	artificial intelligence
AIC	Akaike information criteria
AR	atmospheric revitalization
ASCB	Avionics and Software Control Board
BER	bit error rate
BIC	Bayesian information criteria
CAC	control of airborne contaminant
CAD	computer-aided design
CAM	computer-aided manufacturing
Cat	category of categories
CCAA	common cabin air assembly
CDR	Critical Design Review
CDRA	carbon dioxide removal assembly
CECB	Chief Engineers Control Board
CH_4	methane
CHM	chemical
CLD	causal loop diagram

LIST OF ACRONYMS, ABBREVIATIONS, AND SYMBOLS (Continued)

CM	configuration management
CMS	Competency Management System
CO	collaborative optimization
CO_2	carbon dioxide
cod	codomain
ConOps	concept of operations
DAC	design analysis cycle
DES	discrete event simulation
DM	data management
DOF	degree of freedom
dom	domain
DRL	data requirements list
DRM	design reference mission
DT	design thinking
ECLSS	Environmental Control and Life Support System
EMI	electromagnetic interference
EOI	Earth orbit insertion
EPS	Electrical Power System
ESM	event sequence model
FIR	finite impulse response
fMRI	functional magnetic resonance imaging

LIST OF ACRONYMS, ABBREVIATIONS, AND SYMBOLS (Continued)

GFT	Goal-Function Tree
GSE	global sensitivity equation
H_2O	water
HEU	high enriched uranium
HST	Hubble space telescope
IDF	individual disciplinary feasible
IMV	intermodule ventilation
ISHM	Integrated Health Management System
ISS	International Space Station
JPL	Jet Propulsion Laboratory
K-L	Kullbak-Liebler
KLDI	K-L discrimination information
LEU	low enriched uranium
LH_2	liquid hydrogen
LiOH	lithium hydroxide
LO_2	liquid oxygen
MAVERIC	Marshall Aerospace Vehicle Representation in C
MBSE	model-based systems engineering
MCA	major constituents analyzer
MDF	multidisciplinary feasible
MDOCA	multidisciplinary design optimization coupling analysis

LIST OF ACRONYMS, ABBREVIATIONS, AND SYMBOLS (Continued)

MIT	Massachusetts Institute of Technology
MLE	maximum likelihood estimate; mediated learning experience
MOI	Mars orbit insertion
MSE	mean squared error
NGO	needs, goals, and objectives
NSA	National Security Agency
NTP	nuclear thermal propulsion
OGA	oxygen generation assembly
OML	outer mold line
OTF	optical transfer function
PBS	Product Breakdown Structure
PDF	probability density function
PDR	Preliminary Design Review
POST	Program to Optimize Simulated Trajectories
PRA	probabilistic risk analysis
RBD	reliability block diagram
RCS	reaction control system
RTD	resistance temperature detector
SAM	state analysis model
SEMP	Systems Engineering Management Plan
SFD	stock and flow diagram
SLS	Space Launch System

LIST OF ACRONYMS, ABBREVIATIONS, AND SYMBOLS (Continued)

SM	service module
SME	subject matter expert
SNR	signal-to-noise ratio
SOI	sphere of influence
TCCS	trace contaminant control subsystem
TEI	trans-Earth injection
THC	temperature and humidity control
TIC	Takeuchi Information Criteria
TMI	trans-Mars injection
TMS	Thermal Management System
TP	Technical Publication
vN-M	von Neumann-Morgenstern
V&V	verification and validation
WBS	Work Breakdown Structure
WM	waste management
WRM	water recovery module
WSM	water separator module

NOMENCLATURE

A	area
$[A]$	matrix of the sensitivities of the subsystem outputs with respect to the subsystem inputs
A_{module}	cross-sectional area of vehicle module or capsule
b	number of bits per transmission symbol
$[C]$	damping coefficient matrix
c_p	specific heat at constant pressure
D	distance transported
d	target miss distance; distance of thruster from center of mass
d_{primary}	diameter of primary mirror
$d_{\text{secondary}}$	diameter of secondary mirror
$E_0(T,p)$	Nernst potential
F	force
$[F]$	force matrix
f	mathematical function or physical system
f_1	focal length of the primary mirror
G	universal gravitational constant
g	model of the system; gravitational acceleration
g_0	Earth's gravitational constant at sea level
H	information uncertainty
$H(R)$	entropy in the received information about the system

NOMENCLATURE (Continued)

$H(R\|T)$	entropy (or uncertainty) of the information received given the information transmitted
$H(T)$	entropy in the transmitted information about the system
h	enthalpy
h_1	enthalpy at the beginning of the heating part of the cycle
h_2	enthalpy at the beginning of the cooling part of the cycle
$h_{\mathrm{air,\,in}}$	specific enthalpy of air entering the blower
$h_{\mathrm{air,\,out}}$	specific enthalpy of air leaving the blower
$h_{\mathrm{coolant,\,in}}$	specific enthalpy of coolant entering the precooler
$h_{\mathrm{coolant,\,out}}$	specific enthalpy of coolant leaving the precooler
I	information discriminator
$\bar{I}$	uncertainty that an event occurred
I_c	moment of inertia about the center of a rotating object
$I(S)$	system information
I_{solar}	solar intensity variation
i	exchange current density
i_{A_0}	exchange current density at the anode
i_{C_0}	exchange current density at the cathode
$[k]$	stiffness matrix
L	list of pairs
L_{membrane}	electrolyte membrane thickness
M	mass of vehicle or planet as indicated
$[M]$	mass matrix

NOMENCLATURE (Continued)

M_E	mass of the Earth
M_p	mass of propellant
M_{planet}	mass of the planet
M_{sat}	mass of the satellite
M_{Sun}	mass of the Sun
M_{veh}	mass of vehicle
m	mass; behavioral response from subsystem *m;* remaining board member with inputs from other board members and SMEs
$\dot{m}$	mass flow rate
$\dot{m}_{\text{air}}$	mass flow rate of air
m_{CO_2}	mass of CO_2 vented to space
$\dot{m}_{\text{coolant}}$	mass flow rate of coolant flow through the precooler
$\dot{m}_{\text{H}_2\text{O, in}}$	mass flow rate of liquid water entering the electrolyzer
$\dot{m}_{\text{H}_2\text{O, out}}$	mass flow rate of liquid water leaving the electrolyzer
$\dot{m}_{\text{O}_2\text{, out}}$	mass flow rate of oxygen leaving the electrolyzer
n	*n*th system design variable
P	pressure
p	pressure in the cell stack; specific board member or SME
p_n	probability of occurrence
Q	heat transfer
Q_{aero}	aerothermal heat transfer
$\dot{Q}_{\text{blower}}$	heat loss rate from the blower
$\dot{Q}_{\text{elec}}$	heat loss rate from the electrolyzer

NOMENCLATURE (Continued)

Q_{fg}	heat transfer in converting propellant fluid to gas state prior to combustion
q	number of iterations in the board discussion
qk	subdivided smaller segments
R	mirror radius of curvature
$R_{\text{interface}}$	interfacial resistance
$R_{\text{Sun, planet}}$	distance from the planet to the Sun
$r_{\text{altitude, final}}$	altitude of vehicle at orbital injection
$r_{\text{altitude, initial}}$	altitude of the vehicle on the pad
r_{SOI}	radius of the planet's sphere of influence (SOI)
$r_{\text{Sun, planet}}$	distance of planet relative to the Sun
$r_{\text{vehicle, planet}}$	distance of vehicle relative to the planet
$r_{\text{vehicle, Sun}}$	distance of vehicle relative to the Sun
S_{gen}	entropy generated
s	entropy
s^2	variance parameter
s_1	entropy at the beginning of the heating part of the cycle
s_2	entropy at the beginning of the cooling part of the cycle
$s_{\text{air, in}}$	specific entropy of air entering the blower/precooler
$s_{\text{air, out}}$	specific entropy of air leaving the blower/precooler
$s_{\text{coolant, in}}$	specific entropy of coolant entering the precooler
$s_{\text{coolant, out}}$	specific entropy of coolant leaving the precooler
T	temperature
T_0	reference environment temperature

NOMENCLATURE (Continued)

T_{blower}	operating temperature of the blower
T_{cell}	operating temperature of the cell stack
T_{elec}	operating temperature of the electrolyzer
T_{engine}	engine thrust
Th	thrust
t	time
t_{cycle}	total duration of a full cycle
t_{heating}	duration of the heating part of the cycle
$[U]$	structural displacement matrix
u_{CO_2}	internal energy of CO_2 vented to space
V, v	velocity; value
V_e	velocity of exhaust gas (distance from combustion chamber to nozzle exit)
Vol	volume
$V(T,p,i)$	voltage of the cell stack as a function of temperature, pressure, and current
V_{vehicle}	velocity of vehicle
$V_{\text{vehicle, planet}}$	velocity of vehicle relative to the planet
$V_{\text{vehicle, Sun}}$	velocity of vehicle relative to the Sun
W	work
W_{bl}	work done in engine boundary layer flow
W_{buffet}	work done by system against aerodynamic buffeting
W_{div}	work done in engine flow divergence
W_{elec}	electrical work done by system (e.g., avionics, communication)

NOMENCLATURE (Continued)

$\dot{W}_{\text{elec}}$	power consumption rate of the electrolyzer
W_{ER}	work done by engine energy release
$\dot{W}_{\text{heating}}$	power consumption rate during the heating part of the cycle
W_{KE}	work done by change in kinetic energy
W_{mech}	mechanical work done by system
W_{pointing}	work done by pointing system to maintain trajectory course
$\dot{W}_{\text{rev, blower}}$	reversible work rate limit of the blower
$W_{\text{vibration}}$	work done be structural vibration from engine thrust
X	exergy
X_A	design variable
X_{ACS}	exergy of the atmospheric control and supply
$\dot{X}_{\text{ACS}}$	rate of change of exergy of the atmospheric control and supply
X_{AR}	exergy of the atmospheric revitalization
$\dot{X}_{\text{AR}}$	rate of change of exergy of the atmospheric revitalization
X_B	design variable
X_{des}	system exergy destruction
$\dot{X}_{\text{des, CO}_2\text{, venting}}$	average exergy loss from venting CO_2 to space
$X_{\text{des, cooling}}$	exergy change of the sorbent bed during the cooling part of the cycle
$X_{\text{des, heating}}$	exergy change of the sorbent bed during the heating part of the cycle
$\dot{X}_{\text{des, OGA}}$	exergy destruction rate of the OGA
$\dot{X}_{\text{des, precooler}}$	exergy destruction rate of the precooler
$\dot{X}_{\text{des, sorbent bed}}$	average exergy destruction rate of the sorbent bed

NOMENCLATURE (Continued)

X_{ECLSS}	exergy of the ECLSS
$\dot{X}_{\text{heat in}}$	rate of change of exergy transfer into the system
X_n	specific piece of information
X_{THC}	exergy of temperature and humidity control
$\dot{X}_{\mathrm{THC}}$	rate of change of exergy of temperature and humidity control
X_{WM}	exergy of waste management
$\dot{X}_{\mathrm{WM}}$	rate of change of exergy of waste management
X_{WRM}	exergy of water recovery management
$\dot{X}_{\mathrm{WRM}}$	rate of change of exergy of water recovery management
x	physical variable
x, y, z	Cartesian coordinates
Y_A	output behavior variable
Y_B	output behavior variable
y_{mol}	mole fraction of chemical species (products and reactants) in the combustion process
α	scale parameter; mass fraction of chemical species (products and reactants) in the combustion process; linear coefficient of expansion
ΔT	temperature difference
Δt	time difference
δT	kinetic energy differential change in the system
δV	potential energy differential change in the system
δW	work differential change in the system
$\varepsilon_{\mathrm{CO_2}\ \text{venting}}$	exergy efficiency of venting CO_2 to space

NOMENCLATURE (Continued)

$\varepsilon_{\text{precooler}}$	exergy efficiency of the precooler
η	thermodynamic exergy efficiency of each system
η_{act}	activation over potential
η_{anode}	anode potential
η_{cathode}	cathode potential
$\eta_{\text{cell mismatch}}$	solar cell losses
$\eta_{\text{cover glass}}$	cover glass losses
$\eta_{\text{interface}}$	interface over potential
η_{membrne}	electrolyte membrane over potential
η_{ohmic}	ohmic over potential
$\eta_{\text{parameter calibration}}$	system calibration losses
$\eta_{\text{UV, micrometeorite}}$	degradation losses by years in flight
θ	solar array angle to the Sun; model parameters
μ	location parameter
ρ_{atm}	planetary atmospheric density
σ_{membrane}	electrolyte membrane conductivity
τ	time window
Φ	set with no elements (null set)
ψ_{flow}	flow energy
ψ_{image}	image wave function
ψ_{obj}	object wave function
ω	angular velocity

TECHNICAL PUBLICATION

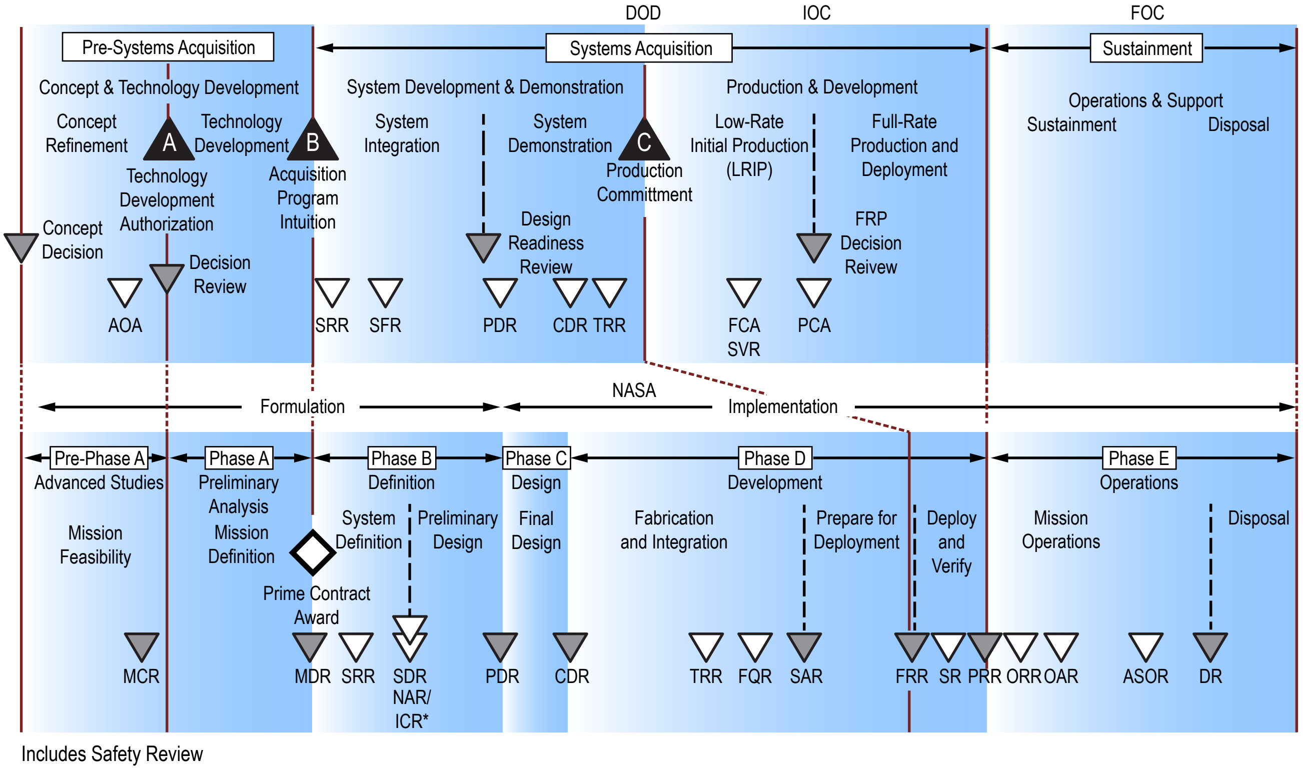
ENGINEERING ELEGANT SYSTEMS: THEORY OF SYSTEMS ENGINEERING

1. ENGINEERING ELEGANT SYSTEMS

Systems engineering today follows a sequence of events to produce a system. The process begins with understanding the application of the system and the desires of the identified system stakeholders. This understanding guides the definition of the system architecture. Systems engineers typically conduct a series of architecture trade studies that culminate in the selection of a specific system configuration. System design proceeds from this configuration definition executed by various engineering disciplines. Systems engineers often assume, sometimes implicitly, that the decomposition of the system functions to these various engineering disciplines is linear. They define this linear decomposition as a set of functional requirements and interface requirements for each subsystem. The separate engineering disciplines design the system functions and subsystems and integrate these functions and subsystems via interfaces into a consolidated system. System interactions are assumed to be contained by the interfaces and generate derived requirements for the subsystems traceable back to the system functions. System production and operations are considered in the development phase, and in turn, development information is used to define system production and operations activities that occur at the end of the design process.

This system engineering process can work well but it can also work poorly, or not at all. Often the approach to managing the sequence of events leads to very inelegant systems. Considering a systems engineering process that can sometimes succeed and sometimes fail brings to the forefront a number of questions, which include “What are we doing correctly?,” and “What are we doing wrong?,” and “How do we address this?” The answer to these questions leads to a new understanding of systems engineering to engineer elegant systems. These questions will be considered separately in the following paragraphs.

What are we doing correctly? Several aspects of systems engineering provide a template to engineer a system: system lifecycle, use of engineering disciplines in design, system testing, and configuration management (CM). The engineering lifecycle has an important sequence of events: Understanding the system application, defining the system architecture, selecting the system configuration, designing the system, testing the system, and then operating the system, is a correct sequence. There are many variations to the system development lifecycle (e.g., waterfall, spiral, block, agile) but they all use this basic construct. Figure 1 illustrates the differences between the NASA and Department of Defense system lifecycles.[1]



AOA	Analysis of Alternatives	IOC	Initial Operational Capability	SAR	System Acceptance Review
ASOR	Annual Systems Operations Review	MCR	Mission Concept Review	SDR	System Design Review
CDR	Critical Design Review	MDR	Mission Design Review	SFR	System Functional Review
DR	Decommissioning Review	NAR	Non-Advocate Review	SR	Safety Review
FCA	Functional Configuration Audit	OAR	Operational Acceptance Review	SRR	System Requirements Review
FOC	Full Operational Capability	ORR	Operational Requirements Review	SVR	System Verification Review
FQR	Formal Qualification Review	PCA	Physical Configuration Audit	TRR	Test Readiness Review
FRR	Flight Readiness Review	PDR	Preliminary Design Review		
ICE	Independent Cost Review	PRR	Production Readiness Review		

Figure 1. System lifecycles.

Another correct aspect of systems engineering is the use of engineering disciplines to develop the various subsystems. This is a well-established practice, based primarily on the physics or logic of the system as applied by the various engineering disciplines. Another valid emphasis in systems engineering is system testing. This is important to confirm engineering assumptions, uncertainties, and sensitivities. In addition, configuration management and associated reviews and analyses of performance and risks associated with the system configuration has also proven to be a durable and successful aspect of systems engineering. The level of effort and detail devoted to each of these practices should be tailored to the system based on factors such as system cost, complexity, and risk to humans or infrastructure. System production and operations activities are based on the system information generated during system development. These are all successful aspects of the engineering process used in system development and are the main reasons that we achieve system success.

So, what is missing? What are we doing wrong? There are several areas that need to be improved in systems engineering: Understanding of stakeholder value, system integration and design, linear assumptions in systems decomposition and configuration definition, identification of system interactions, understanding of production changes on system functionality, understanding of operational interactions, and understanding of organizational influences on system design and integration. System stakeholder values are often not connected well, or only loosely connected, to the actual system implementation. Stakeholder expectations are often biased by what is available rather than what new capabilities can be developed to meet their objectives. This makes new ideas difficult to match with expectations based on the stakeholder's experience. These factors lead to inelegant systems that do not fully meet the expectations for the system. Poorly defined system design and integration (at the system level) leads to more emphasis on subsystem aspects than a balance of the system as a whole. Configuration management, which includes the capture and documentation of mechanical, electrical, and data interface requirements and design, is a necessary but not sufficient aspect of integration. Configuration management covers a large portion of the interfaces yet does not capture all system interactions because it often implicitly uses the previously mentioned linear assumptions. System decomposition and system integration (recomposing the system from its parts) using interface documentation are assumed linear and are usually mapped hierarchically. This is not true for all system functions. For example, nonlinear interrelationships, such as those encoded in software algorithms (including feedback loops) or created by fluid dynamics, can significantly affect systems functions, increasing uncertainty and creating unexpected sensitivities. The hierarchical representation does not reflect the complex interactions that occur among the subsystems and the system environment.

Unrecognized interactions lead to unexpected system responses and to changes in design and function to correct the system design, sometimes late in the development phase. Subsystem testing does not identify system interactions not visible from the disciplinary design perspective, and is insufficient to assess human factor considerations. Designing the system from the contributing discipline perspective is necessary but not sufficient, and is often difficult for a design to close (i.e., successfully achieve all requirements). Finally, system production can create new interactions (e.g., with production tooling), which often lead to design changes after the system enters production. Tooling ability to fit (or interact) with some aspect of the system, human interfaces to the system for manual operations and inspections, material issues (e.g., stress, voids, inclusions (impurities)), or other production-induced defects lead to production changes to the system. System operations may have the greatest need to understand the full set of system interactions properly defining operational sequences and procedures. This requires the full set of system interactions be defined early and system operations to play a considerable role in the design of these system functions and interactions. System interaction with human operators is an important part of this operations understanding, taking into consideration the capabilities and limitations of the human as a functional part of the final system. As discussed, the current approach to system design misses many important system interactions and does not allow a 'best balance' for the system as a whole.

In addition, there are many systems engineering processes that focus on technical management addressing the organizational side of systems engineering. Even within their organizational domain, these processes are limited. They do not address the organization as a whole, do not explicitly address the effects of the organization on the system design, and do not address system information flow through the organization. The technical planning portion of technical management is important early in the program. Although a Systems Engineering Management Plan (SEMP) can include some identification of information flow through the organization, this is not necessarily explicit and not consistently done. If the organizational understanding is not present in the SEMP, then the organization is not well understood and not likely tuned for good system information flow. A poor organization is not likely to create an elegant system.

So, how do we address this? An approach to systems engineering is needed that focuses on the entire system, including the effects from the system's designing and supporting organizations. Approaches and tools are needed to conduct system design from a holistic perspective, complementary with the disciplines perspectives. System stakeholder preferences can be captured by a system value model that enables comparison of the entire system design with the system expectations. System design then needs to be accomplished at the system level, rather than the discipline level. While the disciplines are governed by their discipline physics, what physics governs the integration of these disciplines? A new physics or some combination of the physics emanating from the disciplines and from other sources such as human and social sciences? Physics already contains the system integration relationships necessary to understand the integrated system. The approach to this system-integrating physics is addressed as part of system integration in this Technical Publication (TP). Designing at the system level also requires system level design and analysis. System state variables are the integration points of the system and allow a coupling of the discipline designs into a system as a whole. Multidisciplinary optimization makes use of these state variables to design and analyze the system as a whole, incorporating the full set of system interactions and defining the best balance of all the subsystems for an optimized system design. Engineering statistics allow analysis of the system uncertainties and sensitivities leading to better balance of competing and cooperating influences and disciplines within the system and from outside the system. Full system testing is an important aspect of systems engineering. System level testing confirms the understanding of the system interactions as a whole and builds on the understanding gained from the system design and integration methods discussed in this TP.

Discipline integration is an aspect of systems engineering dealing with the organizational structure and information flow. Since these organizations are social structures, aspects of sociology can help explain how an organization functions, defining characteristics of the organizational structure, and the organizational communication. These characteristics are as important to systems engineering as the system design. Some aspects of the system design reside more in the organization than in the physical/logical design. The information flow through the organization and organizational structure is crucial to the development of an elegant system, as it deals with the integration of the disciplines that develop or operate the system. The flow of information through the decision structure is also important to minimize uncertainty in decisions made about the system. The ability to reconsider decisions outside the decision structure, when warranted, is also an important sociological aspect. Cognitive science and information theory provide important keys to understand system thinking and the flow of information through decision-making boards. A properly constructed SEMP can establish the key principles of discipline integration early in the system development lifecycle and provide a great aide in understanding and managing system information flow through the organization.

This theory TP deals directly with the two systems engineering elements needed for development and operations at the system level: System design and integration and discipline integration. The approaches are defined so that a holistic system development and operational approach can be attained, enabling the engineering of elegant systems. The specific steps to execute these approaches are contained in the companion TP, "Engineering Elegant Systems: The Practice of Systems Engineering."

So, let us begin with a holistic definition of systems engineering, followed by the definition of a systems engineering framework that is supported by a set of postulates, hypotheses, principles and strategies. These are then applied to specific techniques for System Design and Integration, and Discipline Integration. This provides a theory enabling the engineering of elegant systems.

1.1 Definition of Systems Engineering

Systems engineering is the engineering discipline that focuses on the whole system. Webster defines a system as 'a set or arrangement of things so related or connected as to form a unity or organic whole.[2] One can find systems in many contexts and they do not exist in true isolation. There are three system types: physical, social, or logical. Physical systems include those that are mechanical, electrical, chemical, biological, etc. Physical systems operate in environments whose interactions with the system significantly affect the system's functions and performance. Social systems exist in expanding spheres of social environments. Social systems include organizational, corporate, regional, national, and international. These systems have structure and include culture. Regional and national cultures affect the local culture that, in turn, affects organizational social systems. For example, international efforts are affected by multiple national cultures. Software systems contain logical (mathematical) systems in the form of algorithms. Inputs affect logical systems and the physical and social environments affect these inputs when interacting with the human community and physical equipment. These system types exist in interrelationships of many systems that provide the specific services or functions intended by the user. Interactive gaming, for example, involves logical systems coupled through the social system of the users.

Systems engineering must account for the 'connections' or interactions among the system functions and with the system environments. In many cases this includes physical, social, and logical interrelationships and environments. The systems engineer must account for these interrelationships and design them in such a way as to provide reliable, intended results for the system use (i.e., a best-balanced system). A mechanical, electrical, social, or software engineer may design the individual functions within a system. However, the integration of these functions with themselves and with the environment is the domain of the systems engineer.

Systems are unique in their functions, environments, and interactions. Thus, the systems engineer must clearly understand the context for the system application and the defining integrating relationships of the system. One cannot approach systems engineering generically but must consider the unique characteristics and intended outcomes for the system.

Considering all of these characteristics, systems engineering can be defined as: the engineering discipline that integrates the system functions, system environment, and the engineering disciplines necessary to produce and/or operate an elegant system.

1.2 Theory of Systems Engineering

There are many different approaches used to investigate systems theory. Previous theories on systems provide some useful concepts from systems science but none have provided a definitive theory of systems engineering. Ludwig von Bertalanffy developed general systems theory in the 1950's. His empirical approach evaluated different types of systems and sought to define commonalities (i.e., isomorphism) between the different system types.[3] W.R. Ashby followed a deductive approach, starting from the set of 'all conceivable systems' and then looking at their commonalities.[4] Jay W. Forrester approached systems theory from a system dynamics viewpoint,[5] considering how systems change over time and having broad application for all system types. Robert Rosen also worked on dynamical system theory for biological systems.[6]

The sociologist, Robert K. Merton, provided some insights into organizing the various systems theories. He described the idea of middle range theories: theories that bring together empirical relationships forming a middle range of theoretical construction. These are not general systems theories, but he anticipated that they would point toward a general theory as research advances our understanding and as relationships between the midrange theories are better understood. He points to many examples in sociology and physics where midrange theories are prevalent, e.g., William Gilbert on magnetism, Robert Boyle's gas laws, Ludwig Boltzmann's gas distribution, Darwin's theory of the formation of coral reefs, theory of reference groups, theory of role sets.[7] An expansion of this list includes Newton's equations of motion, Maxwell's equations of electromagnetic fields, laws of thermodynamics, etc.

One can expect systems engineering, being an integration of various physical and social disciplines, to use a group of midrange theories. In the context of the entire system, the interrelationships between these disciplines may lead to more general observations, but at present, the midrange theories provide a starting point. The theory of systems engineering, then, consists of many parts derived from the supporting disciplines and system types. The theory of systems engineering is a midrange theory, integrating several different theories together to describe the discipline.

The discipline of systems engineering starts with a rich mathematical basis stemming from the mathematics of each of the system types. The laws of physics govern the individual functions or components of a physical system with all the associated mathematical constructs. Similarly, logical systems are primarily mathematical algorithms producing an output from a specific set of inputs. Social systems, due to their immense complexity and highly coupled individual variables, are still developing in their mathematical constructs, with statistics being a key mathematical tool.

The focus of systems engineering is on the interrelationships and integration of the system functions, not the individual functions themselves (e.g., the mechanical engineer is concerned with the pump as an individual unit, while the systems engineer is concerned with the function of the pump within the system, and the integration of the pump into the broader system context). Thus, systems engineering requires a broad understanding of the engineering and social disciplines and their mathematics, and then goes beyond them to focus on the integration of these disciplines. Mathematical tools of use to the systems engineer include statistics, information theory, matrix algebra, numerical analysis, control theory, Fourier analysis, and category theory. Each of these mathematical tools provides for the construction of integrated system models to support the holistic design and analysis of the system. Category theory provides the mathematical structure to integrate the other mathematical relationships into a complete system representation.

The processes used to organize systems engineering practice are important and vary widely with the specific system and its unique characteristics. The processes organize the engineering approach but they are not the engineering. The focus of the systems engineer is first on the system rather than the process. The process organizes the engineering; it is simply an organizational means to conduct the engineering to achieve the intended system. The social aspects of the system define the processes and often aid in the communication and coordination among groups. Having the process properly structured for the system development and/or operations is necessary for an elegant system. Both the social and physical/logical aspects of the system determine the need for specific aspects of the process.

These concepts form the foundation of systems engineering. A set of postulates, hypotheses, and principles captures these concepts as discussed in section 3. This foundation supports the definition of the systems engineering domain, system influences, mathematical basis, and general application methods.

1.3 Characteristics of an Elegant System

Engineering a system involves the development of a specific system configuration with its set of functions and interrelationships from a group of possible system configurations. Systems engineering intends to generate a system that best meets the needs of the intended system users. System elegance is a descriptive term often given to highly successful systems in this regard.

The idea that the proper goal of systems engineering is to produce an elegant design was first introduced in a speech by Robert Frosch.[8] He noted that he often got no response when he asked systems analysts, "Is it an elegant solution to a real problem?"They did not understand the question. Elegance is something you know when you see it, but is not something easily defined, particularly in the sense of a system. Webster defines elegance as a "dignified richness and grace."[9] This articulates an attitude of intent and a social response to the system. This definition identifies key system attributes. 'Dignified grace' conveys a notable ease of use or operation in a variety of applications. 'Dignified richness' conveys a notable robustness in application, a full achievement of the system intent, and a satisfaction of intent not fully specified. A term that provides further help with this definition is concinnity. Webster defines concinnity as 'a skillful arrangement of parts, harmony, and elegance'. This conveys the idea of a well-organized system with skillfully defined system interrelationships. System aesthetics are accounted for in the idea of richness, grace, and harmony. An efficiency in the system layout and construction is also seen in the 'skillful arrangement of parts, harmony' of the system. A well-structured system is an efficient system. Perhaps one can state a definition of system elegance as 'a system that is robust in application, fully meeting specified and adumbrated intent, is well structured, and is graceful in operation.'

Note that there should be a deep understanding of the system application in meeting intent without full specification in advance. This connotes the idea that, in meeting the intent of the system, there are aspects of the system capabilities where one can naturally extend or configure to meet application needs that are not well defined during system development. Working from options that meet the current system intent, one makes design choices that support natural extension or configuration of the system for future applications that may not be fully known. The evolution of Apple iPods to iPhones to iPads is an example of a system design that supported expanding capabilities not clearly seen at the beginning of the development. The idea of what they could do was there, but the specifics of how these would work in future applications were not fully clear.

This brings to the front the ideas posed in the paper, "How do We Fix Systems Engineering"?[10] This paper defined elegance as a set of system characteristics (sometimes referred to as attributes). These characteristics provide some guidance in the engineering of elegant systems and provide a measure of what a good system design embodies. A set of questions presents the following characteristics:

- Efficacy: How well does the system achieve the intended outcomes?

Efficacy provides a measure of how well the system achieves the intended outcomes. Understanding the context of the system application (or mission) and capturing the system concept as documented in the Concept of Operations establishes the outcome. As the system progresses through development, the systems engineer should check the progress against the intended outcomes. One should also check against intended outcomes during proposed modifications or upgrades.

- Efficiency: How economical is the design in terms of its performance and the resources required to build and operate it, with respect to competing alternatives?

Efficiency deals with the idea of a best solution given the various system configuration options. It embodies the idea that the intended output is obtained from the system inputs with a well-structured system that has no unnecessary capacities. This necessitates a comparison of design options within all constraint and performance boundaries. The ability to compare configuration options early is paramount to selecting the most efficient design option for the intended outcomes. The Concept of Operations captures the selected option. One should update the efficiency measures of the system with each design phase (sometimes executed as design analysis cycles) to ensure the design or operational changes maintain the system efficiency.

- Robustness: How well does the system perform in unanticipated circumstances and in collateral usage?

Robustness deals with the ability of the system to handle unexpected or uncertain events or variability of applications. This measures the system's ability to meet the intended mission objectives (or system uses to achieve goals) in the face of uncertainty or variability. This allows a measure of the system's usefulness in alternative applications (i.e., the intended outcomes of a new or different mission). Robustness in this context incorporates the scope of system resilience, system dependability, and system reliability. Intended and known potential mission variations should be captured by the definition of the Mission Context and documented in the Concept of Operations. Robustness addresses the utility of the system within the Mission Context.

- Unintended Consequences: What does the design do, or produce, that is unanticipated and unwanted?

Unintended consequences are those results of systems development and operation not anticipated by the system design or in system operation. These unintended results have a variety of forms including system failures, environmental impacts, social impacts, legal ramifications, and/or political ramifications. These bring into view the various system constraints such as those arising from budget, schedule, policy, and law, and account for both physical and social consequences in the system development and operations. The source of these unintended consequences is generally human and social. The system's physics do not fail or cease to exist. The system behaves unexpectedly because the system's designers and operators do not fully understand all the interactions that the system can have among its own components and with the environment. Thus, recognizing and managing the factors that lead to unintended consequences is a key role of the systems engineer.

Robert Frosch also indicated some of the qualities of a systems engineer needed to produce a practical, useful system. The Jet Propulsion Laboratory's (JPL's) Gentry Lee[11] who laid out a set of 10 characteristics of the systems engineer greatly expanded these qualities. Happenstance does not achieve elegance. Rather, a systems engineer knowledgeable of the systems integrating physics and mathematics governing the system and skilled in working with the people within the organization producing the system guides the design and organization to an elegant solution.

Remember that the intent is to design, produce, and operate an elegant system. Starting with an elegant concept and ending with a poor system is not elegant. System elegance is an intentional achievement that one must actively and visibly manage during the entire system development and operation life cycles.

1.4 Document Overview

This TP structure addresses different aspects of systems engineering theory. Section 1 provides a definition of elegant systems engineering and lays out its characteristics. Section 2 describes the systems engineering framework, defining the major focuses of systems engineering. Section 3 defines the set of systems engineering postulates, principles, strategies, and hypotheses that are foundational to the discipline. Section 4 provides the theoretical basis for system design and integration. This section presents the theory of the system integrating physics, system state variable approaches, system value modeling, multidisciplinary design optimization (MDO), system information flow, system autonomy, engineering statistics, system of systems, system representations, category theory application, system model integration, and lifecycle application. Section 5 presents the sociological basis for discipline integration, looking at the organizational structure and information flow during system development and operation. Section 6 provides a summary of the systems engineering processes and the importance of defining the system-specific processes in the SEMP. Section 7 provides a short summary of this TP. These sections provide a foundation for *Engineering Elegant Systems: The Practice of Systems Engineering,* the companion volume to this TP. The appendices at the end capture some detailed technical data and a glossary. Appendix A contains the derivation of the rocket equation from the exergy balance equation for a rocket. Appendix B is a summary of system complexity heuristics considered within the consortium, followed by the references.

2. SYSTEMS ENGINEERING FRAMEWORK

Systems engineering as a discipline is comprised of two main elements: system design and integration and discipline integration. In this framework, these two elements encompass four components: mission context, system integrating physics, organizational structure and information flow, and policy and law. Figure 2 illustrates this systems engineering framework.

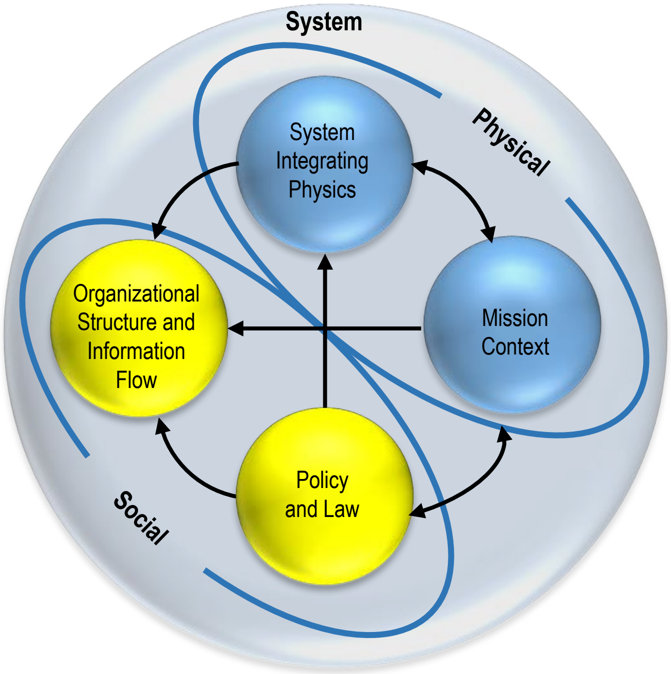

System
Physical
System Integrating Physics
Mission Context
Organizational Structure and Information Flow
Policy and Law
Social

Figure 2. Systems engineering framework relationships.

System design and 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 thus 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 couples 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. The physical/logical choices made for the system can emphasize or amplify the social aspects of the mission context. For example, when a planetary satellite is intended to explore 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 can be seen on the social response.

The social aspects are a major element 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 to which sociological approaches apply. Information flow is a key element in designing and operating an elegant system. Systems engineering, working with program management, 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 are the main organizational concern of systems engineers. Configuration management and data management provide support in this area. Program managers and line managers deal with the fiscal, political, and human capital concerns. The system design and operations represent the knowledge of the system residing in the organizational structure.

Policy and law are 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 origins.

3. SYSTEMS ENGINEERING POSTULATES, PRINCIPLES, STRATEGIES, AND HYPOTHESES

Considering the systems engineering approaches that are working and those that are missing, the NASA Systems Engineering Research Consortium began to consider the basis for the two focuses of systems engineering (System Design and Integration, and Discipline Integration). The concept of elegance drives consideration of how to achieve an elegant system. System interactions (subsystem interactions, environment interactions, manufacturing tooling interactions) are a key element in systems engineering. Social interaction forces that reside within the organization and in the policy and law environment must also be accounted. This led to the definition of basic concepts driving systems engineering.

The Systems Engineering Research Consortium identified a set of postulates, principles, strategies, and hypotheses to articulate the basic concepts that guide systems engineering. The 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 of systems engineering. The application of the postulates led to their expansion as a set of principles. A set of strategies have also been formulated from the papers on the Discipline of Systems Engineering. This definition work leads to the 7 postulates, 14 principles, 8 strategies, and 3 hypotheses stated in this section. The postulates and principles define the domain of systems engineering as well as the system aspects and system influences that are of concern to the systems engineer. The strategies were formulated as concepts to enable an improved form of model-based systems engineering and are linked to the postulates, principles, and hypotheses. The strategies address an approach to putting the postulates and principles into system modeling practice. The hypotheses contain implications stemming from the seeds of a holistic mathematical basis for systems engineering.

3.1 Systems Engineering Postulates

A postulate is something assumed without proof to be true, real, or necessary.[12] The postulates of systems engineering identify the basis for the discipline. These are further expanded by a set of principles in section 3.2.

- **Postulate 1:** Systems engineering is system specific and context dependent in application.

– 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, and 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 stated by the product goals. This includes the aspects of the system needed to operate in an elegant manner and thus considers the entire system lifecycle.

– Evidence: The ubiquitous tailoring of systems engineering approaches provides strong support for this postulate. Tailoring is the manifestation of making the processes fit the system context. 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[13] 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 and the characteristics of the engineering organization. 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, its organization, and 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 whole includes considering the human role in system operations and maintenance, taking into account human capabilities and limitations. This postulate is a mathematical definition of a system containing both the system objects (subsystems) and system interactions. This postulate addresses the system integration aspects of systems engineering. Postulate 3 addresses the discipline integration aspects below.

– Evidence: The individual engineering disciplines generally 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 subsystem functions. This is particularly true when human inputs contribute to the emergent properties, where humans can be both a source of resilience and adaptability as well as a source of error or degradation in system performance (the determination of which is often based on how well the system design incorporates consideration of human factors engineering principles and practices). 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 and human interactions, can drive the design as strongly as the subsystem functions themselves and, when coupled, can potentially create unexpected system responses. Human System Integration (HSI) focuses on the human interactions explicitly. The systems engineer must predict and manage all of these system responses. Note that subsystems can be treated as systems in a limited sense, taking into account the external dependencies on the other subsystems. Subsystems may not be independently functional outside of the system context they are designed. Providing a mathematical basis for the system, this postulate frames the basic entities needed to represent a system.

- **Postulate 3:** The function of systems engineering is to integrate engineering and science disciplines in an elegant manner.

– Description: The systems engineering discipline is its own engineering discipline, but it is not independent from other engineering, science, and social disciplines. Systems engineering seeks to integrate and incorporate the other engineering and social disciplines solutions and designs 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 engineered complex system is developed by multiple engineering (e.g., aerospace, chemical, civil, electrical, mechanical), science, and social disciplines with many social aspects influencing the integration. These engineering disciplines with social influences work in an integrated fashion, formally and informally, to produce these systems.

– Implications: The interaction of these disciplines is a focus of systems engineering. Systems engineers integrate information deriving from the various disciplines via a detailed understanding of their interactions. This requires a basic understanding of each discipline involved in the engineering of the system as well as an understanding of the organizational relationships. Note that for subsystems and assemblies, the system integration can be more engineering discipline based. Systems engineering recognizes and accounts for these disciple integration functions as part of discipline integration. The systems engineer must be cognizant of the organizational and sociological influences on the system development and operations. The systems engineer in conjunction with program management also guides the engineering of 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 research in Biased Information Sharing (sec. 5.2) also shows that system technical 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 and 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 which systems can actually achieve their intents. Similarly, corporate/company policy is influenced by the types of systems the corporation or company chooses to develop, and vice versa.

  – Evidence: Every project has these constraints. Infinite budgets or schedules 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. Physical and logical solutions through these constraints can be assessed by social choice theory. These choices are based on an understanding of the system's abilities to achieve the government and corporate/company executives' intents. This understanding drives the interactions with budget and schedule allocations and the policies put in place. Similarly, the available budget, available expected duration, existing policy and law interact with and can influence choices in the development of a system.

- **Postulate 6:** Systems engineering spans the entire system lifecycle.

  – Description: Systems engineering is not just a development phase activity but continues throughout system operation, decommissioning, and disposal. 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 lifecycle. Operations engineering is responsible for the operation of the system. Systems engineering is responsible for the various changes and upgrades to the system capabilities.

  – Evidence: The necessity of 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 redesigns, etc. In addition, during decommissioning and disposal, systems engineering is essential to deal with the proper decommissioning and dispositioning of the system and supporting infrastructure, ensuring conformance with policy and laws affecting the system disposal.

– Implications: As the system progresses through its lifecycle, 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 and 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 after operations during decommissioning and disposal.

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

– 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.

– 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 once operation begins. This deepening of understanding of the system and its application enables commercial product upgrades or new models.

– 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 upfront. 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 section 4.13.2.

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

– Description: Understanding of the system regresses if organizational changes occur (postulate 4) due to inactivity of an organizational element (loss of experience), retirement of key experienced individuals, or closure of suppliers.

– Evidence: Regression of system understanding can be seen in some lifecycle extension activities. When system understanding is not actively maintained, the basis of system specifications become unclear and some systems have been found not to perform (either underperform or overperform) to their system specifications. This loss of understanding can impair long-term operations as operator errors can increase. In addition, operational procedures can lose their basis and it can become difficult to determine when the system should be retired or maintained as the system ages.

– Implications: If the system basis is not maintained, then the understanding of why certain procedures or specifications were 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.

3.2 Systems Engineering Principles

Systems engineering postulates form the basis of the principles of systems engineering. Principles are accepted truth which apply throughout the discipline. These truths build on the systems engineering postulates and 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.

– Description: This is the application of postulate 5. Systems engineering solutions must address the stakeholder's needs and their constraints. Budget and schedule constrains the development and integration of the system, the operation and maintenance 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. A focus of systems engineering is to keep the system cost within the budget or recommend when the solution space defined by budget and schedule does not meet the intended system application. In addition, other expectations and constraints such as environmental impacts, economic impacts, or social impacts may also affect the system solution options. The systems engineer must account for each of these to ensure a system is developed and operated to satisfy the stakeholder's needs and constraints as captured by the mission context.

– Evidence: Solutions defined in response to stakeholder needs drive system cost, schedule, and other expectations and constraints. System budget and schedule problems result from a lack of understanding of the best balance of the system within the resource allocations provided and the technical needs of the stakeholders. Unexpected consequences can be realized by systems where environmental impacts, economic impacts, social impacts, etc. are not recognized or understood.

– Implications: System solutions account for not only the technical performance (including human factors) but also must fit the allocated budget, schedule for development and operation, and other expectations and constraints (e.g., environmental impact, social impact). The systems engineer must understand the cost, schedule, and other impacts as well as they understand the technical performance of the system. The systems engineer develops this understanding from the initial concept definition and maintains it through the system lifecycle.

- **Principle 2:** Complex systems build complex systems.

– Description: 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 systems engineering. The systems engineer is responsible for both integration of the system functions and the integration of the disciplines developing these functions. The social interaction within organizations working on complex systems is itself complex and is a strong driver in budget and schedule efficiency or inefficiency. Configuration Management (CM) and Data Management (DM) are key systems engineering capabilities providing for effective management of the information about the system from the different disciplines that flow through the complex organizational structure. Postulates 2 and 3 also capture this duality when the systems engineer is responsible for both integration of the systems discipline functions and interactions defined in postulate 2 and the development organization disciplines defined in postulate 3.

– Evidence: Major system failures have occurred due to the lack of information flow through the organization. Organizational structures, particularly for large system developments, are highly socially diverse with diversity in people, the engineering disciplines, and the organizational culture. Projects with more than one company involved see this organizational complexity increase tremendously. It is difficult in some organizational structures to understand how to share the information and what information to share.

– Implications: Complexity resides not only in the system but also in the organization(s) developing and operating complex systems. Thus, systems engineers must deal with both the complexity of the system and the complexity of the development and operation organization(s). Understanding the system integrating perspective (defined in sec. 4.2) provides an engineering basis to understand what information should be shared. This guides the management of information flow. CM and DM provide tools and approaches that aid the systems engineer in managing the complex information flow through the organizational structures.

- **Principle 3:** A focus of systems engineering during the development phase is a progressively deeper understanding of the interactions, sensitivities, and behaviors of the system, stakeholder needs, and its operational environment.

– Description: This principle is the application of postulate 7. What you do upfront does not fully define 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 often do not allow sufficiently deep understanding of system interactions (which we are addressing with tools discussed in secs. 4 and 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 seeks the best balance of performance, cost, schedule, and risk.

– Evidence: In practice, this deepening of understanding is in any system development or operation. 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 also occur in some lifecycle extension activities. When system understanding is not maintained, the basis of system specification becomes unclear and some systems have been found not to perform (either underperform or overperform) 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: Systems engineers derive requirements as the system design progresses. Thus, while systems engineers define the mission requirements (i.e., part of understanding the mission context) at the beginning of development, the system requirements are defined progressively. They are a function of the design choices made and 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 they obtain testing and operational experience. System models gain fidelity as the design progresses and the systems engineer must manage the interaction between subsystem design maturity and system model maturity. These system models become the basis of system operations, as discussed in section 4.13.2. 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.

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

– Subprinciple 3(a): Mission context is defined based on the understanding of the stakeholder needs and constraints.

The understanding and definition of the mission context (i.e., the system application) is essential to a well-developed and operated system. An understanding of the stakeholder needs and constraints on the system defines the mission context. This requires an understanding of the stakeholders' relationship to the system in operation. Different stakeholders have different perspectives on what is important (developer versus operator versus maintainer versus general community). For example, the manufacturer (and developer), the driver, mechanic, and general public are all stakeholders for an automobile. The perspectives that each of these provide is different and can be either enforcing or conflicting. The manufacturer is concerned with production costs and appeal to customers. The driver is concerned with the general appearance, amenities, and ease of operation. The mechanic is concerned with accessibility to the vehicle's engine and components. The general public is concerned with safety and environmental impacts. The definition of the system application must bring together all of these perspectives.

– Subprinciple 3(b): 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 leads to poorly stated requirements, requirement gaps, and inaccurate system models and representations. An objective of system engineering is to understand the system (principle 4(a)) which then leads to the proper specification of requirements and proper representation of the system in the system models.

– Subprinciple 3(c): Requirements are specific, agreed-to preferences by the developing organization.

Preferences are held by individuals. 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.

– Subprinciple 3(d): Requirements and design are progressively elaborated 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 Preliminary Design Review (PDR), and component requirements may not be fully defined until Critical Design Review (CDR).

– Subprinciple 3(e): 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.

– Subprinciple 3(f): A Product Breakdown Structure (PBS) provides a structure to integrate cost and schedule with system functions.

The PBS integrates cost and schedule with the system functions and components. Cost and schedule are defining constraints (postulate 5) on the system and must be clearly integrated to the system functions and operations. The PBS provides the integration of the system functions (defined by the Goal-Function Tree in sec. 4.6.1), the system component cost from the cost modeling tool, and the system schedule from the scheduling tool. Configuration management is important to ensure the PBS reflects the baseline system functions, cost, and schedule. A separate version can be used for trading options. 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 the driving cost components seen through the PBS.

– Subprinciple 3(g): As the system progresses through development, a deeper understanding of the organizational relationships needed to develop the system are gained.

As the organization works through the development activities, new relationships may be defined and the magnitude of these relationships may change as the design matures. Organizational groups that do not have information to share in early development may be critical in sharing information late in the development. Similarly, organizational groups that may be critical at the concept development phase may complete the transfer of information, becoming less critical to information flow as the development matures.

– Subprinciple 3(h): Systems engineering achieves an understanding of the system's value to the system stakeholders.

System success is contingent on stakeholders' expectations, not on the system requirements, models, and design information. System success melds the system as designed and as built with the system as expected by the stakeholders. Often, systems engineers assume 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 stakeholder expectations.

– Subprinciple 3(i): Systems engineering seeks a best balance of functions and interactions within the system budget, schedule, technical, and other expectations and constraints.

In accounting for all of the system needs and constraints defined in the system's mission context (i.e., system's application), the systems engineer seeks to obtain a best balance of all of the stakeholders' differing preferences. This best balance provides a system that most fully meets the system context (i.e., resource allocations, political, economic, social, technological, environmental, and legal factors, and the differing stakeholders' preferences). This balance requires a thorough understanding of the system and its mission context in order to achieve a best balance within the full system context.

- **Principle 4:** Systems engineering has a critical role through the entire system lifecycle.

– Description: This is the application of postulate 6. 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 lifecycle. 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 is well understood during the development phases. During the operational phases, systems engineering is still essential as the system goes through maintenance upgrades, new application adaptations, obsolescence driven redesigns, etc. In addition, during decommissioning and disposal, systems engineering is essential to deal with the proper decoupling of the system and its infrastructure, and ensuring conformance with policy and laws affecting the system disposal.

– Implications: As the system progresses through its lifecycle, 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 systems engineering must deal with these organizational changes. Another organizational change and culture shift occurs during decommissioning and disposal.

A set of subprinciples defines the specific aspects of systems engineering throughout all of the system lifecycle phases:

– Subprinciple 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 possessed by the systems engineer underpins everything they do in terms of engineering the system. This includes understanding of system function and interactions defined in postulate 2 in the system context defined in postulate 1.

– Subprinciple 4(b): Systems engineering defines the mission context (system application).

The systems engineer integrates all of the different stakeholder preferences and resource allocations (budge and schedule) to produce a well-founded understanding of the mission context (i.e., system application). The mission context evolves from this integration and understanding activity and is the essential starting point for system development and operations activities.

– Subprinciple 4(c): Systems engineering models the system.

Systems engineering develops and maintains system-level models to aid in the design and analysis of the system. System modeling provides a means to understand the system including its functions and interactions. Section 4 describes specific system-level modeling approaches.

– Subprinciple 4(d): 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 based on the system-integrating physics/logic.

– Subprinciple 4(e): Systems engineering tests the system.

System engineering is a critical contributor to 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 test objectives. This can involve separate system tests, modification of discipline tests for system-level objectives, or system-level integrated analysis of data from separate discipline tests to obtain a system-level understanding.

– Subprinciple 4(f): 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 subphase is the purview of the manufacturing engineer, the systems engineer must stay involved to understand changes, update models, and perform analysis to ensure manufacturing change impacts are identified and understood at the system level.

– Subprinciple 4(g): Systems engineering has an essential role during operations, maintenance, and decommissioning.

Systems engineering has a key role in system operations which feature a host of 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 deintegrate the system.

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

– Description: There are many types of systems simply categorized as physical systems, logical systems, social systems, or some combination. Since there is not a unified theory of physics, a unified theory of logic, nor a unified theory of sociology, then there is not a unified theory of systems engineering. Instead, systems engineering derives from a set of middle range theories which form the basis of the system and the engineering of the system. As discussed in section 1.2, systems theory exists in various forms (e.g., general systems theory, system dynamics) and seeks to define the unique system aspects of the system. System theory does not replace the physical, logical, or social basis of a system but seeks to look at the interactions among the different aspects of the system. All of these system theoretical bases have a mathematical underpinning. Category theory provides a mathematical structure that integrates the system physical, logical, and social aspects. This system provides the mathematical framework of the system. Systems engineering then has three theoretical bases represented in the subprinciples below. These categories are system specific physics/logic systems engineering theoretical basis, mathematical basis, and sociological theoretical basis.

– Evidence: Systems exist as either physical systems, logical systems, social systems, or some combination of these. These systems incorporate all of the sciences that define their physical, logical, and social nature. Category theory provides the mathematical definition of a system. Category theory provides the mathematical structure to identify the system theoretical aspects from the physical, logical, and social functions and interrelationships of the system. Sociological principles define organizational information flow paths, gaps, and barriers. These principles also provide the basis for understanding system interactions with social systems as part of the system context.

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 the 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. This implies the use of concepts from control theory to control 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 that supports systems understanding and accounts for uncertainties and sensitivities as indicated by postulate 2.

Below are eight theoretical bases for systems engineering. These modeling bases provide a structure to model the various aspects of the system, bringing in the theoretical bases defined in this principle.

(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 and 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 that this is a feasible approach.

(3) Model Basis: System information is represented and maintained in models, and exported to documents when needed. 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. Section 4.6 addresses this explicitly.

(5) Goal Basis: Systems exist to achieve goals, which are represented as constraints on the output state variables of functions. Section 4.6 also 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 improves 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 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.

– Implications: This middle range set of theories provides a complete basis for the systems engineer to understand a system. The specific application will be specific to each system (i.e., the theories needed for a cyber system are very different from those needed to build a ship). This structure provides for these differences and allows the systems engineer to incorporate the theories needed to understand both the system and the organization developing or operating the system. The systems engineer does not need expertise to design each component of the system. The system engineer is the expert in how to integrate these components into the intended system. This requires a broad understanding of several disciplines rather than a deep understanding in only one. The systems engineer must communicate clearly among the engineering disciplines, including understanding terminology differences and the use of similar terms to mean something different to a particular discipline (e.g., to an optical engineer ω is the angular frequency of light while to the mechanical engineer working on the same system it means the angular rotational velocity of a component). Systems engineers should translate terminology and not try to enforce commonality among the engineering disciplines.

– Subprinciple 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 recognize that these differ for different system types (postulate 1).

– Subprinciple 5(b): Systems engineering has a mathematical basis.

Mathematical category theory provides a mathematical structure for systems engineering. A mathematical category provides a definition of a system that provides a structure to incorporate various physical, logical, and mathematical theories into a system representation. Category theory integrates several theories that are important to systems engineering. Systems engineers, in engineering the system, manage information about the system and its interactions, 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. Note that information theory has a set theory basis and naturally extends to the construction of a mathematical category. Systems engineers use information to control the system design or system operations that bring in control theory in a broad scope of controlling the information flow about the system and in defining the control methods to control system states within relevant acceptable ranges over time. Category Theory provides for the interaction structure to show these control relationships for the system. Statistical engineering is also a significant mathematical tool that allows for systems understanding and accounts for uncertainties and sensitivities. Category theory allows for the absence of details within an element and allows for variations of relationships that support the application of statistics in defining system relationships. Category theory provides the mathematical structure to integrate these various theoretical basis into a complete, coherent system representation. This theory is described more fully in section 4.11.

–Subprinciple 5(c): Systems engineering has a sociological basis specific to the organization(s).

Systems engineering incorporates the fundamental sociological concepts specific to the development and operations organization. This is a result of postulates 3, 4, and 5.

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

– Description: The correspondence of the organization to the system (whether the organizational structure mirrors the system structure or not) 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. Successful development of a system by organizations inexperienced in that specific system will require structure that is more formal. 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 model this principle as discussed in section 5.6.1.

– Evidence: The engineering disciplines each create their building blocks of the system in coordination with other engineering disciplines. For example, system dynamics drive the structural loads.

System efficiency increases at the expense of subsystem efficiency. Integrated performance of the system drives the best balance of system performance. Independent subsystem optimization leads to poorer system performance and system efficiency goes down. Appropriate information interchange among the engineering disciplines aids in the recognition of impacts to the overall system balance.

– Implications: Systems engineers are responsible for understanding how the organizational structure and culture affect the flow of information about the system. The systems engineer ensures proper interaction between the engineering disciplines as they produce their aspect of the system. Similarly, in operations, the disciplines must work together to ensure consistent and intended system operation and maintenance. Creating a map of this information flow aides in understanding how this flow occurs within the organization. Where difficulties are identified, the systems engineer should discuss potential changes for improvement with project management and organizational (i.e., line) management. Adjusting systems engineering process flows may handle some difficult situations. Some may require organizational changes by the project manager or line management. These changes may solve one issue and make another information flow path more difficult in complex organizations. The systems engineer should evaluate each change and strive for the best balance of systems engineering process application with project and line organization structures.

- **Principle 7:** Decision quality depends on the system knowledge present in the decisionmaking process.

– Description: 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 need. Decisions are process dependent. Information needed by the decision makers directly drives the decision methods.

– Evidence: Decisions made without a full understanding of the impacts on all phases of the system are known to be flawed in practice. These decisions lead to impacts to subsystems, enabling systems, and interoperating systems when the knowledge of these systems is not present among the decision makers.

– Implications: Good decision quality requires the right knowledge be present in the decision-making process. This drives the membership of boards in the decision-making process, membership on trade study teams, Integrated Product Team structures, and the approach for external coordination. Systems engineers should avoid decision-making processes where the system knowledge needed for the system decision is fragmented. Fragmented decision bodies lead to system decisions that do not properly balance all aspects of the system and the impacts to the enabling systems and interoperating systems.

- **Principle 8:** Both policy and law must be properly understood to not overly constrain or underconstrain the system implementation.

– Description: 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.

– Evidence: 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. There are many examples where policy and law have influenced engineering solutions substantially including the nuclear power industry, field-testing of recombinant DNA in agriculture, and use of growth hormones in dairy cows[14] as well as impacts of the Clean Water Act on Boston Harbor.[15] Policy engineering (understanding both policy and the engineering solutions driven by policy) has been cited as an important industrial effort to ensure policy and law do not overconstrain or underconstrain industry implementation.[16] Proper engineering understanding of policy and law as well as properly written policy and law reflecting engineering solutions in industry are essential for any industries' health. Universities have also developed curriculum to teach engineering students how to properly understand and interpret policy and law.[17]

– Implications: Overconstraining the system due to misunderstanding of policy and law can lead to ineffective and, therefore, inelegant solutions. The system may not have all necessary functions and, in some cases, good solutions may not be seen as viable. Underconstraining the system can lead to exceedances in budget and/or schedule. Underconstrained systems are a source of unintended consequences, particularly with regard to environment or social impacts. It is essential that the systems engineer understand how the policy and law apply to the system appropriately and how the appropriately applied policy and law then constrain the system solutions.

- **Principle 9:** Systems engineering decisions are made under uncertainty, accounting for risk.

– Description: This principle derives from postulates 2, 3, 4, and 7. Systems engineers progressively understand information about the system 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. Uncertainty exists due to the inability to predict the future with certainty and decision which require an understanding of a future system state naturally have a risk in the state not actually being realized. Uncertainty arises from many aspects of systems engineering, including limited knowledge on system environments and social aspects of the organization that affects information maintenance, creation, and flow. Systems engineering must also understand sensitivities to ensure the proper focus on the different uncertainties. Systems engineering models the uncertainty and sensitivities throughout the process. Risk in decision making comes from the need for a sufficient understanding of the system context and the knowledge that uncertainty does exist even as understanding improves.

– Evidence: Systems engineering risk processes exist to address this reality. The inability to predict future decisions and their impacts leads to risk in the decisions about the system. Selected system solutions have assumptions on what factors may or may not manifest themselves. In addition, the unknown factors can drive risk unexpectedly. Systems engineers will recognize many of these factors as the system proceeds through development and operations, but they may not recognize them at the time needed for the decision.

– Implications: Systems engineers are responsible for understanding the system and the system solution implications. Systems engineering must properly identify and track risk factor through the development. Systems engineers may realize new risks at any point in the development or operations lifecycle phases. As the system decisions are made, the risks associated with the decision become apparent.

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

– Description: 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 understanding represented in requirements, models, and designs. This leads to the principle that the proper performance of 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 requirements verification can provide system function verification. If the requirements are not truly level and balanced, then the focus of system verification should be on the system functions directly. By focusing on the proper system functions, a verification approach can be defined for the system that focuses on its successful application.

– Evidence: Testing at assembly and subsystem levels focus on the functions that these parts of the system provide. They are defined based on the design (rather than the requirements) of the assembly or subsystem that embodies the functions that it is to provide. The tests often focus on the outputs of the unit under test for a given set of inputs (e.g., a transfer function test). This testing approach is embodied in discipline engineering and extends directly to the system level. The system level verification includes the holistic set of system functions, their interactions, and the interactions with the environment as stated in postulate 2.

– Implications: System engineers should focus on the system functions and their interactions during system verification. Requirements can be verified through their relationship to the system functions. Focusing on verification of requirement directly can lead to duplication of verification activities (i.e., analysis, inspection, and test) and can miss aspects of the system design defined lower in the detailed design. An efficacious and efficient system verification focuses on the set of verification activities that indicate the system functions, their interactions, and the interactions with the environment are as intended and expected by the design team.

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

– Description: System validation is based on stakeholder expectations, not on the system requirements, models, and design information. It allows the comparison of the system as designed and then the system 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 difficulties with the convergence 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 stakeholder expectations. Section 4.8 discusses this more.

– Evidence: System value models (sec. 4.8.2) are based on a mathematical representation of stakeholder preferences. This mathematical representation provides a basis function to compare the system design attributes directly with the system intentions.

– Implications: By focusing on the value the system provides to the stakeholders, the systems engineer has a clear approach to perform validation separate from verification. System validation now has an important and mathematically distinct complement to system verification.

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

– Description: This principle deals with constraints imposed on engineering decision and system configuration options based on when the system is needed for operations. 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 1933 (mail and passenger transport)[18] was very different from the context found in 1965 (transatlantic jet transport).[19] With the pace of technological advancements, the available solution sets for a given system can change noticeably over as a little as 5 to 10 years, such as seen in the electronics industry over the last five decades. Thus, the decision timeframe is an important aspect of the solution set available to the systems engineer.

– Evidence: A model rocket can be designed and built in a few weeks. A large rocket carrying cargo and/or crew to orbit currently takes 10 years. Similarly, a model airplane can be designed and flown in a few weeks. A modern jet fighter may take 10 years or longer. If you have a year to deliver a system, that is going to limit you to existing (or possibly some emerging) technology. If you have 20 years, you can spend a good amount of time developing new technologies to improve the system performance, manufacturability, etc.

– Implications: Systems engineers need to understand the timeframe they have to deliver a completed system and the implications that has to the system solution set. The timeframe for the system need (whether that is market driven, national policy driven, or natural event driven) is an early filter on solution space. The solutions that are included or excluded must be understood well enough to be able to determine their fit with the system need timeframe. The first step is to gain the understanding necessary to define what solutions fit within this timeframe.

- **Principle 13:** Stakeholder expectations change with advancement in technology and understanding of system application.

– Description: 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 lifecycle 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 lifetime increases. And with more variation in stakeholders and stakeholder preferences, changes can be introduced to the system that can impact the system's ability to adapt to these preferences or stretch out long-duration system developments. System robustness in application can provide pathways for changing system uses. The DC-3 aircraft[20] is a good example of an aircraft that has proven highly adaptable to uses and highly reliable in operation. The changing missions of the B-52 Stratofortress[21] over the lifecycle is another good example of an aircraft that has been robust in application. A key to managing these socially-driven changes is to recognize how the system can be evolved or migrated to new applications and when these shifts indicate the need for a different system, indicating the time for the current system to move into decommissioning.

– Evidence: This is a normal occurrence in the practice of systems engineering. The systems engineering processes deal with the change in stakeholder expectations. These changes are a major source of change in mission context and system requirements.

– Implications: This leads to instability in expectations for the system and in the system requirements. The systems engineers must be aware of these changes and account for them as early as possible. Early identification can provide for lower impacts to system development cost and schedule and to system operational change timeframes. Systems where stakeholder expectations have the potential to change should employ more flexible system engineering process application (e.g., agile systems engineering) to accommodate the changes as the system moves through the lifecycle.

- **Principle 14:** The real physical system is the only perfect representation of the system**.**

– Description: This principle provides a statement of the idea that has long been espoused among statistical modelers. The physical system is the only perfect (complete, full) model of the system. Or stating more simply, the perfect model of the system is the system itself.

– Proof: Kullback-Liebler information provides a definition for 'ideal' information.[22] This information measure indicates how close a particular model matches the real physical system and is defined as:

$$I(f,g) = \int f(x)\log\big(f(x)\big)dx - \int f(x)\log\big(g(x|\theta)\big)dx \ , \tag{1}$$

where

I = information distance between the physical system and the model
f = physical system
g = model of the system
x = physical variables
θ = model parameters.

Note that θ is typically an estimation of the actual system physical variables. Setting this relationship to zero provides a relationship to define the differences in a given model to the real system, and provides proof that the perfect model of the system is the system itself:

$$\int f(x)\log\big(f(x)\big)dx - \int f(x)\log\big(g(x|\theta)\big)dx = 0 \ , \tag{2}$$

and

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

Also note that copies of systems are not physically identical:

$$f_1(x) \neq f_2(x) \neq \cdots \neq f_n(x) . \tag{4}$$

Copies of the system are similar, not identical. This is evidenced where systems do not have identical behavior. Hence, a car can have a manufacturing failure while most will not exhibit this failure. Thus, the physical system only represents itself identically and no 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 that must be recognized. There are various system models that can show various aspects of the system, but no system model can perfectly 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.

3.3 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 modeling that flow out from the mathematical basis defined in subprinciple 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 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 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 subprinciple 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. Section 4.8 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 a starting point for the development of game theory[23] 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, and this measure of optimality needs to use a single scale of value.

(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. It is not always possible to do this, and in such cases, other nonscalar goals and measures can be used. When this occurs, the systems engineer must be aware that this will make the process of coming to agreement on goals, preferences, and requirements more difficult and subject to error.

(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 subprinciple 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 (e.g., state variable models, integrating physics models, value models, statistical models, information models, process 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 lifecycle. 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 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 (principle 14). System models, collectively intended to cover 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 (operational description model) and the Goal-Function Tree (GFT).

- 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 represent abstract component connectivity.

- Failure models (i.e., Fault Trees, Probablistic Risk Assessments (PRA), Failure Models and Effects Assessments (FMEAs)) 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 (i.e., Fault Trees) that are the logical complements of the GFT.

- Performance models come in a variety of types. The main types described here are nonsimulation performance models, such as root-locus analyses in linear control theory or Fourier techniques used in radio frequency system analysis. Any nonbehavioral 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 (State Analysis Models (SAM)), but can also include time-domain simulations that range from purely software simulations with no 'real' hardware or software (software simulations), to simulations that include mixes of the system's actual hardware, software, and humans (Hardware Software Integrated (HSI) simulations), 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 (ABMs)), 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 provides 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 postulates 2 and 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. State Analysis Models provide the representation.

(1) Systems engineering makes use of system state variables in system representations.

System state variables are essential to represent 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 section 4.6.

(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. System development and operation must be tracked to these goals to ensure that the design and operations are meeting the stakeholders' intents. Goal-Function Trees provide this modeling representation. Modeling system goals is discussed in section 4.6.

(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 subgoals). 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 defines 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. Five aspects of the application of control theory in systems engineering is described below:

(1) Systems engineering provides design and performance representations of the system.

This is related to system modeling as discussed under strategy 3 and subprinciple 4(b).

(2) Systems engineering simultaneously, with nominal system design, also addresses 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 the 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 deploys 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 structural margins, or active control through open- or closed-loop control systems.

(4) Systems engineering uses and extends 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 subprinciple 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.

(5) Systems engineering uses control theory to understand decision-making process flows.

Control theory and information theory are used to model information flow in a decision-making process in section 5.3.1.

- **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 uses 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 accepts the variability of human interpretation of acceptable and expected system behaviors.

Individuals can and do differ in their interpretation of system behaviors. For example, prior to the Challenger accident, some engineers at Thiokol were worried that erosion of O-rings discovered after some flights indicated a serious design problem. In other words, they viewed this as a failure. Others believed that the fact that no major solid rocket motor failure had yet occurred despite the erosion indicated that the erosion was a minor problem. While the Challenger accident showed that the 'failure' interpretation to be correct, this was not known until a disaster occurred.[24] Instead of viewing this situation and many others like it as anomalous with some being right and others wrong, these differences are typical of engineering and must be treated as such, both in practice and in theory. Differences of opinion and judgment occur all the time in engineering, and the theory and practice of systems engineering must be designed to account for these differences and use them to an advantage in system success.

(3) Systems engineering models the interaction of the system and the organization to identify information gaps, barriers, and reservoirs.

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. 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. Section 5.6.1 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 methods use probabilistic techniques to address uncertainties of prediction (e.g., PRA), 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 include assessments of uncertainty.

All predictions are uncertain, and hence require estimates of these uncertainties.

3.4 Systems Engineering Hypotheses

The hypotheses are statements that the consortium members are debating and believe can be proven (or perhaps disproven) through research. A hypothesis is a statement of an unproved theory, proposition, or supposition tentatively accepted to explain certain facts to provide a basis for further investigation. 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 include the time frame for the system need as discussed by principle 12 above. This creates a time dependency in the hypotheses.

- **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[25] directly proves this for a thermodynamic system through the relation:

$$\int_{t_1}^{t_2} (\delta T - \delta V + \delta W)\, dt = 0 \ , \tag{5}$$

where

δT = kinetic energy differential change in the system
δV = potential energy differential change in the system
δW = work differential change in the system
t = time.

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.[26] 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 intended goals.

– Description: In each operational context and decision timeframe, the minimum system complexity required to fulfill all the system outputs (accomplish all the system intended goals) 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.[27] 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: A system results from a large set of decisions made by decision makers throughout an organization. To analyze a decision, three key elements are necessary: preference, beliefs, and alternatives. Hence, for a systems engineer to understand how an organization arrives at a particular system, an understanding of the set of decisions, each with their elements, is necessary. Each decision maker may have different preferences, beliefs, and alternatives. While each of these elements are challenging to understand, preferences are of particular interest to systems engineering as they relate to desired system goals. If different preferences are being used to make decisions on a system, then those decisions would be inconsistent with each other, meaning it is possible that given the same beliefs and alternatives, decision makers may decide on different solutions. To enable consistent decision making throughout the organization, systems engineers must elicit, represent, and communicate preferences of key stakeholders to drive to outcomes that the key stakeholder prefers. A mathematical representation supports the modeling of the preferences and enables analysis of the differences and commonalities in the preferences of different stakeholders.

– Evidence: Many systems engineering approaches use a representation of preference to guide decision making. Goals in GFTs, objective functions in multidisciplinary design optimization, payoffs in game theory, and utility functions in value-based engineering are just a few examples of mathematical representations of preferences used in systems engineering approaches. The premise of these approaches is that preferences are mathematically representable and enable a rank ordering of alternatives. Based on these examples, system engineers can create a mathematical function that rank orders alternatives in the same way that a preference does. Decision theory also uses mathematical functions to rank order alternatives as an individual with their preference would, and is widely advocated as a rigorous approach to design and systems engineering.

– Implications: The accurate representation of stakeholder preferences enables the systems engineer to assess how well the system fulfills these preferences as the system progresses through its lifecycle. While many systems engineering approaches assume a mathematical representation of preference exists, accurately representing preferences mathematically is still a significant challenge. The elicitation and formation of mathematical representations must become a significant task undertaken by systems engineers to adopt these approaches. Beyond enablement of approaches that strive to find the best system, mathematical representation of preferences also enables meaningful validation of the system. Mathematical representations of preferences allow comparison of the system characteristics with the stakeholder's preferences, answering the validation question: 'does the system meet the stakeholder's intent.'

4. SYSTEM DESIGN AND INTEGRATION

Systems design and integration is one of the two main aspects of systems engineering. Understanding the physics and mathematics of the system (postulate 2) provides the ability to design and integrate a system, its constituent functions and subsystems with their interactions, and the system's interactions with the environment. The system engineering strategies defined in section 3.3 provide a set of approaches from which to conduct this engineering at the system level. This set of strategies provides an organization of the various systems engineering approaches. Strategies 1 through 6 and 8 address the system design and integration approaches. Section 5 addresses strategy 7.

- **Strategy 1:** System Theory Strategies.

This strategy includes representation of the whole system, the environment in which it operates, and the context for design, manufacturing, and operation. A system has an integrating physics that ties directly to the system goals and integrates the system functions in accomplishing these goals within the operational environment. Specific physics and logic govern the various engineering disciplines and are important for a sufficiently complete understanding of the system. These discipline physics do not generally address the cross-discipline interactions of the system realized in accomplishing the system goals. The system-integrating physics provides a mathematical basis to integrate these differing discipline physics and provide an integrated understanding of the system as discussed in section 4.2.

System design focuses on determining a 'best balance' among the system functions and subsystems. This best balance requires an optimization of the system which may lead to a deoptimization of the subsystems. MDO is an important tool to achieve a system design that provides this best-balanced system as discussed in section 4.7. MDO provides an integrated model of the system to design and analyze system uncertainties and identify system sensitivities.

- **Strategy 2:** Value Theory Strategy.

The system value model provides a mathematical representation of stakeholder preferences. This provides for a comparison of how well the system attributes are satisfying the stakeholders' expectations. A capability model of the system provides an ability to assess how well the system supports various applications (i.e., design reference missions in some industries) as described in section 4.8.4. This provides an indication of the robustness of the system to meet various uses as discussed in section 4.8.3. System goals are best defined with respect to the value of those goals as defined in the value model.

- **Strategy 3:** Model Strategies.

Understanding of the system is inherently generated and represented through models, whether formal or informal. The aim of this strategy is to move towards formal modeling, but only with models that are sufficiently valuable to system design, manufacturing, and operation. Each of the subsections below explicitly define system modeling. Section 4.13 describes the integrated system modeling strategy including the relationships between the models defined in this section and the application throughout the system lifecycle.

- **Strategy 4:** State Strategies and **Strategy 5:** Goal Strategies.

Part of system state variable modeling addresses these two strategies. System state variables are the system parameters that characterize the specific system attributes and are the mathematical entities upon which analyses are performed (principle 5(b)). Different physical systems have different sets of state variables. The GFT provides a construct to integrate the system goals with the system state variables as described insection 4.6.1. This provides an assessment tool and standard to estimate how well the system is achieving the system goals. State variables appear in most disciplinary models, and the aim for improving systems engineering is to extend this practice to system models. Examples of this are system state transition models and time-domain simulations that support analysis of system operation including the response of the system to changing system environments and use conditions. The SAM is a state transition model as discussed in section 4.4.3.

- **Strategy 6:** Control Strategies.

The subsections below cover different aspects of control strategies. System-integrating physics addresses the performance-based system representations. The application of the GFT and SAM can be used to address failures, and to help create failure representations such as fault trees or failure-space directed graphs (a form of mathematical categories discussed in sec. 4.11). The GFT defines the constraints on state variables that define goals, to which the system design must control.

These approaches to system design and analysis provide the basis for systems engineering to engineer the system. System elegance is achieved in engineering a best-balanced system considering the stakeholders' preferences including system performance.

- **Strategy 7:** Knowledge Strategies

These strategies are addressed in section 5.

- **Strategy 8:** Predictive Strategies.

Engineering statistics are essential tools in identifying, quantifying, and understanding system, environment, and context uncertainties and sensitivities. There are a variety of statistical methods that can be employed in understanding a system, the environment, and context including: Information Theoretic, Bayesian, and Frequentist. These are discussed in section 4.5.

4.1 System Application Definition

The definition of the system, its architecture, and configuration starts with an understanding of the system application, sometimes referred to as the mission context. The preferences of the system stakeholders which include the system investors, users, developers, and operators define these applications.

System models provide representations that allow the systems engineer to design and analyze the system. These models incorporate the system goals in fulfilling the applications intended by the system stakeholders. Thus, the preferences of stakeholders are a focus of early system definition work and are key inputs to the system models. These preferences provide the basis for the system goals. Section 4.8.2 represents the system value model. The system value model helps to reconcile conflicting preferences between stakeholders and provides a mathematical representation of the stakeholders' preference which provides a consistent set of system goals.

The preferences of the system value model are the basis for candidate system configurations and concepts of operations. The concept of operations provides a description of the system concept incorporating the system preferences, goals, functions, and applications in a single coordinated narrative. This forms the basis for the system design.

Requirements, which are formal agreements forming the basis for the next level of design (subprinciple 3(c)), formally specify some of the goals defined in the system value model and described in the system concept of operations. As the design evolves and the organization makes design decisions, systems engineers can specify the next level of goals and requirements (subprinciple 3(d)).

4.2 System-Integrating Physics

An important question asked early in development is 'Which is the most efficient system configuration?' The system integrating physics provides the approach to answer this question. As stated in postulate 1, there is a system-integrating physics correlated with the functions specific to the system. This takes the form of an integrating engineering relationship (e.g., thermodynamics, structural mechanics, optical physics, logic, or sociology). This integrating relationship defines the integrating perspective (view of how the system functions are integrated across the different discipline relationships) for the system. Most systems will have this integrating (primary) discipline that ties together the key engineering functions of the system needed to achieve the primary system goals. This primary relationship provides the integration function (or functor as discussed in sec. 4.11.4.4.3) for the other discipline's equations that affect this aspect of the system. Thermodynamic systems abound in this context. Aircraft, electrical power systems, rockets, spacecraft, automobiles, and ships are all examples of thermodynamic systems. Buildings, derricks, and towers are all examples of structural systems. Telescopes and interferometers are examples of optical physics systems. Note that in each of these systems, the specific application depends on the system itself with unique characteristics in the application. Ships are certainly different thermodynamic systems than aircraft or rockets. In this context, the discipline equations are integrated by the integrating relationship. Thus, rockets have an important structural component, but this is not the functional system integrating component. Systems engineers must manage optical systems very carefully due to thermodynamic and structural dynamic effects. The effects of these discipline relationships do not integrate the optical system, however. The impact on image quality measures their effect.

Some systems have more than one integrating physics. Spacecraft capsules, for example, which are control volumes when in space, but treated as control masses during atmospheric reentry. Of course, these are both thermodynamic relationships where the change in environment and function result in a shift of the primary integrating relationship. This is seen for optical systems in which transport environments (e.g., vibration, shock, temperature, humidity) require an emphasis on a special configuration of the system to handle these environments. In this case, the effects of the structural design to handle transportation (whether road, rail, air, water, or ascent) must fit within the system's optical image quality when deployed for operation. Thus, a shift in environment and function can change the integrating relationships or shift the focus of the design. An elegant system is one that designs the structure to meet the optical image quality while enabling support for transportation environments.

4.2.1 System Exergy

For thermodynamic systems, system exergy is the system-integrating physics and serves as the basis for integrated system analysis. Thermodynamic systems include many types of systems: aircraft, rockets, spacecraft, ships, electrical power generation plants, etc. System exergy analysis provides a means for analyzing an integrated system using exergy as a quantifiable attribute of the integrated system. System exergy (sometimes referred to as thermodynamic system availability) is useful in tradeoffs both between subsystems and when comparing systems holistically. Subsystems represent different types of machines (mechanical, electrical, thermal). When integrating different machine types, the common physical input and output is work (whether electrical, mechanical, thermal, kinetic, or potential). The work output of one machine becomes a work input of another. The work relationships are not independent of the physical flows (e.g., thermal, mass, electrical) but contain all the physical parameters that generate the work for each subsystem. Thus, the system is integrated by the work done among the subsystems. This work is the basis of thermodynamic exergy as the integration property for systems. This property integrates all of the work producing terms (i.e., kinetic energy, potential energy, mechanical work, electrical work, thermal work, and flow work) into a balance condition that all systems must meet. The exergy relationship enforces energy balance, entropy balance, and mass balance and is measured against the reference environment in which the system operates.

Exergy is useful as an attribute in a system value function or as the objective function to provide a basis for a subsystem or system-level optimization. System exergy analysis provides a source for information useful in requirements development, tradeoffs, and optimization. Exergy considers the system environment, the system functional interactions, system performance, and losses. Exergy preserves all of the system balance relationships, including mass energy, entropy, momentum, thermal, etc. Exergy analysis treats the losses from all sources such as combustion, friction, etc. in the same manner. Therefore, exergy provides systems engineers with a meaningful measure of efficiency applicable across the whole system and throughout the design process inherently maintaining the physical balance and invoking the combination of performance and losses in the efficiency. The focus of exergy-based optimization provides maximal system efficiency for a thermodynamic system in the operating environment. System exergy analysis provides powerful and useful knowledge throughout the design process, from conceptual to detailed design. For systems in the operations lifecycle phase, application of exergy analysis to existing systems may offer insights to improve the system efficiency by operating the system differently.

4.2.1.1 General Approach. Exergy is a measure of the useful work of a system given the environment in which the system operates. It is derived by subtracting the system entropy balance from the system energy balance making use of mass conservation. A thermodynamic efficiency for the system can also be calculated using the exergy balance relationship. This efficiency accounts for variations in system properties that enable direct and meaningful comparisons of efficiency for different systems. One can minimize the useful work destroyed across the system while maintaining system performance by utilizing the exergy balance equation for a system. Aggregation of system functions is possible since exergy is a superposable property allowing individual subsystem or system properties to be seen in the balance equation maintaining the physical balances. Balancing the exergy relationship enables the systems engineer to determine the design with minimal exergy destroyed in system operation. Additionally, using exergy concepts supports subsystems as well as integrated systems in terms of feasible and optimal performance. This approach is in contrast to the typical design which optimizes each subsystem separately but does not necessarily ensure overall system optimization.

The first step in engineering a system integrated by thermodynamics is defining the system as a control volume or control mass. A control volume is a system that has a flow of mass across the system boundaries while the volume of the system is fixed. A control mass is a system that has a constant mass while the system volume may change. Some systems may be a control volume in some contexts and a control mass in other contexts (or environments). This is illustrated below in the discussion of spacecraft.

With the basic system defined, the second step is to determine the appropriate exergy balance equation using the specific exergy components of the system under analysis. The exergy balance states that the change in system exergy is equal to the sum of the net exergy transfer by heat, work, and mass less the exergy destroyed. Equation (6) gives the steady exergy balance for a control volume system as

$$\left\{X_{\text{heat in}} - X_{\text{heat out}}\right\} + \left\{X_{\text{work in}} - X_{\text{work out}}\right\} + \left\{X_{\text{mass in}} - X_{\text{mass out}}\right\}$$

$$-X_{\text{des}} = \left\{X_{\text{final}} - X_{\text{initial}}\right\}_{\text{system}} \ , \tag{6}$$

where X = exergy.

The various components of these equations represent the exergy values of the different aspects of the system. For the closed system with no mass transfer across the system boundaries, the form of the steady exergy balance for a control mass system is given in equation (7) as

$$\left\{X_{\text{heat in}} - X_{\text{heat out}}\right\} + \left\{X_{\text{work in}} - X_{\text{work out}}\right\} - X_{\text{des}} = \left\{X_{\text{final}} - X_{\text{initial}}\right\}_{\text{system}} \ . \tag{7}$$

The exergy destroyed term is the amount of entropy generated by the system at the reference environment temperature and is written as:

$$X_{\text{des}} = T_0 S_{\text{gen}} \ , \tag{8}$$

where

T_0 = reference environment temperature
S_{gen} = entropy generated.

The rate form of these equations can be obtained by differentiating the equations with respect to time. Time differentiation provides the rate of change for each set of terms and is represented using the over-dot symbol. For example, the rate of change of exergy transfer into the system is denoted as $\dot{X}_{\text{heat in}}$. On the other hand, the rate of change of system exergy is expressed as dX_{system}/dt. This form allows the balance to be stated in terms of the flow rates and rates of change for the system such as mass flow rate, heat transfer rate, etc.

These relationships can be expanded using the different forms of exergy shown in table 1. In this table, the zero terms represent the reference state (sometimes referred to as the dead state, ambient condition, standard temperature and pressure, etc.). Work can only be done by a system when its state is different than its reference environment. When its state equals the reference environment (i.e., $x = x_0$ (x is any state variable)), then all of the exergy terms are zero and $X_{\text{final}} = X_{\text{initial}}$. Exergy relationships thus inherently include the system environment.

Table 1. Exergy relationships

Exergy Relationships	Exergy Term
$X_{KE} = \frac{m_{\text{final}}}{2} V_{\text{final}}^2 - \frac{m_{\text{initial}}}{2} V_{\text{initial}}^2$	Change in kinetic energy
$X_{PE} = m_{\text{final}} g \text{height}_{\text{final}} - m_{\text{initial}} g \text{height}_{\text{initial}}$	Change in potential energy
$X_{\text{heat}} = \sum_n Q_n \left(1 - \frac{T_0}{T_n}\right)$	Heat transfer exergy for *n* heat flows
$X_{\text{work}} = W - P_0\left(\text{Vol}_{\text{final}} - \text{Vol}_{\text{initial}}\right)$	Mechanical work exergy
$\psi_{\text{flow}} = m\Big\{\left(h - h_0\right) + T_0\left(s - s_0\right) + \left(\frac{V_{\text{final}}^2}{2} - \frac{V_{\text{initial}}^2}{2}\right)$ $+ g\left(\text{height}_{\text{final}} - \text{height}_{\text{initial}}\right)\Big\}$	Fluid flow exergy
$X_{\text{static}} = \left(E - E_0\right) + P_0 X_{\text{fluid}} = P_0\left(\text{Vol} - \text{Vol}_0\right) - T_0\left(s - s_0\right)$ where energy = $E = U_{\text{system}} + KE_{\text{system}} + PE_{\text{system}}$	Nonflow exergy

Using the relationships in table 1, the exergy balance equations can be expanded for a control volume system in equation (9) and for a control mass system in equation (10) as:

$$\left(1-\frac{T_0}{T_{\text{in}}}\right)Q_{\text{in}}-\left(\frac{T_0-T_{\text{out}}}{T_{\text{out}}}\right)Q_{\text{out}}+\left(W_{\text{in}}-P_0\left(\text{Vol}_{\text{final}}-\text{Vol}_{\text{initial}}\right)\right)-W_{\text{out}}$$

$$+\Delta m\left(\left(h_{\text{in}}-h_{\text{out}}\right)-T_0\left(s_{\text{in}}-s_{\text{out}}\right)+\frac{v_{\text{in}}^2-v_{\text{out}}^2}{2}+g\left(\text{height}_{\text{in}}-\text{height}_{\text{out}}\right)\right)$$

$$-X_{\text{des}}=X_{\text{final}}-X_{\text{initial}} \tag{9}$$

and

$$\left(1-\frac{T_0}{T_{\text{in}}}\right)Q_{\text{in}}-\left(\frac{T_0-T_{\text{out}}}{T_{\text{out}}}\right)Q_{\text{out}}+\left(W_{\text{in}}-P_0\left(\text{Vol}_{\text{final}}-\text{Vol}_{\text{initial}}\right)\right)$$

$$-W_{\text{out}}-X_{\text{des}}=X_{\text{final}}-X_{\text{initial}}\ , \tag{10}$$

where

Q = heat transfer
W = work
P = pressure
Vol = volume
m = mass
h = enthalpy
s = entropy
v = velocity
g = gravitational acceleration
height = height (or altitude) of the system.

Alternatively, the exergy within a system can be evaluated on a rate basis instead of an interval basis. The general exergy balance equation on a rate basis for a control volume is given in equation (11):

$$\frac{dX}{dt}=\sum_i\left(1-\frac{T_0}{T_i}\right)\dot{Q}_i+\left(\dot{W}_{\text{in}}-P_0\frac{dA}{dt}-\dot{W}_{\text{out}}\right)+\sum_i\dot{m}\psi_{\text{in}}-\sum_i\dot{m}\psi_{\text{out}}-\dot{X}_{\text{des}}\ . \tag{11}$$

The flow exergy, ψ_{flow}, is defined in table 1. Exergy destruction is calculated as a sum of net heat transfer rate, power (rate of work transfer), and flow exergy into the system. Under steady-state conditions, the exergy balance can be recast to provide an expression for the exergy destruction rate equation (12):

$$\dot{X}_{\text{des}} = \sum_i \left(1 - \frac{T_0}{T_i}\right) \dot{Q}_i + \left(\dot{W}_{\text{in}} - \dot{W}_{\text{out}}\right) + \sum_i \dot{m}\psi_{\text{in}} - \sum_i \dot{m}\psi_{\text{out}} \ . \tag{12}$$

In some systems, cyclically operating subsystems coexist with subsystems operating in steady flow processes. To merge these formats, exergy balances for cyclic processes may be cast on a rate basis by averaging the total exergy transfer over the entire duration of the cycle, as given in equation (13):

$$\dot{X}_{\text{des, cycle}} = \frac{\Delta X_{\text{cycle}}}{t_{\text{cycle}}} \ . \tag{13}$$

For example, in the carbon dioxide removal assembly (CDRA) discussed in section 4.2.1.6, carbon dioxide is vented out of the spacecraft during a short interval in each cycle. To convert the exergy destruction into an average rate basis, the exergy content of vented carbon dioxide is divided by the duration, *t*, of the entire cycle.

The third step makes use of the summative property of exergy. One can separate exergy into the individual contributions of the subsystems and then add them together. The example of aircraft and Environmental Control and Life Support Systems (ECLSS) below illustrates this clearly. This allows an allocation of exergy contributions to subsystems during design. The systems engineer will need to adjust these allocations for input/outputs across subsystem boundaries. These may cancel when combined as a system but need to be added/subtracted from the subsystem allocations to maintain the system balance. The total exergy destruction of a system is equal to the sum of the exergy destroyed in each component or the rate at which exergy is destroyed is equal to the sum of the rate of exergy destruction in each component:

$$X_{\text{des, system}} = \sum X_{\text{des, components}} \tag{14a}$$

and

$$\dot{X}_{\text{des, system}} = \sum \dot{X}_{\text{des, components}} \ . \tag{14b}$$

Finally, calculation of the system exergy efficiency is possible. The ratio of exergy recovered to exergy expended defines system exergy efficiency in the interval approach. In the rate approach, fewer irreversibilities occurring during system operation means the system is more exergetically efficient because less exergy is destroyed and more obtainable work is realized. This is also shown as a relationship of exergy destroyed to exergy expended. Equation (15) gives these relationships:

$$\eta_{\text{exergy}} = \frac{X_{\text{recovered}}}{X_{\text{expended}}} = 1 - \frac{X_{\text{des}}}{X_{\text{expended}}} \ . \tag{15}$$

For systems where power is supplied to operate the system, the exergy efficiency can be defined as the exergy destroyed by the component relative to the power supplied to the component:

$$\eta_{\text{exergy}} = 1 - \frac{\dot{X}_{\text{des}}}{\dot{W}_{\text{input}}} \ . \tag{16}$$

Another way to formulate exergy efficiency is to compare exergy transfer to the reversible work limit. The reversible limit is the amount of exergy transferred in the optimal case where no irreversibilities exist. This formulation allows exergy efficiency to be calculated for flow through components as well as other components for which no power is supplied. The exergy efficiency for these cases is given in equations (17a) and (17b):

$$\eta_{\text{exergy}} = \frac{\dot{W}_{\text{rev}}}{\dot{W}_{\text{actual}}} \quad \text{work input} \tag{17a}$$

and

$$\eta_{\text{exergy}} = \frac{\dot{W}_{\text{actual}}}{\dot{W}_{\text{rev}}} \quad \text{work output} \ . \tag{17b}$$

4.2.1.2 Aircraft and Hypersonic Vehicle Exergy. For aircraft and hypersonic vehicles, exergy integrates all work aspects of these systems. Aerodynamic engine thrust, electrical power generation, and system cooling are all accounted for in this analysis.

Following the work of Riggins, Moorhouse, and Camberos, the exergy equations for aircraft and hypersonic vehicles are defined.[28] The exergy balance equation for an aircraft or hypersonic vehicle written in rate form is,

$$\begin{aligned} F_{\text{thrust}} * V_{\text{vehicle}} = {} & \dot{Q}_{\text{flow path}} + \dot{W}_{\text{flow path}} + \dot{m}_{\text{propellant}} \left(\frac{V_{\text{vehicle}}^2}{2} + \frac{\vec{V}_{\text{injected propellant}} \cdot \vec{V}_{\text{injected propellant}}}{2} \right. \\ & \left. + \left\{ h_{0,\text{propellant}} + \int_{T_0}^{T_{\text{propellant injection}}} C_{p,\text{ propellant}} \, dT \right\} \right) + \dot{m}_i \sum\nolimits_{l=1}^{n} \alpha_{l,i} \left[h_{l,i} - T_i \, s_l \left(T_i, P_i, y_{\text{mol}_{l,i}} \right) \right] \\ & - \dot{m}_{\text{wake}} \sum\nolimits_{l=1}^{n} \alpha_{l,i} \left[h_{l,\text{wake}} - T_{\text{wake}} \, s_l \left(T_{\text{wake}}, P_{\text{wake}}, y_{\text{mol}_{l,\text{wake}}} \right) \right] \\ & - T_0 \left(\dot{s}_{\text{vehicle}} - \dot{s}_{\text{wake}} - \dot{s}_{\text{injected propellant}} \right), \end{aligned} \tag{18}$$

where

F = force
V = velocity
h = enthalpy
T = temperature
$\dot{m}$ = mass flow rate
c_p = specific heat at constant pressure
s = entropy
P = pressure
α = mass fraction of chemical species (products and reactants) in the combustion process
y_{mol} = mole fraction of chemical species (products and reactants) in the combustion process.

Equation (18) shows the integrating nature of the exergy balance equation for an aircraft in which the outputs of various engineering disciplines are represented. This balance includes the aerodynamic losses including the energy destroyed by the vehicle in generating the wake. Aircraft with less turbulent flow fields in the wake lose less available work potential and are more efficient when holding all other losses constant. The aerodynamic exergy destroyed directly relates to the vehicle's overall drag. The exergy balance also includes the effects of engine losses reflected in the terms for the mass fraction of the different injected propellant reactants $\left(\alpha_{l,i}\right)$ and the resulting propellant products $\left(\alpha_{l,w}\right)$ exhausted into the wake. If the vehicle is thermally balanced $\left(\dot{Q}_{\text{flow path}} + \dot{W}_{\text{flow path}} = 0\right)$, then this equation can be simplified to:

$$F_{\text{thrust}} * V_{\text{vehicle}} = \dot{m}_{\text{propellant}}\left(\frac{V_{\text{vehicle}}^2}{2} + H_{\text{total}}\right) - T_0\ \left(\dot{s}_{\text{vehicle irreversible}} - \dot{s}_{\text{wake}}\right)\ , \tag{19}$$

where

$$\dot{m}_{\text{propellant}} H_{\text{total}} = \dot{m}_{\text{propellant}}\left(\left\{h_{0,\text{propellant}} + \int_{T_0}^{T_{\text{propellant injection}}} C_{p,\text{ propellant}}\, dT\right\}\right)$$

$$-T_0 \dot{S}_{\text{tanks}} + m_i \sum\nolimits_{l=1}^{n} \alpha_{l,i}\left[h_{l,i} - T_i\ s_l\left(T_i\, ,\ P_i, y_{\text{mol}_{l,i}}\right)\right]$$

$$-\dot{m}_{\text{wake}} \sum\nolimits_{l=1}^{n} \alpha_{l,i}\left[h_{l,\text{wake}} - T_{\text{wake}}\ s_l\left(T_{\text{wake}}\, ,\ P_{\text{wake}},\ y_{\text{mol}_{l,\text{wake}}}\right)\right]\ . \tag{20}$$

Calculation of this balance is possible across the full flight sequence. One can phase the sequences as taxi and take off, accelerate and climb, cruise, decelerate and loiter, accelerate, cruise, decelerate and descend, land and taxi. Integration of the exergy balance for each phase of the flight sequence yields:

$$\Delta m_{\text{propellant}} H_{\text{total}} - \int_{\text{taxi and takeoff}}^{\text{landing and taxi}} T_i \, ds_{\text{total irreversibilities}} = \Delta\left(m_{\text{vehicle}} \frac{V_{\text{vehicle}}^2}{2} \right)$$

$$+ \int_{\text{taxi and takeoff}}^{\text{landing and taxi}} m_{\text{vehicle}} \, g d_{\text{height}} \quad . \tag{21}$$

This form shows that the exergy balance in terms of the propellant work minus the system losses is equal to the change in vehicle kinetic energy and vehicle potential energy (i.e., the work done by the vehicle).

The total efficiency of an aircraft is then:

$$\eta_{\text{exergy}} = \frac{\Delta m_{\text{propellant}} H_{\text{total}} - \int_{\text{taxi and takeoff}}^{\text{landing and taxi}} T_i \, ds_{\text{total irreversibilities}}}{\Delta m_{\text{propellant}} H_{\text{total}}}$$

$$= 1 - \frac{\int_{\text{taxi and takeoff}}^{\text{landing and taxi}} T_i \, ds_{\text{total irreversibilities}}}{\Delta m_{\text{propellant}} \, H_{\text{total}}} \quad . \tag{22}$$

4.2.1.3 Launch Vehicle Exergy. Rockets and launch vehicles are thermodynamic systems. These systems function by converting chemical energy stored in the propellants into thrust to provide the vehicle with kinetic and potential energy changes. Thus, thermodynamics forms the systems-integrating physics for these vehicles.

Rockets and launch vehicles can be considered as control volumes for thermodynamic analyses. Mass flows from the tanks through the engines and is exhausted out of the nozzles. The vehicle outer mold line (OML) remains constant as mass is expelled across the boundary at the nozzle exits. At booster or stage separation, booster or stage mass is removed from the system and the OML (i.e., volume) changes. A control volume characterizes the system best since each stage or booster OML remains constant and the active stage continues to expel mass across the boundaries. The vehicle can be viewed as a combination of control volumes which incrementally drops the separate volumes but maintains the integrity of the individual control volumes as they separate.

For a launch vehicle, the exergy balance equation reflects the propulsion flow exergy, vehicle kinetic energy, and vehicle potential energy as the systems-integrating physics. The propulsion exergy is basically the thermodynamics flow exergy for the propulsion gases leaving the engine nozzle. This flow exergy includes kinetic and potential energy terms. Flow potential energy is not usually accounted for, but exergy balance requires this term in order to balance the mass change in the change of potential energy for the vehicle. Each of the launch vehicle subsystems contribute to one or more of these terms. The propulsion energy is summed over each stage (or booster set). Equation (23) gives the exergy balance equation for a rocket:

$$\sum_{\text{stages}}\left[\Delta m_{\text{propellant}}\left(h_{\text{prop}}+\frac{V_e^2}{2}+\frac{GM_E}{r_{\text{altitude, initial}}}\right)\right]-X_{\text{des}}$$

$$=\sum_{\text{stages}}\left[\Delta KE_{\text{vehicle}}+\Delta PE_{\text{vehicle}}\right]\ , \tag{23}$$

where

$\Delta m_{\text{propellant}}$ = change in propellant mass for each stage or booster
h_{prop} = propulsion enthalpy in the nozzle throat for engines or booster motors
V_e = equivalent exhaust velocity of engines or booster motors
$\Delta KE_{\text{vehicle}}$ = change in vehicle kinetic energy
$\Delta PE_{\text{vehicle}}$ = change in vehicle potential energy.

Note, for a rocket, the V_e term is much larger (typically ~10^3) than h_{prop}. This is one of the key differences between jet engines (discussed in sec. 4.2.1.2) and rocket engines or motors. The ΔKE and ΔPE terms can be expanded yielding the balance equation as,

$$\sum_{\text{stages}}\left[\Delta m_{\text{propellant}}\left(h_{\text{prop}}+\frac{V_e^2}{2}+\frac{GM_E}{r_{\text{altitude, initial}}}\right)\right]-X_{\text{des}}=\sum_{\text{stages}}\left[\left(M_{\text{vehicle, final}}\frac{V_{\text{vehicle, final}}^2}{2}\right.\right.$$

$$\left.\left.-M_{\text{vehicle, initial}}\frac{V_{\text{vehicle, initial}}^2}{2}\right)+\left(\frac{GM_E\,M_{\text{vehicle, initial}}}{r_{\text{altitude, initial}}}-\frac{GM_E\,M_{\text{vehicle, final}}}{r_{\text{altitude, final}}}\right)\right]\ , \tag{24}$$

where

M = mass of vehicle or planet as indicated
V = velocity of vehicle
G = universal gravitational constant
M_E = mass of the Earth
$r_{\text{altitude, initial}}$ = altitude of the vehicle on the pad
$r_{\text{altitude, final}}$ = altitude of vehicle at injection.

The vehicle ΔKE and ΔPE have discrete changes with each mass separation (i.e., booster separation, stage separation, launch abort system jettison, fairing jettison). Also, the h_{prop} and V_e terms are the single engine or motor values. The $\Delta m_{\text{propellant}}$ accounts for the number of engines or boosters as the mass change is based on the mass flow rate over a period of time for the total number of engines or boosters.

The right-hand side of equation (24) is the exergy recovered (i.e., exergy that is recovered in the form of velocity change) by the rocket. The left-hand side is the exergy expended by the rocket. These results may be substituted into equation (15) to solve for the efficiency. Performing these operations yields the exergy efficiency of a rocket when accelerating as

$$\eta_{\text{exergy}} = \frac{X_{\text{recovered}}}{X_{\text{expended}}} = \frac{X_{\text{final}} - X_{\text{initial}}}{X_{\text{expended}}} = \frac{\sum_{\text{stages}} \left[\Delta m_{\text{propellant}} \left(h_{\text{prop}} + \frac{V_e^2}{2} + \frac{GM_E}{r_{\text{altitude, initial}}} \right) \right] - X_{\text{des}}}{\sum_{\text{stages}} \left[\Delta m_{\text{propellant}} \left(h_{\text{prop}} + \frac{V_e^2}{2} + \frac{GM_E}{r_{\text{altitude, initial}}} \right) \right]}$$

$$= 1 - \frac{X_{\text{des}}}{\sum_{\text{stages}} \left[\Delta m_{\text{propellant}} \left(h_{\text{prop}} + \frac{V_e^2}{2} + \frac{GM_E}{r_{\text{altitude, initial}}} \right) \right]} \; . \tag{25}$$

For a vehicle that is braking, such as a planetary lander or Earth reentry vehicle, exergy is expended in order to reduce the amount of exergy in the vehicle (i.e., kinetic and potential energy). Thus, the exergy recovered term changes sign for braking resulting in:

$$\eta_{\text{exergy}} = \frac{-X_{\text{recovered}}}{X_{\text{expended}}} = \frac{X_{\text{initial}} - X_{\text{final}}}{X_{\text{expended}}} = 1 - \frac{X_{\text{des}}}{\Delta m_{\text{prop}} \left(h_{\text{prop}} + \frac{V_e^2}{2} + \frac{GM_E}{r_{\text{altitude, initial}}} \right)} \; . \tag{26}$$

Note, exergy that was expended to accelerate the booster or stage is lost when the booster or stage is discarded, so dropping the booster or stage registers as destroyed exergy and a decrease in efficiency. The system exergy destruction (X_{des}) is comprised of the individual system's exergy destruction in a superposable manner. This enables identification of the source of the loss as well as its magnitude on a subsystem-by-subsystem level and then aggregated to the system level. The process is similar for the various vehicle loss terms for an aircraft. Table 2 gives some of the launch vehicle loss terms that are part of the exergy destruction (or system losses).

Table 2. Launch vehicle loss terms.

Vehicle Exergy Destruction Components	Equation
Vehicle drag	$W_D = F_D d_{\text{trajectory}} = \left(\frac{1}{2} C_D\, \rho_{\text{atm}} A_{\text{vehicle}} V^2_{\text{vehicle}}\right) d_{\text{trajectory}}$
Vehicle aero	$W_D + Q_{\text{aero}} = \left(\frac{1}{2} C_D\, \rho_{\text{atm}} A_{\text{vehicle}} V^2_{\text{vehicle}}\right) d_{\text{trajectory}} + Q_{\text{aero}}$
Structural vibration	$W_{\text{buffet}} + W_{\text{vibration}}$
Engine efficiency[29]	$W_{\text{ER}} + W_{\text{KE}} + W_{\text{div}} + W_{bl} + Q_{fg}$
Vehicle steering, separation, control, and communication	$-W_{\text{mech}} = W_{\text{elec}} - W_{\text{pointing}}$

The exergy balance can then be expanded to show these loss terms explicitly as,

$$\begin{aligned}
&\Delta m_{\text{prop}} \sum\nolimits_{\text{stage}} \left(\left(\text{h}_{\text{prop}} - \text{h}_{\text{atm}}\right) - T_{\text{atm}} \left(s_{\text{exh}} - s_{\text{atm}}\right) + \frac{V_e^2}{2} + \frac{GM_E}{r_{\text{altitude, initial}}} \right) - W_{\text{mech}} - W_{\text{elec}} \\
&\quad - \left(\frac{1}{2} C_D \rho_{\text{atm}} A_{\text{vehicle}} V^2_{\text{vehicle}}\right) d_{\text{trajectory}} - Q_{\text{aero}} - W_{\text{pointing}} - W_{\text{buffet}} - W_{\text{vibration}} - W_{ER} \\
&\quad - W_{KE} - W_{div} - W_{bl} - Q_{fg} = \left(M_{\text{vehicle, final}} \frac{V^2_{\text{vehicle, final}}}{2} - M_{\text{vehicle, initial}} \frac{V^2_{\text{vehicle, initial}}}{2} \right) \\
&\quad + \left(\frac{GM_E\, M_{\text{vehicle, initial}}}{r_{\text{altitude, initial}}} - \frac{GM_E\, M_{\text{vehicle, final}}}{r_{\text{altitude, final}}} \right),
\end{aligned} \tag{27}$$

where

W_{mech} = mechanical work done by system (e.g., separation)
W_{elec} = electrical work done by system (e.g., avionics, communication)
Q_{aero} = aerothermal heat transfer
W_{pointing} = work done by pointing system to maintain trajectory course
W_{buffe} = work done by system against aerodynamic buffeting
$W_{\text{vibration}}$ = work done be structural vibration from engine thrust
W_{ER} = work done by engine energy release
W_{KE} = kinetic energy losses
W_{div} = work done in engine flow divergence
W_{bl} = work done in engine boundary layer flow
Q_{fg} = heat transfer in converting propellant fluid to gas state prior to combustion.

The integrating nature of the exergy balance equation for a rocket can be seen in equation (27) in which the inputs of various engineering disciplines are more clearly represented.

Figure 3 shows the exergy efficiency curve for a Saturn V (Apollo 17) flight produced with MATLAB®. This plot illustrates the integrated system performance for the whole vehicle. Key events (separation events) are shown in the figure.[26]

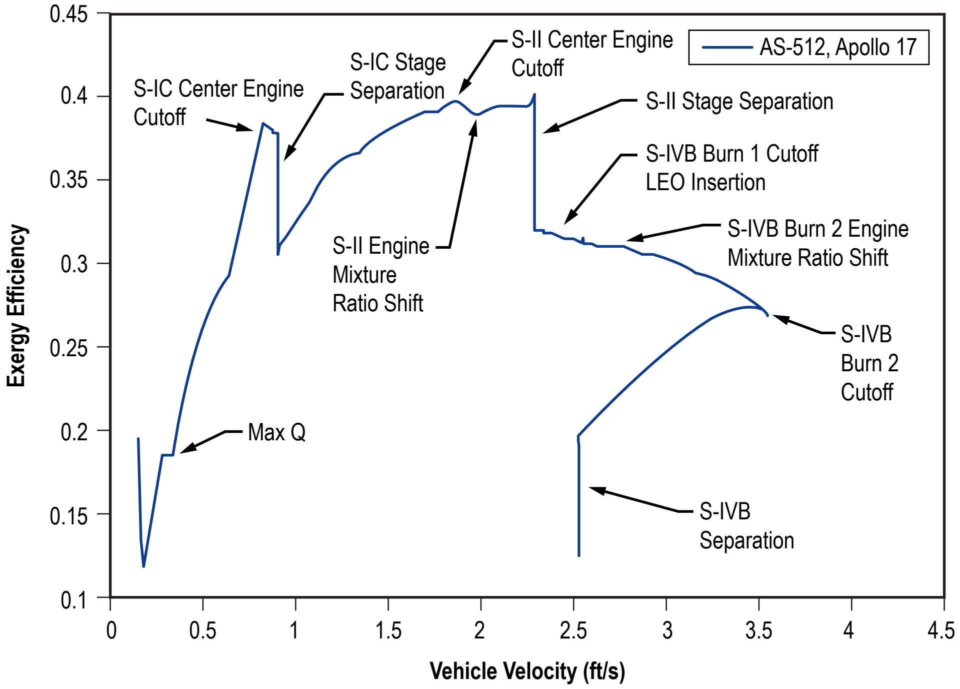


Figure 3. Exergy efficiency curve of a launch vehicle.

A Program to Optimize Simulated Trajectories (POST)[30] simulation file can be used to generate the necessary data for a given configuration. POST is a 3 degree of freedom (DoF) code that calculations the translation of the vehicle in all three directions. For vehicles such as spacecraft, where rotations in three directions are also needed to calculation maneuvers, a 6 DoF code such as Marshall Aerospace Vehicle Representation in C (MAVERIC)[31] can be used. The simulation variables in these codes corresponding to the exergy balance parameters must be identified. Sometimes these code variables must be translated into the necessary parameters. Table 3 gives the POST simulation variables needed to calculate the exergy balance and exergy efficiency. This table also shows the corresponding variable given in equation (24). The exergy balance uses mass remaining (i.e., wprp in POST). POST wprus (propellant used) can also be used for mass calculations but must be converted to remaining propellant by subtracting from the initial propellant mass value. Incremental average flow rate can be calculated by dividing the change in propellant mass for a step in the simulation by the change in time over that step.

Table 3. POST variables for exergy calculations.

POST Variable	POST Description	Exergy Balance Equation Term
time	Used to calculate flow rates for rate-based equations	e.g., $\frac{\Delta m_{\text{propellant}}}{\Delta t} = \dot{m}_{\text{propellant}}$
gdalt	Geodesic altitude	r_{altitude}
weight	Vehicle mass	M_{vehicle}
veli	Vehicle inertial velocity	V_{vehicle}
I_{sp}	Specific impulse	$V_e = \text{I}_{\text{sp}} * g_0$
thrust	Thrust can also be used to find V_e	$V_e = \text{thrust}/\Delta m_p$
wprp	Remaining propellant mass in tanks. wprp(0) –wprp yields propellant used	$\text{wprp}(0) - \text{wprp} = \Delta m_{\text{propellant}}$
wprus	Propellant mass used	$\text{wprus} = \Delta m_{\text{propellant}}$

The rocket equation shown in equation (28) is contained within the exergy balance equation (27):

$$-V_e \ln\left(\frac{M_{\text{vehicle, initial}}}{M_{\text{vehicle, final}}}\right) = \Delta V_{\text{vehicle}} \; . \tag{28}$$

By differentiating the exergy balance equation with respect to V_{vehicle} along with some simplifying assumptions, the rocket equation is obtained. Appendix A contains the complete derivation.

4.2.1.4 Integrated System Exergy: Aircraft Boost for a Launch Vehicle. The power of system exergy as a system integration approach is apparent when applied to launch vehicles that are air launched from an aircraft. The exergy balance, equation (9), allows for the integration of aircraft and launch vehicle terms to produce an integrated balance equation. To determine the exergy efficiency of an air-launch vehicle with an aircraft serving as the booster, a Boeing 747-400 Freighter was chosen as the aircraft booster. The launch vehicle rocket stages are the second and third stages of the integrated aircraft/rocket vehicle. The rocket launches from the aircraft while it is in steady, controlled flight, and then ascends to orbit. The aircraft returns to the landing site. The exergy balance allows both of these flight phases to be considered in the overall balance of the system.

The aircraft exergy balance is constructed from equation (21):

$$\Delta m_{\text{propellant}} H_{\text{total}} - \int_{\text{taxi and takeoff}}^{\text{landing and taxi}} T_i \, ds_{\text{total irreversibilities}} = \Delta\left(m_{\text{vehicle}} \frac{V_{\text{vehicle}}^2}{2}\right)$$

$$+ \int_{\text{taxi and takeoff}}^{\text{landing and taxi}} m_{\text{vehicle}} g d_{\text{height}} \; . \tag{29}$$

The rocket exergy balance is given by equation (24):

$$\sum_{\text{stages}}\left[\Delta m_{\text{propellant}}\left(h_{\text{prop}}+\frac{V_e^2}{2}+\frac{GM_E}{r_{\text{altitude, initial}}}\right)\right]-X_{\text{des}}=\sum_{\text{stages}}\left[\left(M_{\text{vehicle,final}}\frac{V_{\text{vehicle, final}}^2}{2}\right.\right.$$

$$\left.\left.-M_{\text{vehicle, initial}}\frac{V_{\text{vehicle, initial}}^2}{2}\right)+\left(\frac{GM_E\,M_{\text{vehicle, initial}}}{r_{\text{altitude,initial}}}-\frac{GM_E\,M_{\text{vehicle, final}}}{r_{\text{altitude,final}}}\right)\right]. \qquad (30)$$

Combining these equations gives the total balance for the aircraft boosted launch vehicle as,

$$\Delta m_{\text{propellant,aircraft}}H_{\text{total,aircraft}}+\sum_{\text{stages}}\left[\Delta m_{\text{propellant}}\left(h_{\text{prop}}+\frac{V_e^2}{2}+\frac{GM_E}{r_{\text{altitude, initial}}}\right)\right]-X_{\text{des}}$$

$$=\sum_{\text{stages}}\left[\left(M_{\text{vehicle, final}}\frac{V_{\text{vehicle, final}}^2}{2}-M_{\text{vehicle, initial}}\frac{V_{\text{vehicle, initial}}^2}{2}\right)\right.$$

$$\left.+\left(\frac{GM_E\,M_{\text{vehicle, initial}}}{r_{\text{altitude,initial}}}-\frac{GM_E\,M_{\text{vehicle, final}}}{r_{\text{altitude,final}}}\right)\right], \qquad (31)$$

where the vehicle mass, velocity, and altitude are for the total integrated aircraft and rocket during the boost phase.

A phase-specific mission plan (aircraft flight plan and rocket trajectory) is necessary. Solving the exergy balance using equation (31) along the aircraft flight path and rocket trajectory provides the progression of the system performance. This progression can be used to solve for the exergy efficiency along the flight path and trajectory using equations (22) and (25).

4.2.1.5 Planetary Transfer Exergy. Similarly, planetary transfer vehicles (i.e., satellites, landers, transports) are integrated by exergy. This includes their propulsion stages, electrical power systems (e.g., nuclear electric or solar electric), thermal systems, and crew volumes for transporting the crew.[32]

During propulsive trajectory changes, the exergy balance equation is the same as for a launch vehicle as shown by equation (24) in section 4.2.1.3. The propulsion engine (e.g., chemical, electric, nuclear thermal) characteristics (mass flow, enthalpy, exhaust velocity, and electrical power for electric propulsion) are all included on the left of the equation.

For coast phases of the flight trajectory, the exergy balance equation simplifies to the basic orbital mechanics relationship for a balanced system. In this case, the spacecraft energy (and exergy) is constant and the kinetic and potential energies increase and decrease in opposite directions:

$$E_{\text{vehicle}} = \left(M_{\text{vehicle}} \frac{V_{\text{vehicle}}^2}{2} - \frac{GxM_E \, M_{\text{vehicle}}}{r_{\text{altitude}}} \right) . \tag{32}$$

This creates an oscillatory relationship between the vehicle kinetic and potential energies with respect to the dominate body (typically the Sun in interplanetary space). Appendix A.2 shows the derivation of this relationship from the exergy balance equation.

Planetary and solar masses have a large effect on spacecraft exergy in interplanetary space. It is important to ensure an appropriate reference frame is used. A heliocentric reference frame is generally best for analyzing interplanetary space travel in the solar system. When operating within a planetary body's sphere of influence (SOI), the sphere in which the planetary gravitational influence is greater than the Sun's influence, then the solar influence can usually be ignored. In this case, a planetary centric (geospatial reference system for the Earth) can be used. Equation (33), gives the general relationship for the planetary SOI.[33]

$$r_{\text{SOI}} = R_{\text{Sun, planet}} \left(\frac{M_{\text{planet}}}{M_{\text{Sun}}} \right)^{2/5} , \tag{33}$$

where

r_{SOI} = radius of the planet's SOI
$R_{\text{Sun, planet}}$ = distance from the planet to the Sun
M_{planet} = mass of the planet
M_{Sun} = mass of the Sun.

Planetary transfer often uses a Hohmann transfer from Earth to Mars and back. The planetary stay is also important in calculating the possible trajectories. An 11-month stay on the planet is assumed in this discussion with a total mission length on the order of 2 to 3 years. This trajectory contains four main burns: trans-Mars injection (TMI), Mars orbit insertion (MOI), trans-Earth injection (TEI), and Earth orbit insertion (EOI). Four different propulsion systems were analyzed using this basic course: low enriched uranium (LEU) liquid hydrogen (LH_2) nuclear thermal propulsion (NTP), high enriched uranium (HEU) LH_2 NTP, LEU CH_4 (methane) NTP, and a chemical liquid oxygen (LO_2)/LH_2 system.

For the LEU CH_4 NTP and CHM LO_2-LH_2 cases, the mass flow rate for the main engine can be calculated from (I_{sp}) by using

$$\dot{m}_{\text{propellant}} = F_{\text{thrust}} \Big/ \left(I_{sp}\, g_0 \right) , \tag{34}$$

where

F_{thrust} = engine thrust
I_{sp} = engine specific impulse
g_0 = Earth's gravitational constant at sea level.

The mass flow rate of the reaction control system (RCS) thrusters is an important parameter in the maneuvers for the trajectory burns. For the calculations in this section, the RCS mass flow rate is 7 kg/s with an I_{sp} of 291 s.

Figure 4 shows the exergy efficiency of the LEU LH_2 NTP case during the first 500 s of TMI, and shows the decline in the efficiency during the RCS burn. Also visible in this plot is an efficiency drop just after the RCS burn; this corresponds to dropping an empty propellant tank. Exergy that was expended to accelerate the tank is lost when the tank is discarded, so dropping the tank registers as a decrease in efficiency.

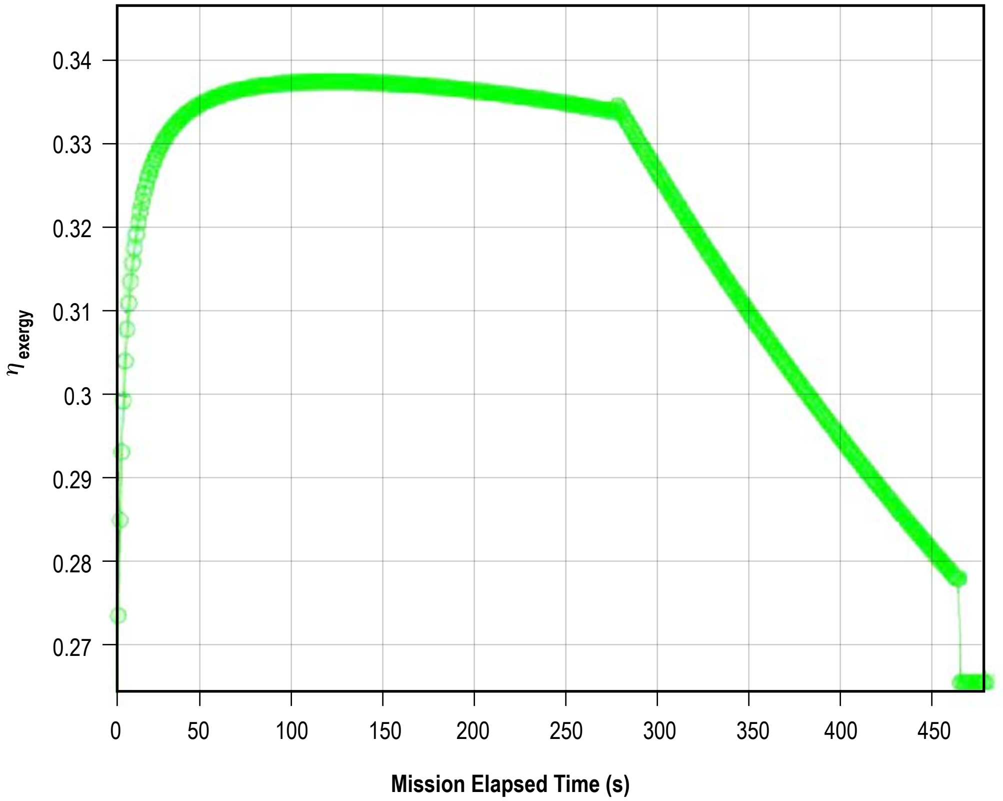


Figure 4. Exergy efficiency during TMI—LEU LH_2 NTP system.

4.2.1.5.1 Planetary Orbit Construction. Exergy calculations are sensitive to changes in position and velocity with respect to the departure and arrival planets, requiring a complete orbital trajectory to calculate exergy efficiency. A patched-conics trajectory or a multibody trajectory is necessary to show the complete system and planetary environments within each planet's SOI and in interplanetary space outside the planet's SOIs.

For a patched conics approach, the departure planet's and arrival planet's position and velocity are important for the periods when the spacecraft is within the planet's SOI. Outside the planetary SOIs, the Sun is treated as the sole gravity source. Acceleration due to the Sun's gravity is broken up into vector components along the interplanetary trajectory path. Figure 5 shows the spacecraft trajectory path and planets' orbital paths during the mission.

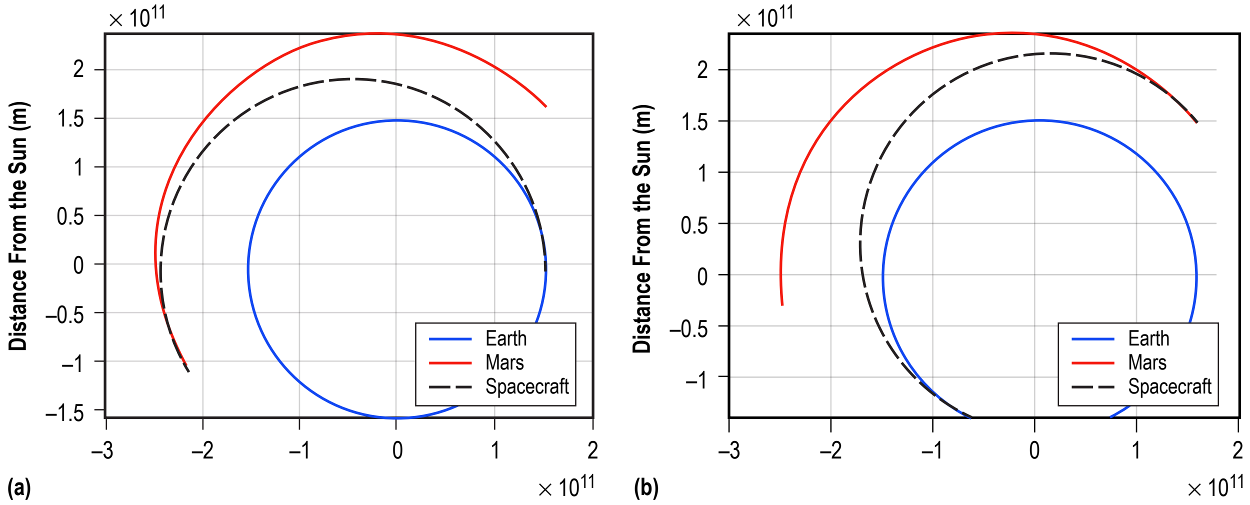


Figure 5. Earth to Mars spacecraft interplanetary trajectory and planet trajectories during the (a) outbound and (b) inbound (return) phases.

Using the planetary positions and the given position of the spacecraft at all points during the mission, the spacecraft's planet-relative distance, speed, and flight angle from the horizon are calculated for the days following the departure burns and leading up to the arrival burns using equations (35)–(37):

$$\vec{r}_{\text{vehicle, planet}} = \vec{r}_{\text{vehicle, Sun}} - \vec{r}_{\text{planet,Sun}} \ , \tag{35}$$

$$\vec{V}_{\text{vehicle, planet}} = \vec{V}_{\text{vehicle, Sun}} - \vec{V}_{\text{planet, Sun}} \ , \tag{36}$$

and

$$\varphi_{\text{planetary horizon}} = \frac{\pi}{2} - a\cos\left(\frac{\vec{V}_{\text{vehicle, planet}} \vec{r}_{\text{vehicle, planet}}}{\left\| \vec{V}_{\text{vehicle, planet}} \right\| \left\| \vec{r}_{\text{vehicle, planet}} \right\|} \right) , \tag{37}$$

where

$r_{\text{vehicle, planet}}$ = distance of vehicle relative to the planet
$r_{\text{vehicle, Sun}}$ = distance of vehicle relative to the Sun
$r_{\text{ planet, Sun}}$ = distance of planet relative to the Sun
$V_{\text{vehicle, planet}}$ = velocity of vehicle relative to the planet
$V_{\text{vehicle, Sun}}$ = velocity of vehicle relative to the Sun
$V_{\text{planet, Sun}}$ = velocity of planet relative to the Sun.

Using the spacecraft's distance from the planet over time, the exact time when it crosses the SOI boundary is interpolated with equation (38). The two points in time used for the interpolation are those just before and after crossing the SOI boundary, the radius of r_{SOI} defined in equation (31):

$$t_{\text{SOI}} = t_i + \left(t_f - t_i\right) \frac{\vec{r}_{\text{SOI}} - \left\|\vec{r}_{\text{vehicle, planet},i}\right\|}{\left\|\vec{r}_{\text{vehicle, planet},f}\right\| - \left\|\vec{r}_{\text{vehicle, planet},i}\right\|} \; . \tag{38}$$

With these values, the spacecraft's planet-relative velocity and flight angle from the horizon at that moment are similarly interpolated using equations (39) and (40):

$$\vec{V}_{\text{vehicle, planet,SOI}} = \vec{V}_{\text{vehicle, planet},i} + \left(t_{\text{SOI}} - t_i\right)\left(\frac{\vec{V}_{\text{vehicle, planet},f} - \vec{V}_{\text{vehicle, planet},i}}{t_f \, t_i}\right) \tag{39}$$

and

$$\varphi_{\text{horizon,SOI}} = \varphi_{\text{horizon},i} + \left(t_{\text{SOI}} - t_i\right)\left(\frac{\varphi_{\text{horizon},f} - \varphi_{\text{horizon},i}}{t_f - t_i}\right) \; . \tag{40}$$

Additionally, a new reference frame is created based on the spacecraft's position and velocity while crossing the SOI boundary, using equations (41)–(43). Planet-centric orbits within the SOI will be plotted in a 2D plane, and this reference frame will track the orientation of the plane relative to the solar ecliptic:

$$\hat{\imath} = \frac{\vec{r}_{\text{vehicle, planet,SOI}}}{\left\|\vec{r}_{\text{vehicle, planet,SOI}}\right\|} \; , \tag{41}$$

$$\hat{k} = \frac{\hat{\imath}\, x\, \vec{V}_{\text{vehicle, planet,SOI}}}{\left\|\hat{\imath}\, x\, \vec{V}_{\text{vehicle, planet,SOI}}\right\|} \; , \tag{42}$$

and

$$\hat{j} = \frac{\hat{k}x\hat{\imath}}{\left\|\hat{k}x\hat{\imath}\right\|} . \tag{43}$$

A transformation matrix is created using the new reference frame2 and equation (44), and will later be used to convert the SOI orbit back to a heliocentric reference frame:

$$T_{\text{transform}} = \begin{bmatrix} \hat{\imath}_x & \hat{\imath}_y & \hat{\imath}_z \\ \hat{j}_x & \hat{j}_y & \hat{j}_z \\ \hat{k}_x & \hat{k}_y & \hat{k}_z \end{bmatrix} . \tag{44}$$

With conditions at the SOI intersection established, the planet-centric transfer and parking orbits within the SOI can be determined. First, the transfer orbit's semi-major axis is calculated using equations (45) and (46):

$$V_{\text{SOI}} = \left\|\vec{V}_{\text{ship, planet,SOI}}\right\| \tag{45}$$

and

$$a_{\text{transfer}} = 1 \Big/ \left(\left(\frac{2}{r_{\text{SOI}}} \right) - \left(\frac{V_{\text{SOI}}^2}{\mu_{\text{planet}}} \right) \right) . \tag{46}$$

The speed and flight angle of the spacecraft at the edge of the SOI is sufficient to define a hyperbolic orbit past the planet. The parking orbit periapsis is established (400 km above the planet's surface, roughly the altitude that the International Space Station (ISS) orbits at over Earth in this example). This is the minimum shift that still puts the spacecraft's trajectory well above the atmosphere to avoid significant drag. Note, that aerobraking (not addressed here) requires an orbital altitude within the upper atmosphere with sufficient drag to reduce the spacecraft velocity (ΔV) to enter the prescribed parking orbit. Equations (47)–(52) are used to determine the apoapsis of the parking orbit for the listed ΔV at that periapsis:

$$e_{\text{transfer}} = 1 - \frac{r_{\text{periapsis}}}{a_{\text{transfer}}} , \tag{47}$$

$$V_{\text{periapsis,transfer}} = \sqrt{\mu_{\text{planet}} \left(\frac{2}{r_{\text{periapsis}}} - \frac{1}{a_{\text{transfer}}} \right)} , \tag{48}$$

$$V_{\text{periapsis, parking}} = V_{\text{periapsis,transfer}} - \Delta V \ , \tag{49}$$

$$a_{\text{parking}} = 1 \Big/ \left(\left(\frac{2}{r_{\text{periapsis}}} \right) - \left(\frac{V^2_{\text{periapsis,parking}}}{\mu_{\text{planet}}} \right) \right) , \tag{50}$$

$$e_{\text{parking}} = 1 - \left(\frac{r_{\text{periapsis,parking}}}{a} \right) , \tag{51}$$

and

$$r_{\text{apoapsis}} = r_{\text{periapsis}} \left(\frac{1 + e_{\text{parking}}}{1 - e_{\text{parking}}} \right) . \tag{52}$$

It is important for the apoapsis to remain within the planet's SOI and should be established to meet the parking orbit period necessary to meet mission objectives. Once the apoapsis and periapsis are established, the parking orbit periapsis is kept as the periapsis of the hyperbolic transfer orbit. This results in an extremely elliptical parking orbit with a very long period (particularly if it extends to the planetary SOI boundary). Equations (47)–(52) can be solved iteratively starting with an initial periapsis estimate and stepping in small increments (e.g., 100-mile periapsis altitude increases) until a reasonable apoapsis is found.

The eccentricity of the hyperbolic transfer orbit, the spacecraft's true anomaly at the SOI boundary, and its periapsis velocity can be calculated using equations (47), (48), and (53):

$$\theta_{\text{SOI}} = a\cos \left(\frac{a_{\text{transfer}} \left(1 - e^2_{\text{transfer}} \right) - r_{\text{SOI}}}{r_{\text{SOI}} \, e_{\text{transfer}}} \right) . \tag{53}$$

By applying the listed ΔV at the new periapsis as a point-thrust burn, the shape of the parking orbit around the planet can be approximated using equations (49)–(51). It is only an approximation because it assumes a point-thrust burn connects the transfer and parking orbit. As long as the chosen propulsion system is sufficiently high thrust, the actual parking orbits will be quite close to the listed values here, as a sufficiently short burn time (on a timescale of minutes) will be negligible compared to the period of the parking orbit.

The parking orbits are only an approximation based on point-thrust burns. To properly calculate the exergy efficiency, plots of the spacecraft's position and velocity during each burn will be needed. Equations (54) and (55) can be used to track the spacecraft forwards or backwards in time from periapsis to establish its trajectory. Another acceleration vector from the spacecraft's engine is added, aimed directly opposite its velocity vector at any point in time for backward tracking. This new vector is split into $\hat{\imath}$ and $\hat{\jmath}$ components for the calculations:

$$r_f = r_i + V_i \Delta t + \frac{1}{2}\dot{V}_i \Delta t^2 \tag{54}$$

and

$$V_f = V_i + \dot{V}_i \Delta t \ . \tag{55}$$

At this point, a complete planet-centric course contains the spacecraft's position and velocity from engine start to SOI exit (or vice versa for entry scenarios). This course is then rotated such that the SOI exit/entry point lies directly on the axis of the planet-centric reference frame. Equations (56) and (57) are then used to plot the spacecraft's heliocentric position and velocity while it is inside the SOI:

$$\vec{r}_{\text{vehicle,Sun}} = \vec{r}_{\text{planet,Sun}} + \left(T_{\text{transform}} \vec{r}_{\text{vehicle, planet}}\right) \tag{56}$$

and

$$\vec{V}_{\text{vehicle,Sun}} = \vec{V}_{\text{planet,Sun}} + \left(T_{\text{transform}} \vec{V}_{\text{vehicle, planet}}\right) \ . \tag{57}$$

4.2.1.5.2 Planetary Transfer Exergy Efficiency. With the modified mass data and orbital data in hand, the actual exergy calculations can begin. During each burn of the mission, changes in expended exergy are calculated in a patched conics context using equation (58) that is taken from equation (23), with mass drops for each time step being calculated from the tank drops and consumable use schedules. The sum of these step changes produces a plot of expended exergy that rises during burns but otherwise stays constant:

$$X_{\text{exp}} = \sum\nolimits_{\text{stages}} \left[\Delta m_{\text{propellant}} \left(h_{\text{ prop}} + \frac{V_e^2}{2} + \frac{GM_{\text{planet}}}{r_{\text{altitude, initial}}} \right) \right] . \tag{58}$$

Note that the propellant potential energy term is important for electric propulsion systems that produce highly efficient thrust in interplanetary space. This term maintains the proper energy term balance between the propulsion system and the vehicle.

In order to calculate destroyed exergy, changes in kinetic and potential energy must be tracked across the entire mission. To determine whether the change in kinetic or potential energy should be positive or negative during a given time step, the ruleset described in table 4 is applied, based on equations (59) and (60):

Table 4. Sign convention for changes in kinetic and potential energy.

Mass	Velocity	ΔKE_{step}	Distance	ΔPE_{step}
$M_f=M_i$	$V_f>V_i$	+	$r_f>r_i$	+
$M_f=M_i$	$V_f<V_i$	–	$r_f<r_i$	–
$M_f>M_i$	$V_f>V_i$		$r_f>r_i$	$+(Y>X)$
$M_f>XM_i$	$V_f=ZV_i$	+	$r_f=Yr_i$	$-(Y<X)$
$M_f>M_i$	$V_f<V_i$	$-(Z^2>X)$	$r_f<r_i$	
$M_f=XM_i$	$V_i=ZV_f$	$+(Z^2<X)$	$r_i=Yr_f$	–
$M_f<M_i$	$V_f>V_i$	$+(Z^2>X)$	$r_f>r_i$	
$M_i=XM_f$	$V_f=ZV_i$	$-(Z^2<X)$	$r_f=Yr_i$	+
$M_f<M_i$	$V_f<V_i$		$r_f<r_i$	$-(Y>X)$
$M_f=XM_i$	$V_i=ZV_f$	–	$r_i=Yr_f$	$+(Y<X)$

$$KE\!:\; m_f V_f^2 - m_i V_i^2 = \begin{cases} >0 \\ <0 \end{cases} \tag{59}$$

and

$$PE\!:\; \frac{m_i}{r_i} - \frac{m_f}{r_f} = \begin{cases} >0 \\ <0 \end{cases} . \tag{60}$$

Changes in the spacecraft's velocity and distance relative to the central body during that time step are taken into consideration when determining the sign. It should be noted that the values X, Y, and Z in the table are all ≥1.

Change in kinetic and potential energy during a given time step is then calculated using equations (61) and (62), where S is the sign taken from the previous table, either 1 or –1:

$$\Delta KE_{\text{step}} = \frac{S}{2}\left| m_f V_f^2 - m_i V_i^2 \right| \tag{61}$$

and

$$\Delta PE_{\text{step}} = S\mu \left| \frac{m_i}{r_i} - \frac{m_f}{r_f} \right| . \tag{62}$$

These step changes in kinetic and potential energy are summed over time to create a running total of energy changes. These sums are subtracted from the expended exergy using equation (63) to calculate the exergy destroyed, which then directly leads to the exergy efficiency, defined in equation (25), at that point in time:

$$X_{\text{des}} = X_{\text{exp}} - \sum \Delta KE_{\text{step}} - \sum \Delta PE_{\text{step}} \ . \qquad (63)$$

When the spacecraft is within a planet's SOI and not burning propellant, efficiency does not stay constant, but fluctuates with the planetary gravity influences as the vehicle and planet both move along their respective trajectories. This is avoided by using a patched conics model for the orbital modifications, where exergy calculations are applied to each SOI independently, not using the heliocentric portion of the trajectory. Whenever the spacecraft crosses into or out of an SOI, the most recent value for the total change in kinetic and potential energy is carried over to the next series of calculations. This ensures that exergy efficiency stays constant whenever the spacecraft's mass and velocity are constant, even across SOIs.

The final exergy efficiency plots over the whole mission for each propulsion system are given in figures 6–9.

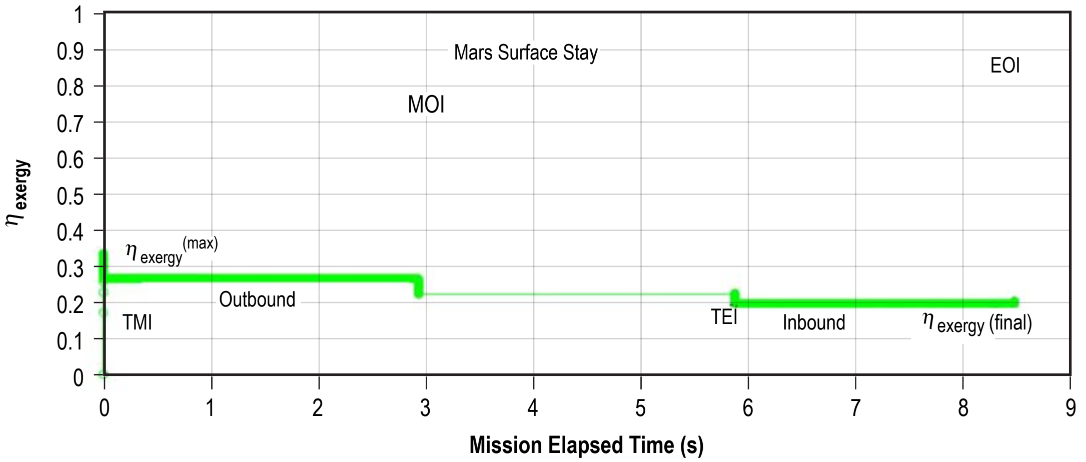


Figure 6. Exergy efficiency throughout the mission using the LEU LH_2 NTP system.

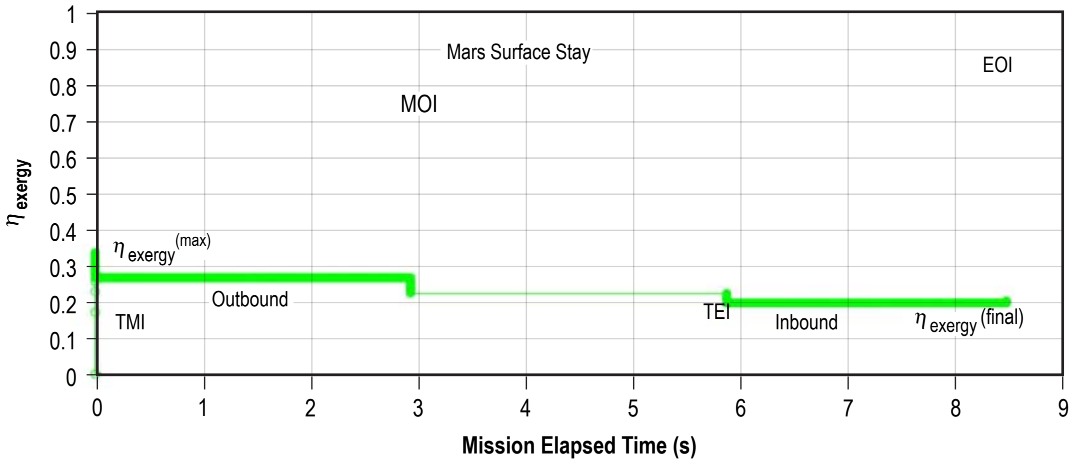


Figure 7. Exergy efficiency throughout the mission using the HEU LH_2 NTP system.

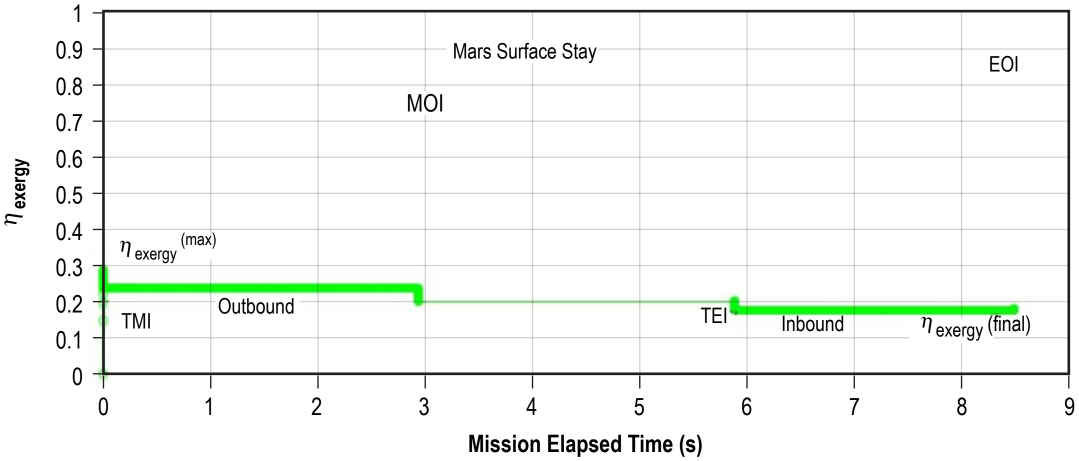


Figure 8. Exergy efficiency throughout the mission using the LEU CH_4 NTP system.

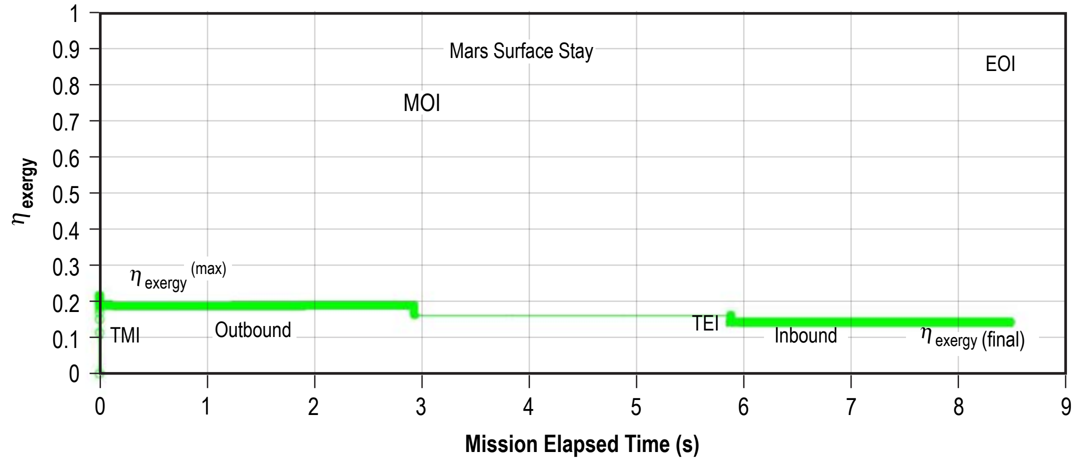


Figure 9. Exergy efficiency throughout the mission using the CHM LOX-LH_2 system.

As seen previously in figure 4, exergy efficiency will rise sharply when using a main engine during a departure burn, and then decrease during the following RCS burn. This is because of the RCS burn's lower I_{sp}, destroying more exergy for the same exergy expenditure, thus lowering the efficiency of that stage of the mission. Efficiency also drops when ejecting an empty propellant tank or spent consumables, as the exergy expended to move those components up to speed is lost when they are discarded.

Unlike the departure burns, braking burns when arriving at a planet show exergy efficiency decreasing during both the main burn and RCS burn. Equation (26) gives the exergy efficiency for the braking burns or maneuvers.

Notable efficiency values are given in table 5. The maximum exergy efficiency achieved (during the TMI burn) is shown in the top row. The second row shows the total exergy efficiency achieved from the TMI departure burn through parking orbit insertion (EOI) burn at the return to Earth.

Table 5. Final exergy efficiency results for all propulsion systems analyzed.

	LEU LH_2 NTP (%)	**HEU LH_2 NTP (%)**	**LEU CH_4 NTP (%)**	**CHM LO_2-LH_2 (%)**
η_{exg} (max)	33.75	33.77	28.89	22.03
η_{exg} (total)	19.57	19.63	17.59	14.23

Overall, exergy efficiency roughly scales directly with I_{sp} and inversely with the total initial mass of the spacecraft. HEU LH_2 NTP achieves the highest efficiencies, but only just barely, since it has the same I_{sp} as the LEU LH_2 NTP case and is only minimally lighter due to reactor sizing to produce the same thrust. CHM LO_2-LH_2 has the lowest efficiencies by far, since its I_{sp} is considerably less than the other cases.

4.2.1.5.3 Multibody Effects. The patched conics approach seeks to isolate the vehicle motion in a reference frame for the dominant gravitational body. Thus, the planets gravitational effects are only considered inside its own SOI. Solar effects are only considered in interplanetary space. In reality, the gravitational forces from all of the planets produce effects on the vehicle. This is the source of the gravitational slingshot used by interplanetary satellites to boost their velocity.

Exergy balance reflects these gravitational forces as potential energy terms being imparted on the vehicle. Force body diagrams are helpful when constructing these balance equations to account for the direction the planetary force is acting on the vehicle. The change in the vehicle potential energy is the negative of the change in planetary potential energy as the two bodies interact. Both the effects on the vehicle and the effect on the body enter into the balance equation as:

$$\Delta m_{\text{propellant}}\left(\left(h_{\text{prop}}-h_0\right)-T_0\left(s_{\text{prop}}-s_0\right)+\frac{V_e^2}{2}+\frac{GM_{\text{Sun}}}{r_{\text{fluid, initial}}}\right)+\frac{GM_{\text{planet}}M_{\text{vehicle, initial}}}{r_{\text{vehicle, body}}}$$

$$-X_{\text{des}}=\left(M_{\text{vehicle, final}}\frac{V^2_{\text{vehicle, final}}}{2}-M_{\text{vehicle, initial}}\frac{V^2_{\text{vehicle, initial}}}{2}\right)+\left(\frac{GM_{\text{Sun}}M_{\text{vehicle, initial}}}{r_{\text{vehicle, initial}}}\right.$$

$$\left.-\frac{GM_{\text{Sun}}M_{\text{vehicle, final}}}{r_{\text{vehicle, final}}}\right)-\left.\frac{GM_{\text{planet}}M_{\text{vehicle, initial}}}{r_{\text{vehicle, body}}}\right) . \qquad (64)$$

Note that the potential energy effects from a gravitational body can be combined with the vehicle effect results in a factor of 2 on these potential energy terms. The complete derivation is presented in appendix A.3. Considering the effects of the Earth, Moon, Sun, and Mars for an Earth-Mars transfer vehicle, the exergy balance can be expanded for each of these body terms as:

$$\sum\nolimits_{\text{stages}}\left[\Delta m_{\text{propellant}}\left(h_{\text{prop}}+\frac{V_e^2}{2}+\frac{GM_E}{r_{\text{altitude, initial}}}\right)\right]+\frac{2GM_E\,M_{\text{vehicle,initial}}}{r_{\text{vehicle,Earth}}}+\frac{2GM_{\text{Moon}}M_{\text{vehicle,initial}}}{r_{\text{vehicle,Moon}}}$$

$$+\frac{2GM_{\text{Mars}}\,M_{\text{vehicle,initial}}}{r_{\text{vehicle,Mars}}}-X_{\text{des}}=\sum\nolimits_{\text{stages}}\left[\left(M_{\text{vehicle,final}}\frac{V^2_{\text{vehicle, final}}}{2}-M_{\text{vehicle, initial}}\frac{V^2_{\text{vehicle, initial}}}{2}\right)\right.$$

$$\left.+\left(\frac{GM_E\,M_{\text{vehicle,initial}}}{r_{\text{altitude,initial}}}-\frac{GM_E\,M_{\text{vehicle, final}}}{r_{\text{altitude, final}}}\right)\right]. \qquad (65)$$

For crewed vehicles, the crew module exergy balance provides another piece of the exergy efficiency and follows the ECLSS relationships derived in section 4.2.1.6.

4.2.1.6 Crewed Spacecraft Exergy. Exergy provides the integrating physics for crewed spacecraft (i.e., modules and capsules). These systems are primarily large crew environment volumes, defined by the functioning of the ECLSS. Their structure, power, and thermal management functions are all defined and managed by the cabin environments. Capsules sometimes also form a dual role and function as a control mass during reentry.

Crewed spacecraft have different exergy relationships driven by the ECLSS within the control volume and driven as a control mass by reentry. The exergy balance for the control volume involve the different steps in the generation and maintenance of the cabin environment, including heat transfer processes, electrical work, and chemical reactions. Reentry systems are control masses, and their exergy balance reflects heat transfer, vehicle kinetic energy reductions, and vehicle potential energy reductions.

For capsules supported by a service module (SM), the SM is both a supply to the ECLSS functions and a propulsive stage. Thus, the SM is also a control volume in which mass is expelled across the system boundary in supplying fluids to ECLSS functions, in controlling the spacecraft attitude, and to propel the integrated spacecraft.

4.2.1.6.1 Crew Module ECLSS Exergy. Considering the ISS Laboratory module, the application of system exergy illustrates the integrating and specific nature of the exergy balance. Focusing on the in-space module aspects, the exergy balance equation for an ECLSS (modeled after the ISS ECLSS[34]) can be written, based on equation (9), as

$$\begin{aligned}\Delta X_{\text{ECLSS}} &= \sum\nolimits_{\text{process}} \Delta m_{\text{fluid}} \left(\left(h_{\text{final}} - h_{\text{cabin}} \right) - T_{\text{cabin}} \left(s_{\text{final}} - s_{\text{cabin}} \right) + \left(\frac{\mathrm{V}_{\text{final}}^2}{2} \right) \right) \\ &- \sum\nolimits_{\text{process}} \Delta m_{\text{fluid}} \left(\left(h_{\text{initial}} - h_{\text{cabin}} \right) - T_{\text{cabin}} \left(s_{\text{initial}} - s_{\text{cabin}} \right) + \left(\frac{V_{\text{initial}}^2}{2} \right) \right) = \sum \left(1 - \frac{T_{\text{cabin}}}{T_{\text{crew}}} \right) Q_{\text{crew}} \\ &- \sum \left(\frac{T_{\text{cabin}} - T_{\text{coolant}}}{T_{\text{coolant}}} \right) Q_{\text{TMS}} + \sum W_{\text{EPS}} - P_{\text{cabin}} \left(\text{Vol}_{\text{final}} - \text{Vol}_{\text{initial}} \right) + \Delta m_{\text{in}} \Big[\sum \left(h_{\text{in}} - h_{\text{cabin}} \right) \\ &- T_{\text{cabin}} \left(s_{\text{in}} - s_{\text{cabin}} \right) + \left(\frac{V_{\text{in}}^2}{2} \right) - \sum \left(h_{\text{out}} - h_{\text{cabin}} \right) - T_{\text{cabin}} \left(s_{\text{out}} - s_{\text{cabin}} \right) + \left(\frac{V_{\text{out}}^2}{2} \right) \Big] - X_{\text{des}} \; . \qquad (66)\end{aligned}$$

The energy transfer rates as well as mass flow rates may be more readily observed if equation (66) is written in rate form as

$$\begin{aligned}\dot{X}_{\text{ECLSS}} &= \sum\nolimits_{\text{process}} \dot{m}_{\text{fluid}} \left(h_{\text{final}} - h_{\text{cabin}} \right) - T_{\text{cabin}} \left(s_{\text{final}} - s_{\text{cabin}} \right) + \left(\frac{V_{\text{final}}^2}{2} \right) \\ &- \sum\nolimits_{\text{process}} \dot{m}_{\text{fluid}} \left(h_{\text{initial}} - h_{\text{cabin}} \right) - T_{\text{cabin}} \left(s_{\text{initial}} - s_{\text{cabin}} \right) + \left(\frac{V_{\text{initial}}^2}{2} \right) = \sum \left(1 - \frac{T_{\text{cabin}}}{T_{\text{crew}}} \right) \dot{Q}_{\text{crew}} \\ &- \sum \left(\frac{T_{\text{cabin}} - T_{\text{coolant}}}{T_{\text{coolant}}} \right) \dot{Q}_{\text{TMS}} + \sum \dot{W}_{\text{EPS}} - P_{\text{cabin}} \dot{\text{Vol}} + \dot{m}_{\text{in}} \Big[\sum \left(h_{\text{in}} - h_{\text{cabin}} \right) \\ &- T_{\text{cabin}} \left(s_{\text{in}} - s_{\text{cabin}} \right) + \left(\frac{V_{\text{in}}^2}{2} \right) - \sum \left(h_{\text{out}} - h_{\text{cabin}} \right) - T_{\text{cabin}} \left(s_{\text{out}} - s_{\text{cabin}} \right) + \left(\frac{V_{\text{out}}^2}{2} \right) \Big] - \dot{X}_{\text{des}} \; . \end{aligned}$$

(67)

The enthalpy and entropy terms in these system level equations contain the enthalpy and entropy changes from the various chemical reactions. This will be the product (output from the chemical reaction) and reactant (input to the chemical reactions) enthalpies and entropies. The chemical reaction products return to the cabin air (desired chemical products) or store (for later return)/vent as waste products.

Equations (66) and (67) involve several variables that are based on the composite functions of the ECLSS. Applying the exergy balance from equation (6) to a system design, the change in exergy of each subsystem may be summed to obtain the change in exergy for the total system, as presented in equation (68):

$$\Delta X_{\mathrm{ECLSS}} = \Delta X_{\mathrm{ACS}} + \Delta X_{\mathrm{AR}} + \Delta X_{\mathrm{THC}} + \Delta X_{\mathrm{WRM}} + \Delta X_{\mathrm{WM}} \ , \tag{68}$$

where

X_{ECLSS} = exergy of the ECLSS
X_{ACS} = exergy of the atmospheric control and supply
X_{AR} = exergy of the atmospheric revitalization
X_{THC} = exergy of temperature and humidity control
X_{WRM} = exergy of water recovery management
X_{WM} = exergy of waste management.

The superposable concept of exergy for subsystems to obtain the system exergy change is clearly illustrated in equation (69) in rate form following equation (67), as

$$\dot{X}_{\mathrm{ECLSS}} = \dot{X}_{\mathrm{ACS}} + \dot{X}_{\mathrm{AR}} + \dot{X}_{\mathrm{THC}} + \dot{X}_{\mathrm{WRM}} + \dot{X}_{\mathrm{WM}} \ , \tag{69}$$

where

$\dot{X}_{\mathrm{ACS}}$ = rate of change of exergy of the atmospheric control and supply
$\dot{X}_{\mathrm{AR}}$ = rate of change of exergy of the atmospheric revitalization
$\dot{X}_{\mathrm{THC}}$ = rate of change of exergy of temperature and humidity control
$\dot{X}_{\mathrm{WRM}}$ = rate of change of exergy of water recovery management
$\dot{X}_{\mathrm{WM}}$ = rate of change of exergy of waste management.

When breaking down into subsystems, the input and output of each subsystem must be included. These values often cancel when combined into the full system equation but must be maintained in the individual subsystem terms to ensure each subsystem is properly characterized.

The atmospheric revitalization (AR) subsystem, is particularly important in the determination of crew module system efficiency because it supplies oxygen to and removes carbon dioxide and particulates from the cabin. Generating oxygen to supply the needs of the crew is done in the oxygen generation assembly (OGA) through the process of electrolysis, which consumes power to separate water into hydrogen and oxygen as illustrated in figure 10. The electrolysis process also generates hydrogen to be used by fuel cells, in which the reverse reaction occurs to generate power.

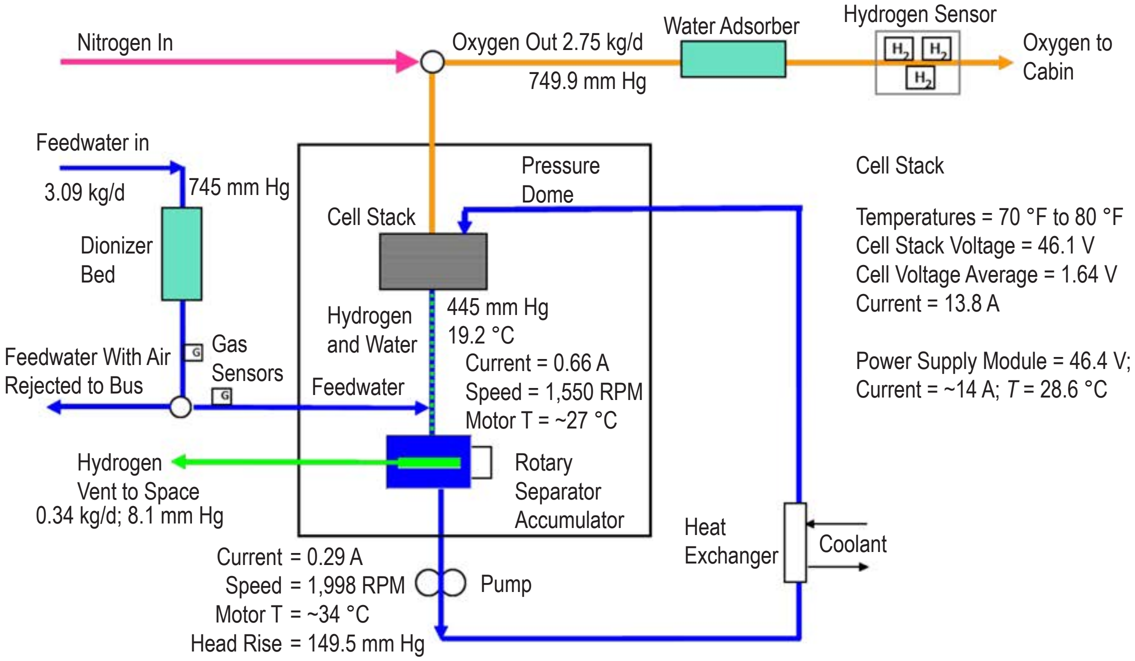


Figure 10. Schematic of the OGA. The electrolyzer cell stack is analyzed in detail.

The OGA electrolyzer was assumed to be insulated in order to simplify analysis. The power necessary to operate the OGA electrolyzer was calculated by multiplying the voltage by the current through the electrolyzer. The voltage of the electrolyzer is a summation of the Nernst potential, the activation over potential, ohmic over potential, and concentration over potential, as given in equation (70). The concentration over potential is neglected because the electrolyzer is never expected to run in conditions where the concentration over potential is a factor. The Nernst potential is the thermodynamic ideal at zero current on the voltage current curve. It is a function of the electrolyzer operating temperature. The activation over potential is the potential needed to drive the hydrolysis reaction and is a function of the exchange current densities at the anode and cathode, as given in equation (71). The ohmic over potential, as given in equation (72), is the potential added by resistance inside the electrolyzer. Membrane resistance and interfacial resistance are the main contributors to ohmic over potential. Here, it is noted that η indicates an electrochemical over potential and not an efficiency.

$$V\left(T_{\text{cell}}, p, i\right) = E_0\left(T_{\text{cell}}, p\right) + \eta_{\text{act}}(i) + \eta_{\text{ohmic}}(i) + \eta_{\text{concentration}} \,, \tag{70}$$

$$\eta_{\text{act}} = \eta_{\text{anode}} - \eta_{\text{cathode}} = \left(\frac{RT_{\text{cell}}}{F}\sin h^{-1}\left(\frac{i}{2i_{A_0}}\right)\right) - \left(-\frac{RT_{\text{cell}}}{F}\sin h^{-1}\left(\frac{i}{2i_{C_0}}\right)\right), \tag{71}$$

and

$$\eta_{\text{ohmic}} = \eta_{\text{membrane}} + \eta_{\text{interface}} = \left(\frac{L_{\text{membrane}}}{\sigma_{\text{membrane}}}\right) i + R_{\text{interface}} i \ , \tag{72}$$

where

$V(T,p,i)$ = voltage of the cell stack as a function of temperature, pressure, and current
T_{cell} = operating temperature of the cell stack
p = pressure in the cell stack
$E_0(T,p)$ = Nernst potential
η_{act} = activation over potential
η_{anode} = anode potential
η_{cathode} = cathode potential
i = exchange current density
i_{A_0} = exchange current density at the anode
i_{C_0} = exchange current density at the cathode
η_{ohmic} = ohmic over potential
η_{membrane} = electrolyte membrane over potential
$\eta_{\text{interface}}$ = interface over potential
L_{membrane} = electrolyte membrane thickness
σ_{membrane} = electrolyte membrane conductivity
$R_{\text{interface}}$ = interfacial resistance.

These resistances are functions of the materials used inside the electrolyzer. The concentration over potential is caused by transport limits that are not expected to be reached in this case and is therefore ignored. From these overpotentials, four parameters that were directly affected by type of material were chosen to be studied: exchange current density, membrane thickness, interfacial ohmic resistance, and membrane conductivity. Values were chosen from literature and estimated if common values could not be found in literature.

While values for power input and mass flow rates are already specified for most components, in the OGA electrolyzer, necessary power input is based on many parameters as shown in the flowchart in figure 11.

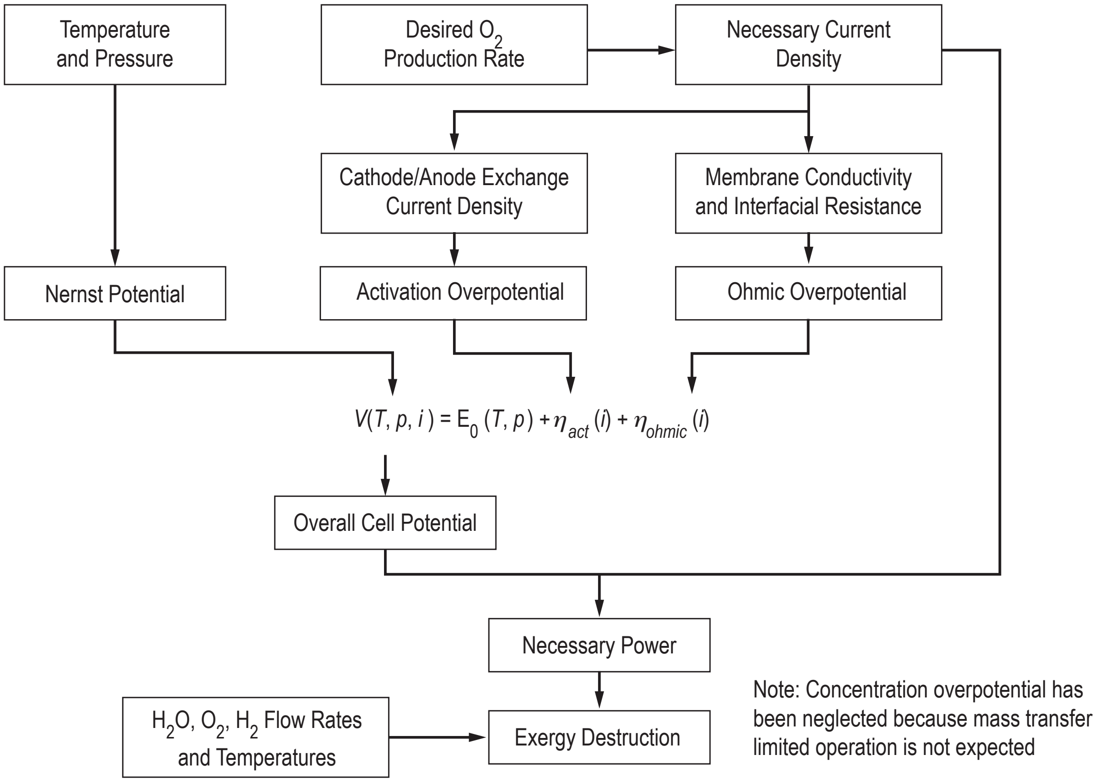

Temperature and Pressure
Desired O_2 Production Rate
Necessary Current Density
Cathode/Anode Exchange Current Density
Membrane Conductivity and Interfacial Resistance
Nernst Potential
Activation Overpotential
Ohmic Overpotential
$V(T, p, i) = E_0 (T, p) + \eta_{act} (i) + \eta_{ohmic} (i)$
Overall Cell Potential
Necessary Power
H_2O, O_2, H_2 Flow Rates and Temperatures
Exergy Destruction
Note: Concentration overpotential has been neglected because mass transfer limited operation is not expected

Figure 11. Flowchart of calculating voltage drop, power consumption, and exergy destruction in the OGA electrolyzer.

After the necessary power input is calculated, it is substituted into the exergy rate balance for the electrolyzer shown in equation (73). In this exergy balance, water is in liquid phase, and hydrogen and oxygen are in gas phase. The electrolyzer is assumed to be adiabatic, and the change in exergy content of unreacted water is neglected.

$$\begin{aligned}\dot{X}_{\text{des,OGA}} = \dot{W}_{\text{elec}} - \left(1 - \frac{T_0}{T_{\text{elec}}}\right)\dot{Q}_{\text{elec}} + \dot{m}_{\text{H}_2\text{O, in}}\left(h_{\text{H}_2\text{O,in}} - T_0\left(s_{\text{H}_2\text{O, in}}\right)\right) \\ -\left[\dot{m}_{\text{H}_2\text{,out}}\left(h_{\text{H}_2\text{,out}} - T_0\left(s_{\text{H}_2\text{,out}}\right)\right) + \dot{m}_{\text{O}_2\text{,out}}\left(h_{\text{O}_2\text{,out}} - T_0\left(s_{\text{O}_2\text{,out}}\right)\right)\right. \\ \left.+\dot{m}_{\text{H}_2\text{O,out}}\left(h_{\text{H}_2\text{O, out}} - T_0\left(s_{\text{H}_2\text{O,out}}\right)\right)\right] ,\end{aligned} \qquad (73)$$

where

$\dot{X}_{\text{des, OGA}}$ = exergy destruction rate of the OGA
$\dot{W}_{\text{elec}}$ = power consumption rate of the electrolyzer
T_{elec} = operating temperature of the electrolyzer
$\dot{Q}_{\text{elec}}$ = heat loss rate from the electrolyzer
$\dot{m}_{\text{H}_2\text{O, in}}$ = mass flow rate of liquid water entering the electrolyzer
$\dot{m}_{\text{H}_2\text{O, out}}$ = mass flow rate of hydrogen leaving the electrolyzer
$\dot{m}_{\text{O}_2\text{, out}}$ = mass flow rate of oxygen leaving the electrolyzer
$\dot{m}_{\text{H}_2\text{O, out}}$ = mass flow rate of liquid water leaving the electrolyzer.

Other components are currently neglected because the electrolyzer exergy destruction term is the dominant term of the total OGA exergy destruction.

The blower in the CDRA, shown in figure 12, is an example of a common application of the general exergy balance. Many components in the ECLSS are similar to the blower in that there is a power input into, heat loss from, and exergy flow(s) through the component. The reversible work limit for the blower is calculated as shown in equation (74). A negative sign is applied to heat loss from the blower, which is defined as a positive value:

$$\dot{W}_{\text{rev, blower}} = -\left(1 - \frac{T_{\text{ref}}}{T_{\text{blower}}}\right)\dot{Q}_{\text{blower}} + \dot{m}_{\text{air}}\left[\left(h_{\text{air, in}} - h_{\text{air, out}}\right) - T_{\text{ref}}\left(s_{\text{air, in}} - s_{\text{air, out}}\right)\right] , \qquad (74)$$

where

$\dot{W}_{\text{rev, blower}}$ = reversible work rate limit of the blower
T_{blower} = operating temperature of the blower
$\dot{Q}_{\text{blower}}$ = heat loss rate from the blower
$\dot{m}_{\text{air}}$ = mass flow rate of air through blower
$h_{\text{air, in}}$ = specific enthalpy of air entering the blower
$h_{\text{air, out}}$ = specific enthalpy of air leaving the blower
$s_{\text{air, in}}$ = specific entropy of air entering the blower
$s_{\text{air, out}}$ = specific entropy of air leaving the blower.

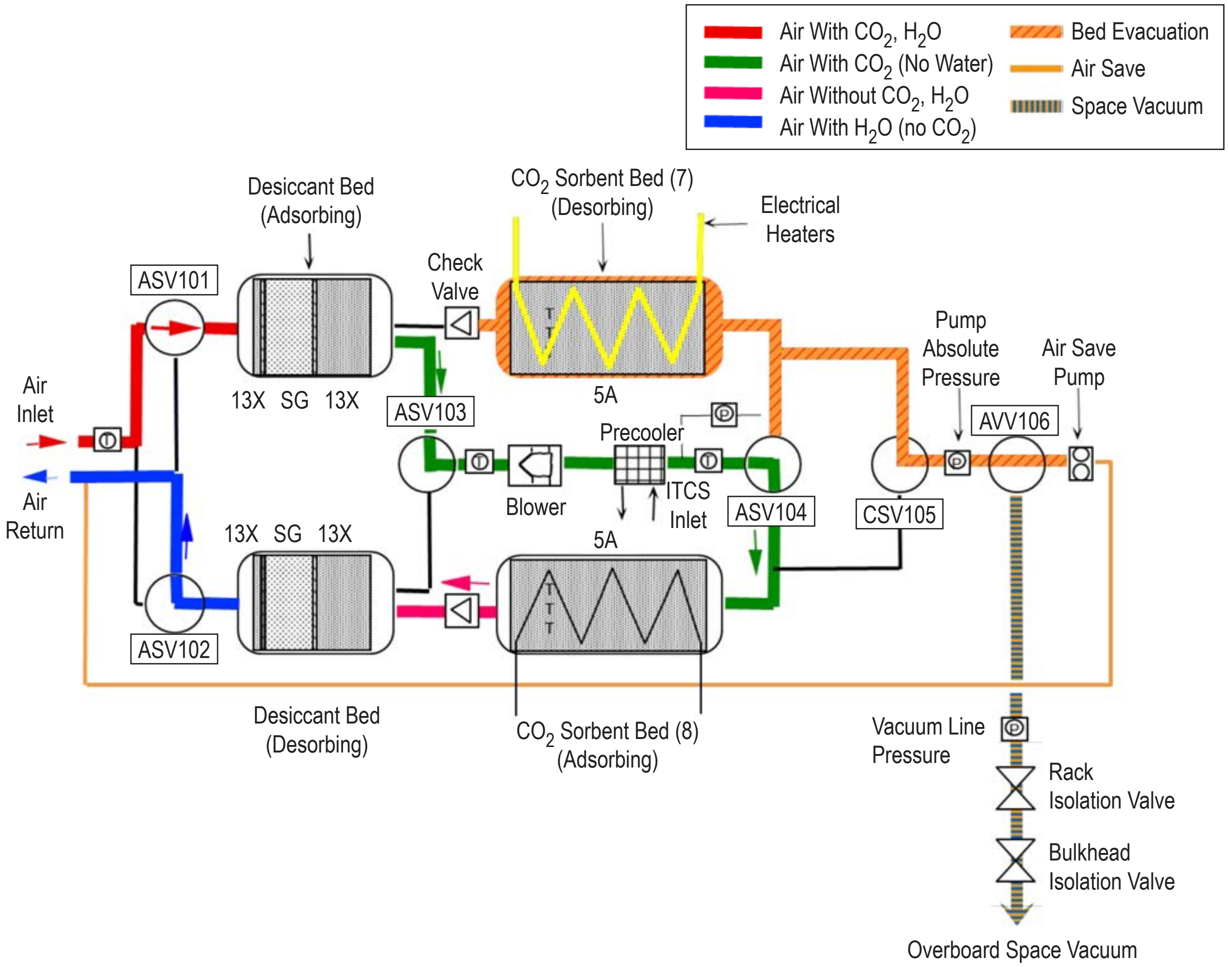


Figure 12. Schematic of the CDRA. The blower, precooler, sorbent bed, and venting process are analyzed.

Subsequently, exergy destruction and exergy efficiency are calculated in equations (75) and (76), in which the power supplied (actual work input) is defined as a positive value:

$$\dot{X}_{\text{des, blower}} \dot{W}_{\text{rev, blower}} + \dot{W}_{\text{actual, blower}} \tag{75}$$

and

$$\eta_{\text{blower}} = \frac{\dot{W}_{\text{rev, blower}}}{-\dot{W}_{\text{actual, blower}}} \ . \tag{76}$$

The precooler is an example of a heat exchanger. There is no power input into this type of component. Instead, the function is usually to remove heat from a hot stream by exchanging heat with a cold stream, which is typically coolant. Exergy destruction is calculated as the exergy content lost by the hot (air) stream that is not gained by the cold (coolant) stream, as given in equation (77). Similarly, exergy efficiency is the exergy gained by the cold stream relative to the exergy lost by the hot stream, as given in equation (78):

$$\dot{X}_{\text{des, precooler}} = \dot{m}_{\text{air}}\left[\left(h_{\text{air, in}} - h_{\text{air,out}}\right) - T_{\text{ref}}\left(s_{\text{air, in}} - s_{\text{air, out}}\right)\right]$$

$$+\dot{m}_{\text{coolant}}\left[\left(h_{\text{coolant, in}} - h_{\text{coolant, out}}\right) - T_{\text{ref}}\left(s_{\text{coolant, in}} - s_{\text{coolant, out}}\right)\right] \quad (77)$$

and

$$\eta_{\text{precooler}} = \frac{\dot{m}_{\text{coolant}}\left[\left(h_{\text{coolant, out}} - h_{\text{coolant, in}}\right) - T_{\text{ref}}\left(s_{\text{coolant, out}} - s_{\text{coolant, in}}\right)\right]}{\dot{m}_{\text{air}}\left[\left(h_{\text{air, in}} - h_{\text{air, out}}\right) - T_{\text{ref}}\left(s_{\text{air, in}} - s_{\text{air, out}}\right)\right]}\ , \quad (78)$$

where

$\dot{X}_{\text{des, precooler}}$	= exergy destruction rate of the precooler
$\eta_{\text{precooler}}$	= exergy efficiency of the precooler
$\dot{m}_{\text{air}}$	= mass flow rate of air through the precooler
$h_{\text{air, in}}$	= specific enthalpy of air entering the precooler
$h_{\text{air, out}}$	= specific enthalpy of air leaving the precooler
$s_{\text{air, in}}$	= specific entropy of air entering the precooler
$s_{\text{air, out}}$	= specific entropy of air leaving the precooler
$\dot{m}_{\text{coolant}}$	= mass flow rate of coolant through the precooler
$h_{\text{coolant, in}}$	= specific enthalpy of coolant entering the precooler
$h_{\text{coolant, out}}$	= specific enthalpy of coolant leaving the precooler
$s_{\text{coolant, in}}$	= specific entropy of coolant entering the precooler
$s_{\text{coolant, out}}$	= specific entropy of coolant leaving the precooler.

The CDRA sorbent bed illustrates converting exergy change over a cycle into a rate basis. Over a cycle, the sorbent bed is heated and cooled while releasing and adsorbing CO_2. Exergy destruction over a cycle can therefore be represented in a rate basis as given in equation (79). Equations (80) and (81), respectively, give the exergy transfer over the heating and cooling periods. State 1 is the beginning of the heating part of the cycle, and state 2 is the beginning of the cooling part of the cycle.

$$\dot{X}_{\text{des, sorbent bed}} = \frac{X_{\text{des, heating}} + X_{\text{des, cooling}}}{t_{\text{cycle}}}\ , \quad (79)$$

$$X_{\text{des, heating}} = \left(\dot{W}_{\text{heating}}\right)\left(t_{\text{heating}}\right) + \left(\left(h_2 - h_1\right) - T_{\text{ref}}\left(s_2 - s_1\right)\right)\ , \quad (80)$$

and

$$X_{\text{des, cooling}} = \left(\left(h_2 - h_1\right) - T_{\text{ref}}\left(s_2 - s_1\right)\right) \ , \tag{81}$$

where

$\dot{X}_{\text{des, sorbent bed}}$ = average exergy destruction rate of the sorbent bed
$X_{\text{des,heating}}$ = exergy change of the sorbent bed during the heating part of the cycle
$X_{\text{des,cooling}}$ = exergy change of the sorbent bed during the cooling part of the cycle
t_{cycle} = total duration of a full cycle
$\dot{W}_{\text{heating}}$ = power consumption rate during the heating part of the cycle
t_{heating} = duration of the heating part of the cycle
h_1 = enthalpy at the beginning of the heating part of the cycle
h_2 = enthalpy at the beginning of the cooling part of the cycle
s_1 = entropy at the beginning of the heating part of the cycle
s_2 = entropy at the beginning of the cooling part of the cycle.

Power is supplied only during the heating part of the cycle, so to convert the total amount supplied over a full cycle to an average in rate basis, it is necessary to average power input over the entire duration of the cycle as shown in equation (82):

$$\dot{X}_{\text{input, sorbent bed}} = \dot{W}_{\text{heating}}\left(\frac{t_{\text{heating}}}{t_{\text{cycle}}}\right) \ . \tag{82}$$

If the system is assumed to be adiabatic, it can be simplified to an exergy sink:

$$\eta_{\text{sorbent bed}} = 1 - \frac{\dot{X}_{\text{des, sorbent bed}}}{\dot{X}_{\text{input, sorbent bed}}} = 0 \ . \tag{83}$$

Carbon dioxide venting is an example of mass leaving the spacecraft. In this case, the exergy content of the mass leaving is treated as being destroyed because it is not recoverable after being vented to space:

$$\dot{X}_{\text{des, } CO_2 \text{ venting}} = \frac{m_{CO_2} u_{CO_2}}{t_{\text{cycle}}} \tag{84a}$$

$$\eta_{CO_2 \text{ venting}} = 0 \ , \tag{84b}$$

where

$\dot{X}_{\text{des, } CO_2\text{, venting}}$ = average exergy loss from venting CO_2 to space
$\eta_{CO_2 \text{ venting}}$ = exergy efficiency of venting CO_2 to space

m_{CO_2}	= mass of CO_2 vented to space
u_{CO_2}	= internal energy of CO_2 vented to space
t_{cycle}	= total duration of a full CDRA cycle.

To calculate the total exergy destruction of the CDRA, the total exergy destruction of all components is summated as shown in equation (85). The overall exergy efficiency of the CDRA is calculated by comparing the exergy destruction to the total power supplied as shown in equation (87):

$$\dot{X}_{\text{des, CDRA}} = \dot{X}_{\text{des, precooler}} + \dot{X}_{\text{des, blower}} + \dot{X}_{\text{des, sorbent bed}} + \dot{X}_{\text{des, CO}_2\text{venting}} \ , \tag{85}$$

$$\dot{X}_{\text{input, CDRA}} = \dot{X}_{\text{input, blower}} + \dot{X}_{\text{input, sorbent bed}} \ , \tag{86}$$

and

$$\eta_{\text{CDRA}} = 1 - \frac{\dot{X}_{\text{des, CDRA}}}{\dot{X}_{\text{input, CDRA}}} \ . \tag{87}$$

A MATLAB program was written to model exergy balances for the ECLSS components, subsystems, and overall system. From these exergy balances, exergy destruction rates and efficiencies were calculated. The exergy balance equations are summarized in tables 6–12 including those for the remaining subsystems and components within the ECLSS not discussed above. Subsystems include the OGA, CDRA, major constituents analyzer (MCA), trace contaminant control subsystem (TCCS), temperature and humidity control (THC), waste management (WM), and water recovery module (WRM).

Table 6. Exergy destruction calculations for the OGA.

Component	Exergy Destruction	Exergy Efficiency
OGA (overall)	$\dot{X}_{\text{des, OGA}} = \dot{W}_{\text{elec}} - \left(1 - \frac{T_0}{T_{\text{elec}}}\right)\dot{Q}_{\text{elec}}$ $+\dot{m}_{\text{H}_2\text{O, in}}\left(h_{\text{H}_2\text{O, in}} - T_0\left(s_{\text{H}_2\text{O, in}}\right)\right)$ $-\Big[\dot{m}_{\text{H}_2\text{, out}}\left(h_{\text{H}_2\text{, out}} - T_0\left(s_{\text{H}_2\text{, out}}\right)\right)$ $+\dot{m}_{\text{O}_2\text{, out}}\left(h_{\text{O}_2\text{, out}} - T_0\left(s_{\text{O}_2\text{, out}}\right)\right)$ $+\dot{m}_{\text{H}_2\text{O, out}}\left(h_{\text{H}_2\text{O, out}} - T_0\left(s_{\text{H}_2\text{O, out}}\right)\right)\Big]$	$1 - \frac{\dot{X}_{\text{des, OGA}}}{\dot{X}_{\text{input, OGA}}}$

Table 7. Exergy destruction calculations for the CDRA.

Component	Exergy Destruction	Exergy Efficiency
Precooler	$\dot{m}_{\text{air}}\left[\left(h_{\text{air, in}}-h_{\text{air, out}}\right)-T_{\text{ref}}\left(s_{\text{air, in}}-s_{\text{air, out}}\right)\right]$ $+\dot{m}_{\text{coolant}}\left[\left(h_{\text{coolant, in}}-h_{\text{coolant, out}}\right)\right.$ $\left.-T_{\text{ref}}\left(s_{\text{coolant, in}}-s_{\text{coolant, out}}\right)\right]$	$\dfrac{\dot{m}_{\text{coolant}}\left[\left(h_{\text{coolant, out}}-h_{\text{coolant, in}}\right)\right]}{\dot{m}_{\text{air}}\left[\left(h_{\text{air, in}}-h_{\text{air, out}}\right)-T_{\text{ref}}\left(s_{\text{air, in}}-s_{\text{air, out}}\right)\right]}$ $-\dfrac{\dot{m}_{\text{coolant}}\left[T_{\text{ref}}\left(s_{\text{coolant, out}}-s_{\text{coolant, in}}\right)\right]}{\dot{m}_{\text{air}}\left[\left(h_{\text{air, in}}-h_{\text{air, out}}\right)-T_{\text{ref}}\left(s_{\text{air, in}}-s_{\text{air, out}}\right)\right]}$
Blower	$-\left(1-\dfrac{T_{\text{ref}}}{T_{\text{blower}}}\right)\dot{Q}_{\text{blower}}$ $+\dot{m}_{\text{air}}\left[\left(h_{\text{air, in}}-h_{\text{air, out}}\right)-T_{\text{ref}}\left(s_{\text{air, in}}-s_{\text{air, out}}\right)\right]$ $+\dot{W}_{\text{actual, blower}}$	$\dfrac{\dot{W}_{\text{rev, blower}}}{\dot{W}_{\text{actual, blower}}}$
Sorbent bed	$\dfrac{\left(\dot{W}_{\text{heating}}\right)\left(t_{\text{heating}}\right)}{t_{\text{cycle}}}+\dfrac{2\left(\left(h_2-h_1\right)-T_{\text{ref}}\left(s_2-s_1\right)\right)}{t_{\text{cycle}}}$	0 (adiabatic) $1-\dfrac{\dot{X}_{\text{des, sorbent bed}}}{\dot{W}_{\text{heating}}\left(\dfrac{t_{\text{heating}}}{t_{\text{cycle}}}\right)}$ (nonadiabatic)
CO_2 venting	$\dfrac{m_{CO_2}u_{CO_2}}{t_{\text{cycle}}}$	0 (all vented gas lost)
CDRA (overall)	$\dot{X}_{\text{des, precooler}}+\dot{X}_{\text{des, blower}}+\dot{X}_{\text{des, sorbent bed}}$ $+\dot{X}_{\text{des, }CO_2\text{, venting}}$	$1-\dfrac{\dot{X}_{\text{des, CDRA}}}{\dot{X}_{\text{input, CDRA}}}$

Table 8. Exergy destruction calculations for the MCA.

Component	Exergy Destruction	Exergy Efficiency
Mass spectrometer	$\dot{W}_{\text{pump}}-\left(1-\dfrac{T_{\text{ref}}}{T_{\text{pump}}}\right)\dot{Q}_{\text{pump}}+\dot{m}_{\text{air}}\,[\left(h_{\text{air, in}}-h_{\text{air, out}}\right)$	$\dfrac{\dot{W}_{\text{rev, mass spec}}}{-\dot{W}_{\text{actual, mass spec}}}$
Pump	$\dot{W}_{\text{pump}}-\left(1-\dfrac{T_{\text{ref}}}{T_{\text{pump}}}\right)\dot{Q}_{\text{pump}}+\dot{m}_{\text{air}}\,[\left(h_{\text{air, in}}-h_{\text{air, out}}\right)$ $-T_{\text{ref}}\left(s_{\text{air, in}}-s_{\text{air, out}}\right)]$	$\dfrac{\dot{W}_{\text{rev, pump}}}{-\dot{W}_{\text{actual, pump}}}$
Heater	$\dot{W}_{\text{heater}}$	–
MCA (overall)	$\dot{X}_{\text{des, mass spec}}+\dot{X}_{\text{des, pump}}+\dot{X}_{\text{des, heater}}$	$1-\dfrac{\dot{X}_{\text{des, MCA}}}{\dot{X}_{\text{input, MCA}}}$

Table 9. Exergy destruction calculations for the TCCS.

Components	Exergy Destruction	Exergy Efficiency
Charcoal bed	$\dot{m}_{\text{air, in}}\left(h_{\text{air, in}} - T_{\text{ref}}\, s_{\text{air, in}}\right) - \left(\dot{m}_{\text{air, in}}\right.$ $\left. - \dot{m}_{\text{deposited}}\right)\left(h_{\text{air, out}} - T_{\text{ref}}\, s_{\text{air, out}}\right)$	$\left(\dot{m}_{\text{air, in}} - \dot{m}_{\text{deposited}}\right)\left(h_{\text{air, out}} - T_{\text{ref}}\, s_{\text{air, out}}\right)$ $/\dot{m}_{\text{air, in}}\left(h_{\text{air, in}} - T_{\text{ref}}\, s_{\text{air, in}}\right)$
Blower	$\dot{W}_{\text{blower}} - \left(1 - \frac{T_{\text{ref}}}{T_{\text{blower}}}\right)\dot{Q}_{\text{blower}}$ $+\dot{m}_{\text{air}}\left[\left(h_{\text{air, in}} - h_{\text{air, out}}\right) - T_{\text{ref}}\, s_{\text{air, in}} - s_{\text{air, out}}\right]$	$\frac{\dot{W}_{\text{rev, blower}}}{\dot{W}_{\text{actual, blower}}}$
Flowmeter	$\dot{W}_{\text{flowmeter}}$	–
Catalytic oxidizer assembly	$\dot{W}_{\text{HTCO}} - \left(1 - \frac{T_{\text{ref}}}{T_{\text{HTCO}}}\right)\dot{Q}_{\text{HTCO}}$ $+\dot{m}_{\text{air}}\left[\left(h_{\text{air, in}} - h_{\text{air, out}}\right) - T_{\text{ref}}\left(s_{\text{air, in}} - s_{\text{air, out}}\right)\right]$	$\frac{\dot{W}_{\text{rev, HTCO}}}{-\dot{W}_{\text{actual, HTCO}}}$
LiOH bed	$\dot{m}_{\text{air, in}}\left(h_{\text{air, in}} - T_{\text{ref}}\, s_{\text{air, in}}\right) - \left(\dot{m}_{\text{air, in}}\right.$ $\left. - \dot{m}_{\text{deposited}}\right)\left(h_{\text{air, out}} - T_{\text{ref}}\, s_{\text{air, out}}\right)$	$\left(\dot{m}_{\text{air, in}} - \dot{m}_{\text{deposited}}\right)\left(h_{\text{air, out}} - T_{\text{ref}}\, s_{\text{air, out}}\right)$ $/\dot{m}_{\text{air, in}}\left(h_{\text{air, in}} - T_{\text{ref}}\, s_{\text{air, in}}\right)$
TCCS (overall)	$\dot{X}_{\text{des, charcoal bed}} + \dot{X}_{\text{des, flowmeter}} + \dot{X}_{\text{des, blower}}$ $+\dot{X}_{\text{des, LiOH bed}} + \dot{X}_{\text{des, HTCO}}$	$1 - \frac{\dot{X}_{\text{des, TCCS}}}{\dot{X}_{\text{input, TCCS}}}$

Table 10. Exergy destruction calculations for the THC.

Component	Exergy Destruction	Exergy Efficiency
Resistance temperature detector	$\dot{W}_{\text{RTD}}$	–
Fans (located in the common cabin air assembly (CCAA), avionics air assembly (AAA), and intermodule ventilation)	$+\dot{m}_{\text{air}}\Big[\Big(h_{\text{air, in}}-h_{\text{air, out}}\Big)-T_{\text{ref}}\Big(s_{\text{air, in}}-s_{\text{air, out}}\Big)\Big]$ $+\dot{m}_{\text{air}}\Big[\Big(h_{\text{air, in}}-h_{\text{air, out}}\Big)-T_{\text{ref}}\Big(s_{\text{air, in}}-s_{\text{air, out}}\Big)\Big]$	$1-\dfrac{\dot{X}_{\text{des, fan}}}{\dot{W}_{\text{fan}}}$
Condensing heat exchangers (located in the CCAA and AAA)	$\dot{m}_{\text{moist air, in}}\Big(h_{\text{moist air, in}}-T_{\text{ref}}s_{\text{moist air, in}}\Big)$ $-\dot{m}_{\text{moist air, out}}\Big(h_{\text{moist air, out}}-T_{\text{ref}}s_{\text{moist air, out}}\Big)$ $-\dot{m}_{\text{H}_2\text{O, out}}\Big(h_{\text{H}_2\text{O, out}}-T_{\text{ref}}s_{\text{H}_2\text{O, out}}\Big)$ $+\dot{m}_{\text{coolant}}\Big[\Big(h_{\text{coolant, in}}-h_{\text{coolant, out}}\Big)\Big)$ $-T_{\text{ref}}\Big(s_{\text{coolant, in}}-s_{\text{coolant, out}}\Big)\Big]$	$\Big[\dot{m}_{\text{moist air, in}}\Big(h_{\text{moist air, in}}-T_{\text{ref}}s_{\text{moist air, in}}\Big)$ $-\dot{m}_{\text{moist air, out}}\Big(h_{\text{moist air, out}}-T_{\text{ref}}s_{\text{moist air, out}}\Big)$ $-\dot{m}_{\text{H}_2\text{O, out}}\Big(h_{\text{H}_2\text{O, out}}-T_{\text{ref}}s_{\text{H}_2\text{O, out}}\Big)\Big]$ $/\dot{m}_{\text{coolant}}\Big[\Big(h_{\text{coolant, in}}-h_{\text{coolant, out}}\Big)$ $-T_{\text{ref}}\Big(s_{\text{coolant, in}}-s_{\text{coolant, out}}\Big)\Big]$
Control of airborne contaminants/ filter	$\dot{m}_{\text{air, in}}\Big(h_{\text{air, in}}-T_{\text{ref}}s_{\text{air, in}}\Big)$ $-\Big(\dot{m}_{\text{air, in}}-\dot{m}_{\text{deposited}}\Big)\Big(h_{\text{air, out}}-T_{\text{ref}}s_{\text{air, out}}\Big)$	$\Big(\dot{m}_{\text{air, in}}-\dot{m}_{\text{deposited}}\Big)\Big(h_{\text{air, out}}-T_{\text{ref}}s_{\text{air, out}}\Big)$ $/\dot{m}_{\text{air in}}\Big(h_{\text{air, in}}-T_{\text{ref}}s_{\text{air, in}}\Big)$
Water separator module	$\dot{m}_{\text{H}_2\text{O}}\Big[\Big(h_{\text{H}_2\text{O, in}}-h_{\text{H}_2\text{O, out}}\Big)-T_{\text{ref}}\Big(s_{\text{H}_2\text{O, in}}-s_{\text{H}_2\text{O, out}}\Big)\Big]$	$\dot{m}_{\text{H}_2\text{O, out}}\Big(h_{\text{H}_2\text{O, out}}-T_{\text{ref}}s_{H_2\text{O, out}}\Big)$ $/\dot{m}_{\text{H}_2\text{O, in}}\Big(h_{\text{H}_2\text{O, in}}-T_{\text{ref}}s_{\text{H}_2\text{O, in}}\Big)$
THC (overall)	$\sum\dot{X}_{\text{des, THC components}}$	$1-\dfrac{\dot{X}_{\text{des, THC}}}{\dot{X}_{\text{input, THC}}}$

Table 11. Exergy destruction calculations for the WRM.

Component	Exergy Destruction	Exergy Efficiency
Water processor (WP)	$\dot{W}_{\text{WP}} - \left(1-\frac{T_{\text{ref}}}{T_{\text{VRA}}}\right)\dot{Q}_{\text{VRA}}$ $+\dot{m}_{\text{H}_2\text{O}}\big(\big(h_{\text{H}_2\text{O, in}} - h_{\text{H}_2\text{O, out}}\big)$ $-T_{\text{ref}}\big(s_{\text{H}_2\text{O, in}} - s_{\text{H}_2\text{O, out}}\big)\big)$ $+\dot{m}_{\text{O}_2}\big(\big(h_{\text{O}_2\text{, in}} - h_{\text{O}_2\text{, out}}\big)$ $-T_{\text{cabin}}\big(s_{\text{O}_2\text{, in}} - s_{\text{O}_2\text{, out}}\big)\big)$	$1-\frac{\dot{X}_{\text{des, WP}}}{-\dot{W}_{\text{actual, WP}}}$
Urine processor (UP)	$\dot{W}_{\text{UP}} + \dot{m}_{\text{urine}}\big(\big(-h_{\text{urine, in}} - h_{\text{urine, out}}\big)$ $-T_{\text{ref}}\big(s_{\text{urine, in}} - s_{\text{urine, out}}\big)\big)$ $+\dot{m}_{\text{coolant}}\big(\big(h_{\text{coolant, in}} - h_{\text{coolant, out}}\big)$ $-T_{\text{ref}}\big(s_{\text{coolant, in}} - s_{\text{coolant, out}}\big)\big)$	$1-\frac{\dot{X}_{\text{des, UP}}}{-\dot{W}_{\text{actual, UP}}}$
Vent	$-\left(1-\frac{T_{\text{ref}}}{T_{\text{heater}}}\right)\dot{Q}_{\text{vent}}$ $+\dot{m}_{\text{H}_2\text{O gas}}\big(\big(h_{\text{H}_2\text{O gas, in}} - h_{\text{H}_2\text{O gas, out}}\big)$ $-T_{\text{ref}}\big(s_{\text{H}_2\text{O gas, in}} - s_{\text{H}_2\text{O gas, out}}\big)\big)$	$\Big(-\left(1-\frac{T_{\text{ref}}}{T_{\text{heater}}}\right)\dot{Q}_{\text{vent}}$ $+\dot{m}_{\text{H}_2\text{O gas}}\big(h_{\text{H}_2\text{O gas, out}} - T_{\text{ref}}\, s_{\text{H}_2\text{O gas, out}}\big)\Big)$ $\Big/\Big(\dot{m}_{\text{H}_2\text{O gas}}\big(h_{\text{H}_2\text{O gas, in}} - T_{\text{ref}}\, s_{\text{H}_2\text{O gas, in}}\big)\Big)$
WRM (overall)	$\dot{X}_{\text{des, UP}} + \dot{X}_{\text{des, WP}}$ $+\dot{X}_{\text{des, vent}} + \dot{X}_{\text{des, other electrical work}}$	$1-\frac{\dot{X}_{\text{des, WRM}}}{\dot{X}_{\text{input, WRM}}}$

Table 12. Exergy destruction calculations for the WM subsystem.

Component	Exergy Destruction	Exergy Efficiency
Commode	$\dot{W}_{\text{fan}}+\dot{W}_{\text{piston}}-\left(1-\frac{T_{\text{ref}}}{T_{\text{waste can}}}\right)\dot{Q}_{\text{waste can}}$ $+\dot{m}_{\text{air}}\left(\left(h_{\text{air, in}}-h_{\text{air, out}}\right)-T_{\text{ref}}\left(s_{\text{air, in}}-s_{\text{air, out}}\right)\right)$	$1-\frac{\dot{X}_{\text{des, commode}}}{\dot{W}_{\text{fan}}+\dot{W}_{\text{piston}}}$
Urinal	$\dot{W}_{\text{fan separator}}-\left(1-\frac{T_{\text{ref}}}{T_{\text{fan separator}}}\right)\dot{Q}_{\text{fan separator}}$ $+\dot{m}_{\text{air}}\left(\left(h_{\text{air, in}}-h_{\text{air, out}}\right)-T_{\text{ref}}\left(s_{\text{air, in}}-s_{\text{air, out}}\right)\right)$ $+\dot{m}_{\text{urine}}\left(\left(h_{\text{urine, in}}-h_{\text{urine, out}}\right)\right.$ $\left.-T_{\text{ref}}\left(s_{\text{urine, in}}-s_{\text{urine, out}}\right)\right)$	$1-\frac{\dot{X}_{\text{des, urinal}}}{\dot{W}_{\text{fan separator}}}$
WM (overall)	$\dot{X}_{\text{des, commode}}+\dot{X}_{\text{des, urinal}}$	$1-\frac{\dot{X}_{\text{des, WM}}}{\dot{W}_{\text{fan separator}}}$

4.2.1.6.2 Atmospheric Entry (Reentry). Entering a planetary atmosphere creates a breaking force on the vehicle due to the atmospheric drag. In cases where the vehicle is not thrusting (such as a crew capsule), the capsule can be considered a control mass (rather than a control volume). Entry into a planetary atmosphere results in body and frictional drag that produces heat on the vehicle surface. This drag results in a slowing of the vehicle velocity (reduction in kinetic energy) and a lowering of altitude (reduction in potential energy). This is represented by balancing the drag force and thermal heating effects with the vehicle's change in kinetic energy and potential energy. This can be written as

$$Q_{\text{aero}}+A_{\text{module}}C_D\left(\rho_{\text{atm, final}}\frac{V^2_{\text{module, final}}}{2}-\rho_{\text{atm, initial}}\frac{V^2_{\text{module, initial}}}{2}\right)-X_{\text{des}}$$

$$=\left(M_{\text{vehicle, initial}}\frac{V^2_{\text{vehicle, initial}}}{2}-M_{\text{vehicle, final}}\frac{V^2_{\text{vehicle, final}}}{2}\right)$$

$$+\left(\frac{GM_E\,M_{\text{vehicle, final}}}{r_{\text{altitude, final}}}-\frac{GM_E\,M_{\text{vehicle, initial}}}{r_{\text{altitude, initial}}}\right), \tag{88}$$

where

Q_{aero} = heat transfer to the vehicle from atmospheric heating
A_{module} = cross-sectional area of vehicle or capsule
ρ_{atm} = planetary atmospheric density.

Note that, in this case, the total amount of heat is transferred to the vehicle. However, the maximum work that could be harvested from this heat is limited to

$$w_{\text{rev}} = \sum_k \left(1 - \frac{T_{\text{ambient}}}{T_{\text{surface}}}\right) Q_k \ . \tag{89}$$

For capsules also used in planetary reentry, the exergy balance for a control mass can be written in rate form as

$$\sum_k \left(1 - \frac{T_{\text{ambient}}}{T_{\text{surface}}}\right) \dot{Q}_k - \dot{X}_{\text{des}} = -\left(\dot{M}_{\text{vehicle, initial}} \frac{V_{\text{initial}}^2}{2} - \dot{M}_{\text{vehicle, final}} \frac{V_{\text{final}}^2}{2}\right)$$

$$-\left(\dot{M}_{\text{vehicle,initial}}\, gr_{\text{altitude,initial}} - \dot{M}_{\text{vehicle,final}}\, gr_{\text{altitude,final}}\right) \ . \tag{90}$$

Note that the ΔKE and ΔPE term sign is reversed for reentry as compared to that for launch vehicle ascent because the vehicle is descending rather than ascending. This equation considers any ablative mass loss during atmospheric reentry and the ablation rates for the capsule heat shields and other exterior structural components are important inputs to this equation (represented in the change in the vehicle mass).

4.2.1.6.3 Service Module Exergy. For a service module (SM), equation (91) in the next section provides the basic exergy balance for the attitude control and propulsion components. The ECLSS storage components should be contained in the ECLSS model (and designated if the component is in the SM).

4.2.1.7 Spacecraft Bus Exergy. Spacecraft typically consist of several physically different instruments integrated on a single spacecraft bus. The bus provides a variety of capabilities including translation (i.e., propulsion) and maneuvering, electrical power, communication, data processing and storage, and thermal management. Each of the instruments individually may have very different integrating physics based on optics (e.g., telescopes, interferometers), lower frequency electromagnetic waves (i.e., radar), mechanical systems (i.e., gyroscopes), fluid physics, or other physical laws. While each of the instruments have a different integrating physics, they are all integrated through the spacecraft bus with the shared capabilities and interact with each other primarily through this bus. Therefore, the spacecraft bus provides the integrating module, and the associated integrating physics of the bus is also the integrating physics of the system.

Looking at the capabilities provided by the spacecraft bus, these are primarily thermodynamic services. Data are the one capability that is not thermodynamic but the electrical characteristics of the data bus are thermodynamic, representing electrical work and providing the integration of the data system electrical work. Thus, the spacecraft bus is a thermodynamic system characterized by thermodynamic exergy.

Propulsion is a spacecraft bus capability, and the spacecraft exergy balance can be started based on the exergy balance defined for a launch vehicle in equation (24):

$$\Delta m_{\text{propellant}}\left(h_{\text{prop}}+\frac{V_e^2}{2}+\frac{GM_E}{r_{\text{altitude, initial}}}\right)-X_{\text{des}}=\Bigg(M_{\text{vehicle, final}}\frac{V^2_{\text{vehicle, final}}}{2}$$

$$-M_{\text{vehicle, initial}}\frac{V^2_{\text{vehicle, initial}}}{2}\Bigg)+\left(\frac{GxM_E\,M_{\text{vehicle, initial}}}{r_{\text{altitude, initial}}}-\frac{GxM_E\,M_{\text{vehicle, final}}}{r_{\text{altitude, final}}}\right). \quad (91)$$

An important capability for spacecraft is attitude control, defined by the kinetic energy of a rotating body as

$$K=\frac{1}{2}M_{\text{sat}}I_c\omega^2+\frac{1}{2}M_{\text{sat}}V^2_{\text{veh}} \quad (92)$$

where

I_c = moment of inertia about the center of the rotating satellite
ω = angular velocity of the satellite.

So, rewriting the right-hand side of equation (91) yields:

$$\Delta m_{\text{propellant, engine}}\left(h_{\text{prop, engine}}+\frac{V^2_{e,\text{ engine}}}{2}+\frac{GM_E}{r_{\text{altitude, initial}}}\right)$$

$$+\Delta m_{\text{propellant, thruster}}\left(h_{\text{prop, thruster}}+\frac{V^2_{e,\text{ thruster}}}{2}+\frac{GM_E}{r_{\text{altitude, initial}}}\right)-X_{\text{des}}$$

$$=\left(M_{\text{vehicle, final}}\left(\frac{I_{c,\text{ final}}\,\omega^2_{\text{vehicle, final}}}{2}+\frac{V^2_{\text{vehicle, final}}}{2}\right)-M_{\text{vehicle, initial}}\left(\frac{I_{c,\text{ initial}}\,\omega^2_{\text{vehicle, initial}}}{2}\right.\right.$$

$$\left.\left.+\frac{V^2_{\text{vehicle, initial}}}{2}\right)\right)+\left(\frac{GxM_E\,M_{\text{vehicle, initial}}}{r_{\text{altitude, initial}}}-\frac{GxM_E\,M_{\text{vehicle, final}}}{r_{\text{altitude, final}}}\right), \quad (93)$$

where $\omega_{\text{vehicle}} = \dfrac{\left\| 2d \times \Delta m_{\text{propellant, thruster}} V_{e,\text{ thruster}} \right\|}{I_c}$ (where d= distance of thruster from center of mass) and $Th\Delta t = \Delta\, m_{\text{propellant, thruster}} V_{e,\text{thruster}}$ (where Th = thrust).

Now, scientific instruments require electrical power and thermal control. The electrical work done can be written as[35]

$$W_{SE,\text{ generated}} = P_{\text{bus}} - X_{\text{des, electric}} = V_{\text{bus}} I_{\text{bus}} \cos(\theta) - X_{\text{des, electric}}$$

$$= \sum\nolimits_{\text{battery}} P_{\text{electric, stored}} + \sum\nolimits_{\text{instruments}} P_{\text{electric, used}} - X_{\text{des, electric}}\ , \tag{94}$$

where

$\theta =$ solar array angle to the Sun and

$$X_{\text{des, electric}} = \left(1-\eta_{\text{cover glass}}\right)\left(1-\eta_{\text{cell mismatch}}\right)\left(1-\eta_{\text{parameter calibration}}\right)$$

$$\times \left(\eta_{\text{UV, micrometeorite}}\right)^{\text{years}} I_{\text{solar}}\ , \tag{95}$$

where

$\eta_{\text{cover glass}}$ = cover glass efficiency losses
$\eta_{\text{cell mismatch}}$ = solar cell efficiency losses
$\eta_{\text{parameter calibration}}$ = system calibration efficiency losses
$\eta_{\text{UV, micrometeorite}}$ = efficiency degradation losses by years in flight
I_{solar} = solar intensity variation.

The thermal work done can be written assuming a radiator system pointing to the background of space (i.e., not accounting for the radiator having a planetary body view) as

$$\sigma Ae\left(T^4_{\text{radiator}} - T^4_{\text{space}}\right) - X_{\text{des, thermal}} = \left(1 - \frac{T_{\text{space}}}{T_{\text{fluid}}}\right)\dot{Q}_{\text{fluid}} = \sum \left(1 - \frac{T_{\text{fluid}}}{T_{\text{instrument}}}\right)\dot{Q}_{\text{instrument}}\ . \tag{96}$$

Combing these relationships together, equations (93)–(96) yield the final spacecraft bus exergy balance as

$$
\begin{aligned}
&\Delta m_{\text{propellant, engine}} \left(h_{\text{prop, engine}} + \frac{V_{e,\text{ engine}}^{2}}{2} + \frac{GM_E}{r_{\text{altitude, initial}}} \right) \\
&\quad + \Delta m_{\text{propellant, thruster}} \left(h_{\text{prop, thruster}} + \frac{V_{e,\text{ thruster}}^{2}}{2} + \frac{GM_E}{r_{\text{altitude, initial}}} \right) \\
&\quad + \sum\nolimits_t \left(\sigma A e \left(T_{\text{radiator}}^{4} - T_{\text{space}}^{4} \right) + V_{\text{bus}} I_{\text{bus}} \cos(\theta) \right) \Delta t - X_{\text{des}} \\
&= \left(M_{\text{vehicle, final}} \left(\frac{I_{c,\text{ initial}}\, \omega_{\text{vehicle, initial}}^{2}}{2} + \frac{V_{\text{vehicle, final}}^{2}}{2} \right) - M_{\text{vehicle, initial}} \left(\frac{I_{c,\text{ initial}}\, \omega_{\text{vehicle, initial}}^{2}}{2} \right.\right. \\
&\quad \left.\left. + \frac{V_{\text{vehicle, initial}}^{2}}{2} \right) \right) + \left(\frac{GxM_E\, M_{\text{vehicle, initial}}}{r_{\text{altitude, initial}}} - \frac{GxM_E\, M_{\text{vehicle, final}}}{r_{\text{altitude, final}}} \right) \\
&\quad + \left(\sum \left(1 - \frac{T_{\text{fluid}}}{T_{\text{instrument}}} \right) Q_{\text{instrument}} + \sum\nolimits_{\text{battery}} + P_{\text{electric, stored}} + \sum\nolimits_{\text{instruments}} P_{\text{electric, used}} \right) \Delta t \ . \qquad (97)
\end{aligned}
$$

Using the spacecraft bus integrating physics as the integrating physics of the complete spacecraft system, the individual instruments can now be integrated with this to have a full view of the system. An example of this is provided in the description of the optical transfer function (OTF) in section 4.2.2.

4.2.1.8 Other Application of Thermodynamic Exergy. There are other applications or potential applications of thermodynamic exergy analysis not discussed in this publication. Electrical power plants are also governed by thermodynamic exergy. Much work has been done on this application in Europe and can be applied to various electrical power generation methods used in space applications.

Sea vessels are also thermodynamic systems that perform a variety of functions. Cargo transport, passenger transport, weapons system platforms, aircraft operations platforms, and subsurface operations are examples of these functions. These bring hydrodynamics into the exergy balance relationship.

4.2.2 Optical Transfer Function

Optical systems are directly affected by thermodynamic effects such as system exergy; however, this is not their primary integrating physics. Optical systems are primarily driven by image quality. Translating the object to the image plane requires highly precise geometric relationships in the imaging systems, including pointing and control. The aberrations that occur in these systems are a function of the thermodynamics of the platform, the environment (e.g., atmosphere), and their manufacture. An example of some of these effects on the Hubble space telescope (HST) are described in figure 13. These effects are captured in the OTF which represents the object, light propagation paths, optical system characteristics (i.e., manufactured materials and shape) and tolerances, and the final image. The atmospheric effects, when applicable for the system, can be captured in the propagation medium relationships while the thermodynamic, mechanical (e.g., vibration), and manufacturing tolerance effects are captured in the spatial filters representing the optical components, stops, and apertures.[36]

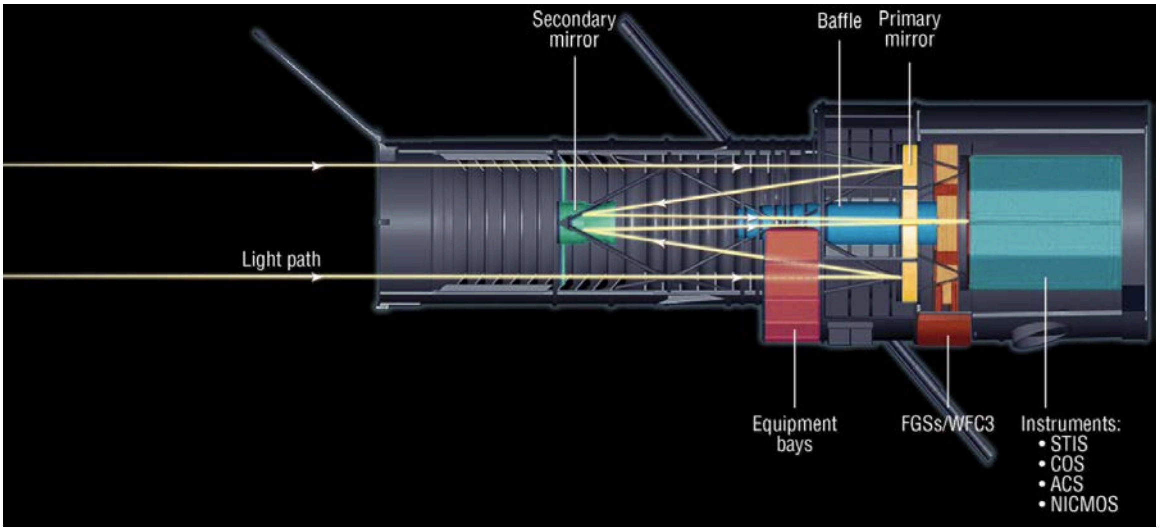


Figure 13. Hubble space telescope image quality effects.

In these cases, image clarity (whether imaging or radiometric, coherent or incoherent light) is driven by geometrical properties of the system. Geometric ray tracing is often used in the design of these system with various tolerances added throughout the geometrical surfaces. While this produces a good analysis of the image given the tolerances, it does not provide an easy means to see the dynamic interaction or provide a fully integrated system equation. This is addressed through the wave optics approach using the OTF. For optical systems, the OTF integrates and describes all the differing aspects (e.g., mechanical, thermal, contamination) leading to acceptable image quality and image aberrations. Thermodynamics, in this case, is an input to the system, but not the systems integrating physics. With this system understanding based on the OTF, the systems engineer is then able to conduct meaningful system analysis and system optimization as the system design progresses. Equation (98) gives the general form of the OTF equation and the definition of variables in these equations:

$$\psi_{\text{image}} = \iint_{-\infty}^{\infty} \psi_{\text{obj}}\, s_f\, dxdy \ , \tag{98}$$

where

ψ_{obj} = object wave function
ψ_{image} = image wave function
s_f = spatial filter (incorporates pupil function(s), propagation distances, etc.).

To examine this approach in more detail, assume a Schmidt-Cassagrain telescope design, similar to HST, shown in figure 14.

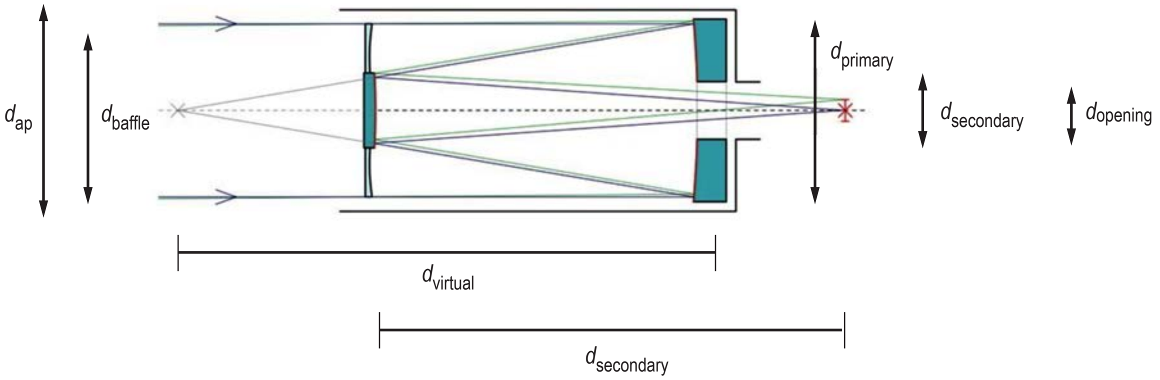


Figure 14. Schmidt-Cassagrain telescope design.

The light propagated from the object to the primary mirror through a distance, d_{obj} (which can be very long). The optical mirror surface produces a Fourier transform of the propagating wave. Thus, after the primary mirror,

$$\iint_{-\infty}^{\infty} \psi_{\text{obj}}\, s_f\, dxdy = \iint_{-\infty}^{\infty} \psi_{\text{obj}}\, e^{j\left(\frac{k_0}{2f_1}\right)\left(x^2+y^2\right)} \left(\text{circ}\left(\frac{x+\Delta x+\delta x}{D_1/2},\ \frac{y+\Delta y+\delta y}{D_1/2} \right) - \text{circ}\left(\frac{x+\Delta x+\delta x}{D_2/2},\ \frac{y+\Delta y+\delta y}{D_2/2} \right) \right) dxdy \ . \tag{99}$$

Note that this relationship has all the effects of the system on the propagating wave, where optical characteristics, including manufacturing aberrations,

$$D_1 = d_{\text{primary}} = \text{diameter of primary mirror,} \tag{100}$$

$$D_2 = d_{\text{secondary}} = \text{diameter of secondary mirror,} \tag{101}$$

and

$$f_1 = -\frac{R}{2} = -\frac{\sqrt{(x+\Delta x+\delta x)^2+(y+\Delta y+\delta y)^2(z+\Delta z+\delta z)^2}}{2} \ , \tag{102}$$

where

f_1 = focal length of the primary mirror
R = mirror radius of curvature.

- Thermal Characteristics—The shape of the optical surface is changed based on the material properties of the mirror and the thermal paths to the mirror. The change in mirror shape is defined through the thermal coefficient of expansion relationship as

$$\Delta x = \alpha x \Delta T,\ \Delta y = \alpha y \Delta T, \text{ and } \Delta z = \alpha z \Delta T \ \ , \tag{103}$$

the effect of the thermal environment on the expansion of mirror in the *x*, *y*, and *z* directions, respectively, where

α = linear coefficient of expansion
ΔT = temperature difference across the mirror.

- Structural Dynamics Characteristic—Structural dynamics lead to vibrations which modulate the mirror surface. These vibrations are found as the solution to Hooke's Law,[37] δx, δy, δz, which provide the structural dynamics motion accounting for vibration. The spacecraft structural characteristics define the particular solution (i.e., overdamped, critically damped, or underdamped) that should be applied to the optical train:

$$[M]\left[\ddot{U}\right]+[C]\left[\dot{U}\right]+[k][U]=0 \tag{104}$$

– $C^2 > 4$ Mk overdamped:

$$\delta x = c_1 e^{-\left(\frac{C}{2M}-\frac{1}{2M}\sqrt{C^2-4Mk}\right)t} + c_2 e^{-\left(\frac{C}{2M}+\frac{1}{2M}\sqrt{C^2-4Mk}\right)t} \tag{105}$$

– $C^2 = 4$ Mk critically damped:

$$\delta x = \left(c_1 + c_2\right) e^{-\left(\frac{C}{2M}\right)t} \tag{106}$$

– $C^2 < 4$ Mk underdamped:

$$\delta x = c_3 e^{-\left(\frac{c}{2M}\right)t} \cos\left(\sqrt{4Mk - C^2}\,t - \varphi\right) \tag{107}$$

$$\tan(\varphi) = \frac{x'(0)}{x(0)\sqrt{\frac{k}{M}}} \tag{108}$$

$$c_3^2 = \sqrt{x(0)^2 + \frac{M}{k}x'(0)^2} \ , \tag{109}$$

where

[U] = [δx, δy, δz] is the structural displacement vector
[M] = mass matrix
[C] = damping coefficient matrix
[k] = stiffness matrix
[F] = force vector.

- Pointing (i.e., spacecraft translational and rotational stability) Characteristics—The pointing of the system is related to error in the object wave front. These can be created by the spacecraft motion or the accuracy which the spacecraft aligns the optical axis with the object. This also includes maintaining the object within the optical system field of view and maintaining the distance from the object within the optical system depth of focus. Starting with the object wave function,

$$\psi_{\text{obj}} = \psi_{\text{obj}}\left(x_0 + \epsilon x,\ y_0 + \epsilon y\right) \ , \tag{110}$$

where x_0 and y_0 are the intended center of the object.

The Abbe Condition defines the imaging resolution where,

$$\epsilon x = 1.22\,\lambda_0 \left(\frac{f_1}{d_0 + \epsilon z + \omega_y \Delta t}\right) + v_x \Delta t + \omega_y \Delta t \tag{111}$$

and

$$\epsilon y = 1.22\,\lambda_0 \left(\frac{f_1}{d_0 + \epsilon z + \omega_x \Delta t}\right) + v_y \Delta t + \omega_x \Delta t \ , \tag{112}$$

where

v = spacecraft translational velocity
ω = spacecraft rotational velocity (note rotation around an axis creates error in the cross terms)
Δt = time for the optical observation of the object.

If an active pointing system is used on the platform, then the motion of this platform can also be incorporated into the OTF. Thus, the OTF represents all surfaces and system effects on the object wave front as it propagates from the object to the image plane. This demonstrates the integrating physics nature of the OTF for an optical system.

4.2.3 Design Analysis Cycle Application

Design analysis cycles (DACs) are a key systems engineering activity during system design. DACs involve all of the relevant engineering disciplines for the system in a coordinated system design effort. One challenge with the DAC is to provide consistent guidance across all of the engineering disciplines such that they can run their particular discipline models to mature the design. Another challenge is to have systems engineering be able to integrate the results into a 'closed' or operable system design. A final challenge is to reduce the number of iterations necessary to achieve a 'closed' design. This is truly an engineering integration activity and involves the integration of results from tens to hundreds of mathematical relationships.

The key to approaching this problem in a consistent and efficient manner is to understand the system integrating physics (or logic for logic-based systems). For a particular system type, there is an integrating physics that ties all other physics relationships together, forming the system integrating perspective. This integration relationship is particular to the system type as discussed in section 4.2.

By applying the appropriate integrating physics relationship (i.e., the integrating perspective), the system engineer can quickly identify the key system parameters from each discipline that contribute to the system integration. The integrating physics also provides a relation to balance the system parameters with a clearer measure of the integrated system balance. This allows clear guidance to be provided to the discipline engineering teams at the beginning of the DAC, knowing that the initial configuration for the DAC is balanced. The systems integrating physics also provide more clear direction to balance the integrated system performance, reducing the amount of iterations necessary to find a closed solution. At the completion of the DAC, this integrating relationship provides the integration approach with the ability to confirm that the system balances (i.e., closes). This provides the systems engineer with a powerful engineering integration tool and improves the DAC by reducing or eliminating time the discipline engineers spend on configuration points that are not balanced (i.e., do not close). This leads to a reduced number of DAC iterations necessary to find a balanced solution.[26]

In addition, SAMs provide a broader testing of the software algorithms than real-time software testing can afford. Thus, software testing can be reduced to those necessary to confirm hardware integration and validate SAM results. This provides more comprehensive software algorithm testing and a reduction in the amount of time needed for software testing.

Note that the application of the system integrating physics and SAM complements and makes use of the design work done by other engineering disciplines. As such, the systems design and integration is dependent on the discipline engineering designs, and not an independent effort as the system progresses through the system lifecycle.

4.3 Engineering Statistics

Systems engineering includes the knowledge of the uncertainties of the system design and operation, as well as the sensitivities to various effects and uncertainties. Statistical thinking is a key aspect of systems engineering that allows individuals to make inferences under conditions of data variability and uncertainty. Three types of statistical thinking may be employed in systems engineering: Information Theoretic, Bayesian Inference, and Frequentist Inference.

(1) Information Theoretic—The information theoretic approaches make use of maximum likelihood estimates to statistically evaluate the information provided by a model (i.e., a set of parameters). Information theoretic is a set of mathematics that includes information theory as originally proposed by Shannon and Weaver (and from which this class of methods derives its name), information entropy, Boltzman entropy, multinomial probability (i.e., logit), and Fisher Information. Figure 15 shows the information theoretic relationships. These techniques provide a comprehensive set of measures of information important in systems engineering.

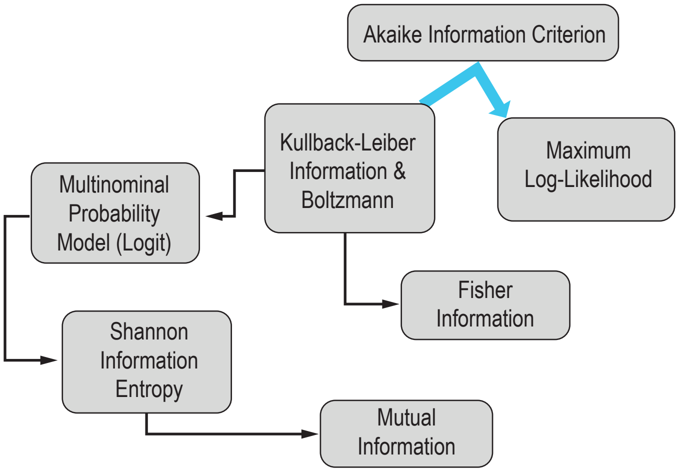

Akaike Information Criterion
Kullback-Leiber Information & Boltzmann
Maximum Log-Likelihood
Multinominal Probability Model (Logit)
Fisher Information
Shannon Information Entropy
Mutual Information

Figure 15. Information theoretic relationships.

(2) Bayesian Inference—The Bayesian inference approach takes a probabilistic view of the unknown quantity (i.e., parameter). Bayesian inference begins with the prior distribution of the parameter (i.e., before the data are seen), then the probability of the prior distribution is induced from the posterior distribution of the data. Typically, there is a description of the posterior distribution of the parameter (i.e., mean and quantiles). The highest posterior density intervals are indicative of the highest posterior probability. In a Bayesian 95% interval, the percentage indicates that there is a 95% probability that the parameter is contained in the interval. In the Bayesian approach,

prior information about the parameter distribution is required. There are several applications of the Bayesian approach including probabilistic risk analysis (PRA), expert systems, and pattern recognition. Applications of particular importance to systems engineers are the PRA, and the mathematical representation of beliefs in Decision Theory approaches.

(3) Frequentist Inference—The frequentist inference approach to statistics is based on sampling theory in which random samples of data are taken in a process to ascertain the underlying parameter of interest. Two primary assumptions are that the process is repeatable and that the underlying parameter remains constant. Often, when individuals talk about statistics they are referring to hypothesis testing and the construction of confidence intervals which take a frequentist approach. In this approach null, and alternative hypotheses are constructed, the type 1 error that the researcher is willing to accept is selected (i.e., the probability of rejecting the null hypothesis when it is true), the parameter is estimated, and the hypothesis is evaluated and/or a confidence interval is constructed. In this approach, when constructing a 95% confidence interval, the percentage indicates that 95% of the confidence intervals constructed this way contain the true value. In the frequentist approach, there is no requirement for prior knowledge about the parameter of interest. The parameter value is estimated from the sample data. There are numerous applications of the frequentist approach that are of value to the systems engineer; the two most prominent are design of experiments and regression analysis, both of which can be applied throughout the lifecycle but are of importance in system testing, verification, and validation.

4.3.1 Information Theoretic Statistics

Kullback and Leibler developed the concept of the ‘information discriminator’ in evaluating models for the National Security Agency (NSA) in the 1950s. The information discriminator was expanded by Akaike in the 1970s to form the Akaike information criteria (AIC).[22] The Kullbak-Liebler (K-L) information discriminator is a more general form of information entropy initially defined by Shannon.[38] This method is very powerful for comparing different system models and providing a measure of the information difference (i.e., the information lost) between the system model and the physical truth of the system. This discriminator is a scalar difference and is a very powerful comparison tool.

The K-L information discriminator makes use of the maximum likelihood estimate (MLE), first developed by Sir Ronald Fisher in 1922.[39] MLE is a statistical analysis of how well a specific system model fits the given system data (i.e., system measurement data, system simulation data, system theoretical model output data). The likelihood function is written as where ‘system’ is the specific system physical truth, ‘data’ are the system data against which the model is evaluated, and θ is the set of data parameters in the system model under evaluation. The likelihood of the individual data parameters, x_i, fitting the system physical truth is the product of the individual data element likelihood functions written as

$$\mathcal{L}\left(\theta \mid x, \text{system}\right) = \prod\nolimits_i \mathcal{L}_i\left(\theta \mid x_i, \text{system}\right) \ . \tag{113}$$

To simplify the product to a summation, one can use the logarithmic

$$\ln\left(\mathcal{L}\left(\theta \mid x, \text{system}\right)\right) = \sum\nolimits_i \ln\left(\mathcal{L}_i\left(\theta \mid x_i, \text{system}\right)\right) . \tag{114}$$

The maximum likelihood estimate (MLE) is the maximum of the likelihood function and can be found by applying the first and second derivatives test which generally requires numerical analysis.

The information discriminator of the MLE from the system physical truth is found by subtracting the logarithm of the model estimate of the system from the logarithm of the system physical truth and finding the expected value of this difference. In continuous form, this is written as

$$I(f,g) = \int f(x)\log\left(\frac{f(x)}{g(x \mid \theta}\right) dx \ , \tag{115}$$

or in discrete form as

$$I(f,g) = \sum f\left(x_i\right)\log\left(\frac{f\left(x_i\right)}{g(x_i \mid \theta}\right) \ , \tag{116}$$

where $g(x \mid \theta)$ is the model derived from data x (x_i in discrete form) and parameter set θ. This gives the measure of the distance between the system physical truth, $f(x)$ (or $f(x_i)$ in discrete form) and the system model.

The system physical truth is not easily determined and is often unknown or is not accurately known (hence, the need for a model of the system, in many cases). This leaves $f(x)$ (or $f(x_i)$ as an unknown. Fortunately, this can be corrected by solving the integral equation and recognizing that, for a given set of models being considered to represent the system physical truth, the $\int f(x)\log\left(f(x)\right)dx$ term integrates to a constant (or $\sum f\left(x_i\right)\log\left(f\left(x_i\right)\right)$ converges to a constant) for all the model comparisons. This is shown by

$$I(f,g) = \int f(x)\log\left(f(x)\right)dx - \int f(x)\log\left(g(x \mid \theta)\right)dx = C - \int f(x)\log\left(g(x \mid \theta)\right)dx \ . \tag{117}$$

Thus, in the K-L information discriminator, physical truth is a constant, and while unknown, the value of $f(x)$ is simply an unknown constant or bias of the scalar information discriminator $I(f,g)$. The relative measure of the information discriminator in evaluating a set of system models against the same system data set is unaffected and a relative comparison of these models is provided. Equation (117) can be rearranged to more clearly show this as

$$I(f,g) - C = -E_f\left(\log\left(g(x \mid \theta)\right)\right) \ , \tag{118}$$

where E_f is the expectation of the logarithm of the model. This information discriminator has also been shown to be equivalent to the MLE minus K, the number of parameters in the model plus the variance parameter. This is shown as

$$\ln\left(\mathcal{L}\left(\theta\middle|x,\text{model}\right)\right)-K=C-E_f\left(\log\left(g\left(x\mid\theta\right)\right)\right)=I\left(f,g\right)\ . \tag{119}$$

Note that K is the degrees of freedom in the model including the estimate of the variance parameter, s^2. For discrete data, the variance is determined using the mean squared error (MSE).

In applying this equation, the data, x or x_i, are viewed as an approximation of the system data allowing for estimation, measurement, and/or communication errors in the data representation. Adjustments to correct the data for this are not necessary in a preprocessing manner but are accounted for in the model representation of data. Kalman filtering is an excellent example, as measurement errors are explicitly contained as a parameter in the model.

Many statistical methods use the Fisher information matrix to understand the amount of information provided by a given model. This matrix is the matrix of second partial derivatives of the model, defining information as the amount of structure (i.e., second derivatives, not zero) in the model. This leads to an *s*-curve type representation of the number of parameters needed, continuously increasing with each parameter added and converging at infinity. The use of the method to determine the number of parameters needed by a model to represent the system to a desired accuracy is difficult to locate. This is often a general point on the inflection point of the curve where the best value is subjectively defined.

The K-L information discriminator defines information as the number of meaningful parameters describing the system from the available system data set. If there are too few parameters, then the data are not well represented. Similarly, if there are too many parameters for the size of the available data set, then the system has too few data points to provide meaningful information on the system. Between these extremes resides an optimum value that is the best information discriminator for a model of the system given the available system data set.

Akaike recognized this and developed a specific measure, which other researchers referred to as the Akaike Information Criteria (AIC). This measure is based on the K-L information discriminator using the system data as a representation of the physical truth (from which the data are obtained). Akaike developed a criterion that is ‘distribution free’ for assessing the value of information. This means that no ‘truth’ is required as a reference. Indeed, Akaike essentially estimates this truth from the data. It is based upon minimizing the expected value of the K-L discrimination information that seeks to minimize the lost information when comparing a supposed distribution relative to the ‘truth’ as specified by the originating function. Akaike instead used the data as the proxy for the originating function as the expectation. Further, candidate models g_i are assumed that best fit the data y_i in the sense of minimizing the negative log likelihood. Akaike[40] then proved that the maximized log-likelihood is an upwardly biased estimator of the model quality criterion. He, thus, reasoned that an approximately unbiased estimator of the relative, expected K-L discrimination information (KLDI) could be written as

$$\text{AIC} = -2\ln\left(\mathcal{L}\left(\hat{\theta} \mid x, \text{system}\right)\right) + 2K, \tag{120}$$

where the –2 factor allows a fit for some statistical χ^2 problems and $\hat{\theta}$ is the value of θ producing the MLE. This criterion provides the adjustment for the number of parameters in the model against the available system data set, x.

AIC can be corrected (AIC$_c$) to compare models with a relatively small dataset. The typical rule of thumb for the application of AIC is 3K ≤ n, where n is the number of data points in the data set. Thus, the number of data points must be at least three times as large as the number of model parameters. For system models and system data sets that do not meet this condition, AIC$_c$ provides an adjustment to accurately measure the information discriminator. This converges to the AIC values as n approaches 3K. AIC$_c$ is written as

$$\text{AIC}_c = -2\ln\left(\mathcal{L}\left(\hat{\theta} \mid x, \text{model}\right)\right) + 2K\left(\frac{n}{n-K-1}\right) \ . \tag{121}$$

Mean square error (MSE) provides a definition of the model fit from the frequentist inference viewpoint. This can be viewed as the representation of a statistical model of the system and AIC and AIC$_c$ can be written then as

$$\text{AIC} = N\log(\text{MSE}) + 2K \tag{122}$$

and

$$\text{AIC}_c = N\log(\text{MSE}) + 2K\left(\frac{n}{n-K-1}\right) \ , \tag{123}$$

where

$$\text{MSE} = \frac{\text{sum of squares of residuals}}{N_{\text{predicted DOFs}}} \, . \tag{123a}$$

When developing these types of statistical models, $N_{\text{predicted DOFs}}$ are the number of degrees of freedom that the model is predicting. K is then $N_{\text{predicted DOFs}} + 2$ (adding DOFs for the constant term and the estimated variance parameter).

As mentioned above, AIC, based on the K-L information discriminator, is a relative measure of the system model. Thus, the actual AIC or AIC$_c$ value is not important (and contains the unknown constant of the system truth). So, model comparison is performed by calculating the AIC or AIC$_c$ value for each model, finding the smallest value and subtracting this from the other values. This delta value is defined as

$$\Delta_i = \ \text{AIC}_{c,i} - \text{AIC}_{c,\min} \, . \tag{124}$$

Note that the smallest or minimum model AIC will have $\Delta_{\min} = 0$. Comparison of the models is straightforward using the deltas. As a rule of thumb, table 13 provides the deltas that should be considered further.

Models with values of four or less should be considered as relatively good models. Combining these models in a weighted average may provide a better representation of the system. The AIC or AIC$_c$ weight can be used for the weighted averages in these cases. Be aware that weights <0.1 indicate a very low contribution by the model and this model(s) may be best excluded from the averaging.

These deltas can also be used to calculate weights for each function. The weights are defined to sum to 1 for the complete set of models and are given by

$$w_i = \frac{e^{-\frac{1}{2}\Delta i}}{\sum_r e^{-\frac{1}{2}\Delta r}} , \tag{125}$$

with the condition

$$\sum_i w_i = 1 . \tag{126}$$

One can obtain a relative comparison by calculating the Δ_i and w_i. These weights can also be used to evaluate the information value of a given parameter that is contained in more than one model. The weight of each model that contains a parameter, say α, can be summed to produce a parameter weight as

$$\sum_j w_j = w_p . \tag{127}$$

The sum in equation (127) will not sum to 1 unless the parameter is used in all the models. In the case where the parameter is ubiquitous, this method does not provide any information on that parameter. However, for parameters contained in some system models under consideration but not all, then an idea of their contribution can be assessed.

The delta between the best (i.e., lowest value) and associated weightings allow models to also be integrated to produce a composite statistical measure for a system.[41] Using the delta's models whose $\Delta_i \leq 2$ as indicated in table 13 can be averaged to give a composite measure of the system. The weights can be used to create a weighted average of these models. Note that this weighted average does not contain all models in the set and cannot be normalized to 1.

Table 13. System statistical model comparison based on AIC or $\mathrm{AIC_c}$ deltas.

Δ AIC or $\mathrm{AIC_c}$ Value	Indication
0 to 2	The system statistical models are almost equivalent representations of the system
>2 to <4	There are significant contributions contained within the models compared with the best model
4 to 7	There may be some important contributions from different model parameters which should be considered based on system integrating physics
>7 to 10	Less significant differences in parameters. Could be some important parameters in the different models but not as compelling
>10	The model is not comparable to the best model in the set being considered

Having calculated a best model or subset of statistical models for the system, two paths are available. Additional models can be formulated to see if a better understanding of the system is obtained through other parameters or relationships; or, assessment of the system can proceed with best model or weighted average of closely related models. Model selection is based on the parameters used in the selected model (single best of weighted average).

Takeuchi Information Criteria (TIC) is an information theoretic approach that may be considered in comparing system models. TIC uses the trace of the product of the Jacobian matrix with the Fisher information matrix. These matrices require some mathematical effort to construct, and the trace of this product generally converges to the number of parameters, K, used in the AIC adjustment parameter. Thus, AIC is generally the recommended approach over TIC.[42]

4.3.2 Bayesian Inference

Bayesian inference provides a statistical approach based on the probability that what you believe to be the truth (your expected value) is the truth. In essence, an initial guess is performed as to what the results are likely to be, then a sample of results is obtained, and the difference in the initial guess and the sample results provides a measure of improvement in the next guess. The implementation of Bayes method is performed in a probabilistic framework, is typically an iterative process, and is dependent upon the initial guess as well as interpretation of the results. The Bayesian information criteria (BIC) was introduced to provide a standardized metric in order to assist with the implementation of the methodology.

A qualitative visualization of Baye's Rule is provided in order to provide a clearer understanding of the Bayes method. The fundamental precept is that the data provide evidence of improved knowledge relative to an initial assumption. The assumption and the data are related statistically through the likelihood. Assume that an initial guess for the probability, or 'prior' for the distribution of data, is provided as (θ), with parameters θ. For example, the Gaussian or normal probability density function (PDF) is characterized by a location parameter, μ, and a scale parameter, σ, thus $\theta = (\mu, \sigma)$. Then this prior is compared to the set of data represented as y. The likelihood of the data, given the parameters $p(y|\theta)$, provides an estimate of an improved distribution known as the posterior, $p(\theta|y)$. The mathematical relationship of Baye's rule is presented in figure 16.

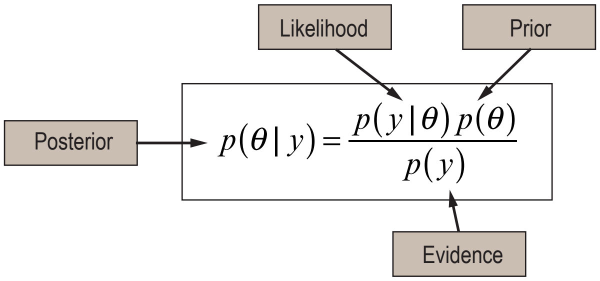

Likelihood
Prior
Posterior
$p(\theta \mid y) = \frac{p(y \mid \theta)\, p(\theta)}{p(y)}$
Evidence

Figure 16. Visualization of Baye's rule.

The BIC makes use of the logarithms of the likelihood for a model with K parameters attempting to fit N observations and is given by

$$\mathrm{BIC} = -2\ln\left(\mathcal{L}\left(\hat{\theta} \mid x, \mathrm{model}\right)\right) + K\ln\left(\frac{N}{2\pi}\right) \ . \tag{128}$$

BIC uses the logarithm of the number of data points as the adjustment factor and is not actually an information theoretic representation because this adjustment is not truly a bias correction like the correction term is for AIC. If the actual true physical representation of the system can be achieved and is in the set of system statistical models, then BIC will always identify this model. This is important but has some limiting assumptions. From a full system perspective, BIC assumes a small number of model parameters (<5). For complex systems, the number of parameters is generally >100 (e.g., a launch vehicle has over 100 state variables in its full system representation). As the number of parameters increases, the results of AIC and BIC become similar. BIC also assumes that all the system statistical models have equal probability prior to application and does not factor in the knowledge of the system which may change the understanding of these probabilities. Thus, the specific system being analyzed and the size of the statistical model needed to represent the system are important factors in determining whether to use BIC or AIC.

4.3.3 Frequentist Inference

Frequentist inference is based on the frequency of occurrence of events. The expected values and other statistical measures are based on how often you could expect a certain event to occur within a set of occurrences (or time). The frequentist approach is not strictly probabilistic like the AIC and BIC methods. In fact, the frequentist approach relies upon the relative number of events that meet a certain prescribed, fixed, criteria set compared to the total observed data set. For example, given a data set of N observations of the data, the number of occurrences (frequency) of a target value, say $y = y_{\mathrm{target}}$, may be represented by $n_{y_{\mathrm{target}}}$ and is given by

$$n_{y_{\mathrm{target}}} = \sum\nolimits_{i=1}^{N} \left\{ y_i = y_{\mathrm{target}} \right\},\ i = 1, \ldots, N \ . \tag{129}$$

The proportion of the number of target values is then determined as

$$P\left(y = y_{\text{target}}\right) = \frac{n_{y_{\text{target}}}}{N}. \tag{130}$$

The actual probability that $y = y_{\text{target}}$, $\mathbb{P}\left(y = y_{\text{target}}\right)$, is statistically only relevant for the total population or infinite set, and is given by

$$\mathbb{P}\left(y = y_{\text{target}}\right) = \lim_{N \to \infty} \frac{n_{y_{\text{target}}}}{N}. \tag{131}$$

One of the major assumptions employed using the frequentist approach is that the proportion is equal to the probability. This is generally not true and may cause some issues with the interpretation of the results, especially when the N is small and the sampled data are not likely to be representative of the population itself. In such cases, one set of samples may be quite different from another with notably different conclusions. For example, there might be three catastrophic events, $n_{y_{\text{target}}} = 3$, out of ten attempts, $N = 10$, and yields a proportion of 0.3 catastrophes in the set. However, this proportion should not be interpreted as a probability of occurrence in any future attempt, but is only an indication of historical events. Sometimes all that is available is this proportion, but its predictive power may be limited, and decisions based upon the proportion rather than the probability assume a higher level of uncertainty risk for that decision.

4.4 System State Variable Modeling

System integration has primarily been a natural language-based set of cross-checking and traceability processes. By contrast, both system analysis and system integration are based on system models that use mathematics, physical laws, and logical reasoning. These system models are constructed from system and environment state variables to enable mathematical and logical representation of the system and the environment.

The use of natural language, while important for system description, is not accurate for specific system integration and prone to communication error and terminology differences between engineering disciplines. Section 5.1 discusses the aspects of terminology in discipline integration. System integration requires the use of system models based on math, physics, and logic which greatly aides in communication and enables a system representation that can represent all the disciplines necessary for the system functions and application. State variables are necessary to achieve these integrated system models and enable mapping between multiple kinds of models about the system. These form the system state basis.

Most systems are a combination of hardware and software functions. These states are not independent in the operation of the system but are coupled. The system states encompass both the hardware states and the software states. The state variables which are shared among the different system functions and subsystems are the system state variables. The state evolution encompassed by system state variables describe these system functions and interactions. System modeling approaches based on these variables provide considerable understanding of the system design. These modeling approaches include the GFT and the State Analysis Model (SAM).

The GFT has been employed by the Space Launch System (SLS) to assess failure detection coverage in an integrated fashion. The GFT is a top-down, formal, hierarchical representation of stakeholder intentions for the system. While in appearance it resembles a traditional functional decomposition, it is far more rigorous and useful due to its use of state variables and associated constraints as the output of functions. The constrained state variables are goals, which when translated into formal statements are requirements. A system concept of operations, when rigorously defined, specifies a sequence of desired or required events, each of which specifies a top-level goal for that portion of the sequence. A GFT can then be constructed to define all the supporting goals and functions needed to achieve each goal or event defined in the concept of operations. The use of state variables ensures that the tree structure is physically and causally valid, which in turn means that the GFT can be employed for a variety of analytical uses. In the GFT, a path from any selected goal or function moving up the tree defines a scenario that can be used for testing. Because any failure that can threaten a top-level goal must threaten either that goal or goals below it in the tree (if there is no redundancy), failure detection coverage can be assessed by placing detections to identify failure to achieve goals along every path that can lead to a top-level goal. Taking the logical complement of the GFT creates the beginnings of the system fault tree, which in turn can be used for safety/hazard analysis or for probabilistic risk assessment.

Another important model emphasizing the use of state variables and their states is SAM. This model builds on the idea that systems are described by the evolution of states encompassed by their state variables. The model is constructed representing the vehicle hardware and software states. Vehicle execution and state transitions can then be modeled and the system evaluated across all subsystem functions for proper sequencing, and expected and unexpected interactions.

4.4.1 Goal Function Tree

The GFT, illustrated in figure 17, is a representation of the system intent contained in the concept of operations.[43] Models of intention define the top-level goals that a prospective design is to fulfill. The models of intention are mapped to design models and drive each other progressively ‘deeper’ to greater detail in a spiral process. As this representation of system intent is a model of functions and goals, the GFT inherently shows the traceability of requirements from higher to lower levels simultaneously in ‘goal space’ and ‘function space’ that does not directly represent the design. Even though the specifics of the design are not directly represented, the rigorous use of state variables ensures that the intention of how the system is to achieve goals through the use of physics is nonetheless modeled.

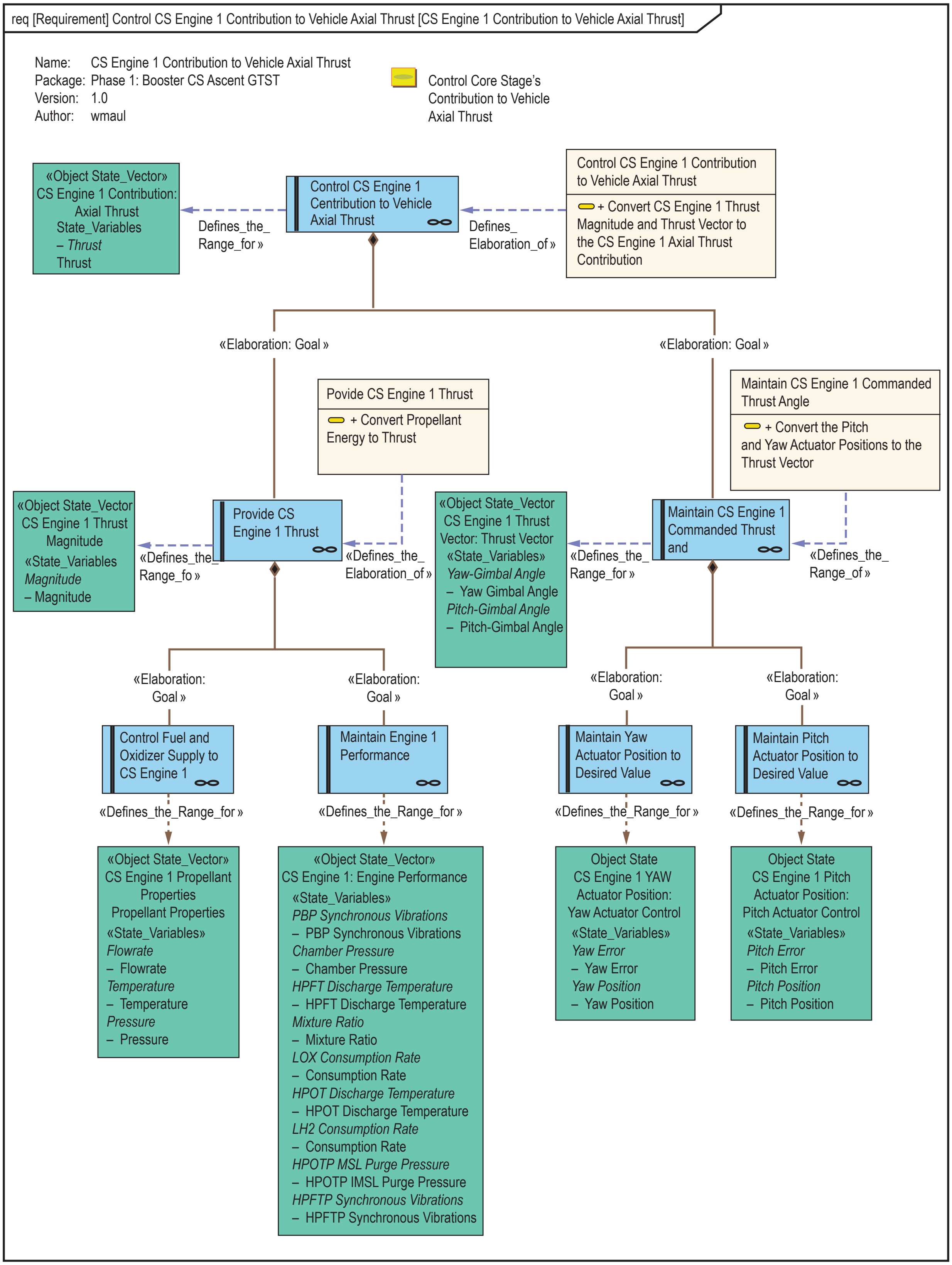

req [Requirement] Control CS Engine 1 Contribution to Vehicle Axial Thrust [CS Engine 1 Contribution to Vehicle Axial Thrust]
Name: CS Engine 1 Contribution to Vehicle Axial Thrust
Package: Phase 1: Booster CS Ascent GTST
Version: 1.0
Author: wmaul
Control Core Stage's Contribution to Vehicle Axial Thrust
«Object State_Vector» CS Engine 1 Contribution: Axial Thrust State_Variables – Thrust Thrust
Defines_the_ Range_for »
Control CS Engine 1 Contribution to Vehicle Axial Thrust
Defines_ Elaboration_of »
Control CS Engine 1 Contribution to Vehicle Axial Thrust
+ Convert CS Engine 1 Thrust Magnitude and Thrust Vector to the CS Engine 1 Axial Thrust Contribution
«Elaboration: Goal »
«Elaboration: Goal »
Povide CS Engine 1 Thrust
+ Convert Propellant Energy to Thrust
Maintain CS Engine 1 Commanded Thrust Angle
+ Convert the Pitch and Yaw Actuator Positions to the Thrust Vector
«Object State_Vector CS Engine 1 Thrust Magnitude «State_Variables Magnitude – Magnitude
«Defines_the_ Range_fo »
Provide CS Engine 1 Thrust
«Defines_the_ Elaboration_of »
«Object State_Vector CS Engine 1 Thrust Vector: Thrust Vector «State_Variables» Yaw-Gimbal Angle – Yaw Gimbal Angle Pitch-Gimbal Angle – Pitch-Gimbal Angle
«Defines_the_ Range_for »
Maintain CS Engine 1 Commanded Thrust and
«Defines_the_ Range_of »
«Elaboration: Goal »
«Elaboration: Goal »
«Elaboration: Goal »
«Elaboration: Goal »
Control Fuel and Oxidizer Supply to CS Engine 1
Maintain Engine 1 Performance
Maintain Yaw Actuator Position to Desired Value
Maintain Pitch Actuator Position to Desired Value
«Defines_the_Range_for »
«Defines_the_Range_for »
«Defines_the_Range_for »
«Defines_the_Range_for »
«Object State_Vector» CS Engine 1 Propellant Properties Propellant Properties «State_Variables» Flowrate – Flowrate Temperature – Temperature Pressure – Pressure
«Object State_Vector» CS Engine 1: Engine Performance «State_Variables» PBP Synchronous Vibrations – PBP Synchronous Vibrations Chamber Pressure – Chamber Pressure HPFT Discharge Temperature – HPFT Discharge Temperature Mixture Ratio – Mixture Ratio LOX Consumption Rate – Consumption Rate HPOT Discharge Temperature – HPOT Discharge Temperature LH2 Consumption Rate – Consumption Rate HPOTP MSL Purge Pressure – HPOTP IMSL Purge Pressure HPFTP Synchronous Vibrations – HPFTP Synchronous Vibrations
Object State CS Engine 1 YAW Actuator Position: Yaw Actuator Control «State_Variables» Yaw Error – Yaw Error Yaw Position – Yaw Position
Object State CS Engine 1 Pitch Actuator Position: Pitch Actuator Control «State_Variables» Pitch Error – Pitch Error Pitch Position – Pitch Position

Figure 17. Goal function tree of a rocket engine control system.

Arguably, the most crucial and common type of traceability in systems engineering practice tracks requirements from the top system level to lower levels in terms of components and of the organizations that build these major components. Requirements (agreements) must be agreed to with organizations, and thus a crucial issue is how to translate hierarchical goals modeled in the GFT to organizations building system components. The process is to map goals and functions represented in the GFT to designs represented in directed graphs, and then to the design organizations, which are typically represented using a hierarchical model. The use of state variables in the intention and design representations facilitates this mapping. The organizational mapping is associated with the assigned responsibilities for the represented system functions.

The concept of operations and the GFT model contain the functions and goals, specified in terms of input and output state variables of functions, and the constraints on the output state variables. Since designs are simply mechanisms to achieve functions, the output state variables of a design mechanism must be the same as the outputs of the function that the mechanism is implementing, and the constraints on the state variables must be similar (though not necessarily identical). Mapping from the concept of operations and GFT to the design is simply a matter of identifying the state variables and constraints with the exact same state variables and with appropriate similar constraints in the directed graph design models. Proper state variable and constraint naming conventions are crucial.

Mapping to organizations requires the use of a physical containment model. The physical containment model uses state variables at the interfaces of the components that define the 'containment regions.' Each containment region is defined by a physical and/or logical boundary in which other components reside. For example, a launch vehicle contains stages, and the stages contain subsystems. Subsystems contain components, and so on. The physical containment model has a strong correlation between the containment components and the organizations that build those components. It is generally true that a single organization is responsible for integrating all subcomponents that exist inside a component. Thus, the existence of subcomponents inside a component can hierarchically mirror the authority structure of the organization that builds the component and subcomponents.

The traceability and allocation of requirements from top to bottom and from function to organizations therefore move from the concept of operations and GFT model to the design models by identification of state variables. The next mapping is from the directed graph design models to the physical containment model, once again through identification of the same state variables that exist both in the directed graph and physical containment models. Finally, the organization model closely mirrors the physical containment model. Both can be hierarchical, allowing mapping from physical design components arranged hierarchically to the organizations that build these components, which are also arranged hierarchically. We thus have complete traceability from function to organizational hierarchy.

The GFT also provides important information that can be used to support testing. Each path moving from the bottom of the tree to the top represents a related set of intended requirements and functions that should be tested. Because every goal is defined by constraints on the relevant function output state variables, these constraints specify the required performance of the function to achieve the goal. These can be extracted from the tree to be used as the required constraints on observed/measured state variables in a test of that set of functions and goals.

4.4.2 Failure and Risk Assessment

By its nature, the GFT is a success-space model, but it can be used to assess failure-related issues such as failure detection coverage against system goals. Design models for any system always begin as representations of the nominal design, but are then frequently used to generate off-nominal models for the system. These models have important relationships that can be exploited to decrease costs and increase consistency of the models.

One example is the GFT, which can be used as the starting point for system fault tree(s). Since every goal in the GFT (and in the concept of operations) is defined as constraints on the output state variables of functions over a certain time span, failure of the goal is simply defined as the state variable being outside this range. Since the GFT is hierarchical, taking its logical complement creates a fault tree. Failure models inherently represent a larger number of behaviors than the GFT success-space models, because some failures create failure effect paths that do not exist in the nominal design.

The first type of failure model is the fault tree, which can be constructed, in part, by taking the logical complement of the success-space GFT as discussed in section 4.4.2. In failure space, OR gates represent the default relationship of the inputs of each node, with AND gates representing redundancy. Because more links between components exist in failure space than success space, there are more tree branches in the fault tree than exist in a success tree like the GFT. These models extend the success tree to incorporate failure-induced interactions not intended in the design. Thus these are a type of model of intention.

Being hierarchical, a fault tree must have some principle or attribute that is the basis for its hierarchical structure. Usually, fault trees are implicitly ‘intentional’ in that they represent ways in which intentions for the system fail to be achieved. The top branches of the fault tree represent failures of top-level goals. Like traditional functional decompositions, fault trees can have the same issues with use of natural language in that it is possible to model the system in ‘nonphysical’ ways. Fault trees that use probability, such as in probabilistic risk assessment, are typically rigorous in the sense that the relationships between failures in the tree are arranged so that they are probabilistically correct. However, this does not guarantee that they are ‘causally correct’ in the sense of the time-arrow of physical effect propagation through the system and its functions. Fault trees can also be constructed independent of risk probabilities. This is typical of fault trees used for safety analyses, in which the purpose is to identify functions that cannot be allowed to fail, and then determining proper ‘controls’ to minimize the possibility of the function failing.

A full exploration of system behaviors thus requires failure-space models, not merely success-space models. The fault tree is larger than the corresponding success tree because the fault tree has all the GFT branches but adds new failure effect branches. Examples of such new failure branches include representations of short circuits and of debris from explosions. In these cases, new electrical and geometrical connections between components are created by the failure that otherwise do not exist.

Once the fault tree is constructed by adding failure paths from the initial complement of the GFT, the fault tree can be used to generate an initial functional failure modes and effects analysis (FMEA) and list of system failure scenarios. The bottom-most node of the fault tree that is assessed as 'failed' is a failure mode for the FMEA, and the effect fields of the FMEA are the compromised functions moving up the tree from a given failure mode. This becomes possible because the GFT, and hence its logical complement fault tree, are always constructed with 'state variable rigor,' which ensures proper physical causality.

Because causality is properly modeled, each path from the bottom or from the middle of the fault tree defines a failure scenario. Therefore, these can be automatically extracted simply by searching and documenting all paths up the tree. These can later be used for system verification. Failure scenarios represent the suite of behaviors represented by a path from any location on the fault tree up the tree until it either meets a redundancy node or continues to the top of the tree.

The same kind of relationships exists between the nominal design directed graph and its logical complement failure space directed graph that also adds failure paths. Failure scenarios are modeled by selecting a node in the failure space directed graph and allowing the failure effects to propagate from node to node through the directed arcs that link them. In general, the directed graph set of failure scenarios should be compared with the hierarchical representations of the fault trees to determine if there are any differences between the two representations. If there are, one or both representations are likely to be deficient or erroneous in some way. Similarly, the success-space directed graph representing nominal connections between components can be compared to the GFT. Finally, the state machine models (discussed in the next section) should generate the same sequence of nominal or failure events through its bottom-up modeling technique.

Failure models can also use directed graphs and related models such as reliability block diagrams (RBDs). These models are not hierarchical, and, in general, are constructed so that components are related to each other in ways that represent the actual component connectivity of the design. Thus, these can be considered as a kind of design model. When modeling in failure space, directed graph design models often represent the propagation of failure effects through the system. RBDs implicitly do the same. For example, when a set of redundant hardware is modeled in an RBD, the existence of a mechanism to 'vote out' a bad string of components or ensure a good string of components is used is assumed, even if not explicitly modeled. The modeling technique assumes that the failure effects generated by a bad string of components will not propagate beyond the (often implicit) voting mechanism.

Once the failure scenarios are constructed, they can be analyzed through a variety of quantitative and qualitative techniques to determine the effectiveness of the detections and responses to failure. These, in turn, feed system-level quantitative estimates of meeting system dependability goals, as the performance of the fault management is one of the key metrics for system dependability.

4.4.3 System State Analysis Model

The system SAM,[44] illustrated in figure 18, is a type of system behavioral model. State machine models have the significant advantage of generating many complex behaviors from many simple models. That is, while many models require analysts to predict and model behaviors directly, state machine representations consist of many simple models of the state transitions of individual components, which can be triggered by the state transitions of other components, with relevant time delays as appropriate. The models are generally driven by an external script that defines how the system is commanded and operated. When environment models are added, the state machine executive executes the component models. This provides the state machine with the ability to model emergent behavior when sufficient fidelity is present to capture interactions between the system functions and with the system environment.

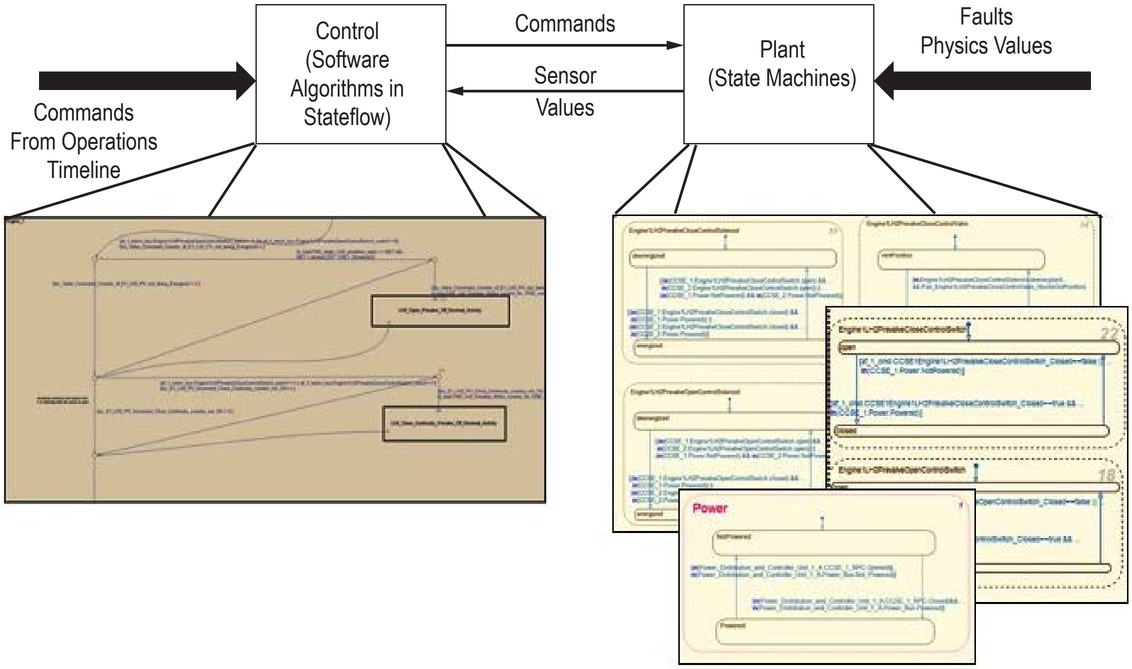


Figure 18. System analysis model structure.

SAMs were originally developed and used to assess software. They expand the software approach to include the entire system and its states. State machine models are built from the bottom up, such that ‘the model’ consists of many simple models, each representing how a single component (which can be hardware, software, or humans) changes output state based on changes to input states. When ‘executed,’ an output from an individual component is tracked by the execution software, which then searches for any other components that use the new output state as an input. For any component that does, then that component is ‘executed’ to determine what output states change based on that input. This process is repeated as often as needed to yield the total set of system state changes across all components over time. Because the state machine model is a very simplified, all-software representation, it can be executed much faster and across many more kinds of situations than more realistic physics-based simulations or tests.

State machine models are useful for assessing the ways in which the system's software and command structures operate and interact with each other and the various hardware, software, and operational components. Being simpler and less expensive to run than 'regular simulations or tests,' these models can be used to explore a much larger set of potential behaviors than regular testing can normally accomplish. These explorations, particularly when enhanced by formal methods to comprehensively model the state combinations, provide a key means of generating and verifying system behavior analytically.[22] System state machines are one of the best types of models to search the system off-nominal space and to search for undesired and unexpected interactions. State machine models create scenarios naturally, simply by inserting a failure behavior (which is a state) and allowing the model to execute to generate the resulting behaviors.

4.5 System Information Flow

Understanding the physics of the system is a key to designing and integrating the system. Part of the system is the information needed to manage and control the system functions. This is derived from the knowledge of the system state variables (see sec. 4.4). State variables define the system's attributes. Some of these attributes are observable and some are monitored to provide measurements of their states to the management and control functions to enable system execution and fault management. Figure 19 illustrates the relationship of the system information to the system physics.

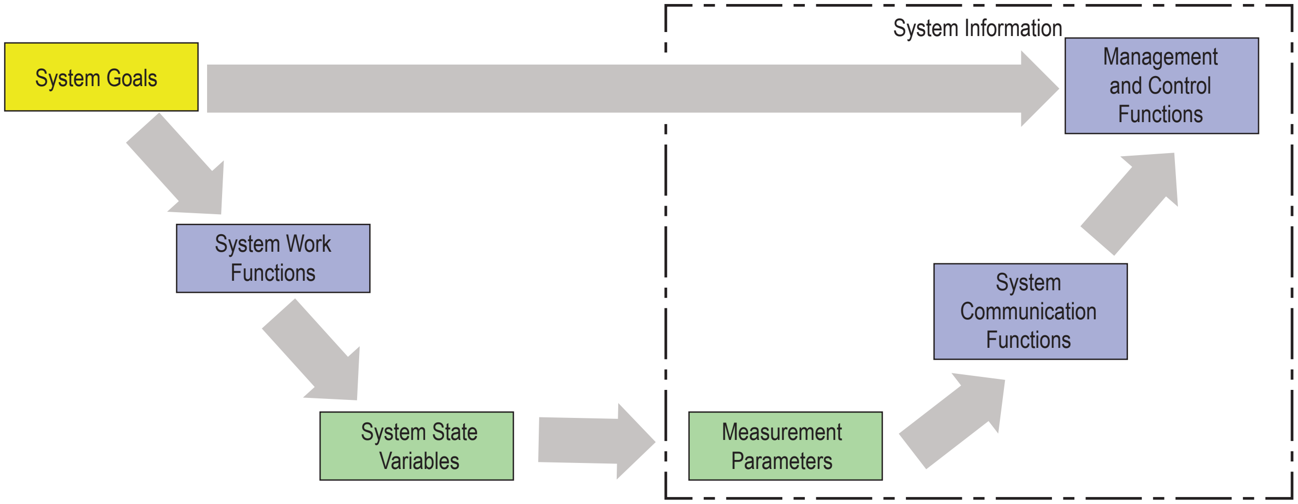


Figure 19. System information flow derived from system state variables.

The flow of the system information is governed by information theory. Information theory addresses information flow in the communication channels, the impact of adding processing nodes, the way in which these nodes are configured, and the way in which information is collected.

Channels (i.e., free space, microwave channels, electrical cables, and optical cables) are not error free, but have many sources of noise that affect the quality of the information flow. These sources of noise include both natural and system-induced electromagnetic interference (EMI).

Natural sources include atmospheric charging and discharging (lightning), triboelectric (skin charging through the atmosphere), solar-induced EMI, cosmic radiation, etc. System-induced environments include electronic transmissions from various system components, system faults which induce unintended electrical signals, mechanical connection chatter, etc. These are generally managed by ensuing that the intended signal levels are greater than the noise. The power levels of the signal and noise are measured as a ratio in the signal-to-noise ratio (SNR). Various coding methods are addressed in information theory that can improve the quality of signals in the communication channels. The quality of the data in coded and uncoded systems is measured in terms of the bit error rate (BER).

Channel SNR degrades with distance from the source of the signal. The length of cables or the distance from the receiver is an important consideration. For cables, there is a limiting distance based on the resistance of the cable over which it can transmit the signal. For long systems (e.g., rockets, ships, buildings), signal power boosters may be necessary to connect these longer distances. Electrical integration must take this into account.

For free space transmissions, the distance between the transmitter and receiver drive the amount of power necessary to maintain a minimal SNR of the signal at the receiver. The power requirements driven by this affect the transmitter size and mass, transmitter power, and correspondingly can drive the system power generation needs. For this reason, satellite transmitters require more power and have more mass than ground station receivers. Free space communications require the system to be in view of the receiving station. Satellite systems often have a better view (the satellites cover a much broader area) but are much further distant. Therefore, systems engineering must be aware of the balance needed between the various free space communication options distances and view.

The number of processing and measurement nodes is also addressed by information theory. The more state variables (parameters) there are in a system, the more information there is about the system, and the higher the systems uncertainty. Adding additional nodes can actually increase the uncertainty in the complete system state as these individual nodes add system state variables. These changes in uncertainty are sometimes apparent in system reliability estimates that should take into account the results of information theory as well as hardware reliability and software error estimates. Information theory provides mathematical tools to define the correct processing configuration based on the system physics work functions and the set of state variables which need to be measured to manage and control the system. Information theory shows that the information about a system is defined by the sum of subsystem or function information. This can be written as

$$I(S) = -\sum_{n} p\left(S_n\right) \log_2 p\left(S_n\right) \ , \tag{132}$$

in which the probability of the state of the *n*th subsystem is given by $p\left(S_n\right)$ and information about the *n*th subsystem is given by $p\left(S_n\right)\log_2 p\left(S_n\right)$.[45]

This system information, *I*(*S*), is also related to the uncertainty, or information entropy, as

$$I(S) = H(T) + H(R) - H(R|T) \ , \tag{133}$$

where

$H(T)$ = entropy in the transmitted information about the system
$H(R)$ = entropy in the received information about the system
$H(R|T)$ = entropy (or uncertainty) of the information received given the information transmitted.

Note that as the amount of information about the system increases (i.e., as subsystems, functions, communication nodes, processing nodes are added), the uncertainty of the information about the system increases, which is given by the relationship:

$$H\left(S_{n+1}\right) \geq H\left(S_n\right) \ , \tag{134}$$

where $S_{n+1} > S_n$.

Adding processing or communication nodes to a system adds millions of possible states (consider a million gate integrated circuit, for example) and corresponding bits to represent these states. Information theory shows that, in general, the uncertainty of a system increases as the system is subdivided. This is the relationship:

$$H(S) \leq H\left(s_1\right) + H\left(s_2\right) + H\left(s_3\right) + H\left(s_4\right) + \cdots + H\left(s_n\right) \ , \tag{135}$$

where s_1, $s_2 \ldots s_n$ represent different subsystems.

Thus, information theory shows that the number of subsystems to accomplish the system intent should be kept as small as possible in equation (134) (hypothesis 2) and that the number of processing nodes should also be minimized in equation (135) (hypothesis 2). This must be balanced with the considerations of communication system view and transmission distance as discussed above.

The other important aspect about system information communication is the bandwidth (or channel capacity) which the system information is transmitted. The larger the bandwidth, the greater and, therefore, quicker information about the system can be relayed. This bandwidth defines the data rate, bits per second (bps), that the communication system operates. Information theory defines the maximum channel capacity as

$$C = \max I[T,R] = \log_2 b \ , \tag{136}$$

where b is the number of bits per transmission symbol.

The bit rate can then be defined as the number of bits transmitted per second by dividing the channel capacity by time:

$$C_t = \frac{c}{t} = \frac{1}{t}\log_2 b \ , \tag{137}$$

where t is the time it takes to transmit the channel symbols.

So, the amount of information needed to represent the system is defined by the system state variables as shown in equation (132). The design of the communication system is then derived from this system information, $I(S)$. Additional information is added to this set by the communication system processing nodes. These nodes are necessary to transfer the information needed to manage and control the system. This system information set, $I(S)$ plus communication system information, should be minimized to keep the uncertainty about the system to a minimum.

4.6 Autonomous Systems

Autonomous and automated systems often incorporate artificial intelligence (AI) into the management and control of complex systems. Autonomous systems operate independently of other management and control systems, though may include human operators (i.e., crew) as part of the operation. Crewed spacecraft, aircraft, and ships are examples of autonomous systems. Automated systems are independent of human operators though their management, and control may be centralized or distributed. The basis of these autonomous and automated systems are defined through the system integrating physics. The AI requires a detailed understanding of the system as a whole, its goals, and its goals associated with functions and state variables that are connected to the physics and logic of the system. This information is directly provided by the GFT in a hierarchical structure of system intention, and is supported by information resulting from analyses using the system integrating physics. In addition, an understanding of each of the system functions, subsystems, and environments is necessary for AI decisions to be sound with regard to the system operations.

It is important to understand the distinction between autonomy and automation. Automation is simply the replacement of human action by machine action. This automation can either be a part of the system or separate from the system in a control center. Autonomy refers to the relative physical and/or functional separation of decision making and action or response capabilities (e.g., independence of action from a control location). For example, an autopilot in an aircraft is an automation of the flight control functions. The flight crew, which can also manage the flight control functions (i.e., fly the aircraft), are autonomous from ground control centers. They do have certain flight safety rules to follow in their flight paths and do have some level of response to an air traffic control station, yet they are free to determine the best (i.e., safest) course of action for the aircraft in any number of flight emergencies including mechanical, medical, security, or weather related.

Many factors that are part of the mission context define the need for automated or autonomous system operation. Time-of-response constraints are a big consideration. If a system response has time for the system to communicate with a human operator, locally or remotely, then less automation is necessary versus a quick response where automation is necessary. System decisions needed for a quick response (to avoid a safety incident) lead toward automation. System decisions that affect long-term viability lead to informed human decision making. A long-term system can be expensive to manually sustain. In these cases, automation provides for a more cost-effective operation and reduces the required workload on small crews (actually enable small crew sizes). Accessibility of the system also drives both automation and autonomy. Systems in difficult to reach locations (e.g., spacecraft, submerged vehicles and platforms, systems in hazardous environments) will need more automation to handle local operations, maintenance, and repair responses. Systems that need local control to respond to failures, weather, threats, etc. need more autonomy than systems where remote control can sufficiently respond to events. In all of these cases, there will generally be a mix in automated and autonomous operations of systems. It is generally not a question of automated or not, autonomous or not, but rather a question of how much automation and autonomy fit the needs defined by the needs of the stakeholders and the operational environment.

There are several aspects that must be considered in an AI system.[45,46] The AI must have a system executive to manage the overall functioning of the system with respect to the current system status, the environmental state, and operational constraints (i.e., the system's goals) to maintain system safety (for humans involved and for the system itself) and for reliable operations. Operational goals include human safety, system reliability (e.g., structural integrity), consumable efficiency (e.g., power, water, breathable air, food, replacement components), and environmental impacts. A planning function is also involved which considers the current system state, near-term objectives, and long-range objectives. The planning function provides a course of action for the system to maximize the mission objectives (i.e., system success). Health management (i.e., failure detection and response) functions prevent systems from coming to an abrupt halt and provide diagnostics and prognostics on current system states. The planning system uses the current system configuration and historical system operational data to develop the short- and long-term course of action. This planning is based, in part, on the GFT to link system functions with system goals. This provides information on which functions are necessary to achieve the system goals. Both automated and human operators determine the changes in configuration, goals, and operations needed to continue system operation. The difference between use of automated operation and human operations is based on communication timeframes, response time needed (i.e., short term (tactical) versus long term (strategic)), significance of deviation from system goals (humans must be the decision makers for human safety or long-time system repurposing in the current and near-term state of the art) for the system decisions.

System control functions are at both the system and the subsystem levels. Similarly, the health management functions are at both the system and the subsystem levels. The system level functions primarily integrate functions of each of the subsystem control inputs or health management inputs. This level of system management provides control inputs for each of the subsystems and manages conflicts between the subsystems. These conflicts include automated responses leading to logic races or conflicting responses resulting in system failures. The health management functions at the subsystem level include the system monitoring functions, the diagnosis functions for current subsystem states, and the prognosis functions for future subsystem states. At the systems level, diagnostics and prognostics provide the integrated system status, making use of the integrated system physics to properly characterize the system state. Figure 20 illustrates the system level relationships and figure 21 illustrates the subsystem level relationships.

The system integration physics provides the structure for the autonomous algorithm to properly interface with the system. This integrating relationship identifies the parameters necessary to determine and control system performance. Using this integrating perspective supports a balanced response by autonomous algorithms to changes in system status.

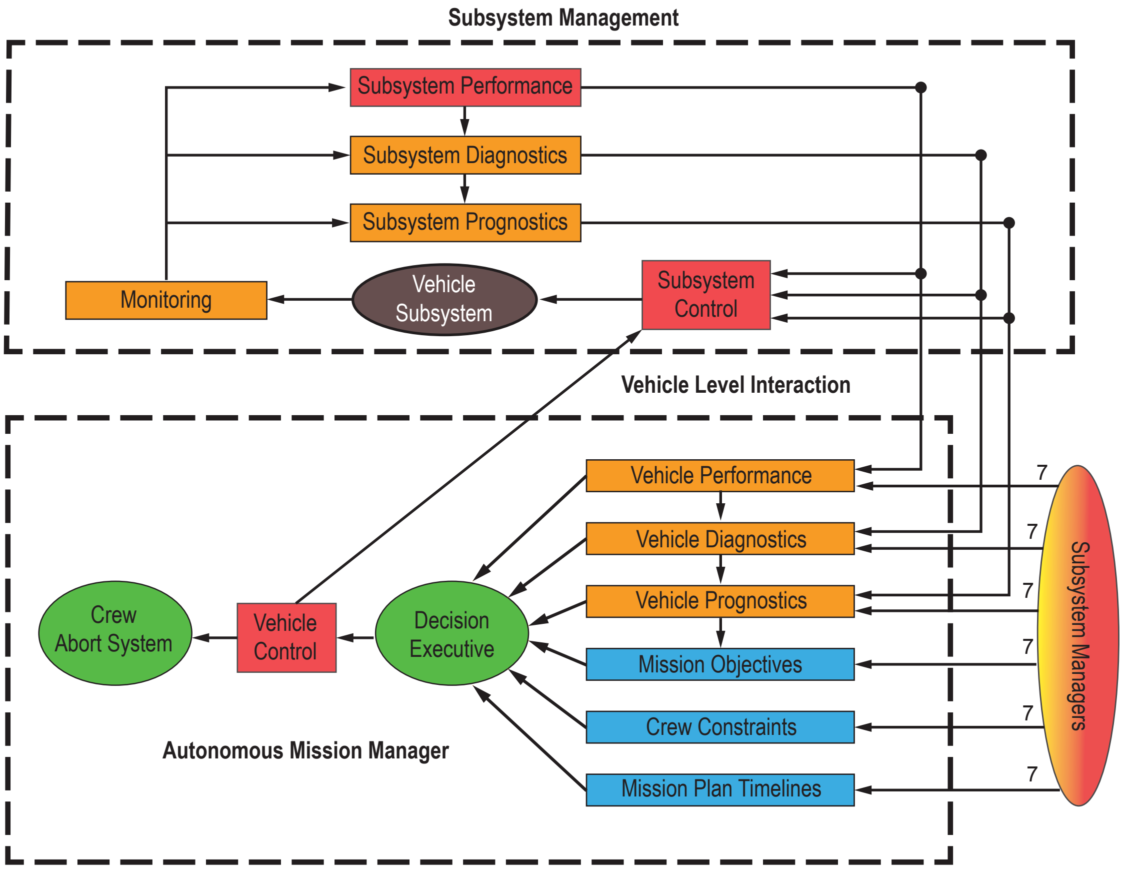


Figure 20. Integrated system autonomy.

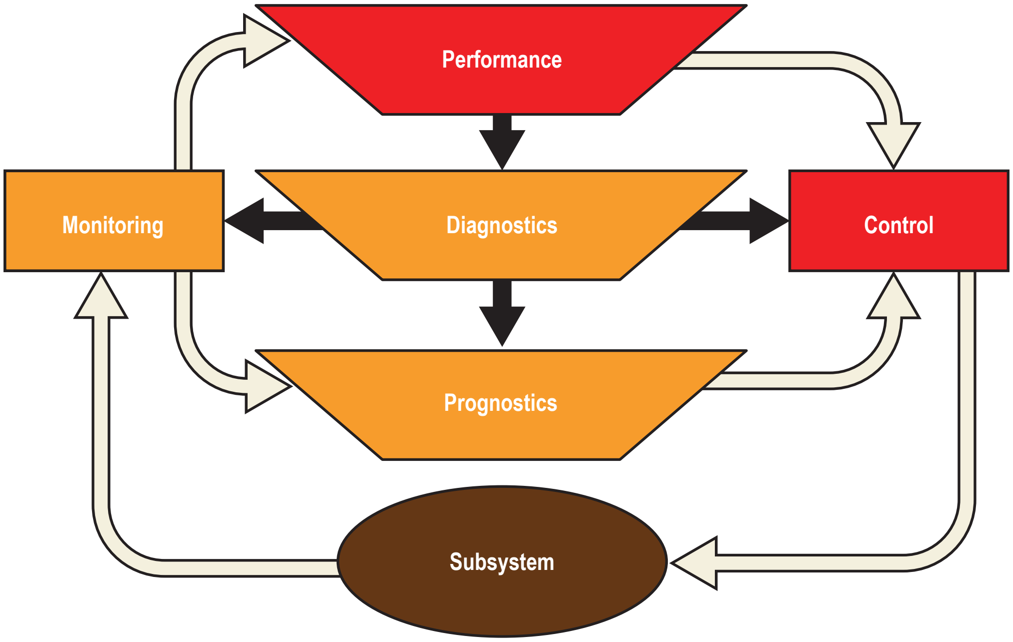


Figure 21. Subsystem management and control.

The relationship between the different functions can be constructed in an autonomous system showing the hierarchy of the algorithms shown in figure 22. The system management loops are shown within the stack showing the progressively broader levels of management loops that exist within the autonomous system stack.

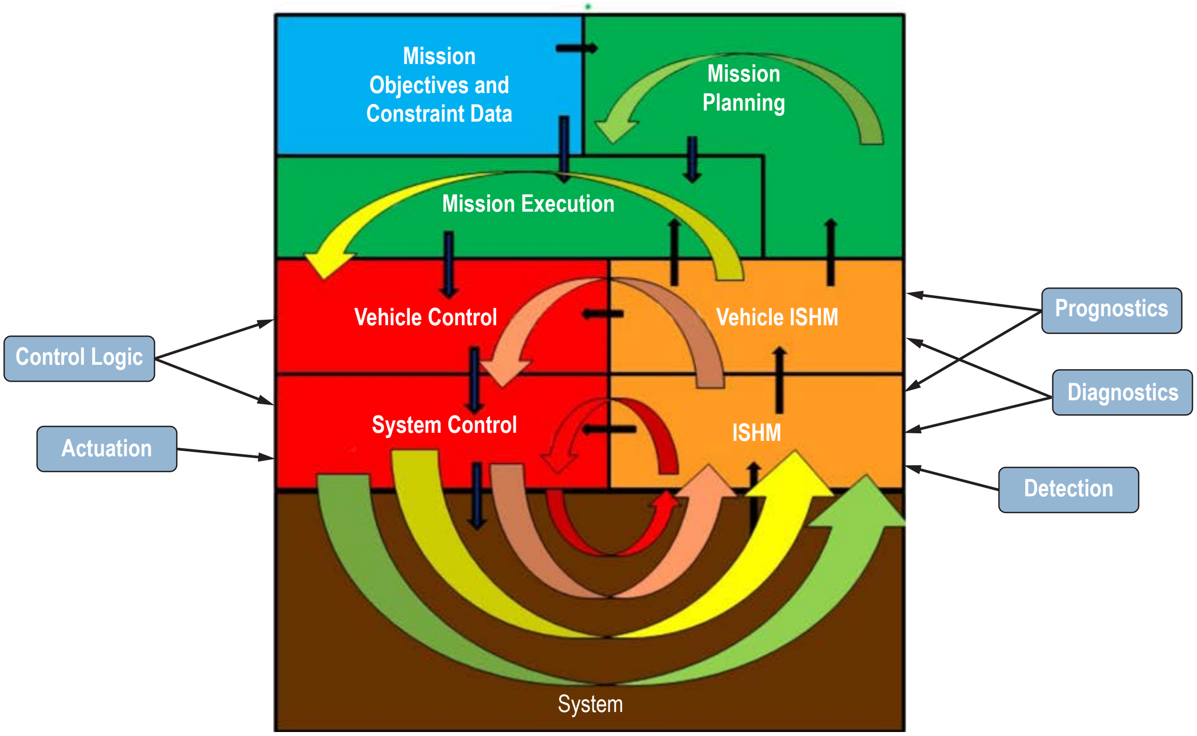


Figure 22. Autonomous system stack.

4.7 Multidisciplinary Design Optimization

Typical systems engineering processes focus on hierarchical decomposition of design and development tasks. This provides a linear structure of simple relationships but is quickly overwhelmed by system interactions in complex systems. System couplings provide the medium in which to understand the interactions of the system functions and the interaction of the system with the environment. There are many types of physical and logical couplings for a system. These couplings are not always obvious and can lead to emergent behavior which can modify the system couplings (i.e., adding new or changing the response of known couplings). System analytical techniques which support the full set of system interactions include MDO and its application for MDO coupling analysis (MDOCA).

MDO was developed in the early 1980s to address the interactions in designing large-scale complex engineering systems.[47,48] A key aspect of MDO research has focused on developing frameworks that enable a system optimization in which the inherent couplings in the physics and the analysis are appropriately modeled[49] (referred to here as MDOCA). Computer simulations are heavily used in modeling subsystem interactions, which enable a system analysis that is employed within the larger system optimization. Constraints are used to reflect the requirements for the design. The objective function is chosen to represent system design preferences.

Many different MDO frameworks exist. Common frameworks include multidisciplinary feasible (MDF), individual disciplinary feasible (IDF), all-at-once, and collaborative optimization (CO). Each framework has strengths and weaknesses. Some frameworks, such as MDF, provide a set of system behaviors that are consistent with the set of design variable values at each optimization step. Other frameworks, such as IDF, are not consistent at each optimization step, but do offer the ability to compute the subsystem behaviors in parallel. Each system must be examined to determine the appropriate MDO framework to use. Hence, the best MDO framework is problem specific. Given the inherently interdependent physics models, an iterative process is required to converge the system analysis, which is critical for assessing the impacts that design variables have on subsystems and the overall system. MDO achieves this convergence by initializing a system with a set of design variables and iterating through the coupled analysis until converged. This is illustrated for a simple system with two subsystems in figure 23, in which design variables are denoted by $\left(X_{\mathrm{A}}, X_{\mathrm{B}}\right)$ and the behavior variables, which represent the couplings, are denoted by $\left(Y_{\mathrm{A}}, Y_{\mathrm{B}}\right)$. Then, the sensitivities of these subsystem couplings are used to determine the overall system impact. The system design variables and their impacts are analyzed through implementation of a coupling strength analysis. One can analyze and leverage the local and global derivatives by using the global sensitivity equation (GSE) approach.[49]

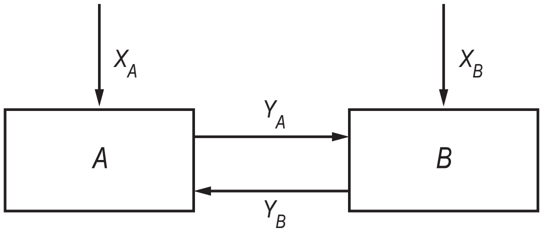


Figure 23. MDO two-subsystem example.

The GSE method provides an efficient approach to obtain the first-order sensitivity of the system behavioral response with respect to the system design variables using a process in which the larger system is decomposed into smaller subsystems. These system sensitivities are $\left(\frac{dY_A}{dX_A}, \frac{dY_A}{dX_B}, \frac{dY_B}{dX_A}, \frac{dY_B}{dX_B}\right)$ for the two-subsystem example shown in figure 23. The systems engineer can base these sensitivities on subsystem behavioral response sensitivities, $\left(\frac{\partial Y_A}{\partial X_A}, \frac{\partial Y_B}{\partial X_B}, \frac{\partial Y_A}{\partial X_B}, \frac{\partial Y_B}{\partial X_A}\right)$.[49] The goal is to solve for the total derivative matrix of the system where the influence of changes in one subsystem's outputs are accounted for by the variation of design variables of another subsystem as well as its own. The advantage of the GSE is that a system convergence need only be implemented once. Subsystem sensitivities are then found simultaneously within each subsystem. Each partial derivative and total derivative in equation (138) is a submatrix, with dimensionality being driven by the number of behavior variables in each subsystem as well as the number of design variables.

$$\begin{bmatrix} 1 & \dfrac{-\partial Y_A}{\partial Y_B} \\ \dfrac{-\partial Y_B}{\partial Y_A} & 1 \end{bmatrix} \begin{bmatrix} \dfrac{dY_A}{dX_A} & \dfrac{dY_A}{dX_B} \\ \dfrac{dY_B}{dX_A} & \dfrac{dY_B}{dX_B} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial Y_A}{\partial X_A} & 0 \\ 0 & \dfrac{\partial Y_B}{\partial X_B} \end{bmatrix} \text{ or } [A]\left[\frac{dY}{dX}\right]=\left[\frac{\partial Y}{\partial X}\right]. \qquad (138)$$

These system sensitivities are then used in MDO in gradient-based optimization frameworks. Additionally, however, they can then be used to determine the relative importance of the system couplings to identifying which are critical in impacting system behavior in MDOCA.

4.7.1 System Coupling Strength Approach

An ability to quantify the strength of the couplings in a systems context (i.e., not just limited to local interactions), and to subsequently understand how these couplings impact the overall value of the system, can be a valuable means for accomplishing trades amongst participants in the design process and can identify where undesirable behaviors might arise. Two basic methods have been offered to quantify the coupling strengths—a local sensitivity-based approach and a system sensitivity-based approach.[48] Previous work[49] demonstrated that local sensitivities (i.e., based solely on partial derivatives) provide a significantly incomplete view of the system impacts. System coupling strengths must be developed to reflect the impact of each local coupling $[A_{ij}]$ on system requirements and preferences as represented by constraints and the objective or value function.

Local sensitivity-based coupling strengths focus on the subsystem to subsystem (or component to component) interactions. Once normalized, these sensitivities can be used as local coupling strengths, providing insight into relative strengths in a local design space.[50] As an example, teams generally understand which couplings are critical in a system analysis. A coupling strength analysis, however, might demonstrate the importance of a coupling which was previously thought to be insignificant. Hence, implementation of the coupling strength analysis can provide valuable insight early in the design process to aid in appropriately addressing what must be included in a system analysis.

While the local coupling strength approach provides valuable insight, it is insufficient to fully understand the extent to which the elimination or suspension of a coupling will affect a system-level metric[51] or how variations will impact the system. This requires a system-level coupling strength metric[51] which can then be used to guide decisions on what interactions must be modeled.[52–54]

Consider the GSE form previously shown in equation (138), where A is the matrix of the sensitivities of the subsystem outputs with respect to the subsystem inputs. Since the equation is linear by nature, the partial derivatives of the total derivatives with respect to elements within $[A]$ may be found with equation (139):

$$[A]\frac{\partial}{\partial A_{ij}}\left[\frac{dY}{dX}\right]+\frac{\partial}{\partial A_{ij}}[A]\left[\frac{dY}{dX}\right]=\frac{\partial}{\partial A_{ij}}\left[\frac{\partial Y}{\partial X}\right] . \tag{139}$$

Given that the right-hand side is zero yields the following relationship:

$$[A]\frac{\partial}{\partial A_{ij}}\left[\frac{dY}{dX}\right]=-\left[\frac{\partial}{\partial A_{ij}}\right]\left[\frac{\partial Y}{\partial X}\right] . \tag{140}$$

Matrix $\left[\frac{\partial A}{\partial A_{ij}}\right]$ is empty everywhere, except at element A_{ij} within matrix [*A*], resulting in unity. Equation (140) provides an indication of the extent to which a change in the value of a coupling, A_{ij}, impacts the system behavioral derivatives with respect to system design variables.

If a coupling is removed (i.e., considered unnecessary in modeling the system physics), the local sensitivity becomes zero, resulting in the following change in the total derivative because of a coupling's elimination or suspension:

$$\Delta\left(\frac{dY_m}{dX_n}\right)=\frac{\partial\left(\frac{dY_m}{dX_n}\right)}{\partial A_{ij}}\bullet\left(0-A_{ij}\right) . \tag{141}$$

In equation (141), *n* indicates the *n*th system design variable while *m* indicates the behavioral response from subsystem *m*.

To understand the impact such an elimination of a coupling has on a system-level preference, represented here as *V* (value), the derivative of this system preference with respect to the system design variables, resulting from the elimination can be found. This can then be used to model the resulting change in *V* due to a change in $\left(\frac{dY_m}{dX_n}\right)$ when the coupling is suspended. Normalizing the relations provides equation (142):

$$\frac{dV_{A_{ij}}}{dX_n}=\sum_{m=1}^{M}\frac{\partial\left(\frac{dY_m}{dX_n}\right)}{\partial A_{ij}}\left(-A_{ij}\right)\frac{dV}{dY_m} . \tag{142}$$

Then, one can normalize the absolute change in *V* to represent a percent change in system preference as an error estimate due to the elimination or suspension of coupling, A_{ij}:

$$\Delta V_{A_{ij}}(\%) = \sum_{m=1}^{M} \sum_{n=1}^{N} \text{ABS}\left[\frac{\partial\left(\dfrac{dY_m}{dX_n}\right)}{\partial A_{ij}}\left(-A_{ij}\right)\frac{dV}{dY_m}\Delta X_n\left(\frac{1}{V}\right)(100\%)\right]. \tag{143}$$

This allows assessment of how both local sensitivities as well as global sensitivities will impact the system, providing insight on fidelity of couplings required to make informed system design and operations decisions. It also enables a mechanism to better understand the costs associated with inclusion or elimination of a coupling—cost in terms of computational time as well as the cost to system accuracy, thereby enabling trade decisions.

4.8 System Value

System value is the worth of the system to a stakeholder. These preferences are represented in a system value model that enables the preferences of different stakeholders to be compared and conflicts or contradictions resolved. This forms the value basis of the system.

The use of a quantification of a system's worth is present in many different systems engineering approaches. In MDO frameworks, the optimization algorithm assumes an objective function is available in order to find the best system. In decision-based design, a utility function is assumed to be available to find the best system. In other systems engineering approaches, a quantification of system worth may not be needed, but the comparison enabled through a quantification is required. In Pugh matrices and house of qualities, it is assumed that aspects of a system can be compared to one another, such as system attributes. If a system's worth can be quantified, then rigorous approaches can be adopted to determine the optimal system.

System cost is an important aspect in the system value, both the development cost and the operational cost. Cost models are specific to the system (principle 1) and should be based on the specific aspects of the system's components, development approach, and production and operations processes.

System capabilities are another aspect of the value of the system and tie to the stakeholder preferences as well. These system capabilities can also be used to assess the system's ability to perform various applications (or support various design reference missions). This assessment provides an indication of the system's robustness.

4.8.1 System Cost Modeling

The primary component of the system cost is labor, not materials, and is based on the tasks necessary to achieve the systems-integrating physics or logic that defines the system functions. A PBS, which captures the architectural view of the system, provides a system-based cost structure for the system cost. This PBS provides a basis to understand the major system cost elements, the regulatory costs incurred on the project. Using this cost structure, a story of the system's value and benefits to various system stakeholders (including manufacturers, operators, and users) can be constructed.

Note that specifying detailed budget requirements may overconstrain the system, leading to inelegant solutions or cost conflicts. The component costs are not important, the total cost is. At the component level, it is important to understand the cost drivers so these can be managed to keep the total system cost within budget. The PBS provides the cost structure to identify the cost drivers and analyze the alternatives. Stakeholders generally do not buy parts of the system, they buy the whole system.

4.8.2 System Value Model

A system value model is a mathematical representation of the preference of a stakeholder. This representation is a function of attributes of a design, relating them to the value the system provides.[55] A value model can provide the integration of the physical and organizational aspects of the system in a single model under uncertainty. This model can provide a mathematical representation for the value of the system to relevant decision makers (e.g., operators, users, investors, manufacturers, government regulators). This enables decision makers to make decisions that are consistent with one another, striving to improve the same metric. Where system value preferences vary among different groups of stakeholders, the value model indicates the contradiction in the preferences held by the different stakeholder groups. The system value model can provide a mathematical basis for system validation, clearly differentiating system validation from system verification.

The primary challenge with the system value model is not the use of them, as many different approaches already assume a mathematical representation of preference exists, but is in the formation of the value model. The best method of forming system value models is still an open research question, but research has been conducted in the area to provide suggestions on how to proceed. A general approach adopts an iteration of steps to ensure that new knowledge is captured in the value model as the design process proceeds. This includes steps of templating (brainstorming on key attributes and their relationships), evidence gathering/analysis (using data from similar projects, physical equations, etc. to inform the attribute relationships), and stakeholder feedback (use information from content analysis of documentation, questionnaires, interviews, etc. to align the value model with the stakeholder preferences). These steps are iterated to continuously improve the value model and ensure alignment with the stakeholder.

The use of data from previous missions or related projects is the primary technique currently used in value model formation. This can be seen in diverse projects such as lunar mining missions, NASA funding allocations,[56] electric vehicles,[57] small satellites,[58] and nurse staffing.[59] Content analysis of documentation has also been used to elicit top-level stakeholder preferences. This can be seen in eliciting desires found in NASA habitat documentation[60] and from NASA's 2017 Strategic Plan.[61] Questionnaires and interviews are also actively being researched as techniques to determine the stakeholder preferences.[62] No matter the technique, the goal is the same: form value models from evidence in order to have a strong foundation as a basis to vet the value model.

When forming value models, there are typically two high-level attributes: cost and benefit. For commercial organizations, the benefit is typically revenue, with significant research available to support appropriate revenue models. For noncommercial organizations, the model of benefit is much more complex. Attributes such as knowledge, prestige, and avoidance of catastrophes[61] may need to be modeled. Evidence from past data may help to quantify these attributes, but may heavily be project specific.

The system value attributes vary between stakeholders (including the development organization) and may be tracked from initial stakeholder definition, through design, and finally to system validation. This provides a measure of the system capabilities, tied directly to the system design, which can guide decisions during system development. The system value model can provide a mathematical basis for system validation, clearly differentiating system validation from system verification.

4.8.3 System Robustness

System robustness is a challenging concept to define and quantify. Many authors offer no explicit definition for robustness. There are various terms used synonymously for robustness such as resiliency[63] or antifragility.[64] The few schemes reported in the literature for quantifying robustness are mutually exclusive with each other and are in conflict with other established theory, such as expected utility decision making. As exemplified by Taguchi's robust design[65] and Kitano's notion of robustness for biological systems,[66] the most common approach is to associate robustness with low variability or spread in performance or other attributes of interest. Although such robustness measures may lead to some insights about a system, they are mathematically incompatible with rigorous decision theory and can lead to decisions that contradict stated preferences.[67] Established and mathematically rigorous decision theory provides a means to incorporate preferences for system robustness into systems development processes. The definition of what makes a system robust is based, in part, on the capabilities the system provides.

System robustness may be defined with reference to the system keeping stable outputs under wider ranges of input state variations or the addition of inputs (such as when a system is applied to a different application or environment). This is observed in Russian launch vehicles that operate across much wider environmental variations (temperature, winds, etc.) than American launch vehicles. Thus, Russian launch vehicles are often stated to be more robust to launch weather than American launch vehicles. For this example, the levels of robustness of American versus Russian launch vehicles is quantifiable. From this example, a mathematically quantifiable definition of robustness is possible, when delimited against specifically stated criteria. If those criteria are derived from value models and associated preferences, then the value of robustness for a given system can be quantified as well.

To integrate robustness into a rigorous decision-theoretic framework, it is necessary to characterize a candidate system's attributes under uncertainty. In this context, an attribute is a property or figure of merit of the system that serves as an input to the system value model. This leads to the use of the system capability model as a means to characterize system robustness.

4.8.4 System Capability Model

A system capability model is a description of the key attributes of candidate system architecture as a function of its application environment and specific applications (i.e., missions). When provided with environmental and mission models, one uses the system capability model to predict a probability distribution over system attribute values. These serve as an input to the value model. Nonrobust systems will tend to have low certainty equivalence and therefore would not be favored in decision making.

4.8.4.1 Single Versus Multiple Value Scenarios. Engineered systems are often designed to produce value to the decision maker through some specific value-producing scenario. When this is the case, that value-producing scenario provides the context for the system value model. This context dictates the way in which system capabilities are modeled because the only attributes that are important are those which are salient to the primary value-producing scenario of the system. The system capability model is, in a sense, integrated into the value model along with any assumptions inherent to the value-producing scenario of the system.

Take, as an example, a typical commercial system, such as an aircraft developed for sale to commercial transportation companies. From the point of view of the firm designing the aircraft, the system produces value by selling to their customers for a higher price than the cost of development and manufacture. The value-producing scenario for this system is that of being sold to commercial transportation companies for a profit, and the firm can use net present value of profit as a measure of value. It is important to recognize the differences between the business-oriented decision maker versus the engineering-oriented decision maker. The business manager tends to value profit more than performance, making performance an input to the overall calculation of profit. The engineering decision maker tends to focus more on performance, given a proper context has been established and passed down as mission requirements. For a commercial system where profit is the primary measure of value, the system engineer ensures that the engineering system performance context is well defined with respect to the business profit context.

Many systems can be used in a variety of contexts. An aircraft may be designed as a commercial air transport and then applied in a military context. Suddenly, attributes that may have only been important within the restricted regime or may not even have been included in the context of commercial air transport are now important to properly determining the system's value for military applications. The aircraft may now benefit from resistance to small arms fire or from radar stealth systems. The value model and model of capabilities, previously sufficient in the commercial case, is now no longer a complete picture of system value and will need revision to consider the preference changes in the new context.

The need to consider multiple value-producing scenarios complicates the construction of a value model significantly. The traditional approach of integrating the system capability model with the value-producing scenario and its assumptions does not produce a complete picture of system value because there is no longer a single value-producing scenario to use as a reference. The example used in this case was a profit-driven system, but this situation can also arise when dealing with systems that produce intrinsic value in noneconomic forms. Table 14 summarizes some general characteristics of systems with multiple value-producing scenarios.

Table 14. Characteristics of systems ill-fitted for a traditional value model.

Characteristic	Description
Multiple value scenarios	The system will be used in a variety of distinct and independent value-producing scenarios
Varied capability utilization	Each value scenario may utilize different aspects or regimes of system capability to produce value
Some capability irrelevance	An aspect of system capability vital to one value scenario may be irrelevant to another
Varied scenario frequency	Value scenarios may vary widely in frequency, with some common and others comparatively rare

How are systems engineers to accurately evaluate these systems? One approach would be to construct a traditional value model for each value-producing scenario separately, integrating into each scenario value model only the aspects of system capability that are relevant to it. This approach is problematic for several reasons. It is inefficient to reconstruct the system capability model for every single value scenario, and propagating design changes through a multitude of slightly different capability models is time-consuming, creating opportunities for ambiguity or errors. Additionally, this approach offers no guidance as to how to aggregate these value models. A major strength of value-driven design is the ability to obtain a single scalar score for each system—a methodology that produces multiple scores that requires a heuristical trade is of less use.

A composite value model taking value-producing scenarios into account modeling both system capability and value generation accordingly provides a more comprehensive value structure. This structure centers on developing a single system capability model that is independent of any given value scenario but compatible with any of them. Each value-producing scenario has its own value model. While the structure of the valuation is given in the following sections, the details of implementation are not always strictly prescribed. This is intentional to allow for a degree of freedom in the specific implementation of the structure, which can vary greatly per context. The basic structure of this capability-based framework is presented in figure 24.

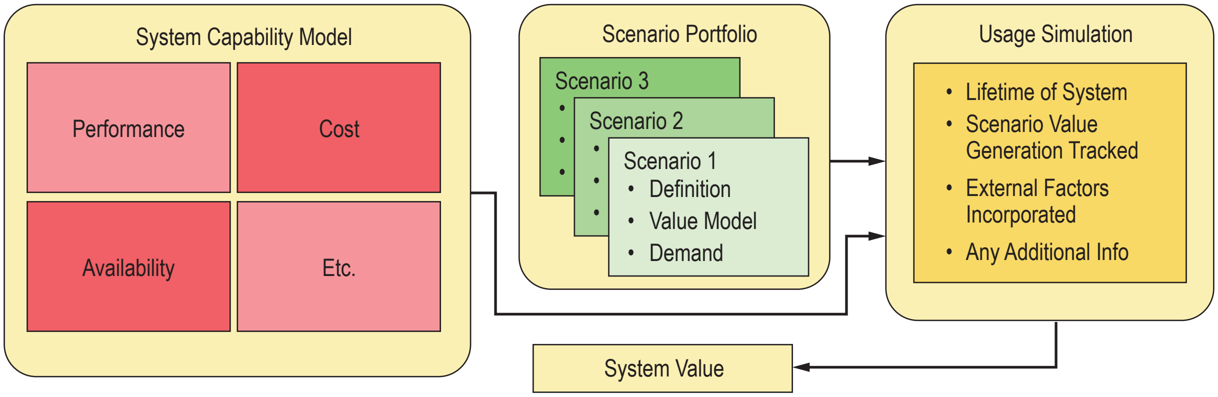


Figure 24. Basic structure of capability-based value framework.

4.8.4.2 Modeling System Capability. The cornerstone of this value structure is the system capability model—a comprehensive collection of all top-level system attributes with direct impact on value in any value scenario. This capability model is quite like the system capability model used in the traditional approach to value-driven design with a few additional considerations considered. Because the system produces value through a variety of different and distinct scenarios, the capability model must be broad enough to be used as an input to all value-producing scenarios while remaining independent of any given scenario. In many cases, the capability model for the complete value structure may look very different from the system attributes model one would construct for a given application (i.e., use case).

The simplest mathematical form of the capability model would be a multidimensional vector of scalar system attributes. However, the nature of the problem often necessitates a more complicated representation. Often, certain aspects of capability that could be expressed as scalar values or vectors for any single value scenario must be represented as curves or envelopes in the general capability model because the existence of multiple value scenarios necessitates the additional information encapsulated by the relationships. This is demonstrated for a launch vehicle by the 'delta-V versus payload mass' curve, which would not be needed in its entirety for any single mission value model (only a single point is necessary for a specific mission) but must be included in the general capability model for it to be compatible with any general mission.

In short, the capability model should contain a complete picture of system capability that contains all information about the system necessary to evaluate the value it produces when operating under any individual value scenario.

4.8.4.3 Value Scenario Portfolio. The other major component of the capability-based value structure is a portfolio of value-producing scenarios. This portfolio should be developed systematically. First, all unique value-producing scenarios should be identified. Deciding what qualifies as a unique scenario for the purposes of this structure is not always trivial. A good rule of thumb is if two modes of operation use cases, missions, or other similar divisions in system functionality utilize different regimes of system capability or have substantially different relationships between system attributes and value produced, they should be classified as separate value-producing scenarios. There may be multiple useful ways to construct a portfolio of value scenarios—there is not necessarily a single 'correct' answer.

Once the scenarios have been identified, they must be modeled. Two pieces of information are needed at a minimum to define each scenario: an independent value model capable of mapping from system attributes to value, and information about demand for the scenario or how often the scenario will be utilized.

Each unique scenario requires a value model specific to that scenario. These value models are not substantially different from traditional value models—they take system attributes as inputs and output a scalar measure of value. Although each scenario value model will pull attributes from the capability model as inputs, there is no requirement that they all pull the same attributes. In fact, the nature of the multiscenario value framework suggests that they will not do so. It is precisely because of this that a general capability model, developed independent of any single scenario but capable of serving all scenarios, is necessary.

The other major piece of information needed to model each scenario is information on how often each scenario will be utilized (i.e., demand). The multiscenario value framework allows considerable latitude as to how specific this information must be. It can be expressed in terms as simple as '50% of the time, system will operate under scenario A' or as complex as 'no more than once per 2 years, system will have a 15% chance of operating under scenario B for a period of 1 month.' The representation of demand should be as complex as is necessary to accurately portray the relative frequency or weight of each scenario.

4.8.4.4 Evaluating System Value. After the capability model and the scenario portfolio have been developed, the value of the system is evaluated by simulating its usage across some time horizon of interest considering all relevant value-producing scenarios. Information about relative scenario utilization frequency should be used to structure the simulation, and any external or environmental factors affecting the operation of the system should be included in the simulation in whatever detail is necessary to capture their impact on value generation. The capability model is used as an input to all individual scenario value models, and both benefits incurred and value generated through all value scenarios are tracked and (if relevant) adjusted for time discounting. If the capability model, value models, external factors, or any interaction between them is stochastic, an appropriate method should be used to estimate the distribution of system value production and determine the anticipated value of the system.

4.9 System of Systems

A system of systems involves all of the system integrating principles in the development of a new or extended capability from a set of independent system capabilities. The basic approach is to consider the capability desired from the integration of the systems which involves identifying the system of systems integrating physics. The integrating system physics may be similar or different from the individual systems that make up the system of systems. In addition, a single point measurement (e.g., specific efficiency of a system) will not describe a system of systems. Instead, a distribution function is involved in considering the new or extended system capability from the different constituent systems.

Examples of systems of systems include military capabilities, transportation capabilities, and communication capabilities. Military system of systems can be measured in some applications as delivering amount of physical force that exceeds the opposing system or system of systems structural load capacity (i.e., (lbf/ft^2) ordinance > (lbf/ft^2)structural limit). This is a thermodynamic relationship, thus, this particular example is a thermodynamic system of systems. This simple relationship works well in defining the effect of a single system impact on another system. However, when dealing with multiple systems, a distribution of capability must be considered corresponding to the distribution of the opposing system of systems. This brings in engineering statistics to represent the distributions and uncertainty in the parameters of the system of systems. The result is the updated relationship:

$$\Gamma\left({F_n}\big/{A_n^2}, x_n, y_n, z_n, d_n, \tau_n \right) \geq \Gamma\left({F_m}\big/{A_m^2}, x_m, y_m, z_m, d_m, \tau_n \right), \quad (144)$$

where

Γ = distribution function
F = force
d = target miss distance
A = area
τ = time window
x, y, z = location of each ordinance or opposing structural system.

Transportation system of systems involves the delivery of passengers and/or cargo to specific locations (or over a specific distance). This can be measured as lbm/mi. Note, that for transportation systems, passengers are typically accounted for as transported mass for the individual systems to be balanced correctly. The distribution function for this system of systems involves the location of both the departure and the destination, as well as the exergy efficiency of the systems. This can be represented as:

$$\Gamma\left(\eta_n, d_n, x_n, y_n, z_n, x_m, y_m, z_m, \tau_n\right) \leq \text{objective function}, \tag{145}$$

where

Γ = distribution function
η = thermodynamic exergy efficiency of each system
d = distance transported
x, y, z = location of each departure point *(n)* and destination *(m)*.

The objective function in this case could be a variety of functions including minimal distance, maximum efficiency, maximum passengers or cargo to locations, minimum time, or some combination. Thus, the relationship could be greater than or equal to as well. The value model can be used to determine the objective function to balance the transportation system of systems.

Information systems are very different types of system of systems. They can employ various information-gathering functions (e.g., optical, radar, data input, stored data retrieval, sensor readings) and have various forms of media transport (e.g., radio, microwave, optical). This brings in several different systems with different physics being integrated to form a system capability. A measure of information is the concept of information triples that can be used to characterize the system. A triple consists of subject, predicate, object as a construct for representing information at a higher level than basic data patterns. The system is integrated then by defining the triples generated by each data gathering system in order to produce meaningful information. The distribution function for this can be complicated which hundreds, thousands, or millions of individual systems comprising the system of systems. Value modeling is an invaluable tool to construct the objective function in which to determine the performance of these systems.

State variable methods are important to show the interacting parameters between the individual systems within the system of systems. The GFT provides the structure to accomplish the mapping of the system of systems goals with the functions (i.e., systems) and their state variables. In a GFT framework, each system has its own goal structure with its associated functions and state variables. One system can be ‘used’ by another system only if a state variable in the using system matches a state variable in the ‘used’ system, and the constraints on those state variables are compatible. This generally means that the using system’s state variable constraint is equal to or looser than the same state variable in the ‘used’ system. The state variables can be associated with very different functions and goals in the two systems, but still be compatible as long as their constraints are compatible. State machine modeling can provide an understanding of the state transitions that each system progresses through as it functions within the system of systems.

MDO provides an outstanding tool to design and analyze a system of systems. Each system can be modeled separately and the integrated operation forming the new or extended system capability can be balanced through the shared variables representing the system interactions within the system of systems.

4.10 Human Systems Integration

Systems engineering strives to view the system as a whole. The individual components and their interactions with the system result in emergent properties that may be difficult to anticipate or assess without evaluating the integrated system, including all of its components assembled and working together as intended. In systems that are operated by, maintained by, or communicate with humans, the human is a key component of the overall system, and one with unique capabilities and limitations. Humans are adaptable and capable of responding to unforeseen consequences in innovative ways, allowing for system resiliency to unanticipated failures or events. Humans can also have limitations in cognitive processing capability, mental and physical workload, situation awareness, and anthropometric variables (strength, reach, range of motion, etc.). If the system does not accommodate for human capabilities and limitations, mismatches can occur between the inputs and outputs of the system and what the human can manage or provide, resulting in system failure. Incorporating these considerations as part of the systems engineering process is referred to as Human Systems Integration (HSI):

- HSI definition: An interdisciplinary integration of the human as an element of the system to ensure that the human and hardware/software components cooperate, coordinate, and communicate effectively to perform a specific function or mission (application) successfully.

- HSI scope: HSI covers all aspects of the system with human interactions. This includes manufacturing, operations, and maintenance. HSI considers how the system interacts with humans through communication, human computer interfaces, physical access and interfaces, and social structures incorporating the system.

- HSI benefits: Proper HSI practices enhances human system design, reduces system operations and maintenance cost, and optimizes total system performance. Through the process of inclusion of technical disciplines and domains, HSI provides a capability that ensures the limitations and abilities of humans are adequately addressed in the system capabilities.

Failure to address the human interactions with the system can lead to substantial cost and schedule impacts. These impacts result from the system not being useable, operable, or maintainable in an economically feasible or time-dependent manner.

To implement HSI, many of the considerations put forth in the postulates and principles of this document are key including consideration and integration of multiple technical disciplines and an understanding of the sociological concerns. From a technical discipline perspective, HSI focusses on incorporating six 'domains' of technical expertise: Human factors engineering (HFE), operations resources, maintainability and supportability, habitability and environment, safety, and training. These domains, shown in figure 25 and given in table 15, are not exclusive, and have significant overlap in their technical content.

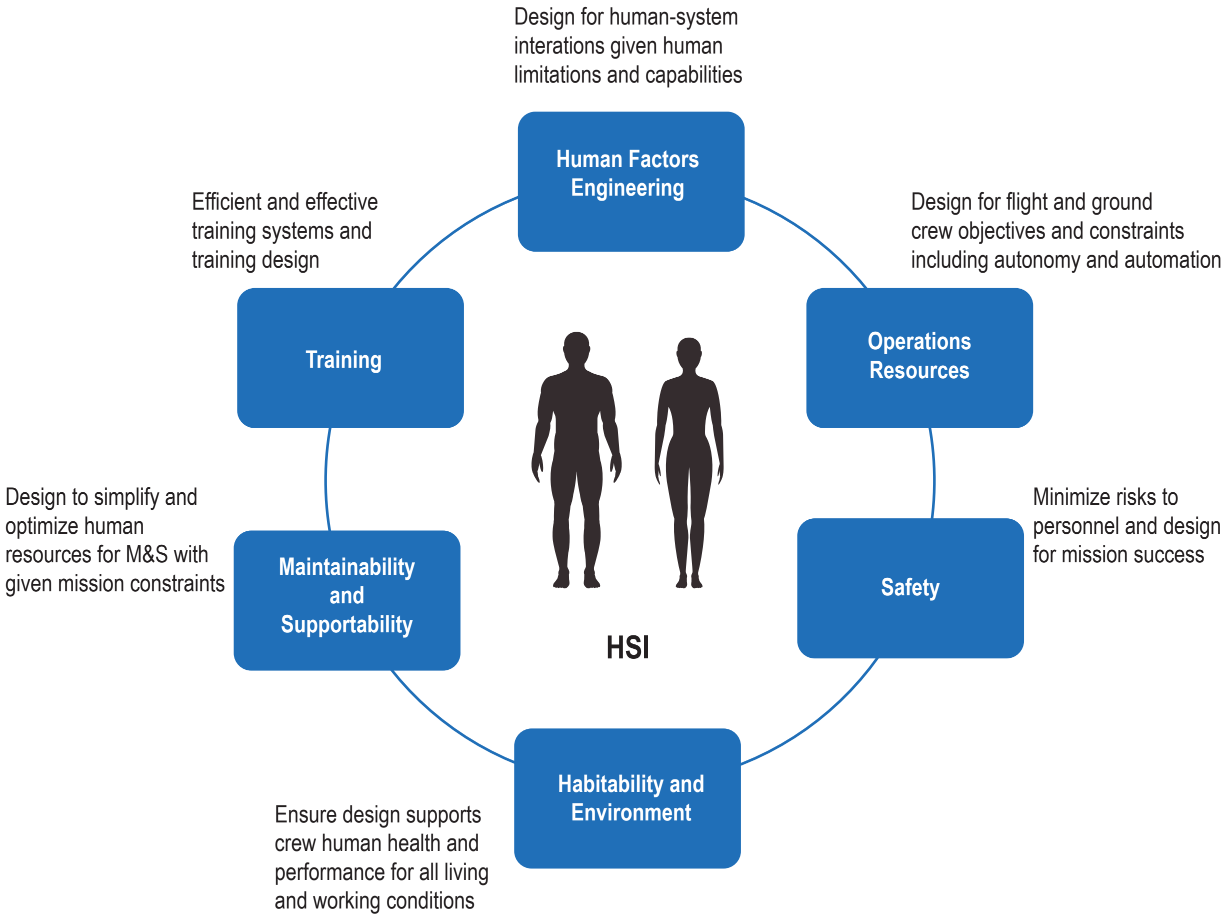

Design for human-system interations given human limitations and capabilities
Human Factors Engineering
Efficient and effective training systems and training design
Design for flight and ground crew objectives and constraints including autonomy and automation
Training
Operations Resources
Design to simplify and optimize human resources for M&S with given mission constraints
Minimize risks to personnel and design for mission success
Maintainability and Supportability
Safety
HSI
Habitability and Environment
Ensure design supports crew human health and performance for all living and working conditions

Figure 25. NASA HSI technical domains.

Table 15. NASA HSI technical domains.

Domain	Definition	Examples of Expertise
Human Factors Engineering (HFE)	• Designing hardware and software to optimize human well-being and overall system safety, performance, and operability • Done by designing with an emphasis on human capabilities and limitations as they impact and are impacted by system design across mission environments and conditions (nominal, contingency, and emergency) • This supports robust integration of all humans interacting with a system throughout its lifecycle • HFE solutions are guided by three principles: – System demands shall be compatible with human capabilities and limitations – Systems shall enable utilization of human capabilities in nonroutine and unpredicted situations – Systems shall tolerate and recover from human error	• Task analysis • Human performance measures – Workload – Usability – Situation awareness • HFE design – Anthropometry and biomechanics – Crew functions – Habitat architecture • Human in the loop (HITL) evaluation • Human error analysis • Human-system interfaces • Systems design • HFE analysis
Operations resources	• The considerations and resources required for operations planning and execution • This includes operability and human effectiveness for flight and ground crews to drive system design and development phases, as well as trades for function allocation, automation, and autonomy	• Operations process design for both ground and flight crew • Human/machine resource allocation • Mission operations • Resource modeling and complexity analysis • Flight operations • Procedure development • Crew time • Staffing/qualifications analysis
Maintainability and supportability	Design to simplify maintenance and optimize human resources, spares, consumables, and logistics • These are essential due to limited time, access, and distance for space missions	• In-flight maintenance and housekeeping • Ground maintenance and assembly • Sustainability and logistics
Habitability and environment	• Safety factors ensure the execution of mission activities with minimal risk to personnel • Mission success includes returning the crew following completion of mission objectives and maintaining the safety of ground personnel	• Medical • Crew health and countermeasures • Environmental • Radiation • Toxicology • Nutrition • Acoustics • Architecture • Lighting • EVA physiology
Safety	• Safety factors ensure the execution of mission activities with minimal risk to personnel • Mission success includes returning the crew following completion of mission objectives and maintaining the safety of ground personnel	• Safety analysis • Reliability • Quality assurance • Factors of survivability • Human rating analysis • Hazard analysis
Training	• Design training program to simplify the resources that are required to provide personnel with requisite knowledge, skills, and abilities to properly operate, maintain, and support the system	• Instructional design • Training facility development • Onboard training

The implementation of HSI is documented in the Human Systems Integration (HSI) Practitioner's Guide, NASA/SP-2015-3709. This document provides a full discussion of these HSI concepts.[68]

4.11 Category Theory: Mathematical Basis of Systems Engineering

As stated by postulate 2, the domain of systems engineering consists of subsystems, their interactions among themselves, and their interactions with the system environment. This defines systems as consisting of two main components: subsystems (or functions) and interactions. Looking into mathematics, category theory provides a mathematical structure to describe these two components; i.e., to describe the system.

4.11.1 Mathematical Category Definition

A mathematical category consists of objects and the relationships between the objects. A category has both a domain and a codomain. The relationships create the map, or define the interactions, between the objects. The source (i.e., initiating) object is the domain (dom), and the terminating object (resultant object arrived at when a relationship is applied to an object) is the codomain (cod). Mathematical categories also have three properties that must be included in the definition:

(1) Identity: The relationship that terminates at the source object (a): $I(a) = a$.

(2) Composite: A resultant object when two relationships, f and g, are combined. The two relationships are related such that the dom(f) = cod(g). A composite can be several different functions (e.g., addition, subtraction, multiplication, division, powers) and is written generally as $f^{\circ} g$.

(3) Associativity: There are both associative and nonassociative relationships for systems. When the composites are defined (exist), the associative property applies to the operation of the composites: $f^{\circ} (g^{\circ} h) = (f^{\circ} g)^{\circ} h$.[69]

These properties of a category provide some important implications for systems. The Identity property implies that a system, where no interactions occur, remains unchanged. Note, that environmental interactions are constant, so no systems actually exist in this state.

The Identity property also indicates that copies of systems are similar yet separate categories. Manufacturing variations yield small differences that change some of the relationships. The changes may be subtle but this provides some mathematical rationale as to why different units from a manufacturing line can have different performance characteristics that lead to a variety of statistical variances in the system properties.

The Composite property provides for the application of various mathematical properties that define the physical relationships in systems. This property also provides for the mathematical definition of unexpected interactions. System objects (i.e., components, assemblies, or subsystems) may seem isolated, yet have subtle interactions that yield unexpected results (i.e., emergent properties). The Composite property provides the construction of the path relationships between two objects (components) and a clear definition of the actual composite interaction. Category Theory provides the mechanisms to construct these composite relationships.

Associativity provides some insight into systems functions. From an assembly perspective, associativity tends to hold, but from a system function perspective, associativity does not hold. System emergence is a property that is not the sum of the individual properties of the subsystem. Emergence results from the integration of the individual properties yielding new functions of the system. Controlled flight is an emergent property of the vehicle subsystems. Controlled flight is not achievable by the subsystems alone, but when the subsystems are integrated, the vehicle has this emergent property.

Categories have three types of correspondences: Function, Relation, and Equivalence. Construction of these relations and correspondences are system specific as defined by postulate 1. These correspondences are defined as follows:

(1) Function—A function is a correspondence where for each unique element in the source (domain) there is a unique element in the target (codomain). A source element maps to only one target element. While more than one target element can map to a single source element (i.e., many to one), there cannot be one source element mapping to more than one target element (i.e., not one to many).

(2) Relation—A relation is a unique correspondence between two objects. These can have both many to one and one to many relationships for any given source. A relation which is not unique is not considered a mathematical correspondence.

(3) Equivalence—An equivalence relation can have two forms:

(a) Reflexive, identity, or diagonal relation is an equivalence relation: $X \sim X$.
(b) Symmetry relations are also equivalence relations: $X \sim Y = > Y \sim X$.

Equivalence relations hold to the transitive property which states if $X \sim Y$ and $Y \sim Z$, then $X \sim Z$.

Equivalence produces an equivalent result, not necessarily a same or equal result. This is an important system property and allows copies of systems that are not identical yet produce equivalent results within the variance of the system properties. This relates well to the concept that copies of the same system are similar but not identical, as shown in hypothesis 4.

4.11.2 System as a Category

A system is a form of a mathematical category. The objects, or elements, of the system are its components, assemblies, and subsystems. The relationships are the physics interactions between these objects. This set of objects and relationships defines the system as a category. Category theory then can be used to organize and understand the structure and relationships of the system. Category theory does not define the system but provides the mathematical structure to understand the system. The design of the system is done through the definition of the system subsystems and interactions through the application of engineering processes and methods. Category theory provides the mathematical structure to organize and understand the system.

By knowing the objects and relationships, one knows the basis of the system. Changing the objects or relationships changes the mathematical category and, thus, is a different system. Therefore, a specific system must have, or is required to have, a specific set of objects and relationships. Thus, the definition of these objects and relationships form the set of requirements necessary to define the system. By defining the mathematical category of the system, the requirements are completely defined.

Category theory provides a structure to identify potentially hidden relationships as the system is defined and the category structure is filled out. This structure allows one to look at all of the objects and explore the potential relationships that exist to all other objects.

The underlying structure of a category can be seen as a directed graph of the objects and relations. Directed graphs are used quite extensively in engineering applications and provide a way for engineers to visualize the basic structure of a category. Directed graphs become categories when compatibility conditions are applied. The nodes (i.e., objects) and lines (i.e., relations) in the directed graphs cannot be arbitrary. The compatibility conditions define the limits on the graph structure (i.e., object and relations). A system also is defined by these limits that are well represented by the category.

As noted in principle 3(c), you define system requirements as you progress through the system. An important property of categories is that the objects of large categories can themselves be categories. This is seen in the example of the category of categories, cat, which is the mathematical category (a super category) that contains all other mathematical categories. So, early in the system definition and design phases, subsystems may be represented as objects in the system category without defining any of their internal properties (i.e., objects and relationships). Only the external subsystem properties may be initially defined. Thus, the subsystem is a smaller category that fits within the larger system category. This provides a mathematical approach to general engineering 'black box' analysis where the external relationships with the box are known but not the internal functionality (i.e., objects and their relationships). As the design progresses, category theory supports the further design of each of these subsystem categories.

Functors between categories preserve the relationships between the internal objects of the category. For a system composed of subsystems, the Functor provides for the interconnection of the subsystems while maintaining the internal structure and relationships of the subsystems. For example, thermodynamic exergy provides for a system balance equation which preserves the subsystem thermodynamic relationships including the mass balance, energy balance, and entropy balance. Exergy can be expanded to show all of the constituent equations or abstracted to show only the subsystem interactions. This abstraction always preserves and is dependent on the internal relationships of the subsystems.

The objects of the category defining the system are important for the system engineer to identify and understand. These are an integral part of the system as seen in the definition of a category. However, a category can be described abstractly as only the relations between the objects, thus hiding the objects. The objects are identified in this case by the identity property that yields the source object back. This abstract idea illustrates the focus of the system engineer not on the objects individually, but on the relationships between the objects (i.e., system interactions). Having the objects in view, however, is necessary to clearly represent the system. This full view of both the object and the relation is the power of category theory to fully describe a system.

Systems are composed of both fixed objects and expendable or consumable objects. Category theory provides a representation of this through the concept that a value of zero does not mean the object no longer exists. The object contributes the properties of the zero identity and multiplication by zero to any relationship associated with the object. Thus, the zero value affects the value of the relations but not the existence of the object or relations. This is fundamental to representing items that are 'empty' such as tanks, batteries, or bins.

In contrast, if a component is removed from a system, then the structure of the category changes with changes in the object content and the relationships. For example, a battery that is drained yields a value of zero electrical power to the connected system relations. However, a battery that is removed from the system no longer exists as part of the system. These two conditions are different and category theory treats them as different conditions.

4.11.3 System Integrating Relationship

As noted in section 4.2, there is an integrating physics (i.e., an integrating relationship) for a given system type. Category theory provides some important mathematical concepts to address the definition of these integrating relationships.

Equivalence relations, defined in section 4.11.1, can be used to simplify categories to their normal form. The normal form provides the underlying structure of the category with more complex structures reduced to their direct relationships (reduces composites to the source and terminating relations). This provides a mathematical approach to simplify the structure of a complex system and obtain the underlying integrating perspective (i.e., direct view of the system integrating relationship). For finite sets, which physical systems are, the normal form is isomorphic to the overlaying category. Thus, the normal form and the category are the same for physical systems.

Systems are constructed from materials that occur in nature (even software is stored and executed as electrical charges). Natural constraints (constraints which occur naturally) within these systems are present in the normal form. However, constraints or interactions that are enforced beyond the natural constraints (e.g., maintaining a nonequilibrium condition) may make the normal form unsolvable for the system as the complexity imposed by the nonnatural constraints may make the structure difficult or impossible to resolve. In addition, if the natural transformation cannot be performed from the system to the natural environment, this may indicate missing objects or relations in the system category structure. These can lead to stopping problems in software algorithms trying to resolve the category to its normal form.

The 'Forgetful Functor' maps higher dimensional structures to lower dimensional structures while maintaining the basic constraints and limits that exist on the lower dimensional structure. The functor essentially 'forgets' the higher dimensionality structure while maintaining lower dimensional structure contained in the specific category (i.e., system). This provides a mechanism to identify underlying relationships of the system in lower dimensional levels.

Functors between categories preserve the relationships between the internal objects of the category. For a system composed of subsystems, the Functor provides for the interconnection of the subsystems while maintaining the internal structure and relationships of the subsystems. For example, thermodynamic exergy provides for a system balance equation which preserves the subsystem thermodynamic relationships including the mass balance, energy balance, and entropy balance. Exergy can be expanded to show all of the constituent equations or abstracted to show only the subsystem interactions. This abstraction always preserves and is dependent on the internal relationships of the subsystems.

A 'Natural Form' of the category is one that is independent of any basis or coordinate system. This only considers the objects and relations of the category and not its representation in any basis system. This is important for systems as they are not defined by a coordinate system but can be represented in many different coordinate systems. Objects and relations that naturally occur in nature have this property. Thus, the natural form of the category provides a map of the category to the natural environment.

Cocones map the category onto a new set of indices. A set of parts contained in assembly bins are mapped into a final assembly by a Cocone. The term 'cone' or 'cocone' comes from the category representation of the different objects mapped onto a resulting object (the assembly is now a new object of the system category) which resembles a geometrical cone. A colimit also exists for this mapping that defines the direct limit (direct mapping) of the source object to the resultant object. This construction represents translating the design (a category let us call the Design Category) to the physical system (let us call the Physical Category). The cocone provides the mapping from a set of functors that show the assembly step for a given object (i.e., part) to the final assembly. The complete mapping of these functors is the cocone that results in the completed assembly. The Natural transformation, as mentioned above, leads to the identification of the resulting Category structure to the natural environment.

From a directed graph perspective, the Functors are each directed graphs of the parts into the assembly. The colimit is the integration of these individual part-directed graphs into the final assembly. This includes the sequencing of the parts in the assembly where one Functor then has a dependency on another Functor and order is important. Executing in the wrong order would not result in the intended final assembly, and some parts may not be included as a result.

4.11.4 Mathematical Category Theory Introduction

The mathematics that support the category definition of a system are presented in this section. The approach taken here views category theory as an organizational tool for concepts concerned with the design of structures at all levels of size and complexity (as found in system architectures). Such concepts include physics, mathematics, computational techniques, and the management of group collaborations consisting of possibly diverse subgroups; however, these notes will emphasize the mathematical and computational aspects of the applied theory.

To begin, the important notions of functions, partitions, equivalence relations, and quotient sets will be discussed. These notions are important in ways that will become evident in all that will follow. The initial discussion leading to the definition of categories will be in terms of directed graphs (digraphs). This approach and some notation involving it has been influenced by the classic reference by Mac Lane, for example, chapter I, paragraph 2.[69]

Specifically, a quick review of set theoretic concepts, an intuitive categorical view of sets and functions, will be given. Following that, directed graphs (digraphs) and a number of finite digraph-based examples will be given and categories will be defined. Then some basic mathematical structures that fit into the categorical framework developed above will be presented. This will include, semigroups, monoids, groups, rings, fields, modules over a ring, vector spaces, and algebras over a 'ground ring.' These examples will be used to provide intuition for notions of morphisms that generalize functions, functors that are correspondences between categories that generalize functions, and natural transformations that are correspondences between functors. The notion of natural transformations first appeared in the influential paper, "General Theory of Natural Equivalences,"[70] which formally introduced category theory.

4.11.4.1 Review of Set Theory. Informally, a set X is a collection of objects which are called elements. If x is an element of a set X, we write $x \in X$. If an element in a set is included more than one time, only one copy is considered and the others are ignored. So elements in a set are considered to be distinct. The order in which elements appear in a set is irrelevant. Thus, the set $X = \{2; 7; s; c; c; a; 2\}$ is considered to be the same as $\{7; a; s; c; 2\}$, etc.

A subset of X is a set A such that every element of A is an element of X, i.e., A is 'contained in' X. The notation $A \subset X$ is used to denote that A is a subset of X and not all of X itself. To denote that A is a subset that might also be the same as X, the notation $A \subseteq X$ is used.

The cardinality, i.e., the number of elements, of a set X is denoted by Card$|X|$ or just $|X|$ when the context is clear. The context is clear when the number of elements of X is finite. When X is infinite, the situation is more complicated. Intuitively, if the elements in X can be listed in order indexed by the natural numbers $N=\{0; 1; 2; : : :\}$, we say that X is countably infinite. Set theoretically, there are cardinalities larger than 'countably infinite' such as the cardinality of the real numbers.

The union of two subsets A and B of a set X is defined to be

$$A \cup B = \{x \in X \mid x \in A \ \text{ or } \ x \in B\} \ . \tag{146}$$

The intersection of two subsets A and B of a set X is defined to be

$$A \cup B = \{x \in X \mid x \in A \ \text{ and } \ x \in B\} \ . \tag{147}$$

It can occur that A and B have no elements in common. In that case, their intersection is empty and the sets are said to be disjoint. The set with no elements is denoted by Φ, the null set. So, A and B are disjoint if and only if $A \cup B = \Phi$.

The complement of a subset $A \subseteq X$ is

$$A^c = \{x \in X \mid x \notin A\}, \tag{148}$$

where the symbol $\notin$ is read 'not an element of.' Another notation that is often used is $A^c = X - A$.

4.11.4.1.1 Functions. A function f from a set X to a set Y is a correspondence such that to each element $x \in X$ there exists a unique element $y \in Y$ that corresponded to it (by assignment). This correspondence is written $y = f(x)$. We also say that $y \in Y$ is 'hit by x' when $y = f(x)$. This is consistent with the convention of calling X the source and Y the target of f. That convention will be used throughout section 4.11.4.

- Remark 1—Another way of rephrasing the definition above that is useful in defining functions in particular cases of the statement, 'if $x = y$ in X, then $f(x)=f(y)$ in Y.' When a correspondence f satisfies this definition, i.e., is a function, the correspondence is said to be 'well defined.'

The notation $X \xrightarrow{f} Y$ is quite often used to denote that f, a function from x to Y and the assignment of $f(x)$ to x is denoted by $x \mapsto f(x)$ which is read, 'x maps to $f(x)$".

- Remark 2—A function $X \xrightarrow{f} Y$ is said to be onto if every element in Y is of the form $f(x)$ for some element in X.

The function is said to be one-to-one (or just one-one) if $f(x) = f(x')$ implies $x = x'$.

- Remark 3—An important 'operation' involving functions is called composition. Given two functions $X \xrightarrow{f} Y$ and $Y \xrightarrow{g} Z$, there is an associated function $X \xrightarrow{g \circ f} Z$ defined by $(g \circ f)(x) = g(f(x))$ for each x in X.

In addition, as pointed out in section 4.11.4.1.5, for any set X, there is the identity function $X \xrightarrow{id_x} X$ defined by id$X(x)=x$ for all $x \in X$. It is noted here that, with respect to composition of functions, for a function $X \xrightarrow{f} X$ we have that

$$A^c = \{x \in X \mid x \notin A\}, \tag{149}$$

and

$$\left(f \circ id_x\right)(x) = f(x) \tag{150}$$

so that we always have $id_x \circ f = f$ and $f \circ id_x = f$ for functions f that map X to X.

4.11.4.1.2 Inverse Image and Partitions. Given a function $X \xrightarrow{f} Y$ and $y \in Y$, the inverse image of y is the subset

$$f^{-1}(y) = \{x \in X \mid f(x) \in Y\}. \tag{151}$$

Consider the set

$$P=\left\{f^{-1}(y)\,|\,y\in Y\right\}. \tag{152}$$

- Remark 4—If $f^{-1}(y)\cap f^{-1}(y')\neq\Phi$, then there is an element $x\in X$ such that $f(x)=y$ and $f(x)=y'$ and since f is well defined (see remark 1), this implies that $y=y'$. This means that the distinct elements of the set of inverse images P above are all pairwise disjoint.

Also, note that the union of all of the sets in P is all of X since the union is clearly a subset of X, but if $x\in X$, then obviously x $\in f^{-1}(f(x))$ so that x is in the union of all inverse images.

A set of pairwise disjoint subsets of a set X whose union is all of X is called a partition of X.

4.11.4.1.3 Products of Sets. If $X_1, X_2, \ldots, X_n$ is a list of sets for any $n\geq 2$, the product of these sets in the given order is

$$X_1\times X_2\times\ldots\times X_n=\left\{(x_1,x_2,\ldots,x_n)\,|\,x_i\in X_i, i=1,\ldots,n\right\}. \tag{153}$$

The element $(x_1, x_2, \ldots, x_n)$ is called an n-tuple. When $n=2$, it is usually called an 'ordered pair.'

4.11.4.1.4 Relations. A relation in X is any subset $R\subseteq X\times X$. The notation $x\underset{R}{\rightarrow}x'$ if and only if $(x,x')\in$ R will often be used in section 4.11.4.

- Remark 5—The relation

$$\Delta=\left\{(x,x)\in X\times X\,|\,x\in X\right\} \tag{154}$$

is the relation of equality. A given element $x\in X$ is related to an element x' if and only if $x=x'$. The equality relation Δ is also called the diagonal.

Note that the graph $G_f=\left\{(x,f(x))\,|\,x\in X\right\}$ of a function f is a relation in X. So functions may be thought of as special kinds of relations. In fact, the graph of the identity function

$$X\xrightarrow{id_x}Y \tag{155}$$

and

$$x\mapsto x \tag{156}$$

is Δ which is the geometric reason it is called the diagonal.

- Remark 6—Given a relation $R\subseteq X\times X$, the opposite relation is

$$R^{op}=\left\{(x',x)\,|\,(x,x')\in R\right\}. \tag{157}$$

4.11.4.1.5 Equivalence Relations and Quotient Sets. A relation $E \subseteq X \times X$ is called an equivalence relation if and only if it satisfies the three properties given by:

(1) $\Delta \subseteq E$,i.e., $x \xrightarrow[E]{}$ for all $x \in X$, (reflexive)

(2) $E^{op} \subseteq E$,i.e., $x \xrightarrow[E]{} x'$, then $x' \xrightarrow[E]{} x$, (symmetric)

and

(3) $x \xrightarrow[E]{} x'$ and $x' \xrightarrow[E]{} x''$, then $x \xrightarrow[E]{} x''$. (transitive)

A relation E that satisfies (1) above is called reflexive. It is called symmetric if it satisfies (2), and transitive if it satisfies (3).

An equivalence class for an equivalence relation E on X is defined for each element of X to be

$$[x]_E = \left\{ x' \in X \mid x \xrightarrow[E]{} x' \right\} . \qquad (158)$$

When the context is clear, the subscript E is dropped from $[x]E$ and we simply write $[x]$ for the equivalence class of x.

- Remark 7—The set of equivalence classes in X is defined to be the set of subsets

$$X \mid E = \{ [x] \mid x \in X \} . \qquad (159)$$

In fact, $X|E$ is a partition of X. To see why, assume that $[x] \cap [x'] \neq \Phi$. Then, there is some element z in both classes, i.e., if $x \xrightarrow[E]{} z$ and $x' \xrightarrow[E]{} z$. Since $z \xrightarrow[E]{} x$ by symmetry and transitivity, we have $x' \xrightarrow[E]{} x$. Similarly, $\xrightarrow[E]{} x'$.

So, suppose that $x'' \in [x]$. Then symmetry and transitivity with the above implies $x'' \xrightarrow[E]{} x'$ so that $[x] \subseteq [x']$. Analogously, $[x'] \subseteq [x]$ since every element x'' will also be in $[x]$ by the same reasoning and so $[x] = [x']$, hence $X|E$ is a partition.

The set X/E is called the quotient set of X by E. It is an important construction in mathematics. Note that if x and x' are related in X by E, then $[x] = [x']$ in X/E. Thus, intuitively, taking quotient sets turns 'equivalence' to 'equality.'

- Remark 8—While it is true that equivalence 'becomes' equality at the level of set theory, one has to be careful not to overgeneralize such a notion to other situations that will be encountered later in category theory.

Let $\mathbb{Z} = \{ \ldots, -3, -2, -1,\ 0;\ 1,\ 2;\ 3, \ldots \}$ denote that set of all integers.

Recall the result of the division algorithm. Given an integer $a \in \mathbb{Z}$ and a nonnegative integer n, there is an integer $q \in \mathbb{X}$ (called the quotient) such that $a = q\,n + r$ where $r \in \mathbb{Z}$ and $0 \leq r < n$.

Fix a nonnegative integer n. The relation mod_n,

$$a \xrightarrow[\text{mod}_n]{} b \quad \text{if and only if } a = b + qn \text{ for some } q \in \mathbb{X}\ , \tag{160}$$

defines an equivalence relation on $\mathbb{Z}$.

For each $[a] \in \mathbb{Z}/\text{mod}_n$ there is a unique nonnegative integer r such that $[a] = [r]$ so that the set of equivalence classes $\mathbb{Z}/\text{mod}_n$ is in one-to-one correspondence with the finite set $\{0, 1, \ldots, n-1\}$.

Note that we usually simply write $\mathbb{Z}/\text{mod}_n$ as $\mathbb{Z}/n$.

- Remark 9—Note the intersection of any number of equivalence relations is an equivalence relation. Thus, if $R \subseteq X \times X$ is any relation, there is a smallest equivalence relation containing it, viz, the intersection of all equivalence relations containing it. (Note that $X \times X$ is an equivalence relation that contains any such R.) There are algorithms, however, that do construct such a smallest equivalence relation 'extension' of a given R. The easy part is making a given R reflexive and symmetric in a minimal way. If it is not reflexive, simply take $R' = \Delta \cup R$ where Δ is the diagonal. If that is not symmetric, take $R'' = R' \cup (R')^{op}$. R'' will then contain R and be reflexive and symmetric. The 'tedious' part is in making R'' transitive if it is not already so. Doing that in a minimal way is said to produce the transitive closure of R'' which will then be an equivalence relation. There are efficient algorithms for calculating the transitive closure. Of note is Warshall's algorithm and more recently parallelized versions which may also be found.[71,72]

- Remark 10—Note that since the entire product set $X \times X$ is an equivalence relation, any relation $R \subseteq X \times X$ is contained in a unique equivalence relation, namely, the intersection of all equivalence relations in X that contain R.

For an interesting discussion of quotient sets in topology, the interested reader should see paragraph 4.5, Adjunction spaces, in reference 73.

4.11.4.2 Some Observations Concerning Sets. There are logical problems in thinking about the notion of a set of all sets. One cannot simply allow any proposition to define a set. The classic example is Russell's paradox[74] which is about the specification of a set X which does not contain itself as an element (one asks if X is an element of X and considers the consequences. Then one asks if X is not an element of X to see the paradox). For many reasons, including such paradoxes (there are more than just Russell's), there are axiomatic treatments of set theory[75] that avoid such paradoxes. There are various conventions for talking about a container for all sets and other collections that are 'too big' to be sets. A discussion of such conventions is given in chapter I, paragraphs 6 and 7 of reference 69. The convention taken here is to talk about a kind of universal container, called a class, which may contain all sets without logical difficulties. The formal reasons that this can be done require a proper reading of references such as the three mentioned above and will be left to the interested reader. A class that is not a set will be called a proper class. A class that is not a proper class is called a small class.

4.11.4.2.1 Sets as Nodes, Functions as Arrows. A notation for specifying a function *f* from a set *X* to Y, viz. $X \xrightarrow{f} Y$ has already been discussed. It suggests a kind of graph structure on the class of all sets. Indeed, in section 4.11.4.1.1, *X* has already been called the source and *Y* the target of *f*, so we alternately call a function *f* an arrow in this context. This leads to a view of sets and functions as 'nodes' and 'arrows' of a graph structure (a rather large one admittedly).

4.11.4.2.2 The Set of All Arrows From One Set to Another. We will often denote the set of all functions from a set *X* to a set *Y* by [*X*, *Y*]Set. Here, we will consider the number of possible functions from one finite set to another; i.e., the cardinality of the set [*X*, *Y*]Set when *X* and *Y* are finite. To begin, consider the set $X = \{0\}$ with one element and the set $Y = \{0, 1, \ldots, n-1\}$ with *n* elements. Clearly, the number of choices of a correspondence of zero to a unique element in *Y* is exactly *n*. We may denote these functions by $f_i(0) = i$ for $i = 0, \ldots, n-1$.

Now consider all functions from $X = \{0, 1\}$ to itself. All such functions may be conveniently denoted by

$$f_{i,j} = \begin{pmatrix} 0 & 1 \\ i & j \end{pmatrix}, \tag{161}$$

where $i, j \in Y$ with the possibly that $i = j$. It is easy to write down all of the choices, viz.

$$\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}. \tag{162}$$

Continuing, the distinct functions from *X* to $Y = \{0, \ldots, n-1\}$ may be enumerated by filling in the symbols *i*, *j* in the expression in equation (162) above where this time *i*, *j* in $Y = \{0, \ldots, n-1\}$ (with the possibility that i =j again). Obviously, there are *n* choices for associating (or 'mapping') 0, and following that, there are *n* choices for mapping 1. In all, this makes $n \bullet n = n^2$ choices.

Finally, consider the number of functions from $X = \{0, \ldots, n-1\}$ to $Y = \{0, \ldots, m-1\}$ which can be enumerated by the expression

$$\begin{pmatrix} 0 & \cdots & 1 \\ i_0 & \cdots & i_{n-1} \end{pmatrix}, \tag{163}$$

where $i, j \in$ Y with possible repeats. Again, clearly, there are *m* choices for 0, *m* choices for 1, and so on, making $m \bullet m \ldots \bullet m = m^n$ choices in all.

- Remark 11—A moment of thought shows that the same argument can be made for any finite sets X and Y in terms of counting functions because the counting argument does not depend upon what the elements of a finite set are labeled. The counting is the same whether the elements are numbered 0, 1, 2 or a, b, c, etc.

Thus, we see that the cardinality of $[\{0, \ldots, n-1\}, \{0, \ldots, m-1\}]\underline{\mathrm{Set}}$ is exactly m^n. In other words,

$$\left|[X,Y]_{\underline{\mathrm{Set}}}\right| = |Y|^{|X|}. \tag{164}$$

Because of this result, the notation Y^X is sometimes used to denote $[X, Y]\underline{\mathrm{Set}}$.

4.11.4.3 Directed Graphs and Free Path Algebras. A directed graph (digraph) $\mathcal{G} = \{G_0, G_1; s, t\}$ consists of two sets G_0 and G_1 along with two functions, $G_1 \xrightarrow{s} G_0$, $G_1 \xrightarrow{t} G_0$. The function s is called the source function and t is called the target function. The following notation will be used for this situation:

$$\begin{tikzcd} G_1 \arrow[r, shift left, "s"] \arrow[r, shift right, "t"'] & G_0 \end{tikzcd} . \tag{165}$$

The elements of G_0 are called nodes or vertices. The elements of G_1 are called arrows.

An arrow $a \in G_1$ such that $s(a) = t(a)$ is called a loop arrow or just a loop (at the node $s(a)$).

The following implicitly defines a typical digraph:

$$G_0 = \{a, b, c, d, e\} \tag{166}$$

and

$$\mathrm{G}_1 = \{|a, b|, |b, e|, |c, c|, |c, d|, |d, c|, |e, e|, |e, d|, |e, a|\}. \tag{167}$$

The arrows are denoted by lists of the form $\alpha = [x; y]$ where $s(\alpha) = x$ and $t(\alpha) = y$. This is not a universally used convention, but in some instances, it is convenient. Thus, the loops in the above graph are $[c, c]$ and $[e, e]$. Figure 26 illustrates the corresponding digraph.

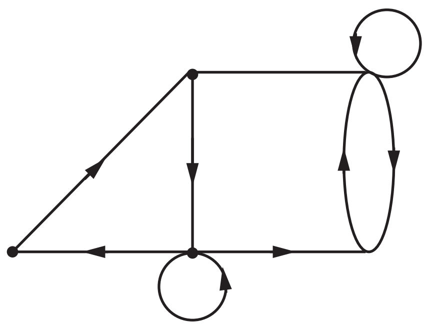


Figure 26. A typical digraph.

4.11.4.3.1 The Free Nonassociative Path Algebra. The reason for the title word 'nonassociative' will be explained in the next section.

Given a digraph $\mathcal{G} = \left\{G_0, G_1; s, t\right\}$ defines a sequence of sets $\widehat{G_{1,n}}$ inductively as follows:

- $\widehat{G_{1,1}} = G_1$.
- $\widehat{G_{1,2}}$ = the set of all parenthesized juxtapositions (ab) of elements in $\widehat{G_{1,1}}$ such that $s(b) = t(a)$, i.e., $\widehat{G_{1,2}} = \left\{(a,b) \mid a,b \in \widehat{G_{1,1}}, s(b) = t(a)\right\}$.
- For $(ab) \in \widehat{G_{1,2}}$, define an extension of the source and target functions by $s((ab)) = s(a)$ and $t((ab)) = t(b)$.

Suppose that we have $\widehat{G_{1,k}}$ as well as extensions of s and t for $2 \le k < n$:

- $\widehat{G_{1,n}}$ is the set of all parenthesized juxtapositions of elements (cd) where $c \in G_{1,i}$ and $d \in G_{1,j}$ with $1 \le i, j$ and $i + j = n$, and $s(d) = t(c)$.

Note that every element $\alpha \in \widehat{G_{1,n}}$ consists of a sequence $a_1, a_2, \ldots, a_n \in G_1$ with parentheses in various positions between the a_i's and $s(\alpha) = s(\alpha_1)$, $(\alpha) = t(\alpha_n)$.

Now define

$$\widehat{G_1} = U_{n \ge 1} \widehat{G_{1,n}} \tag{168}$$

and let

$$\widehat{G_1} \times_{G_0} \widehat{G_1} = \left\{(a,b) \in \widehat{G_1} \times \widehat{G_1} \mid s(b) = t(a)\right\}. \tag{169}$$

Finally, define an operation called 'composition' by

$$\widehat{G_1} \times_{G_0} \widehat{G_1} \to \widehat{G_1} \tag{170}$$

and

$$(a,b) \mapsto (ab). \tag{171}$$

This operation is often denoted by $a \circ b = (ab)$.

Note that the operation is well defined since any element a must be an element of some $\widehat{G_{1,\imath}}$ and similarly, b must be an element of some $\widehat{G_{1,j}}$ so $(ab) \in \widehat{G_{\imath+j}} \subseteq \widehat{G_1}$.

4.11.4.3.2 Degree of a Word. As defined above, every element $\alpha \in \widehat{G_1}$ consists of a sequence $a_1, \ldots, a_n \in G_1$ with parentheses in various positions between the a_i's for some n. We call α a word in the indicated elements. We define the degree of such an element to be $\deg(\alpha) = n$. More can be said however, by construction, α is actually a concatenation of an element from $\widehat{G_{1,i}}$ and $\widehat{G_{1,j}}$ where $i + j = n$. We define the bidegree of α to be (i,j).

4.11.4.3.3 The Free Path Algebra. Associativity: A function $S \times S \overset{m}{\rightarrow} S$ such that $m(m(a, b), c) = m(a,m(b, c))$ for all $a, b, c \in S$ is called an associative operation. If we write $m(a, b) = ab$, this becomes the more familiar rule that $a(bc) = (ab)c$ as is known to hold for the usual operation of multiplications of integers and composition of functions. Because these two expressions are equal, one can unambiguously write abc for either one of them, thereby forgetting about any parentheses in products of such elements.

The free associative path algebra on graph $\mathcal{G} = \{G_0, G_1; s, t\}$ is a mathematical structure that is much easier to visualize than the free, nonassociative one because of the comments above. In $\widehat{G_2}$, there are distinct elements of the form $(a(bc))$ and $((ab)c))$ which indeed indicates that the path product m is not associative in general. But if associativity is assumed, all parentheses can be removed and all words of degree n are simply concatenations of n elements from G_1 such that the source of a factor of the concatenation whose position is >1 is the target of the previous factor.

Generally, we call the free associative path algebra simply the free path algebra, dropping the word 'associative.' If we need to refer to the nonassociative path algebra, the word 'nonassociative' will be explicitly used.

The free path algebra on a graph $\mathcal{G} = \{G_0, G_1; s, t\}$ will be denoted by $\mathcal{P}(\mathcal{G})$.

- Example 1—Consider the digraph defined by $G_0 = \{1\}$, $G_1 = \{\mathrm{a}\}$, and $s(a) = 1$; $t(a) = 1$. This graph is illustrated in figure 27. Clearly, in this case, the set of all path products is $\widehat{G_1} = \{a, aa, aaa, aaaa, \ldots\}$. The notation can be abbreviated by writing $a_1 = a$ and $aa \ldots a$ (n-times) as a_n. By the definition of path product and associativity, we then clearly have $a^n \circ a^m = a^{n+m}$.

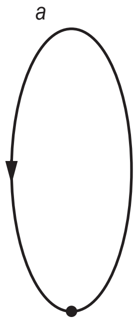


Figure 27. One node, one arrow digraph.

- Remark 13—A set S with an operation

$$S \circ S \overset{\circ}{\to} S \ , \tag{172}$$

which is associative is called a semigroup. Thus, in the example above, the free path algebra $\widehat{G_1}$ is an example of a semigroup.

4.11.4.3.4 Identities. In ordinary multiplication of integers, $\mathbb{Z}$, the distinguished element 1 has the property that $1 \bullet n = n$ and $n \bullet 1 = n$ for all $n \in \mathbb{Z}$. Such an element is called an identity element. The path algebra on a graph $\mathcal{G}$ may possess identity elements 'at each node' if there are distinguished arrows that satisfy the property that, in addition to the source and target maps s and t, there is a function

$$G_0 \overset{\text{ids}}{\to} G_1, \tag{173}$$

such that

$$s \circ ids = ids_{G_0}, \text{ and } t \circ ids = id_{G_0}. \tag{174}$$

These conditions imply that, at each node $n \in G_0$, there is an arrow that will be denoted by $id_n = ids(n)$ such that $s(id_n) = s(ids(n)) = n$ and $t(id_n) = t(ids(n)) = n$, i.e., that each id_n is a loop at n for all nodes in G_0.

With the conditions above, it is assumed that the id_n loops act as identity elements in the path algebra $\mathcal{P}(\mathcal{G})$. This combined structure is described succinctly in section 4.11.4.3.5.

4.11.4.3.5 The Free Path Algebra With Identities. The free path algebra with identities has the following structure:

$$\begin{tikzcd}\mathcal{P}(\mathcal{G}) \arrow[r, shift left, "s"] \arrow[r, shift right, "t"'] & G_0\end{tikzcd} \tag{175}$$

along with

$$G_0 \overset{ids}{\to} \mathcal{P}(\mathcal{G}) \tag{176}$$

such that

$$s \circ ids = id_{G_0} \tag{177}$$

and

$$t \circ ids = id_{G_0} \tag{178}$$

with an induced associative operation

$$\widehat{G_1} \times_{G_0} \widehat{G_1} \xrightarrow{m} \widehat{G_1} \tag{179}$$

such that the loops id_n for $n \in G_0$ act as identities.

- Example 2—Consider the digraph defined by $G_0 = \{0\}$, $G_1 = \{id_0, 1\}$, $s(1) = 0$; $t(1) = 0$, $s(id_0) = 0$; $t(id_0) = 0$ and identity $ids(0) = id_0$. This graph is illustrated in figure 28 (with the identity arrow id_0 omitted but assumed). In this case, the operation of path composition will be written as +.

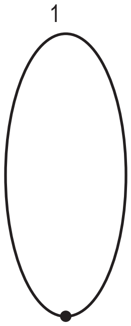


Figure 28. One node, one arrow identity digraph.

Thus, the set of all path compositions is $\widehat{G_1} = \{id_0, 1, 1+1, 1+1+1, \cdots\}$. Again, we can abbreviate the notation, but this time, we will write $1+1+\ldots+1$ (n-times) as n. By the definition of path composition and associativity, we then clearly have $n \circ m = n + m$ where, on the left we mean $1+\ldots+1$, $(n+m)$-times. Note that in this case, we also have $n+m = m+n$ as is clear from the definition of path product. Also, by the properties of identity loops, we have $id_0 + n = n$ and $n + id_0 = n$. We denote id_0 in this case by 0.

- Remark 14—A set M with an operation

$$M \times M \overset{\circ}{\rightarrow} M, \tag{180}$$

which is associative and has an identity element, is called a monoid. Thus, in the example above, the free path algebra $\widehat{G_1}$ with identities is an example of a monoid (where we write the operation as + instead of $\circ$ in this case. We have that

$$\widehat{G_1} = \{0, 1, 2, \ldots, n, \ldots\} \tag{181}$$

with $0 + n = n$, and $n + m = m + n$. This monoid is the same as the set of nonnegative integers with addition and identity element zero.

4.11.4.4 Categories. The above free (associative) path algebra with identities is a model for the general definition of a category; however, while this mathematical structure was constructed artificially, we have seen a naturally-occurring example, viz. sets, functions, composition of functions, and identity maps. This last comment needs some explanation and that is given below where sets and functions as a category are presented.

In general, while we take the free path algebra with identities as a model for categories, we do not require a category to be exactly of this form. In fact, the definition of a category is as follows:

- Definition 1—A category C consists of two classes C_0 and C_1 and two well-defined correspondences:

$$C_1 \overset{s}{\underset{t}{\rightrightarrows}} C_0 \tag{182}$$

so that $s \circ ids = id_{C_0}$ and $t \circ ids = id_{C_0}$. It furthermore is supposed that there is an operation $\circ$ of the form

$$C_1 \times_{C_0} C_1 \overset{\circ}{\to} C_1 , \tag{183}$$

$\left(\text{where } C_1 \times_{C_0} C_1 = \left\{(f,g) \,|\, C_1 \times C_1, s(g) = t(f)\right\}\right)$, which is associative and which there are identities with respect to this operation:

$$C_0 \overset{ids}{\to} C_1 \ . \tag{184}$$

Categories will always be denoted with an underline as in $\underline{C}$.

- Remark 15—The free path algebra with identities satisfies this definition with respect to $\circ$ given by the path composition. Thus, $\underline{G} = \left(\widehat{G_1}, G_0, s, t, ids\right)$ is a category. It is called the free category on the digraph with identity loops $\left(G_0, G_1, s, t, ids\right)$.

- Example 3—Here is an interesting yet very simple category.

Consider the digraph given by $G_0 = \{n\}$, $G_1 = \{id_n, 0, 1\}$, $s(0) = n$; $t(0) = n$; $s(1) = n$, $t(n) = n$, and identity $ids(n) = id_n$. This graph is illustrated in figure 29 (again, with the identity loop omitted but assumed).

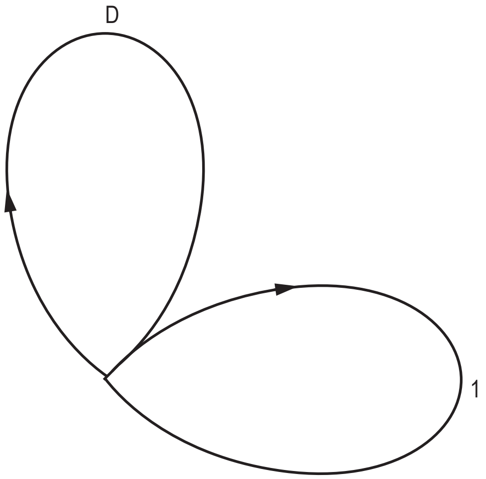


Figure 29. One node, two arrows, and one identity arrow.

This time, $\widehat{G_1}$ consists of the identity loop id_n and all words in 0 and 1. Note however that 01 is not the same as 10. Nonetheless, this set is a monoid.

- Example 4—Consider the digraph with Set$_0$ equal to the proper class of all sets and Set$_1$ equal to the class of all functions from a set to any other set. For a function $X \xrightarrow{f} Y$, we have $s(f) = X$ and $t(f) = Y$. We interpret the path product as composition of functions. The identities are $ids(X) = id_X$. Since composition of functions is associative, this structure, which will be denoted by Set, is a category.

- Remark 16—Consider again the category from example 3 which we will denote by Mach. If we interpret the words not equal to the identity idn as encoding words in English via the ASCII encoding,[76] the phrase

  01001000 01100101 01101100 01101100 01101111
  01110111 01100111 01110010 01101100 01100100

which reads ‘Hello world’ is encoded in the category Mach. It is interesting to note that all of written history may be encoded in this small category.

4.11.4.4.1 Some Standard Terminology. Let $C = (C_0, C_1, s, t, ids)$ be a category. The class of nodes C_0 is often called $Ob(C)$ and the class of arrows C_1 is called $Arr(C)$. For two objects (nodes) $c_1, c_2 \in Ob(C)$, the class of all arrows from c_1 to c_2 is denoted by either $\left[c_1, c_2\right]C$ or hom$C\left(c_1, c_2\right)$.

4.11.4.4.2 A Strong Correspondence Between Two Categories. Consider two one node, two loop categories generated freely by the graphs in figure 30. Let *C* denote the category generated by the digraph on the left.

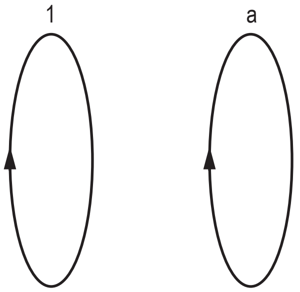


Figure 30. Two one node, one arrow, one identity digraphs.

As in example 2, let *n* denote $1 + :::+1$ (*n*-times) and let 0 denote id_0.

The set of arrows of $\underline{C}$ is

$$Arr(\underline{C}) = \{id_0, 1, 2, \ldots, n, \ldots\} \tag{185}$$

and the path product is $n \circ m = n + m$.

As in example 1, let a^n denote $aa \ldots a$ (*n*-times). For this example, however, we include the identity id_1 which we denote by 1. The set of arrows is

$$Arr(\underline{C'}) = \{1, a, aa, \ldots, a', \ldots\} \tag{186}$$

and the path product is $a^n \circ a^m = a^{n+m}$.

Now, it is quite apparent that these two categories are essentially the same. However, the word 'essentially' needs to be made more clear. To make a complete comparison, we need to compare nodes, identity loops, and other arrows of both categories. That means that we need not only a correspondence between $Ob(\underline{C})$ and $Ob(\underline{C'})$, but also one between $Arr(\underline{C})$ and $Arr(\underline{C'})$.

Figure 31 gives such correspondences; namely, we define

$$Ob(\underline{C}) \overset{\exp}{\rightarrow} Ob(\underline{C'}) \tag{187}$$

$$0 \mapsto 1 \tag{188}$$

and

$$Arr(\underline{C}) \overset{\exp}{\rightarrow} Arr(\underline{C'}) \tag{189}$$

$$n \mapsto a^n . \tag{190}$$

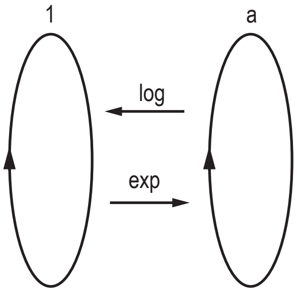


Figure 31. Comparing two categories.

Both of these correspondences are well defined, and furthermore, they are clearly one-one and onto and have inverses. In fact, we have that the inverse functions are given by

$$Ob(\underline{C'}) \xrightarrow{\log} Ob(\underline{C}) \tag{191}$$

$$1 \mapsto 0 \tag{192}$$

and

$$Arr(\underline{C'}) \xrightarrow{\log} Arr(\underline{C}) \tag{193}$$

$$a^n \mapsto n . \tag{194}$$

Note, furthermore, that these functions preserve path compositions namely, $\exp(n+m) = a^{n+m}| = a^n a^m = \exp(n)\exp(m)$ and $\log(a^n a^m) = \log(a^{n+m}) = n + m = \log(a^n) + \log(a^m)$. Because of this, we say that the two categories are isomorphic; i.e., they are essentially the same category.

4.11.4.4.3 Functors. A functor between two categories is a generalization of the above example; however, in general, they are much weaker than being isomorphisms.

A functor $\underline{C} \xrightarrow{f} \underline{D}$ consists of correspondences of the form

$$Ob(\underline{C}) \xrightarrow{f} Ob(\underline{D}) \tag{195}$$

and

$$Ob(\underline{C}) \xrightarrow{f} Ob(\underline{D}) \tag{196}$$

for every pair of objects $C,D \in \underline{C}$ such that compositions are preserved, i.e., on morphisms, $f(\alpha \circ \beta)=f(\alpha) \circ f(\beta)$. *hom* represents the set of relationships (homomorphisms) for the subscripted category.

Such a functor is sometimes called a covariant functor because it preserves the direction of arrows. Shortly, we will see an example of a functor that reverses the direction of arrows. That kind of functor is called a contravariant functor.

4.11.4.4.4 The Dual of a Category. Given a category $\underline{C}=(\underline{C_0}, \underline{C_1}, s, t, ids)$, the dual category $\underline{C^{op}}$ has the same class of objects as $\underline{C}$, but its arrows are reversed; i.e., $s^{op}=t$ and $t^{op}=s$.

Note that a contravariant functor $\underline{C} \xrightarrow{f} \underline{D}$ is the same as a covariant functor $\underline{C^{op}} \xrightarrow{f} \underline{D}$.

4.11.4.4.5 Relations in a Category. Relations in a category $\underline{C}$ are relations R_{c_1,c_2} in $\hom_{\underline{C}}(c_1,c_2)$ for all $c_1 c_2 \in Ob(\underline{C})$. Thus, $R_{c_1,c_2} \subseteq hom_{\underline{C}}(c_1,c_2) \times hom_{\underline{C}}(c_1,c_2)$. As with sets, a relation E_{c_1,c_2} is called an equivalence relation if it is reflexive, symmetric, and transitive. We say that the family $\left\{E_{c_1,c_2}\right\}$ is preserved by compositions if the correspondence

$$hom_{\underline{C}}(C,D) \xrightarrow{f} hom_{\underline{D}}(f(C), f(D)) \tag{197}$$

given by

$$[g] \circ [f] = [g \circ f] \tag{198}$$

is well defined (see sec. 4.11.4.1.1).

Being well defined in the case directly above means that the correspondence

$$([g] ,[f]) \mapsto [g \circ f] \tag{199}$$

actually defines a functional correspondence. In other words, the assignment is unique, and in still other words, that if $([g],[f])=([h],[k])$ then $[g \circ f]=[h \circ k]$.

The above amounts to saying that if g is related to g' and h is related to h', then $g \circ f$ is related to $h \circ k$. So there is nothing really to 'prove' here; it is an exercise in unraveling definitions.

- Remark 17—If it is required to design a category for some specific use, one way to proceed consists of specifying an appropriate digraph, specifying relations in the corresponding free category that the digraph generates and extending those relations to equivalence relations that preserve compositions.

The category formed by taking the quotient classes of the arrows by the given equivalence relations is then the desired end category. It is possible to extend relations as above to equivalence relations that preserves compositions. Earlier, in Remark 9, it was noted that there is a smallest equivalence relation containing a given relation. The same consideration may be used to construct a smallest equivalence relation that preserves compositions (when they are present). The interested reader should also see chapter I, paragraph 8, in Mac Lane.[69]

4.11.4.5 Some Basic Mathematical Categories. The following define some basic mathematical categories that are of use in the definition of systems.

4.11.4.5.1 The Category of Semigroups. The category SemiGp has objects consisting of all semigroups and arrows (or morphisms as they are also called) all functions $S_1 \xrightarrow{f} S_2$ on the underlying sets that satisfy $f(st)=f(s)f(t)$. The set of arrows is denoted as usual by either $[S1, S2]_{\underline{\text{SemiGp}}}$ or $hom_{\underline{\text{SemiGp}}}(S_1, S_2)$. Such morphisms are called semigroup maps or semigroup morphisms.

4.11.4.5.2 The Category of Monoids. The category Monoid has objects consisting of all monoids and arrows (or morphisms as they are also called) all functions $M_1 \xrightarrow{f} M_2$ on the underlying sets that satisfy $f(st)=f(s)f(t)$ and $f(1)=1$. The set of arrows is denoted as usual by either $[M_1, M_2]_{\underline{\text{Monoid}}}$ or $hom_{\underline{\text{Monoid}}}(M_1, M_2)$. Such morphisms are called monoid maps or monoid morphisms.

4.11.4.5.3 The Category of Groups. A group G is a monoid such that every element $g \in G$ has an inverse denoted by g^{-1}. That means that $g\,g^{-1}=1$ and $g^{-1}g=1$.

The category Grp has objects consisting of all groups and arrows (or morphisms as they are also called) all functions $G_1 \xrightarrow{f} G_2$ on the underlying sets that satisfy $f(ab)=f(a)f(b)$ and $f(1)=1$. It follows that $f(g^{-1})=f(g)^{-1}$.

The set of arrows is denoted as usual by either $\left[G_1, G_2\right]_{\underline{\text{Grp}}}$ or $\text{hom}_{\underline{\text{Grp}}}\left(G_1, G_2\right)$. Such morphisms are called group maps or homomorphisms (a word that, historically, no doubt inspired the word 'morphism' in general).

- Remark 18—In all cases of semigroups, monoids, and groups, if the operation satisfies an additional condition called commutativity, namely, xy = yx for all x and y in the underlying set object, the operation is denoted by '+' instead of '–' (or just juxtaposition) and the identity element is denoted by 0 instead of 1.

Thus, if G is a commutative group, then $x+y=y+x$ for elements of G and for all x in G, we have $x+0=x$ and $x+(-x)=0$. In general, we define $x-y=x+(-y)$ in a commutative group.

Commutative semigroups, monoids, and groups are often called 'abelian' in honor of the mathematician Niels Henrik Abel (1802–1829).[77,78]

4.11.4.5.4 The Category of Abelian (Commutative) Groups. The category of abelian groups, AbGprp consists of all commutative groups with morphisms exactly the same as in Rgp.

- Remark 19—Given a set X with an operation •, we denote the mathematical system consisting of the set with its operation • by (X, •). If there are any distinguished elements like an identity element 1, we denote the system by (X, •,1), etc.

- Example 5:

  – The system $(N, +, 0)$ where $N = \{0, 1, 2, \ldots\}$ is the set of nonnegative integers and the operation is the usual one of addition and the identity element is 0 is an abelian monoid.

  – The system $(\mathbb{Z}, +, 0; -)$ where $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$ and + is the usual operation of addition, 0 is the identity element, and – is the usual operation for negation for inverses is an abelian group.

  – The monoid in Remark 16 consisting of all noncommutative words in 0 and 1 with identity element id_n is a monoid that is not abelian.

  Let $\mathbb{Q} = \frac{p}{q} | p, q \in \mathbb{Z}$ be the set of rational numbers, $\mathbb{R} = \{n.a_0 a_1 a_2 \ldots | n \in \mathbb{Z}, a_i \in \{1, 2, 3, 4, 5, 6, 7, 8, 9\}\}$ be the set of real numbers, and $\mathbb{C} = \left\{a + b\ i \mid a, b \in \mathbb{R};\ i = \sqrt{-1}\right\}$ be the set of complex numbers:

  – The systems $(\mathbb{Q}, +, 0, -), (\mathbb{R}, +, 0, -)$, and$(\mathbb{C}, +, 0, -)$ are all abelian groups with the commutative operation of usual addition, identity element 0, and – as inverse operation for addition.

4.11.4.5.5 The Category of Commutative Rings With Identity. A commutative ring is a system of the form $(R, +, 0, -, *)$ where $(R, +, 0, -)$ is a commutative (abelian) group and $R \times R \overset{\circ}{\rightarrow} R$ is a commutative (abelian) semigroup such that the following distributive laws hold:

$$r * (s + t) = r * s + r * t \tag{200}$$

and

$$r * (s + t) = r * s + r * t \tag{201}$$

As usual, the multiplication operation is often written simply as juxtaposition.

A commutative ring with identity is a system $(R, +, 0, -, *, 1)$ (R, +, 0, –, *) is a commutative ring and $(R;, *, 1)$ is a commutative (abelian) monoid.

The category $\underline{\text{AbRng}_1}$ has objects all commutative rings with identity and morphisms all functions $R \xrightarrow{f} S$, where R and S are objects in $\underline{\text{AbRng}_1}$ which are abelian group morphisms with respect to + and abelian monoid morphisms with respect to *.

Clearly, the systems $(\mathbb{Z}, +, 0, -, *, 1)$, $(\mathbb{Q}, +, 0, -, *, 1)$, $(\mathbb{R}, +, 0, -, *, 1)$, and $(\mathbb{C}, +, 0, -, *, 1)$ are all commutative rings with identity with respect to the usual operations indicated. Recall the quotient sets $\mathbb{Z}/n$ in remark 7. Then $(\mathbb{Z}/n, +, 0, -, *, 1)$ is a commutative ring with identity for every nonnegative integer n with the operations given by

$$[a]+[b]=[a+b] \ , \tag{202}$$

$$[a]*[b]=[a*b] \ , \tag{203}$$

and

$$-a=[-a] \tag{204}$$

and constants $0=[0]$, $1=[1]$. This amounts to showing that the functions defining these operations are well defined—a simple exercise in arithmetic (when the problem is organized properly).

- Remark 20—While noncommutative rings are studied in mathematics, only commutative rings with identity will be considered in this particular application.

4.11.4.5.6 The Category of Fields. A field *F* is a mathematical system such that $(F,+,0,-,*,1)$ is a commutative ring with identity and $(F,-,0,\_,1,/)$ is a commutative group. Note that the inverse is conventionally written as $r^{-1}=1/r$ and a/b is equal to $a*b^{-1}$. Each of $\mathbb{Q}, \mathbb{R}$, and $\mathbb{C}$ are fields with respect to the usual operation of division. The ring of integers is not a field. The category Fld has all fields as objects and arrows all functions that preserve the operations as arrows.

4.11.4.5.7 The Category of Modules Over a Ring. A module *M* over a ring *R* (or an *R*-module as it is also called) is an abelian group structure $(M,+,0,-)$ along with an operation (sometimes called an 'action') of R on M, $R\times M \overset{\mu}{\rightarrow} M$ that satisfies the properties that, writing $\mu(r,m)=rm$,

$$r(m+m')=rm+rm' \ , \tag{205}$$

$$(r+s)m=rm+sm, \tag{206}$$

$$rsm=r(sm), \tag{207}$$

and

$$1m=m, \text{ if } m \text{ has an identity element.} \tag{208}$$

- Remark 21—It is worth noting at this point that if M is an abelian group, then the set of arrows (morphisms) homAbGrp(M,M) has the structure of a ring with identity element. The operation of addition is given by

$$(f+g)(m)=f(m)+g(m) \tag{209}$$

and the operation of multiplication is given by

$$fg=f\circ g \ , \tag{210}$$

where ∘ denoted composition of functions.

The ring $\hom_{\underline{\text{AbGrp}}}(M,M)$ is usually denoted by End(M) and called the endomorphism ring of M.

Note that, in general, the multiplication to make End(M) into a ring with identity is not commutative. Later, this will become clear when we identify End(M) as a ring of matrices in special cases.

Furthermore, note that if M is an R-module, then End(M) is also and R-module where $(rf)(m) = rf(m)$ for a module morphism f. Finally,

$$R \overset{\rho}{\rightarrow} \text{End}(M) \tag{211}$$

and

$$r \mapsto (m \mapsto rm) \tag{212}$$

is a ring with identity morphism. The morphism ρ is called a representation of R on M. The category of modules over a ring is denoted by $\underline{R_{\text{Mod}}}$.

4.11.4.5.8 The Category of Vector Spaces Over a Field. For basic information about vector spaces, matrices, and linear algebra in general, the textbooks in references 79 through 81 are useful. Consider a vector space of module V over a field F. The extra operation of division in F considerably enriches the 'computational power' of vector spaces compared to modules and this partially accounts for it prevalence as a computational tool in engineering. However, do not be misled into thinking that modules are not important computationally as well. The operation of division is quite useful when it is available, but that is not always the case. The category of vector spaces over a field F is denoted by $\underline{{}_F\text{Vect}}$ so that, of course, $\underline{{}_F\text{Vect}} = \underline{{}_F\text{Mod}}$. Much more will be said about this category shortly.

4.11.4.5.9 The Free R-Module on a Set X. There is an explicit construction of a type of R-module that has useful properties shared with vector spaces. Recall that every vector space V over a field F has a basis $B \subseteq V$. Bases are characterized by the fact that every linear combination of the form $\sum_{i=1}^{n} r_i b_i$, where $r_i \in F$ and $b_i \in B$ is unique. If X is a set and R is a ring, we can construct an R-module M with this 'basis property' P with respect to $X \subseteq M$, i.e., every linear combination of the form where $r_i R$ and $x_i X$ is unique.

Here is the construction. Let M be the set of all formal linear combinations of the form $\sum_{i=1}^{n} r_i x_i$, where $r_i \in R$ and $x_i \in X$. To make things completely unambiguous, one can take such a linear combination to mean a list L of pairs (r, x) where $r \in R$ and $x \in X$. Thus, for example, the linear combination above is represented by the list

$$L = \left[(r_1, x_1),\ (r_2, x_2),\ \ldots, (r_n, x_n)\right], \tag{213}$$

and to be even more explicit, we can take lists to be ordered n-tuples in n-time products of the set $R \times M$. The additive structure on such linear combinations is exactly as it is for vector spaces and the action of R (also called scalar multiplication) on M is also as it is for vector spaces. We need the convention that in this context, there is an ‘empty’ list and that is the zero element of M. This construction will be called the free R-module on X and denoted by FreeMod(R,X).

4.11.4.6 A Categorical Properties of Modules Over a Ring. Everything done in this section will be true for modules over a ring as well as vector spaces over a field, i.e., no reference or reliance upon the operation of division or, more notably, on the existence of a basis will be necessary.

Consider the correspondence $Ob(RMod) \xrightarrow{D} Ob(RMod)$ given by

$$D(M) = \hom RMod(M, R). \tag{214}$$

This set of morphisms is indeed another R-module since we can add such morphisms by ‘adding pointwise,’ i.e., $(\alpha+\beta)(m) = \alpha(m) + \beta\,(m)$ and scalar multiplying by $(r\,\alpha)(m) = r\,\alpha(m)$ and the necessary relations can easily be checked to see that this turns $D(M)$ into an R-module. $D(M)$ is most often denoted by M^* and is called the dual module of M. It turns out that this correspondence is the first part of a contravariant functor. On hom sets (i.e., arrows), the correspondence $D(f) = f^*$ is given as follows. If $V \xrightarrow{f} W$ is a morphism and the morphism $W \xrightarrow{\beta} R$ is given, the corresponding morphism $V \rightarrow R$ is given by $f^*(\beta)(v) = (\beta \circ f)(v) = \beta\big(f(v)\big)$.

4.11.4.7 Cocones and Colimits. A diagram in a category C is the image of a functor $D \xrightarrow{f} C$. A cocone (due to the cone shape of the diagram shown in eq. (215)) in C over F with vertex $C \in C$ is a correspondence collection of morphisms $F(z) \xrightarrow{c_x} C$ such that all diagrams of the form

$$\begin{tikzcd} F(x) \arrow[rr, "\xi"] \arrow[dr, "c_x"'] & & F(y) \arrow[dl, "c_y"] \\ & C & \end{tikzcd} \tag{215}$$

where $\xi = F(f)$ and $f \in hom_{\underline{D}}(x,y)$ commute, i.e., $c_x\,\xi = c_x$.

The vertex f a cocone as above is said to be a colimit if over morphism from an object C to another object C' in C which is also a vertex of a cocone over F is uniquely determined by morphisms from the $F(z)$ to C' for which the maps within the cocone for C' are ‘compatible’ with the morphisms in the cocone for C in a sense that will be made precise in the example below.

The corresponding cocone involves the category generated by the digraph (D_0, D_1, s, t, ids) where $G_0 = \{x, y\}$, $G_1 = \{idx, idy;\ \alpha;\ \beta\}$, $s(a) = x$ and $t(a) = y$ for a $\in G_1$. Thus, G has the shape

$$x \underset{\beta}{\overset{\alpha}{\rightrightarrows}} y\ \cdot \tag{216}$$

Since there are no path products other than the compositions with the identities, the entire category $\underline{D}$ generated by G has the same shape. So, a diagram in $\underline{C}$ over a functor F is of the form

$$\begin{tikzcd} A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & B, \end{tikzcd} \tag{217}$$

where $F(x)=A$, $F(y)=B$, $F(\alpha)=f$, and $F(\beta)=g$.

A cocone with vertex C then consists of a commutative diagram

$$\begin{tikzcd} A \arrow[rr, "\xi"] \arrow[dr, "c_x"'] & & B \arrow[dl, "c_y"] \\ & C & \end{tikzcd} \tag{218}$$

where $\xi \in \{f,g\}$. Now note that given such a situation, we may define a diagram

$$\begin{tikzcd} A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & B \arrow[r, "c"] & C, \end{tikzcd} \tag{219}$$

where $c=c_y$ and then we have $cf=cg$ from the cocone condition. Conversely, however, if we have a diagram such as the one above, we may form a commutative diagram

$$\begin{tikzcd} A \arrow[rr, "\xi"] \arrow[dr, "c_x"'] & & B \arrow[dl, "c_y"] \\ & C & \end{tikzcd} \tag{220}$$

where $\xi \in \{f,g\}$ and $c_x = cf$ and $c_y = c$. Thus, the two situations are equivalent. We call the object C in the situation in equation (218) a colimit if for any other diagram of the form

$$\begin{tikzcd} A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & B \arrow[r, "c'"] & C', \end{tikzcd} \tag{221}$$

where c' with $c'f=c'g$ there is a unique morphism $C \overset{\phi}{\to} C'$ such that $c' = \phi \circ c.$ Note that this means that any morphism from C to another object C' is completely determined by a map c' from B to C' for which $c'f=c'g$.

Essentially any map from a colimit object is given uniquely by a map from the ‘parts’ of the cocone of which it is the vertex as long as some compatibility conditions are satisfied. Colimits in $\underline{\text{Set}}$ and many other categories can be constructed using co-equalizers. Obviously, for this to be useful, one needs to know how to construct colimits in such categories. The basic idea is to take $C = B/E$ where E is the equivalence relation generated by the relation $f(a) \xrightarrow[E]{} g(a)$ for all $a \in A$.

- Definition 2—A cocone is atomic if all the nodes comprising it except the vertex are considered to be indecomposable, i.e., not vertexes of cocones themselves.

A system is the vertex of either an atomic final cocone or a final cocone whose nodes are recursively systems.

4.12 System Model Integration

The various system models provide different views and understanding of the system. These different system models for a given system have common points that allow the sharing and transfer of information between the models. This also provides a common basis for the system models ensuring consistency in their application.

4.12.1 System Representations

Systems engineering constructs and maintains a suite of models that represent various aspects of the system (subprinciple 5(b) and strategies in sec. 3.3.2). One of the most important questions for systems engineering is determining the minimum number of models, or model types, necessary to perform systems engineering functions. In addition, systems engineering must address both success and failure (achievement of goals and nonachievement of goals). The hypothesized minimum suite of model types include value, intention, design, failure, behavior, performance, and agency. Finally, we assume that the suite of models for systems engineering must inherently connect to existing models, which generally are either disciplinary or product focused. Existing models reflect discipline engineering structures such as mechanical engineering, electrical engineering, control engineering, etc., or models that relate to the product that a given institution delivers, which often combines information from the disciplines within that organization.

Figure 32 shows the proposed set of minimum model types, and the typical flow of information between these models. These models are arranged in a loop, indicating that models are developed and updated in a typical circular pattern. Over the lifecycle of the system, multiple passages through the loop forms a spiral as models are progressively refined and eventually compared to the real system.

The engineering disciplines operate with formal models built, refined, and used to inform their portion of the system design as well as assess its performance and help to verify and validate it. Systems engineering should be constructed and functions in the same way, but its model set integrates the various discipline models, which exist ‘outside’ but nonetheless ‘inform’ the system models shown in figure 32.

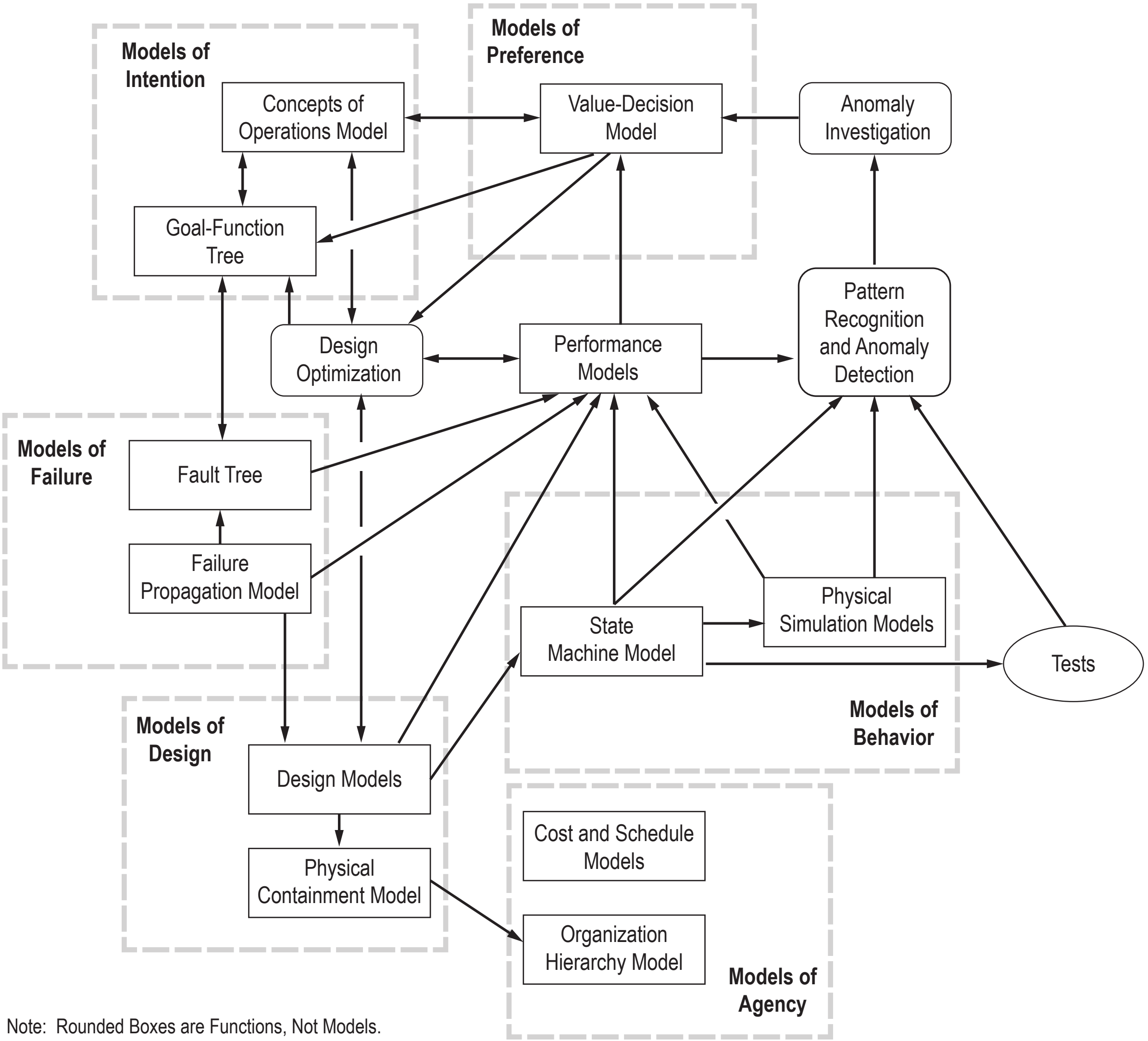


Figure 32. System models and model types in closed loop.

After a discussion about hierarchical and flat representations, the following subsections describe the major representation types, and the specific models within those types.

4.12.1.1 Hierarchical Versus Flat Representations. There are two major classes of representations: hierarchical and flat. This section discusses some general observations about their differences. These two classes will be encountered frequently in the ‘model types’ discussed in later portions of this section. An engineered system, unless its design structure really is a ‘tree,’ does not have a hierarchical structure (subprinciple 3(d)). Components connect to each other in a variety of ways. Loop structures are very common for feedback control. Components are connected to many other components, which exist in many other larger scale groupings like subsystems or volumes that contain components. Component connections can change over time depending on the system’s different operating modes and can also change based on failures. A simple example is a staged rocket, in which some components after use are simply discarded. Failures sometimes create new and undesired connections, such as a short circuit creating a new electrical path, or an explosion creating patterns of debris that ‘connect’ normally ‘nonconnected’ components. The manner of representation that enables components (and the functions being performed by them) to connect to each other in complex ways that exist in a system is a ‘flat’ representation.

By contrast, hierarchical representations inherently places some components or functions ‘over’ others in the hierarchy. The existence of hierarchy implies a value or attribute that is given ‘priority’ over others, enabling some functions or components in the success tree or fault tree to be ‘higher’ in the hierarchy than others. One such attribute is intention, in which the system’s overall purpose, or (in a fault tree) failure of that purpose, is the highest level node in the inverted tree structure. Physical containment models give priority to components that physically ‘surround’ others as the hierarchical principle. Organizational hierarchy models give priority to authority and institutional control. In theory, one can use almost any attribute as its organizing hierarchical principle. However, in practice, only a few of these are used; the three mentioned above are typical.

4.12.1.2 Value Models. Whenever requirement and design choices must be made, values are inherently involved. To ensure that choices are properly based on explicit values, and the values are addressed in a way that allows rational choices to be possible, a value model must be developed as discussed in section 4.8.4. This value model is subject to the constraints of vN-M’s theory.[23] This model, like other models of systems engineering, will become more detailed and refined over time, to enable rational, explicit choice at progressively more detailed levels of the design. Other sections of this TP address the nuances of value and choice models. The point of emphasis here is that information produced from the other system models frequently highlights issues or problems. These, in turn, imply the need for changes in intention, design, or operations, which the value model helps to determine. Systems engineering intends to drive the use of value models as far as is practicable to greater degree of detail in the design process.

4.12.1.3 Models of Intention. Every system has one or more purposes that must be defined. While value models prioritize preferences, these must ultimately be turned into specific intentions and goals for the system. In traditional systems engineering, a concept of operations (ConOps) provides a description of how the system is to be operated in its environment. Requirements and constraints are typically extracted from the ConOps by detailed reading and interpretation of this document.

In systems engineering, it is expected that a written, natural language description of system operations will still be generated. However, systems engineering will provide rigor by developing an event sequence model (ESM) to formally represent the operations described in the natural language ConOps. The ESM representation defines operations using system state variables (discussed in section 4.4) to provide descriptive precision and formality, and to ensure that the operations described in the ConOps document are assessed for potential gaps, overlaps, and contradictions. Major events in the ESM can be translated or extracted to define requirements that can be exported to a document if desired. Images of the event sequence(s) can also be exported to the ConOps document.

The ESM is a contributor to the generation of a GFT model (and depending on the time-order of how these two models are developed, vice versa), which provides the formal hierarchical representation of goals and functions in success space, associated with each other through the definition of state variables and constraints on the state variables. The GFT provides a precise functional decomposition of the system in a joint single representation with its corresponding hierarchical system goals as described in section 4.4.1. This functional decomposition is physically accurate, hence, it can be used for a variety of analytical purposes. It inherently provides a top-down trace of goals (and when formally documented, requirements) and functions. If desired, some or all the goals can be exported from the GFT to generate a significant portion of a top-level system requirements document.

Part of the modeling of intention is the modeling of system constraints, as these are necessary to translate preferences into goals. Some constraints are directly developed for specific goals and functions as part of the GFT and/or ConOps. However, not all constraints are attributes of specific functions. Some are system-level constraints that do not directly associate with a single function or function-related goal. These kinds of constraints are sometimes called 'extensive attributes,' which are associated with all or some system components, and are 'added up' with the number of components. These constraints include cost, mass, power, reliability, computer processing memory, computer processing speed, propellant mass, and the like. Many of these attributes are additive per component. An example is mass, in which each component (except for software) has a certain amount of mass, which adds linearly to the masses of other components to generate the system mass. Not all attributes add in this way. Attributes such as system reliability or safety have properties such that it is possible for the system to have more components, each has a reliability number, but because they are placed in parallel to other components, system reliability improves. Costs add in a manner related to the number of components, but also are associated with operational tasks as well as physical components and other factors.

4.12.1.4 Models of Design. Systems engineering requires high-level system design models. At least two types of design models are needed: directed graph and physical containment.

The directed graph is a representation that requires a node for each component, and an arc (arrow) between components. Thus, directed graphs can be a form of mathematical category as discussed in section 4.11.4.3. Loops are allowed, which makes the directed graph nonhierarchical. Directed graphs in success space and failure space are needed. Success-space directed graphs assume AND gates represent the default relationship of the inputs of each node, with OR gates representing redundancy. While directed graph arcs are normally associated with the linkage of electrical, data, and fluid-flow components, they can and should be used in a more general sense. One can consider three-dimensional analytic models such as structural finite element models, electromagnetic field models, and explosion models as directed graph models with a finer, three-dimensional mesh. Thus, directed graphs are an excellent candidate for the role of a representation or common 'language' that is 'abstract enough' to translate models that exist at a finer grain of detail into common system-level representations that can connect to each other.

The physical containment model is a hierarchical representation of which components exist within the physical boundaries of other components. Thus, for a staged chemical-propellant launch vehicle, the top level of the physical containment model separates the vehicle 'segment' from the ground 'segment.' Inside the vehicle segment are the first stage, boosters if applicable, second stage, and so on. Inside each of the stages are various components such as propellant tanks, propellant feed lines, electrical power and data wiring, computers, structures, propulsion, and so on. Inside of each of these are lower-level components such as valves, chips, wiring, etc. The physical containment model is crucial to connect organizations to designs.

4.12.1.5 Models of Failure. Failure models could potentially be grouped into two distinct groups of intention and design. While they have features of these prototypes, they also have some unique features as discussed in section 4.4.2.

4.12.1.6 Models of Agency. Agency models represent the 'agents' that manage, design, build, test, and analyze the system, and various attributes related to these agents. These include representations of the organizations and individuals involved with these activities, and include critical management representations such as cost, schedule, and organization hierarchy models. Agency models are essential to describe and assess critical attributes and performance of the organizations that create the system.

Since organizations implement the components of the prospective system design, models related to these organizations are necessary. The most obvious purpose of an organization model is the need to allocate requirements to the relevant organizations. These types of models are addressed in section 5.

One typical organizational model is the organization chart. It is typically hierarchical, but for different reasons than the GFT or other intention models. The reason that the organizational chart is hierarchical is the hierarchical structure of institutional control over the project or system. During the early phases of a system lifecycle, usually one organization oversees the entire system and its integration. This organization typically hires other organizations to build the next level of components, and it integrates these components and contractors. These contractors hire subcontractors to build the next tier of components further down the managerial hierarchy, and so on until all components are allocated to organizations that build or purchase components. Organization models that mirror the organization chart are important because of the need to allocate requirements, and ultimately funds, to organizations to design, build, test, or operate their portions of the system. These allocations, at least in government organizations and contracts, are often implemented through the WBS and numbering systems.

Another set of critical agency models are representations of cost and schedule. Managing the organizations that create the system is just as crucial as understanding the system itself. The major proxy measures used by management to judge the performance of these organizations are cost and schedule. Cost models come in at least two types: those linked on a day-to-day basis with the ongoing project development and related labor, travel, and materials costs, and those generated through historical data based on analogy for long-term prediction of overall project cost in proposals and checks against proposals. These are not mutually exclusive but are often separate due to their differing purposes. Schedule models can be as simple as waterfall or Gantt charts, or more sophisticated directed graph methods that search for and highlight the project's critical path. Day-to-day management of the project links these cost and schedule models to the system configuration through the configuration control board in a process called configuration management.

There are systems engineering tools that have the potential to facilitate the development of products in a multidisciplinary environment, requiring the integration of social systems. Systems dynamics models and ABMs provide an approach to model the social aspects of the actors of the agency. Systems dynamics models capture the flow of information through the organization. These models represent the information flow as stocks (information stores) and flow (transfer of information) and provide a construct in which organizational and system design interactions can be represented and analyzed. ABM models the agents that work in the development or operations organization, addressing the social choices and influences in the organizational choices about the system. These models bring in the social sciences aspects of systems engineering and are discussed in sections 5.6.1 (System Dynamics) and 5.6.2 (Agent-Based Models).

Historically based cost models bias the system cost toward the historically referenced system, as discussed in section 4.8.1. This bias can create cost errors of more than 2 times. If a historical basis is used, the systems design, manufacturing, testing, and application must be highly similar. If not, another cost approach must be used, which is typically the case.

4.12.1.7 Models of Systems Behavior. Along with cost and schedule criteria, ultimately, the success or failure of a system depends on whether it achieves its goals. This means that its functions are properly performed, which in turn means that the system 'behaves as it should.' It is no surprise then that models that generate system behavior are a critical aspect of systems engineering. Behavioral models allow exploration of the state space implicit in the design, when driven by appropriate inputs to stimulate the many possible states of the system in both nominal and off-nominal scenarios. Exploration of the state space is necessary to ensure the design performs as intended, and more generally, to verify the system.

One major type of behavioral model is the state machine as discussed in section 4.4.3. State machine models have the significant advantage of generating complex behaviors from many simple models. That is, while many model types require analysts to predict and model behaviors directly, state machine representations consist of multiple simple models of the state transitions of individual components, which can be triggered by the state transitions of other components, with relevant time delays as appropriate. The models are generally driven by an external script that defines how the system is commanded. When environment models are added, the state machine executive executes the component models.

Another critically important set of models that generate behaviors are those that drive simulations. These are generally physics-based models, which can mix combinations of models of the system and actual components of the system, up to and including the entire system, within a simulated environment. Driven by inputs that create nominal and off-nominal scenarios, simulations create huge amounts of data that must be assessed. Simulations are generally more expensive than state machine runs, both in cost and time. This is because they have greater fidelity to the actual system, though this can vary significantly. As simulations incorporate more elements of the actual system design, these transition over to tests. Software integrated testing and hardware/software integrated testing often employ these high-fidelity system simulations. When the entire system is operated in its real environment, no more simulation exists and we proceed into the realm of pure testing.

4.12.1.8 Models of Performance. Simulations are often used to assess system performance. However, there are other kinds of models that also assess the performance without generating system behavior. A classic example is the root locus analysis of control theory. This method assesses projected stability of the control system without explicitly generating system behavior over time. Another example is models used to assess performance of Fault Management Control Loops, in which models are used to estimate performance, expressed in terms of fault management metrics that are inspired by, but not directly comparable to, classical control theory. Some models exist to provide a structure in which to collect and relate empirical data in an analytical framework. Some of these models exist in the engineering disciplines, with system integrating physics models providing this at the system level as discussed in section 4.2.

4.12.1.9 Discipline Models. Information from discipline models must be integrated into system models. System integrating physics models provide this function as discussed in section 4.2. When considering the types and content of systems engineering representations, information coming from discipline models must be integrated and used in system models. In general, this can be done because both discipline and system models typically use state variables in which their inputs and outputs are expressed. This enables mapping of discipline representations to system representations.

4.12.2 Model Integration Points

The system model relationships for the model set currently being explored are seen in figure 33. This figure shows the relationships between the models at a high level. Each of these models provides different views of the system used to understand the system and to aid in decision making. Within the set of system models, system integrating physics, state variables, system value, and other system characteristics are determined. Some of these characteristics are, or are related to, system attributes. Attributes are also related to GFT and state analysis approaches. The attributes seen in the green dashed box are related to the goals of these two approaches and are inputs to the value model. A value model is an abstraction of the desires of the stakeholder, enabling the determination of the value of a system that has a set of attribute measurements. The stakeholder also informs the goals of the GFT and state analysis. The value model offers a system view from a preference perspective and is most powerful when combined with an optimization approach. MDO provides a framework to incorporate the system value as an objective function in order to find the optimal system with respect to a set of design variables. By changing the design variables, the system itself changes, and the system models must be reanalyzed.

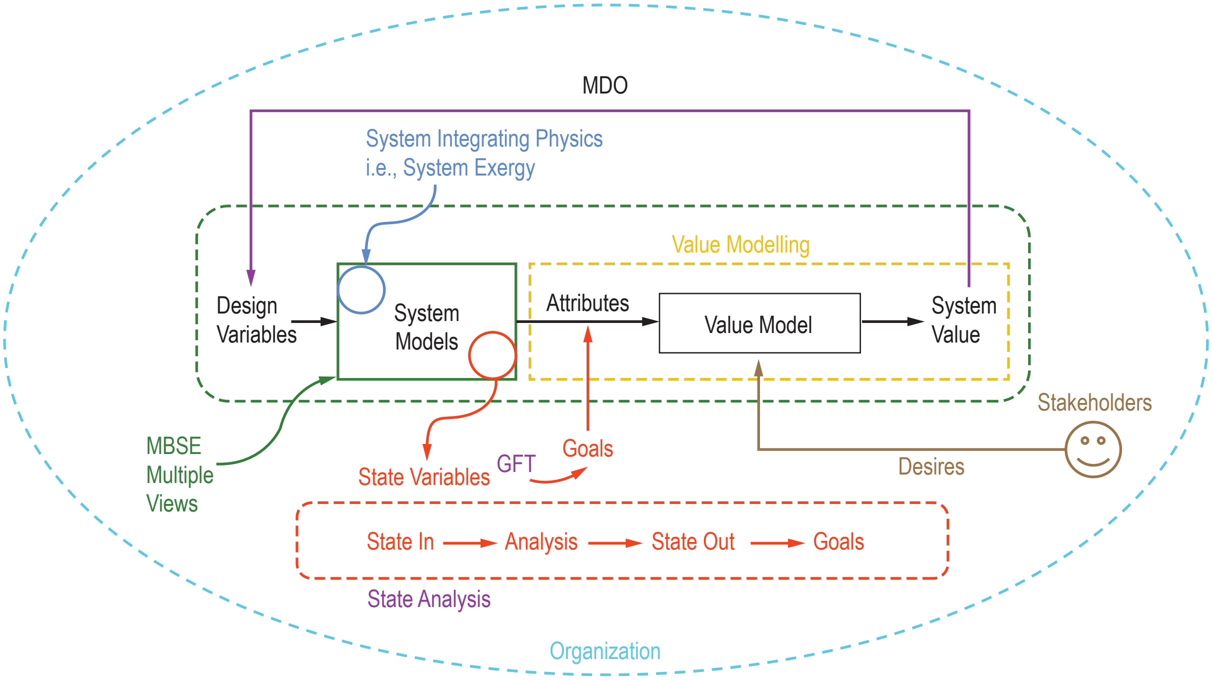


Figure 33. System model relationships.

An important reminder is that these models, which generally focus on the system, are being viewed and interpreted by people. These people are influenced by their organization's culture, structure, policies, etc. A comprehensive understanding of the system expands the system definition to include the organization that is involved in its lifecycle.

Preliminary work has identified how the models interact with each other as shown in figure 34. System goals are the apparent ties that bind the models together. Since engineered systems exist to achieve intended goals, the models used to design and build these systems must link to their stakeholders' intended goals. System goals provide benefits used in value models to represent stakeholder preferences. Those same benefits are seen in MDO through the objective function (value model) as well as when analyzing the couplings of the system. The system goals are the end states in state analysis and the objectives in GFTs. In MBSE, the system goals trace to requirements. System physics models are influenced by system goals through limits that the goals place on what the physical system is allowed to do. Goals are directly or indirectly represented in each of the models and serve as a centerpiece when linking the models together.

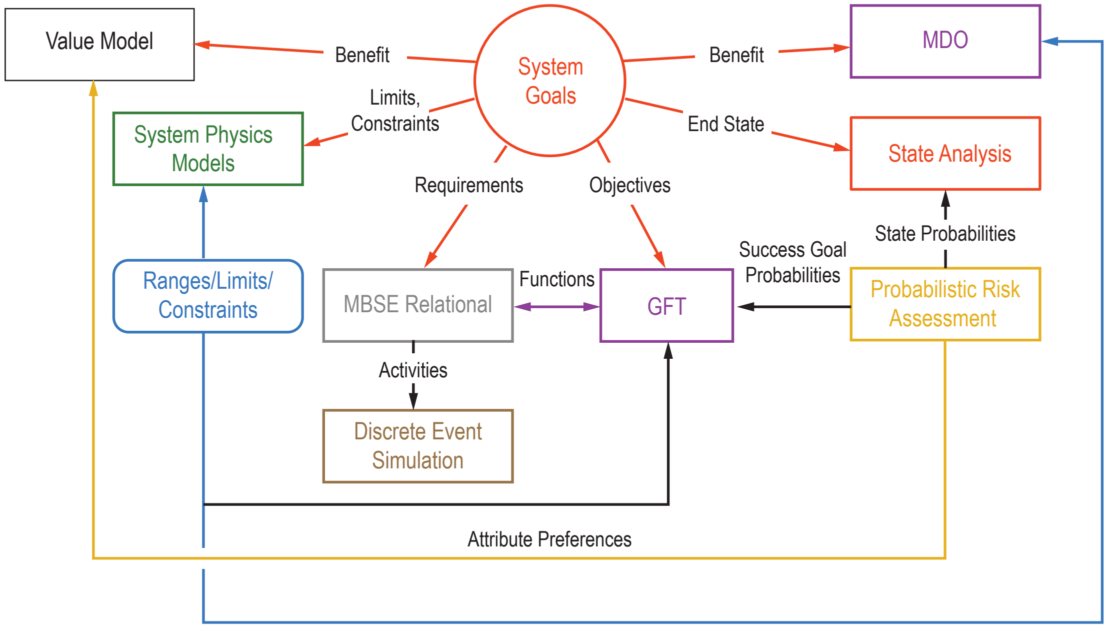


Figure 34. System model integration points.

Ranges, limits, and constraints restrict the design space in system physics models, the GFT, and in the optimization approaches in MDO. The ranges, limits, and constraints must be consistent across models to enable consistent decisions. Inconsistent constraints will result in different design spaces for the modelers to make decisions within. This is also true for inconsistent representation of goals in the models. In effect, this inconsistency causes different views of different problems. While modelers may be making decisions that are well informed by their models, their decisions may be in conflict with decisions based on other models. While this conflict may occur due to the views themselves, inconsistent goals or constraints will inflame the problem.

Probabilistic risk assessment informs value models of preferences associated with attributes that can lead to an improved utility function. PRA also informs GFTs and state analysis of goal and states. The integration points mentioned in this subsection are just a few of the ways that the different models of the systems interact.

4.13 Systems Engineering of Elegant Systems: System Development and Operation

The application of system integrating physics is specific to the system (postulate 1). The use of each of the approaches in this section is related to the system and to the lifecycle phase the system is in. 'Engineering Elegant Systems: The Practice of Systems Engineering' provides guidance on how to apply the postulates, principles, and approaches defined by the theory. Before putting these into practice, the relationship of each of the approaches is important to understand. This section addresses these for both the development phases and the operations phases of the system. Figure 35 illustrates the System Design approach for an elegant system.

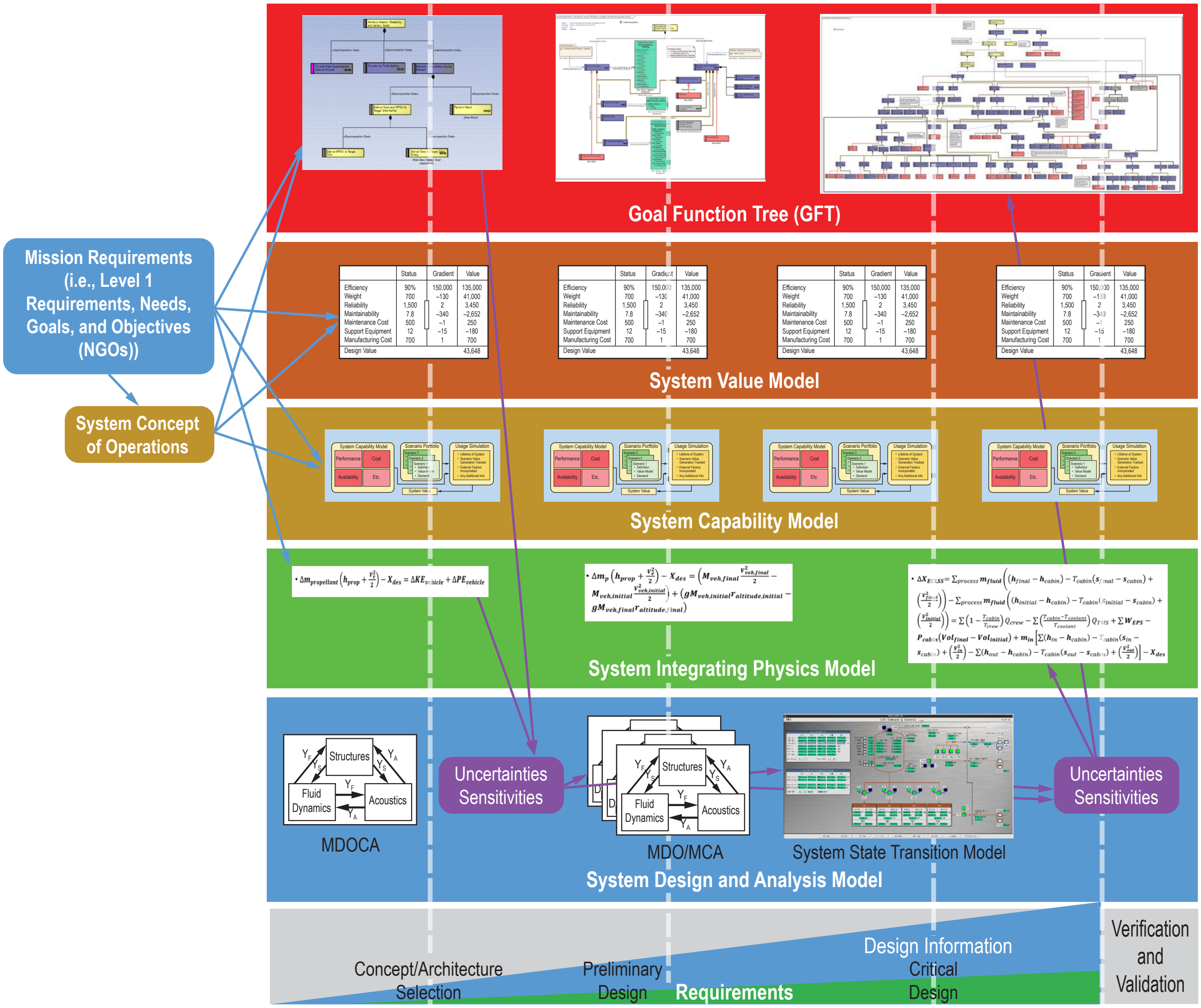
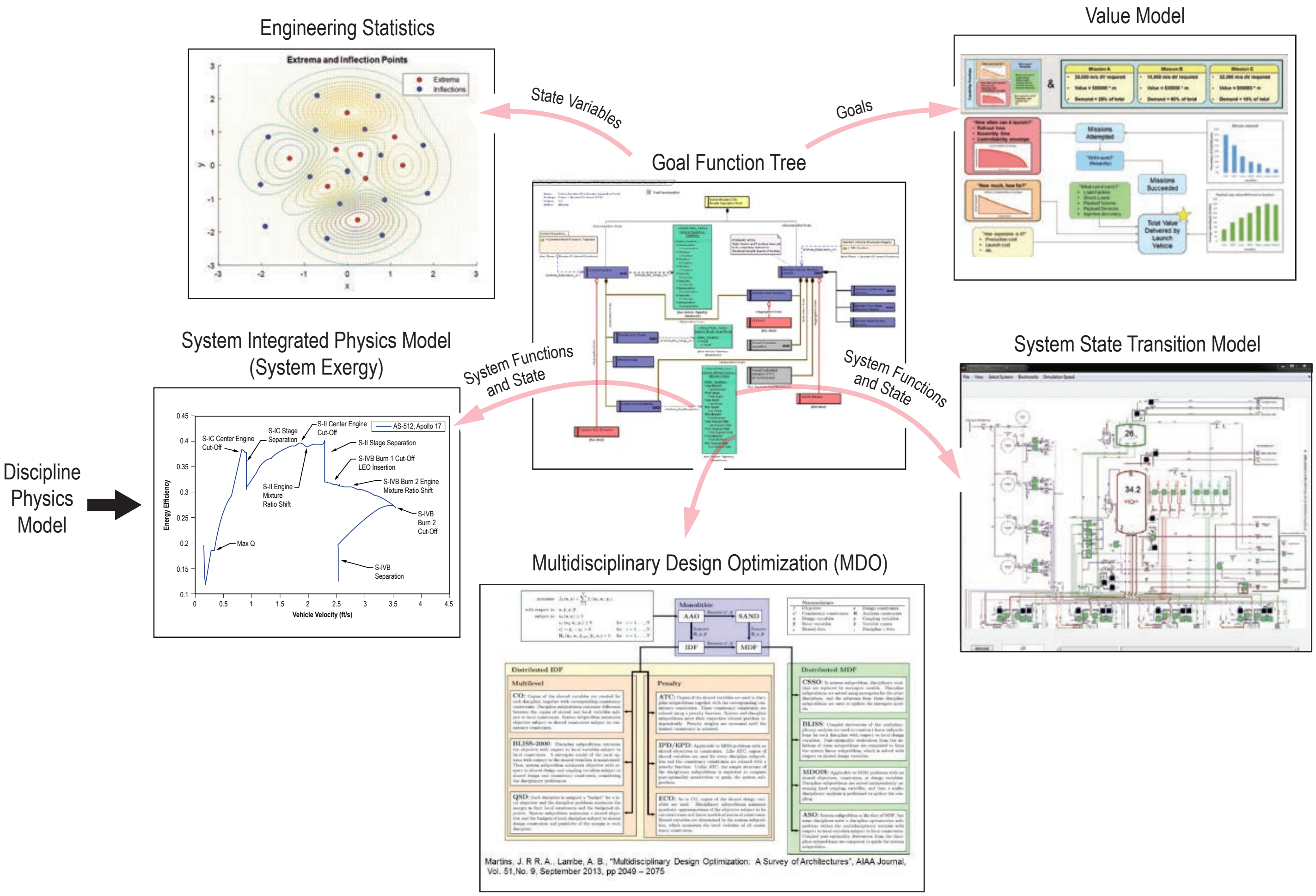

Mission Requirements (i.e., Level 1 Requirements, Needs, Goals, and Objectives (NGOs))
System Concept of Operations
Goal Function Tree (GFT)
System Value Model
System Capability Model
System Integrating Physics Model
Uncertainties Sensitivities
MDOCA
Structures
Fluid Dynamics
Acoustics
MDO/MCA
System State Transition Model
Uncertainties Sensitivities
System Design and Analysis Model
Concept/Architecture Selection
Preliminary Design
Requirements
Design Information
Critical Design
Verification and Validation

Figure 35. Elegant system design.

4.13.1 Engineering an Elegant System During Development

The various system models developed for system design, analysis, and integration as described above form an integrated set of system representations as illustrated in figure 36. The key to the integration of these models is the structure of goals, functions, and state variables contained in the GFT. Each of the system models then makes use of various aspects of GFT information to consistently model the system for a specific viewpoint (e.g., performance, value, behavior) as discussed in section 4.12.



Martins, J. R. R. A., Lambe, A. B., "Multidisciplinary Design Optimization: A Survey of Architectures", AIAA Journal, Vol. 51, No. 9, September 2013, pp 2049 – 2075

Figure 36. System model relationships.

The GFT provides an integrating structure that gains depth as the understanding of the system increases. Early in concept selection, the system goals are derived directly from the mission context requirements and the stakeholder's intent. This captures the intended uses and functions of the system. The GFT then grows as the design matures. As concept selection is made, a set of top-level state variables are identified and added to the tree. The system functions become visible in the tree as these are necessary to accomplish the goals. This growth continues to expand as the subsystem and component definitions mature.

A system value model can be constructed during mission definition activities. This captures the stakeholders' intents and preferences. These intents and preferences are also the goals in the GFT at this early stage of system definition. There may be more than one class of stakeholders such as investors (e.g., stockholders or Congress), users, operators, organization management, etc. A value model of each of these group's preferences should be developed and compared with each other. This provides early identification of conflicts in expectations and the ability to resolve these conflicts before design decisions are made. As the design progresses, changes in the value model are tracked, typically at major system lifecycle milestones. A value model based on system design decisions can be compared to the value model of the stakeholders providing identification of deviations of the system design to the stakeholder's preferences. This is a key tool for the systems engineer, as socially, the design organizations' preferences can be very different from the external stakeholders (e.g., users, operators, investors) and this can lead to very inelegant systems in operation and use. The system value model enables these differences to be identified and design adjustments made to correct for this.

In government design organizations, performance may become much more valuable than operational cost efficiency, whereas the stakeholders may desire operational cost efficiency much more strongly. This can potentially lead to two very different system designs. This divergence can also occur in a commercial setting. Commercial design organizations can be much more cost attuned than the stakeholders who may have a strong preference for specific system features. This also leads to two very different designs. System elegance is dependent on knowing how well the system meets the intents/desires of the system stakeholders.

Part of the input to the GFT and the value model is the system-integrating physics and the preference that the stakeholders have for the systems performance. The system-integrating physics, whether system exergy, optical transfer function, structural loads, algorithm mathematics, social statistics, etc., provide the initial set of state variables for the GFT and the performance measures for the determination of value preference by the stakeholders. The evaluation of system configuration options is conducted using the relationships and properties of the system integrating physics. This evaluation provides a quantitative measure of which system is the most efficient from a physics/logic standpoint.

Based on the system goals and functions, a top-level system capability model can be developed to compare against potential system applications (e.g., design reference missions (DRMs)). This provides an understanding of the system robustness in application (which includes the concepts of system resilience and system reliability). The capability can be updated as the design matures and further system design decisions are made, allowing system robustness to be tracked and system design changes updated to maintain or improve the robustness of the system. System reliability is also tracked, sometimes through a PRA, which can be generated using the GFT as a starting point. The GFT and its logical complement fault trees provide the integration of system reliability aspects of system robustness with the system success goals.

As the design progresses, design decisions and subsequent technically-derived requirements are guided by the system integrating physics. This enables a measure of the system performance changes as the design matures. Each design decision is evaluated with respect to the system integrating physics (e.g., exergy efficiency, image quality, total loads with margins, computational efficiency). This also enables a first look at system sensitivities and uncertainties. Uncertainties of the system state variables defined by the system integrating physics are key system parameters to consider for sensitivities. As the depth of understanding in the design deepens, subsystem choices may provide additional state variables for incorporation in the GFT and for further understanding of system uncertainties and sensitivities. The system integrating physics provides a framework in which to identify system interactions that can then be input to the GFT.

Once the initial configuration is selected, preliminary design and then detailed design ensues. MDO/MDOCA provides a method to optimize the system, accounting for all the system internal and environmental interactions (i.e., physical, logical, human). These methods allow design at the system level to be performed, providing guidance to subsystem designs and integrating the results of subsystem decisions. Analysis can be performed on manufacturing, operations, and maintenance features (including human system interaction) of the design to bring together an elegant system.

Statistical analysis of the complex system interactions also provides an important set of tools for system design including information theoretic, Bayesian, and frequentist methods. These methods support the design and optimization of complex system interactions such as sensor configurations, system protuberances, social interactions, etc.

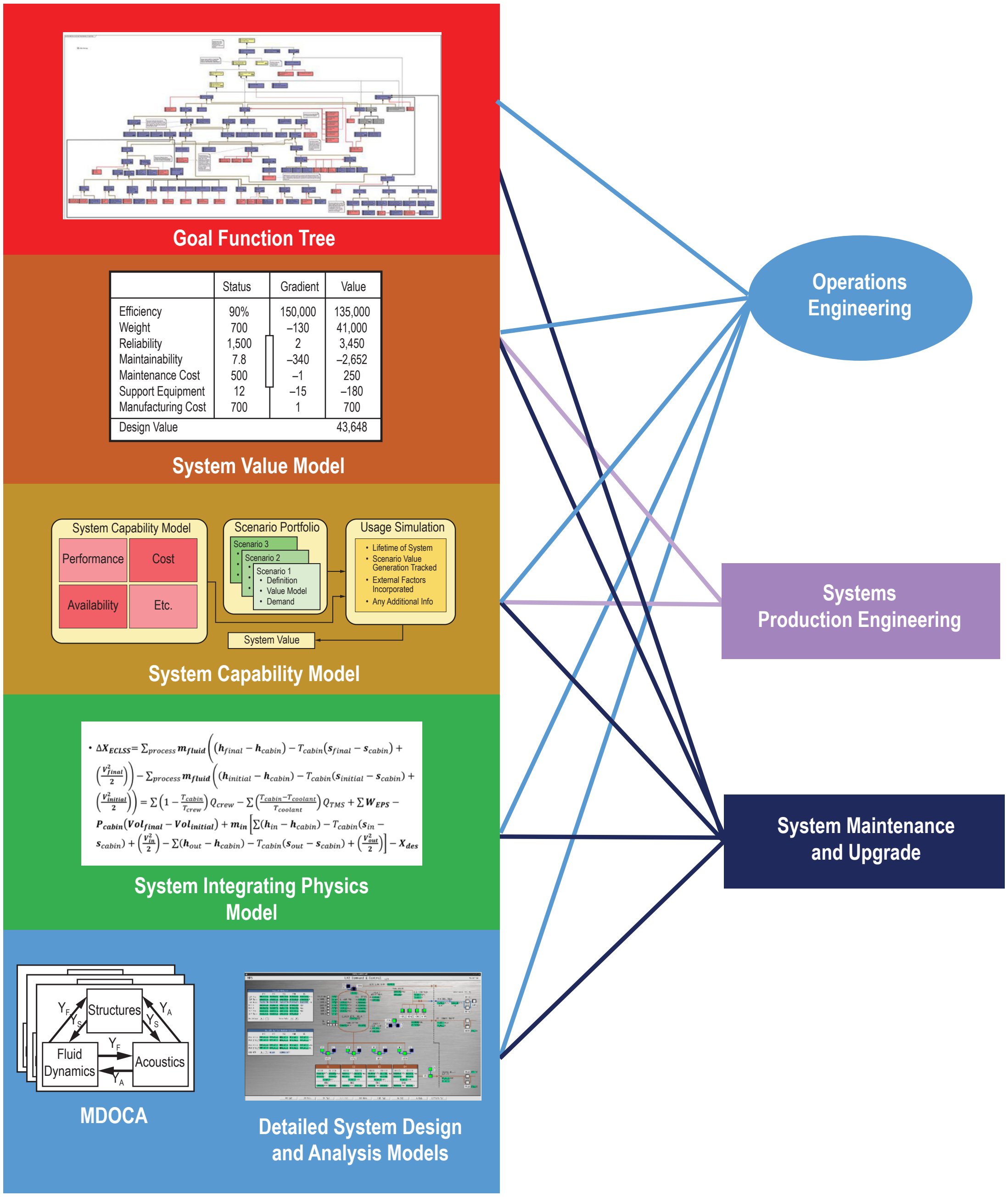
Cost and schedule models (i.e., development, production, and operations) are driven by the design decisions and are updated as the understanding of the design progresses. These models should support all options considered in configuration selection, preliminary design, and detailed design. PBS, which follows the structure of the GFT, provides a method to view the cost relationships and determine the cost drivers of the design. These cost models provide input to the system value model to assess the design's satisfaction of the stakeholder's preferences.

As the design progresses through preliminary design, a system state transition model can be developed to model the systems state changes to various operation sequences. This provides an integrated system understanding of the system hardware, software, and human responses in system operation, and enables exploration of the state space beyond what testing can accomplish. This provides an integrated system model and system medium to design hardware/software integration as well as some aspects of human system integration.

System verification and validation (V&V) occurs progressively throughout the design process as the understanding of the system deepens (postulate 7, principle 10, and principle 11). This V&V is supported by system models in two ways: system models are validated by comparison to test data, and the system design is verified and validated by comparison to analytical data output from the system models. Inspection is also an appropriate verification method where the system is visually compared to a model (e.g., drawing or code list). At each lifecycle review, the system physics models verify the system design (sometimes referred to as the design 'closing'). The system value for the stakeholders and for the system and the system capability model are compared with the system value generated by the system design providing incremental system validation. The system models themselves are validated (in terms of accurately representing the system interactions and responses (i.e., system behavior)) against test data and are used to understand unexpected system interactions with the test fixtures and test environment. The state variables captured in the system integrating physics models, GFT, and system state model provide guidance in the proper structuring of final system verification at the end of the development phase. Separately, the system value model provides guidance in the establishment of the final system validation (comparison of as-built design with stakeholders' expectations). These results of the system V&V provide the basis to accept the system or correct the system design or production processes prior to system release (e.g., first flight, system fielding, system deployment, market release date).

4.13.2 Engineering an Elegant System During Operations

System operations makes use of the system models developed during system development as discussed in section 4.13.1 and illustrated in figure 37. The system operational phase includes system production, system operation (use), and system maintenance and upgrade. Each of these phases uses the input from the system models constructed during system development.



	Status	Gradient	Value
Efficiency	90%	150,000	135,000
Weight	700	–130	41,000
Reliability	1,500	2	3,450
Maintainability	7.8	–340	–2,652
Maintenance Cost	500	–1	250
Support Equipment	12	–15	–180
Manufacturing Cost	700	1	700
Design Value			43,648

- $\Delta \boldsymbol{X}_{\boldsymbol{ECLSS}} = \sum_{process} \boldsymbol{m}_{\boldsymbol{fluid}} \left((\boldsymbol{h}_{final} - \boldsymbol{h}_{cabin}) - T_{cabin}(\boldsymbol{s}_{final} - \boldsymbol{s}_{cabin}) + \left(\frac{V_{final}^2}{2} \right) \right) - \sum_{process} \boldsymbol{m}_{\boldsymbol{fluid}} \left((\boldsymbol{h}_{initial} - \boldsymbol{h}_{cabin}) - T_{cabin}(\boldsymbol{s}_{initial} - \boldsymbol{s}_{cabin}) + \left(\frac{V_{initial}^2}{2} \right) \right) = \sum \left(1 - \frac{T_{cabin}}{T_{crew}} \right) Q_{crew} - \sum \left(\frac{T_{cabin} - T_{coolant}}{T_{coolant}} \right) Q_{TMS} + \sum \boldsymbol{W}_{\boldsymbol{EPS}} - \boldsymbol{P}_{\boldsymbol{cabin}}(\boldsymbol{Vol}_{\boldsymbol{final}} - \boldsymbol{Vol}_{\boldsymbol{initial}}) + \boldsymbol{m}_{\boldsymbol{in}} \left[\sum(\boldsymbol{h}_{in} - \boldsymbol{h}_{cabin}) - T_{cabin}(\boldsymbol{s}_{in} - \boldsymbol{s}_{cabin}) + \left(\frac{V_{in}^2}{2} \right) - \sum(\boldsymbol{h}_{out} - \boldsymbol{h}_{cabin}) - T_{cabin}(\boldsymbol{s}_{out} - \boldsymbol{s}_{cabin}) + \left(\frac{V_{out}^2}{2} \right) \right] - \boldsymbol{X}_{\boldsymbol{des}}$

System Integrating Physics Model
Structures
Fluid Dynamics
Acoustics
Y_F
Y_S
Y_A
MDOCA
Detailed System Design and Analysis Models
Operations Engineering
Systems Production Engineering
System Maintenance and Upgrade

Figure 37. System development model mapping to system operations functions.

System production primarily uses the system models to produce the system. A standard approach has been to use the computer-aided design (CAD) models of the hardware, CAD-produced drawings, and software object code to produce the system. Advances in production including computer-aided manufacturing (CAM), additive manufacturing, and plant automation are dependent on system models. These models provide a near-complete representation of the system to develop production processes and verify the accuracy of the system's physical and logical characteristics. Decisions on changes or modifications to production processes depend on a check against the stakeholder preferences to ensure production changes do not impact the expected functionality of the system. These decisions make use of the system value model and system capability model to identify any changes (i.e., system cost, capability differences, application differences) induced by the production process modifications.

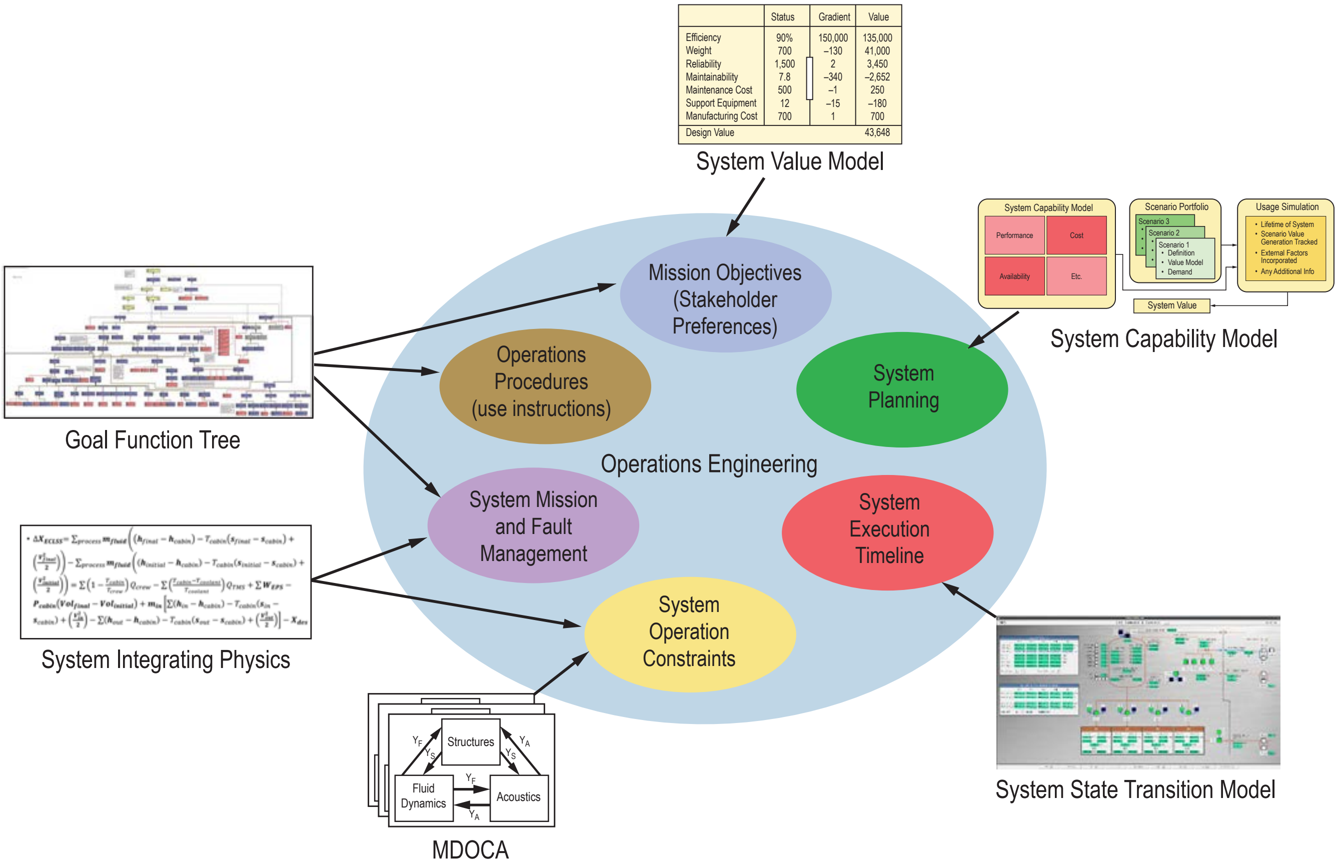
System operations can include operators (e.g., a drawbridge operator) and system users (i.e., the motorist using the drawbridge to cross the channel). In some systems, operators are different than users. The flight crew on a rocket or aircraft is the actual user of the system (as well as any passengers). The ground operations team is the system operator. Typically, these lines are not as clearly drawn (i.e., the flight crew is a user and may be the operations team or part of the operations team) as these simple examples assume.

Systems operations are generally the role of operations engineering. The procedures and tools used for system operations are based on the system models discussed in section 4.13.1. Figure 38 illustrates these relationships. The GFT and SAM based on the Concept of Operations provide a structure for the system operations. This provides a basis for the development of operational plans, resource allocations, and data flows. The system SAM provides a tool to check executable sequences and to monitor the execution to aid in the detection of variances indicating an anomaly or emergent behavior. This state model provides a basis for the operational procedures and aids in operational troubleshooting.

Systems engineers can monitor system performance through the health of the subsystems for most degradations. However, the impact of any subsystem failure on the total system can only be seen in the context of the system integrating physics. In addition, system interaction effects can cause degradation in system performance even though the subsystems are not indicating an off nominal condition. Thus, the system integrating physics provides a basis to periodically track the system's performance.

The system capability model used for evaluating system robustness is the basis for operational decision on new system applications. This model can be used by system engineers to advise the operations engineers on the suitability of the system for the new application including any operational constraints that will need to be imposed or relieved in the new application.

Statistical models are updated periodically by systems engineering, based on system operational performance measures. The operational history provides a basis for identifying and quantifying uncertainties in a system. This knowledge enables the system engineer to focus on reducing those uncertainties with the largest values and the greatest impact upon a system state. This potential reduction in uncertainty increases the depth and quality of the understanding of the system, particularly in the operational environment.



	Status	Gradient	Value
Efficiency	90%	150,000	135,000
Weight	700	–130	41,000
Reliability	1,500	2	3,450
Maintainability	7.8	–340	–2,652
Maintenance Cost	500	–1	250
Support Equipment	12	–15	–180
Manufacturing Cost	700	1	700
Design Value			43,648

System Value Model
System Capability Model
Performance
Cost
Availability
Etc.
Scenario Portfolio
Scenario 3
Scenario 2
Scenario 1
• Definition
• Value Model
• Demand
Usage Simulation
• Lifetime of System
• Scenario Value Generation Tracked
• External Factors Incorporated
• Any Additional Info
System Value
System Capability Model
Goal Function Tree
Mission Objectives (Stakeholder Preferences)
Operations Procedures (use instructions)
System Planning
Operations Engineering
System Mission and Fault Management
System Execution Timeline
System Operation Constraints
$\Delta X_{ECLSS} = \sum_{process} m_{fluid}\left((h_{final} - h_{cabin}) - T_{cabin}(s_{final} - s_{cabin}) + \left(\frac{v_{final}^2}{2}\right)\right) - \sum_{process} m_{fluid}\left((h_{initial} - h_{cabin}) - T_{cabin}(s_{initial} - s_{cabin}) + \left(\frac{v_{initial}^2}{2}\right)\right) = \sum\left(1 - \frac{T_{cabin}}{T_{crew}}\right)Q_{crew} - \sum\left(\frac{T_{cabin} - T_{coolant}}{T_{coolant}}\right)Q_{TMS} + \sum W_{EPS} - P_{cabin}(Vol_{final} - Vol_{initial}) + m_{in}\left[\sum(h_{in} - h_{cabin}) - T_{cabin}(s_{in} - s_{cabin}) + \left(\frac{v_{in}^2}{2}\right) - \sum(h_{out} - h_{cabin}) - T_{cabin}(s_{out} - s_{cabin}) + \left(\frac{v_{out}^2}{2}\right)\right] - X_{des}$
System Integrating Physics
Structures
Fluid Dynamics
Acoustics
Y_F
Y_S
Y_A
MDOCA
System State Transition Model

Figure 38. System model mapping to operations engineering activities.

DES provides a tool to model organizational activities and system responses to discrete events in the operational flow as discussed in section 5.6.3. DES supports Monte Carlo analysis and can be used for manufacturing flows, operational processing flows, supply chain flows, and flows of information through an organization.

Systems engineering uses the system cost model to compare against system production and operations costs and identify areas that cost can be reduced or improved. Operational experience generally creates a learning curve as production and operations personnel learn how to effectively produce and operate the system. The system cost model will need to incorporate this experience.

System maintenance actions, obsolescence upgrades, or planned block capability upgrades are the systems engineering activities during operations. These activities are all based on the baseline system models discussed in section 4.13.1. These models will be updated with the new information as necessary for the planned system changes. MDO/MDOCA are important system design models primarily for these efforts.

System value models are also updated during maintenance and upgrade activities by systems engineering. These activities can potentially lead to changes in the system operations as compared to stakeholders' (in particular, operators and users) expectations. In addition, stakeholders' expectations or preferences can change with their experience in the systems application. These changes are identified through the system value model and will need to be incorporated into the system design.

4.14 Summary

System design and integration is the aspect of systems engineering focusing on the system itself. This focus is specific to the system as defined by postulate 1. There is not one, but several strategies to understanding the system. These strategies provide an approach to understand the system integrating physics, system value, system goals and states, system control, and system statistics.

These strategies define the various system modeling types which aid in system understanding. The integration of these different system models provide a more complete understanding of all the system's aspects.

Important to future advances in systems engineering theory is the mathematical basis provided by category theory. This basis supports the system strategies and models to provide a complete understanding of a system.

5. ORGANIZATIONAL STRUCTURE AND INFORMATION FLOW

Systems engineering is responsible for integrating the various disciplines within an organization to develop or operate a system. This aspect is a parallel aspect with system design and integration that involves understanding the system's integrating physics, value, state variables, relationships, and statistics. If one understands the system, but not the organization structure and interrelationships, the system may never get developed. If you understand the organization but not the system's design and integration, the system will not work. Systems engineering must deal with both the system design and integration and the discipline integration aspects to design an elegant system. The remaining system strategy involves knowledge of the system.

- **Strategy 7**—Knowledge Strategies. Discipline integration is a highly sociological function and brings in aspects not traditionally thought of as engineering. Nonethelesss these aspects are essential as complex systems are developed by complex systems (organizations). The systems engineer must understand how the organization is structured, how communication flows, and how information about the system is maintained. Information maintained within the organization is not always readily identified in the system design. Managing this is a crucial role of systems engineers. Discipline engineers are reservoirs of information, information which they generate and maintain. The systems engineer manages the channels between these reservoirs, ensuring the right information is provided to the right discipline when needed. This brings in information theory as a key element in understanding information flow throughout the organization. Information theory applies not only in the design process but also governs the decision-making structure within the organization. Configuration management and data management, from the discipline integration viewpoint, are important tools for the systems engineer to manage the system information.

Systems engineers are concerned with how information about the system flows through the organization. This is a focused view on one aspect of the organizational structure and culture. In general, the project manager is responsible for the organization of the project. The project manager deals with the full organizational structure and culture picture. The systems engineer supports the project manager with recommendations on organizational structures that have efficient information flow for a specific system and organization. In addition, line management may also exist separately from the project structure in some organizations (i.e. matrix organizational structure). The systems engineer also coordinates with line management as necessary to recommend improvements and work information flow issues within the line organization structure. The systems engineer identifies gaps and barriers to information flow, which can arise from either organizational structure or cultural issues, and recommends solutions to project management and line management.

5.1 Sociological Principles

Systems engineering facilitates many sociological functions in the development or operation of a system.[82,83] Systems engineering provides appropriate opportunity structures[84] for the maturing of design concepts and ideas as part of discipline integration. Opportunity structure represents the social paths for success in the organization. These structures include technical review forums (such as status meetings separate from decision-making structures), task teams, working groups, communities of practice, etc. where new ideas can be explained, discussed, challenged, and matured, leading to a fully-formed idea for the formal decision-making process.

The systems engineer has a set of roles within the organization. This role set[85] includes system expert (in system interactions), system analyst, discipline integrator, team leader (study teams, task teams, etc.), advisor (to the program manager, chief engineer), and employee (to line management). These are not different roles conducted by the same individual but a set of roles associated with the position of systems engineer within the system design or operation organization.

Systems engineering must also engage in creating a balanced engineering effort appropriate for the system under development or in operation. Systems engineering must address imbalances that can be created sociologically in accumulating advantage or disadvantage. This imbalance occurs when a highly successful organization is rewarded with higher priority, more resources, and improved pool of skills to address a problem, while struggling organizations loose priority, resources, and skilled engineers. The needs of the system regulate the balance and overemphasizing successful organizations or deemphasizing struggling organizations can lead to an imbalance in the design or operation of the system. Managing organizational engineering deficiencies and imbalance is crucial for a successful system. Incentives may be defined to encourage communication in some cases. The wrong emphasis in the organizational efforts can lead to very imbalanced, and hence inelegant, design.

Part of the complication of discipline integration is the cultural subsets that exist for each discipline. Each discipline contains a unique set of understandings and terminology for their discipline. Consistent use of terminology is important within the sociological function of systems engineering. Terms communicate ideas within an organization and can develop specialized meaning within certain cultures. Systems engineering ensures that terms are used with consistent and recognized (agreed to) meaning. Sometimes different terms are used for similar, not necessarily the same, meanings. Other terms may be used for different meanings. Within a discipline, the meanings are consistent, but across the disciplines, they are not. Systems engineering must translate the differing cultural meanings into a consistent terminology and understanding across the entire organization. Using different terms for the same meaning creates confusion in the organization and can lead to unanticipated system design or operation errors. Similarly, having a single term represent different meanings in different contexts also leads to organizational confusion and system design or operations errors. These, in turn, can lead to system failure or to increases in cost and delays in the schedule. Systems engineering should avoid both types of terminology confusion that can occur in discipline integration.

The specific cultural aspects of each individual discipline can also make discipline integration challenging. The different disciplines typically have different team structures or processes. They may use team structures or processes in different ways or view the significance of a team structure or process differently. Systems engineering functions to create a blended team structure across the disciplines so that communication and information can flow without misunderstanding in meaning or significance of an activity.

Both manifest and latent social functions will be present in discipline integration. Manifest social functions are defined as 'objective consequences contributing to the adjustment or adaptation of the social system that are intended and recognized by participations in the social system.' Written norms of conduct which are adhered to by those in the organization (adherence is the important distinction) are an example of manifest social functions. Unwritten norms which are adhered to by those in the organization are also manifest social functions. Latent functions are defined as 'those [social functions] which are neither intended nor recognized.' These are difficult to identify and are not clearly seen by those in the organization. Organizational biases against certain solutions or approaches can be a latent social function. These can lead to positive or negative system impacts yet the organization does not recognize the bias. The result of the organizational actions can demonstrate the bias against a system aspect. The systems engineer must be alert for these effects and work with organizational line management and program management to address latent functions which have a negative (limit or defeat) impact on the elegance of the system design or operation.

Social dysfunction is 'any process that undermines the stability of survival of a social system.[86] Systems engineering helps to mitigate dysfunctions that can cause information about the system to be suppressed or inaccurately communicated. These dysfunctions are a risk factor for the system and can greatly affect the ability of an organization to accomplish a given system design or operation. Innovative approaches to accomplishing a system entail a social change that can be very disruptive to organization's sociological values. This can mean that a given organizational culture is not able to develop the system that embodies values contrary to what the organization has come to believe as most important. Innovative system approaches often entail the formation of an entirely new organizational structure and culture with a different view on what is most significant in the system in order to succeed. Examples of innovation disruptions can be found in various industries including the computer industry (mainframe versus networked workstations), heavy equipment industry (steam-driven systems versus hydraulic systems), medical practice, etc.[87] Automation of previously manual operations is a historically significant and still current topic in the United States culture (i.e., uncertainty about drone applications).

Sociological ambivalence is an 'incompatible normative expectation of attitudes, beliefs, and behavior assigned to a social status (i.e., position) or a set of statuses in a society.' An ambivalence can be created if a discipline or position within the organization is confronted with conflicting norms. Manifest and latent functions can lead to these conditions at times. This condition can pose a threat to the system's success. There are six types of sociological ambivalence (quotes in this list are taken from Merton[88]):

(1) Inherent in the social position—Government employee relationships with contractors is an example where government ethics demands disinterest while social etiquette requires personal interest. There are many examples of these types of cases in the literature.[89]

(2) A conflict of interests or values—These may arise when a person is a member of two different organizations such as in a matrix organizational structure or when a person is working two projects. If the normative values of the organizations are different, the person can become socially ambivalent. For example, a conflict can arise when one project has a norm 'to do what it takes to solve a problem' that results in a conflict with the time agreed to spend on another project.

A conflict in time priority arises where one cannot satisfy both norms. These can also arise between organizational values and values from a person's life outside the organization.

(3) Conflict between roles associated with a particular position—These are conflicts in cultural norms that occur inherent to a given job position. These conflicts can occur in discipline integration where a representative to the system team may find oneself in conflict between norms of the system team and norms of their discipline team. Another example may be in procurement, balancing the norms of the procurement office with that of what the program views as necessary for success.

(4) Contradictory cultural values—These risks to the system occur when different cultural values collide. For example, an emphasis on high reliability can conflict with the need for techniques seen as credible. It takes special effort to determine the engineering basis to determine if a technique is credible or not in a specific application. In addition, the engineering basis must include an assessment on how the technique will improve reliability or not. Often, cultural bias can predisposition the trust or distrust of certain techniques with a determination of the engineering basis. The systems engineer ensures that the engineering basis is defined and known by the relevant decisionmakers. As discussed below, mathematics provides an integrating function for social functions and should be used in evaluating these types of conflicts. Look for engineering representations, not subjective logic.

(5) The disjunction between culturally prescribed aspirations and socially structured avenues for realizing these aspirations—This illustrates a disconnect between social expectations and the structure to achieve these expectations. An example is when a quick change is needed in the system design or operation and the organizational structure does not support a quick assessment and implementation of the change. The engineer is faced with either allowing a larger impact to the system later or moving ahead of approval with a change. Systems engineering is to identify and recommend to project management decision-making structures that are efficient and that mechanisms are in place for the types of disjunctions in this example (see sec. 5.3).

(6) That which 'develops among people who have lived in two or more societies and so have become oriented to differing set of cultural values'—This occurs when an engineer worked in different disciplines or supported projects with very different cultural values. The varying cultural values experienced can lead to ambivalence to cultural values in the current system that conflict or contradict what has been successful in the engineer's past. This can lead to a strong disinterest in the social structure of the system development or operation. These types of issues should be brought to line management or project management (at the appropriate level) to address. It is important that the members of the organization have an agreed-to set of values or sociological dysfunction can develop within the sociological structure of the organization.

Sociological ambivalence can lead to a failure to deal with or possibly to acknowledge conditions that affect system reliability and success. Systems engineering must be aware of these conditions when they occur in the organization and work with project management to find a new balance for the norms. This may involve the precedence of conflicting norms elevating one as more prominent to resolve a conflict or finding a common understanding that balances the norms, and addressing the concerns that may be suppressed in the ambivalent situation.

An extreme sociologically ambivalent context leads to sociological anomie. In this case, an individual in the culture can become normless or rootless.[90] A no-win situation has been perceived when an individual moves outside the organizational structure and opposes the organizational norms to achieve what the sociological culture calls for and the organizational structure is preventing. This imbalance occurs when the emphasis on success goals of the system are much greater than the emphasis on the institutional means to achieve these goals.

Within a social structure, people will adapt to the cultural expectations (i.e., norms) in some manner. These social adaptations can be positive for the system's success or negative. These adaptations can be generated by the organization (i.e., unreasonable expectations or contradictory expectations) or by an individual person in response to cultural pressures in some form. Systems engineers should be aware of these and work with project management and line management to address negative responses by the culture or by the person. There are five types of individual adaptations to the social structure:

(1) Conformity—Most people seek to conform to the cultural norms and the social structure to achieve these norms. They will try to stay within these bounds as they work in the organization.

(2) Innovation—Individuals caught in a conflict between the cultural norms and social structure may try to create a new path through the social structure. This typically involves violating some minor cultural norm or organizational constraint to resolve the conflict (or organizational pinch) that they are in. An example may be in skipping a level in the chain of command or bypassing an approval cycle to move forward. There are many creative ways that people may find to move forward in satisfying a cultural norm when that norm is not facilitated by the social structure. Social structure, in this case, would include the formal approval cycle (e.g., decision-making boards).

(3) Ritualism—In some cases, the frustration in conflicts can lead to an abandonment or reduction in importance to achieving a cultural norm. This leads to a ritualistic following of the organizational structure and processes. This ritualism can prevent effective solutions of engineering problems. This can be dangerous to the system as ambivalence has developed and conflicts in the system design or operation may not be identified as discussed above.

(4) Retreatism—Occasionally an individual will retreat from both the cultural norms and the organizational structure. A person who is in such an ambivalent situation simply withdraws from significant participation in the system development or design. When this occurs, the person should seek a different position in the project or with another project where the conflict they have encountered does not exist. These cases should be discussed with line management or project management at an appropriate level. Systems engineering is focused on the success of the system. When these deep sociological conflicts develop, line management is primarily responsible to help the individual deal with the conflict. Systems engineering evaluates these conflicts to identify issues in organizational culture values or structure that need to be addressed. If these conflicts are organizational in nature, then the systems engineer coordinates with the project manager and line management to develop corrections to support the clear flow of system information through the organization.

(5) Rebellion—This is the most radical of the responses to sociological anomie situations. Rebellion is a strong form of social dysfunction attempting to bring about a new social structure within the organization. This can occur in cases where an organization views the success of system based on different values than are required for the system in application. This can occur when a traditional organization attempts to adapt to a disruptive technological approach. As discussed above, this may mean the organization is not suited to the system development or operation.

The engineering reconsideration process (sometimes referred to as a reclama process) is one means to accommodate individuals caught in these difficult social norm conflicts. Project management is responsible to ensure that an effective reconsideration path is available and impartial to the social norms. The use of line management can be an avenue unless the line management is driven by the same cultural norms (both manifest and latent) as the system development or operation organization. In these cases, other pathways (external organization, external arbiter, etc.) will need to be found for relief of the anomie generating situations. Systems engineering advocates for an effective reconsideration path that provides a relief valve for social pressures in the organizational structure. This is important to the system success, as major failures have occurred in systems where effective reconsideration paths were not available or went through a chain driven by the same cultural norms that generated the conflict.

One possible response to social ambivalence, dysfunction, and anomie is in the self-fulfilling prophecy.[91] This sociological concept deals with the expected behavior of a group. If the culture defines a group as socially different, the organization can establish expectations and ascribed motives to the group, whether they accurately describe the group or not. The organization interprets the actions of the group in terms of the organization's expectation and ascribed motives. Thus, no matter how the group behaves, their actions always 'confirm' the expectations of others in the organization. This form or prejudice or bias can lead to serious limitations of the system and can lead to certain disciplines, organizational units, or engineering approaches being shunned by the organization even when a needed and positive contribution is offered. Sociologically, 'the self-fulfilling prophecy where fears are translated into reality operates only in the absence of deliberate institutional control.[82] Thus, systems engineering needs to be aware of these types of bias. When these are perceived, systems engineering advocates with project management and line management to ensure that the organization addresses the social biases that may be expressed. Testing these biases against a sound engineering basis is a good method to help identify and control this bias. This is one sociological reason why the engineering reconsideration system may need to be outside the development or operational organization.

Understanding of system interactions, sensitivities, and uncertainties requires application of sociological specification of ignorance. This is defined as 'the express recognition of what is not yet known but needs to be known to lay the foundation for still more knowledge'[92] of the system. Systems engineering needs to know what is not understood about the system as well as what is understood. This fosters system analysis to gain understanding of these interactions, sensitivities, and uncertainties that are not defined. This also supports systems engineering identification of unknown unknowns as the understanding of the system expands. Sociologically, analysis is not conducted on areas that are believed to be known or areas that are not known to be unknown. Practicing the specification of ignorance enables the system engineer to identify areas that may otherwise be overlooked within in the organizational culture.

Socially expected durations play a big role in how people within the organization behave and react to the progress on the system development or operation. These expected durations are not actual durations and can be quite different. There are three kinds of expected durations.[93] Socially prescribed durations are the formal system schedules and timelines. Organizational authorities responsible for the systems establish these formal system schedules and timelines that engineers within the organization recognize. Collectively expected durations are uncertain in terms of being able to specifically write them down. They are the collective attitude of a development organization or discipline in how they anticipate the development to proceed. Systems engineering is attuned to differences in the prescribed and collectively expected durations. These can indicate problems with the schedule and may indicate that an overly ambitious schedule has been established based on experience. There should be good engineering rationale associated with the formal system schedule with agreement from the disciplines that they can achieve the durations. The opposite can also occur, when the prescribed schedule is much longer than the collectively expected schedule. This can indicate problems in equipment availability, skills availability, funding availability, etc., that are not being expressly stated. These types of issues indicate the system is not achievable when needed which keys a review of the projects viability. The third kind of expected duration is 'patterned temporal expectations.' These types of durations occur in areas such as contracting, business transactions, etc. There is an expectation of how long an agreement or transaction should take to be finalized. Socially expected durations drive how people behave within the organization. If they expect a short timeframe, they may accelerate contributing efforts. If they expect a long timeframe, they may slow down or procrastinate contributing efforts. Systems engineering coordinates the contributions provided by the disciplines focusing on getting these in a timely manner, maintaining a balance in the organizations effort as it progresses on the system schedule.

Different sets of knowledge exist in different social groups (i.e., disciplines or organizations). This is not just technical knowledge but all kinds of cultural knowledge including history of various efforts. Access to knowledge is free in most societies, but it is not practical or possible for a single individual to have all knowledge. This leads to the formation of small groups of knowledge within a society. Within engineering, these are the disciplines. As systems become more complex, the ability to contain all knowledge of the system by a single individual is quickly surpassed and a larger group is needed. Thus, for most system developments or system operations, the knowledge of the system is contained by the organization and not just an individual. This is a natural sociological function and can be a very positive social structure when used cooperatively.

Sometimes, though, the differences in beliefs brought about by the different knowledge sets in each group can lead to contradictions in beliefs between the groups. These can be beliefs on which configuration or version is best, which approach will or will not work, and even how the project should be managed. When this happens the communication between the groups can break down and a strong distrust between the groups can develop. In its extreme, this leads to a questioning not of why the belief of the other group is wrong, but a questioning of what is their motive to bring such a 'palpably implausible' belief statement into the discussion.[94] This social polarization greatly hinders the progress on the system and can defeat the elegance of the system as effort is spent managing social conflicts within the organization rather than attention to elegant solutions for the system. Polarized political beliefs are a good example of this phenomenon.

Social polarization leads to functionalization of thought interpreted only 'in terms of its presumed social or economic or psychological sources and functions.' This can lead to an exclusive view of information or knowledge contained by a group (i.e., 'insiders') with others viewed as 'outsiders.' This also sets up a self-fulfilling prophecy about those outside the group being unable to understand the knowledge possessed by the insiders. A monopolistic view of knowledge contained by the insiders is maintained, breaking down communication altogether with outsider groups. The belief becomes one that you must be an insider to understand the knowledge or situation. The outsider groups are assumed to have different beliefs even when this is not true.[95]

Systems engineering supports project management and line management to mediate these different beliefs and to foster trust within the organization, among the disciplines, and with external organizations. Openness to discussion and allowing expression of beliefs to be made is important in this difficult context. Seeking points of commonality is a starting point to bring the groups to a common understanding. Note, there are individuals who may not be able to move past the distrust due to their own personal experiences, relationships, etc. This can also occur at the organizational level indicating that a group or organization may not be compatible for some reason. Changing a group or groups, if possible, can be a solution if an alternative is available. Normally competing organizations attempting to work together can experience this very strongly.

5.1.1 Unintended Consequences

Unintended consequences are based on human action or inaction resulting in unanticipated outcomes. The systems integrating physics do not fail; we simply do not always recognize the consequences of our actions/decisions. Robert Merton[96] developed four sources of unintended consequences. These sources have been decomposed into six sources of unintended consequences in systems engineering. These sources provide a framework to understand and manage these difficult results:

(1) Ignorance—Ignorance is a limited knowledge of the problem leading to unexpected (i.e., anomalous) performance or failure of the system. Failing to understand the system's interactions within itself and with its environment is a major source of ignorance leading to poor (though not realized) design decisions. Often the engineering and science is not well understood and the system models do not capture these interactions. The systems engineer must realize where knowledge is lacking and manage the risks, uncertainties, and sensitivities to these unknown or poorly understood interactions. Conducting tests, where possible, provides a method to reduce the ignorance in system interactions. Validation of models is also a crucial method to reduce the uncertainty in the system models used for design and operation.

Another form of ignorance is on the effect of the system, its fabrication, or its operation in the environment or local cultures. This can lead to effects which limit or eliminate design, fabrication, or operational configuration options. This requires a good understanding of policies and laws (e.g., Environmental Protection Agency regulations) as discussed in section 5.5, Policy and Law.

(2) Historical Precedent—Historical precedent (i.e., confirmation bias) is characterized by expecting previous efforts to result in the same outcomes. This bias often overlooks the changes that

may make the previous efforts inapplicable. This is frequently encountered when using 'heritage' components in similar applications. These components will need to be requalified for the new applications and new environments which often lead to design changes. NASA cost models are generally based on historical precedents and are frequently inaccurate due to the presence of this unrecognized bias. New methods, procedures, and materials change the basis of the cost models but were not anticipated in the previous efforts and are not explicit in the model structure. This bias can be mitigated by recognizing the differences in the system application and environments from previous uses and accounting for these differences in the design, budget, schedule, and operations.

(3) Error—Errors are simply undetected mistakes in the design or operation of a system. This encompasses mistakes in calculations, mistakes in communications, and working from habit. Error is different from ignorance in that the correct solution is known but not implemented. The systems engineer should recognize the sources of error and develop checks for these. Verification and validation are part of the checks for errors in design. Model validation plays a crucial part in ensuring that the model accurately represents the systems integrating physics or logic (software). In complex designs, independent evaluations provide a check against errors in the system. This is often done for flight-critical software.

Communication errors are a significant concern in organizational information flow. The systems engineer should ensure that the correct information is provided to the designers and decision makers for the design to achieve the intended consequences. If inaccurate or incomplete information can propagate through the organizational structure, design or operation decisions can be adversely affected resulting in unintended consequences.

(4) Short-Sightedness—Short-sightedness (what Robert Merton called the 'imperious immediacy of interest') is focusing on near-term consequences and ignoring long-term consequences. Government-driven projects are particularly susceptible to this type of unintended consequence. This can lead to budget, schedule, and performance issues or failures. Budget cycles are annual and the consequences of the budget in 5 or 10 years are often not credibly considered (since situations in the country will be different then). This leads to an emphasis on next year's budget at the expense of the budget in the next decade. Although not intended, design decisions are driven by near-term cost savings that could increase long-term costs. Accountability in the Government system reinforces this and it must be actively and consciously addressed in all phases of the system design and operation by the systems engineer. The systems engineer will also need to work with program management and organizational management to keep this tendency in check. Part of balancing an unintended consequence is to have visible and explicit metrics and cost data to provide a current view of the long-term consequences.

Schedule can also be driven by short-sightedness, by making decisions to achieve near-term milestones that may delay future efforts by months or years. Mission dates are often well known, providing a clear and explicit target of the future consequences. Planetary missions, in particular, have 'must achieve' dates that keep the near-term decision drivers in check. Understanding the relationships of today's decisions to future limitations is a major factor in ensuring the system effectiveness in achieving its intended consequences.

System performance is well understood if the system application context is well defined. This provides a guard against focusing on near term over longer term performance. System performance can be adversely affected by budget and schedule decisions and the consequences of these decisions needs to be clearly understood as part of the decision. There can be a tendency to look past these consequences which can result in a system with reduced capability or failure to meet the mission objectives. The systems engineer is responsible for ensuring these consequences are identified and discussed as part of the decision process. Without this understanding, it is easy to reduce or remove major system capabilities unintentionally.

(5) Cultural Values—Cultural values can lead to cultural bias regarding what can and cannot happen. Cultural values exist in every organization. Many of these values are positive and improve the organization's success in its execution of system design and operation. Cultural values, though, can also create blind spots for the organization. If the organization believes that a consequence will not occur, then it will not guard against the consequence (if negative) or pursue the consequence (if positive). The Columbia Accident Investigation Board[97] and the Roger's Commission[98] both cited strong cultural biases leading to the Columbia and Challenger accidents. The organizational culture supported the belief that the failure sources were not credible and did not adequately protect against these failures, resulting in the two disasters. The systems engineer must first recognize the organizational culture and then protect the system design from any cultural bias that may exist. A key to this is considering consequences with objective facts, recognizing uncertainties and sensitivities, and providing systems integrating physics-based or mathematically-based answers. Cultural bias can be stated more subjectively and may not be based on facts directly relevant to the current system. This difference must be understood to avoid culturally-induced unintended consequences.

(6) Self-Defeating Prophecy. The self-defeating prophecy (i.e., by stating the hypothesis, you induce a set of conditions that prevent the hypothesized outcome) is a strong, yet subtle, form of bias in system design and operations decisions. A simple example of this is the statement, 'All colors being equal, I like blue.' The hypothesis, 'All colors being equal,' is immediately defeated by the statement, 'I like blue.' Thus, placing stronger consideration of blue over all other colors, meaning they are now not all equal. The systems engineer will need to recognize these subtleties in meetings and discussions and ensure that problem statements do not contain subjective statements that bias the solution.

The corollary of the source, the self-fulfilling prophecy, must also be guarded against. The self-fulfilling prophecy is complex in action. It involves declaring an option to be the best (or the worst) and working hard to show that option is the best while not giving the same effort to other options. This creates a bias in the design or operations team for or against the option without adequately considering all aspects of that or the other options. This can be a dangerous bias and can lead to system failures or other unexpected/undesirable results.

The normalization of deviance (i.e., that the deviated results are expected) is another similar construct where the abnormal performance becomes expected and then becomes the normal course of the system operation.[24] This was cited as a factor in the Columbia accident.[97] The systems engineer must guard against such normalization, keeping the uncertainties and sensitivities before the system design or operations team.

5.1.2 Sociological Systems Theory

The field of sociology has developed its own systems theory, called sociological systems theory, which is useful when facilitating discipline integration. This theory is an offshoot of general systems theory, though sociological systems theory is rather new, having been primarily developed in the last few decades. On the most basic level, sociological systems theory understands systems as systems of communication with self-referential operations that influence their environment.[99] Many sociologists have contributed to sociological systems theory, and Niklas Luhmann is one of the pioneers. It is his interpretation of the theory that is outlined here for use in discipline integration.

In the application of fundamental concepts of sociological systems theory through various systems engineering tools (see sec. 5.6), we can foster the engineering of elegant systems in a multidisciplinary environment. Sociological systems theory can be applied in any social system, but it is especially helpful in resolving conflict in organizations where many social subsystems interact.[100]

Functional differentiation, polycontexturality, and polyphony/heterophony are three important concepts in sociological systems theory,[101] explained and illustrated in the context of discipline integration:

(1) Functional Differentiation. Through the lens of functional differentiation, the world is many operationally closed, autonomous, communicative systems, coexisting, each with their own specific functional purpose. Examples of these societal systems include law, economy, politics, art, education, science, religion, love, medicine, and mass media. No function can control, dominate, or substitute for another function.

(2) Polycontexturality. Modern society is polycontextural. Social systems have many unique, overlapping functional logics and are influenced by their own historical evolution. These contextures develop, adapt, and change, and a major part of their evolution is the communication between different contextures. Each system mentioned above (law, economy, etc.) is monocontextural and views the world per its own binary code. These systems overlap and interact and form a polycontextural society.

A polycontextural construction with a product that requires discipline integration in its development can be illustrated. Society is the greater system that holds all social systems. Figure 39 shows several examples of functionally differentiated social systems (engineering, law, government, and academia), each with its own requirements, but which interact to produce a system.

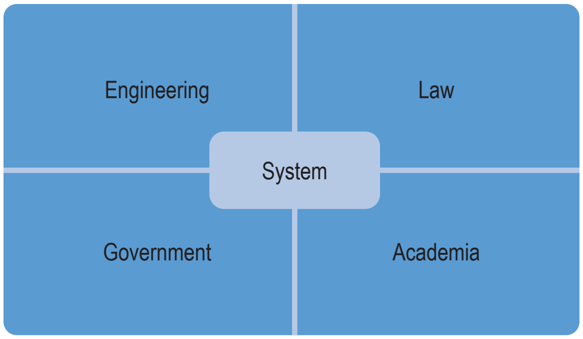


Figure 39. Society—the product as a polycontextural construction.

(3) Polyphony/Heterophony. Polyphony, also referred to as heterophony, refers to the fluidity of the relationships/links between the function systems. On the organizational level, many function systems interact, and the interaction between these systems is not static. In large research organizations, with legal, economic, engineering/technical, scientific, and political systems (among others) interacting, the relationships between them evolve. They are in a constant state of change. Positions in large research organizations, especially those at the interface of different function systems, are most subject to this; for example, with the role of engineering department head in a research organization, operating between the technical and economic function systems.

Functional differentiation, polycontexturality, and polyphony are concepts that are like system concepts in other system science/engineering disciplines. As they are specific to social systems, however, they offer a foundation in which to foster engineering elegant systems.

5.2 Biased Information Sharing

Biased information sharing involves the social negotiation for the use of resources (e.g., funding, schedule, information, labor) in the development of a system. This is not an explicit characteristic identifiable in the system design but resides within the organization's social structure.

Early in the SLS program, a series of interviews were conducted with the senior design engineers within the organization. A key finding from these interviews is that design margin is maintained by the organization, and not in the design. This finding led to further study and the development of simulations to model how the margin is negotiated and shared. Several factors in the sharing of design margin were identified.[102,103] Note that because 'design margin' has a specific meaning in the design research literature, it is referred to more generally as 'bias.'

Several themes emerged from analysis of the results presented above. First, the interview data clearly demonstrate the use of biases and decreasing bias over time between subsystems in the organization. All the negotiation structures in the organization, both formal and informal, are susceptible to this type of behavior. The framework used in the simulations is derived from this information. Second, the use of biases leads to both suboptimal and an increased number of iterations in simulations. Third, this behavior was observed across a variety of multiobjective problem types and structures.

The use of a decreasing bias strategy was described by almost all of the subsystem engineers and by the system integrators and was identified as a possible cause of system suboptimality. In practice, subsystem engineers report that they provide conservative, worst-case estimates of design parameter, and point design information in discussions with other subsystems. Interviews indicated that this was due to a desire to underpromise and overdeliver. It may have also been driven by a competition for resources (i.e., personnel or money) between the different subsystems. The scheduling of limited resources puts pressure on the organization not to pursue improved estimates. If a conservative estimate meets the need(s) of other disciplines, then the organization is not likely to pursue further reductions in the conservative estimates. This use of people and money to reduce conservatism only occurs when an event, such as a request from another discipline, triggers the development of less conservative estimates. Decreasing biases is one strategy for ensuring a subsystem has the resources it needs to complete the required tasks and be robust to unexpected design constraints.

This can be an effective strategy at the subsystem level, but the simulations demonstrated that it may lead to system-level issues. For example, figure 40 (no bias) shows the final system design to be directly on the Pareto Frontier. While in figure 41, the final system design found using a static bias strategy, from the same starting point, is further away from the Pareto Frontier and clearly less optimal. Likewise, the decreasing bias condition shown in figure 42 did not lead to suboptimal results but did take more iterations. Although commonly used to compare optimization algorithms, the number of iterations is also an important metric when considering the design process. An increased number of iterations reflects a longer overall design process and is of interest, because time is an important resource in any design project. For example, time constraints can be viewed as limiting a design team to a fixed number of design iterations. A team using the decreasing bias strategy may reach a suboptimal result given the same number of iterations when compared to a team using no biases, especially if the number of iterations required to reach the Pareto Frontier is large. However, given an infinite amount of time and other resources, the decreasing bias strategy actually may be preferable to the no bias case because it reaches the same level of optimality and the ‘refinement’ period near the end giving the design team more confidence that they are still in the feasible region.

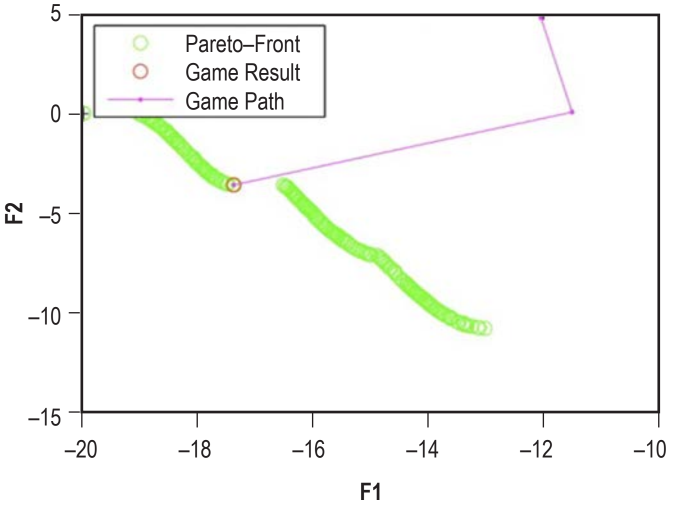


Figure 40. Solution path in the no bias condition.

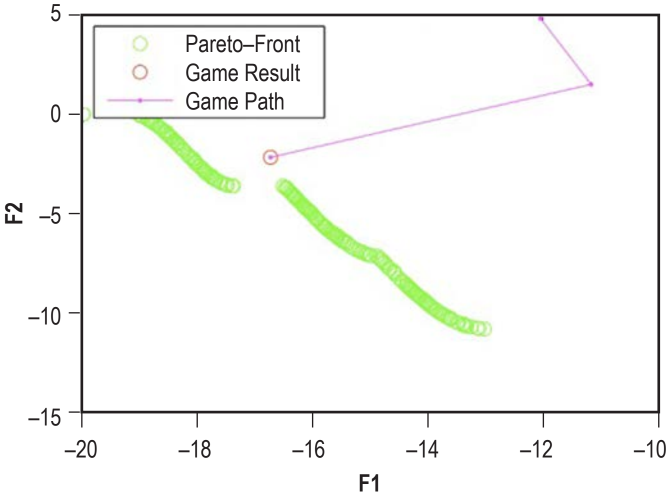


Figure 41. Solution path in the static bias condition.

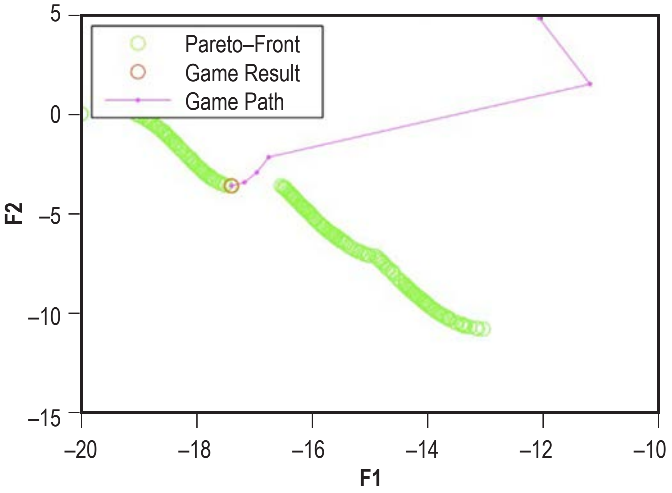


Figure 42. Solution path in the decreasing bias condition.

The results demonstrated use of biased information passing throughout the organization studied at the subsystem level. This reportedly led to suboptimal system-level results. Simulations of three conditions—no bias, fixed bias, and decreasing bias—showed significant changes in system behavior with the addition of biases. Two types of errors were observed regarding speed and optimality.

Biased information is an important aspect in managing the information in the organization. The following three questions address some of the practical implications of this phenomenon:

(1) What strategies do real-world aerospace designers and engineers use when negotiating design parameters with other subsystems? Practitioners interviewed reported negotiating tradeoffs between subsystems in styles that may be described as MDO and Game Theoretic structures. Lower

level negotiations were done informally in a Game Theoretic structure, while higher level negotiations were done formally in front of upper management committees. Interviewees also reported the use of biased information passing between subsystems during negotiations at all levels.

(2) What impact might these strategies have on system-level optimality? Although the size of the effect was problem dependent, biased information passing negatively affected system-level optimality across all problem types tested. Solutions that resulted from strategies incorporating fixed biased information passing negatively affected system-level optimality to a high degree. Solutions resulting from strategies incorporating a decreasing bias had the same level of optimality as those with no bias.

(3) What impact might these strategies have on the speed of optimization? The speed as measured by the number of system iterations was not affected using a fixed bias in most test problems. However, a decreasing bias strategy increased the number of iterations significantly and the amount increased for more complex problem types.

5.3 Decision-Making Structures

Decision making is a formal process in systems engineering that takes the outcomes of the lower level opportunity structures and makes formal agreements on the system design or system operations. There are several aspects of decision-making structures the systems engineer should be cognizant and then provide input to project management to ensure an efficient information flow and limit the uncertainty in the decision-making process. Information theory provides a mathematical basis for structuring decision boards. The overlap of delegated board scopes can lead to uncertainty in the decision process and is an aspect that the systems engineer should work the project manager to ensure the board is properly structured.

5.3.1 Decision-Making Processes

The SLS decision-making processes were studied for three different change requests. A survey was administered to the participants in the decision task team and in the change request evaluation. Change requests are a formal method to request a change in an approved engineering baseline. SLS uses task teams to develop change requests, evaluate options, and provide some vetting of the requested change prior to presenting to the board for a decision. It is interesting that all three changes evaluated had very different affected groups and all three were conducted differently in the task team phase. Yet, participants in all three changes indicated high-quality evaluations and proper decisions were made. All three decisions followed the same process during the change request portion of the evaluation. Thus, the importance of having a complete and knowledgeable set of reviewers appears to be the most important factor, although the process varied quite significantly. This study also resulted in the following three primary findings:

(1) There is a need to include all involved parties in the discussion of the comments on changes. This requires additional upfront resources but may solve issues in the long term. Changes could be made to a change request during discussion of a comment which were not known to those not included in the discussion. This could result in unintended consequences.

(2) The difference between NASA's schedule and the contractor's schedule could impact the effectiveness of the decision process and result in a suboptimal decision. A decision-making approach that has the contractor and contracting organization schedules in synch facilitates needed changes and quick response.

(3) The time and resources to review, understand, and completely assess all the change requests is limited. Engineers feel the pressure to respond quickly but this has the risk of overlooking a problem or implementing a conservative answer/comment.

5.3.2 Information Theory of Decision Structures

Information flow through an organization in the development or operation of a system is an important aspect of systems engineering.[104] Systems engineering ensures that the correct information is provided to the correct engineers when or before it is needed. This is performed by understanding and managing the information about the system which resides in the design and in the organization. Information theory provides the tools to understand and manage this flow and the organizational decision structures that utilize this information.

Information theory provides a mathematical structure to model a decision-making body (e.g., decision board). Warren Weaver discussed the basic concepts supporting this at the beginning of the work of Claude Shannon and Warren Weaver in The Mathematical Theory of Communication.[38] Webster defines information theory as "a theory that deals statistically with information and the measurement of its content in terms of its distinguishing essential characteristics or by the number of alternatives from which it makes a choice possible, and the efficiency of processes of communication between humans and machines."[105] Expanding this definition to include human communication encompasses the organizational communications and, hence, decision-making bodies. The decision-making body essentially operates as a communication system that information is presented and shared in an open forum. Figure 43 illustrates a basic communication system model.

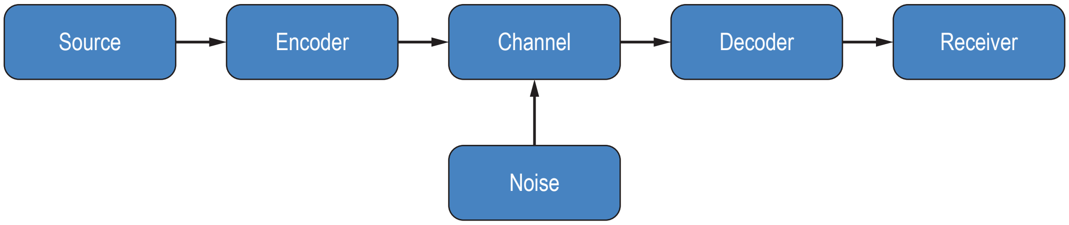


Figure 43. Communication system model.

Information transmitted through this communication system model is a logarithmic function:

$$I = -\log p_n \ , \tag{222}$$

where p_n is the probability that message f_n was sent. Taking the average, this is the measure of uncertainty that the transmitted information represents a specific event,

$$\bar{I} = H = -\sum_n p_n \log p_n \, . \tag{223}$$

$\bar{I}$, representing the uncertainty that an event occurred, is also defined as the Information Entropy, H, of the communication system.[38]

5.3.2.1 Single Board Structures. In the context of a board structure, each board member acts as a source and encoder, contributing information to the discussion. Each board member also acts as a decoder and receiver, receiving information and understanding (or interpreting) the meaning of the information. In this model, the board members include the board chairperson. The chairperson has the final decision authority in the board setting. Subject matter experts (SMEs) often present information to the board or can be additional sources contributing information to the board discussions. The channel is the board meeting. Noise includes many factors including uncertainty in the information presented to the board, distractions (i.e., side conversations, board members working other issues on e-mail or text), or physical noise in the room or on phone lines. Following this structure, a board can be modeled as illustrated in figure 44.

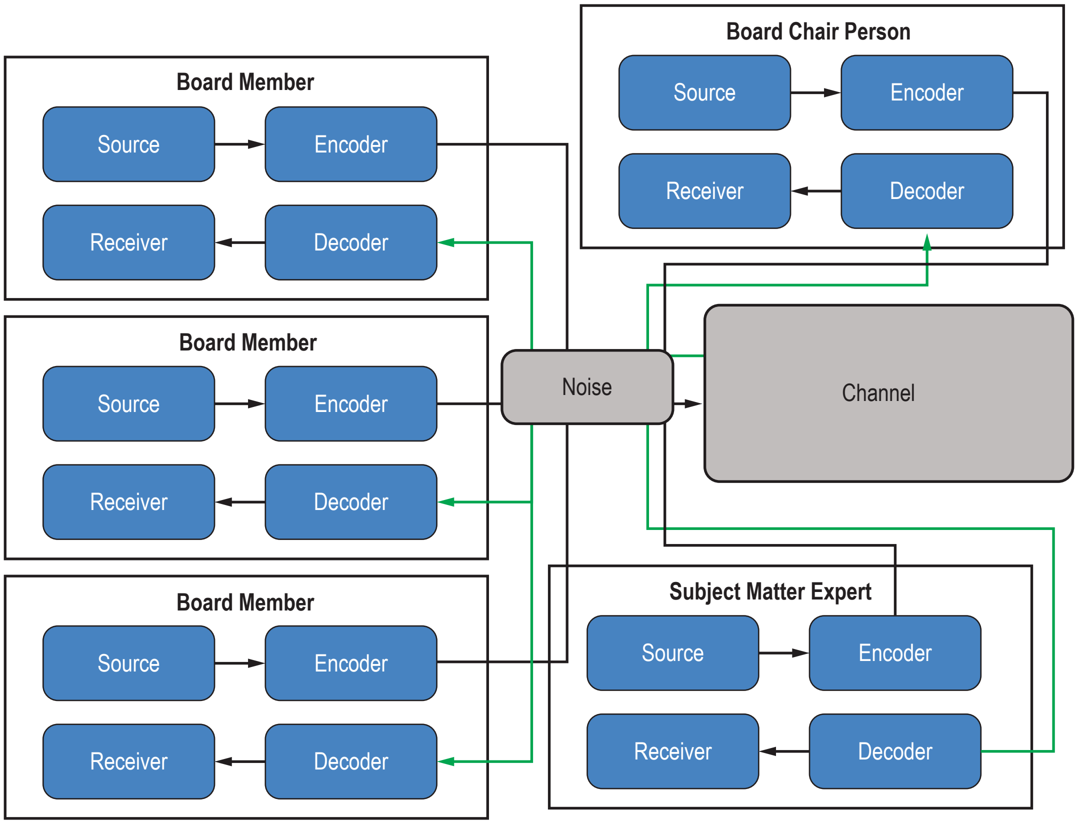


Figure 44. Information theory board model.

This model provides for the inclusion of cognitive aspects of the board members. Each board member must present information in a clear and understandable manner. The extent of their skill in this is represented by the encoding of the knowledge that they possess. In addition, the decision to share or withhold information is a cognitive aspect of the board member. Similarly, the ability of each board member to understand what is being discussed is represented by the decoding of the information (understanding). Many cognitive factors influence the decoding (understanding) of the information including education background, experience, intuitive ability, etc. Cognitive science, then, can be used to establish the distribution functions for the knowledge, encoding, and decoding of each board member and SME.

In this simplest form, this board model assumes that all information needed to decide is provided to the board and that the information is properly and completely understood. Therefore, the uncertainty in the decision is zero, and the information entropy $H = \bar{I} = 0$. This does not mean that no information is conveyed by the board, but that there is no uncertainty in the board decision. In this simple model, the uncertainty (or absence of uncertainty) is absolute in the sense that the decision is fully understood and is not subjective. This is not a valid assumption in practice where various types of uncertainty exist in decision making about systems. Understanding the decision outside the board is not addressed in this model and can lead to uncertainty in the larger context as well.

There are many sources of uncertainty in board decisions. These include hidden (or withheld) information, cultural biases (creating blind spots on certain topics or ignoring factors),[96] ignorance (not understanding aspects of the topic),[92] and missing information in the board discussion. As decision-making bodies, decision boards are chartered with controlling a particular program, project, system, etc. As such, control theory applies to the basic functions of a board as part of the system control strategy. Boards can be modeled as a finite impulse response (FIR) system. Each board member comes to the board with information on a given topic. This information is cognitively processed forming preferences (i.e., weightings), relationships with other information, etc. These cognitive processing functions are quite complex. The board member then communicates with other board members during the board meeting and adds this information with their initial thoughts to create or modify their position. Thus, each board member's thought processes can be modeled as a cascade filter with feedback as shown in figure 45.

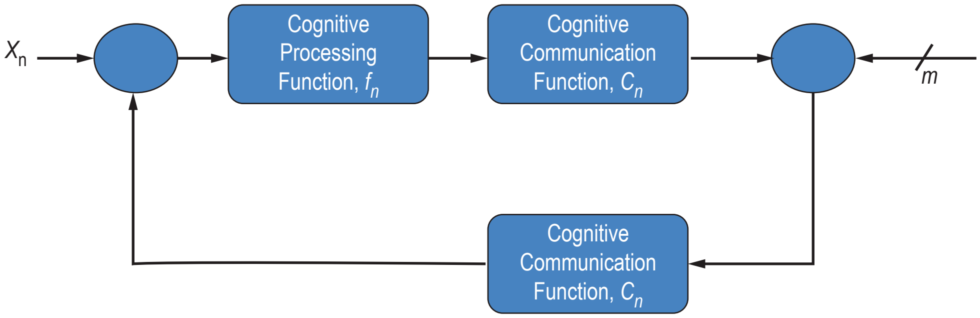

X_n
Cognitive Processing Function, f_n
Cognitive Communication Function, C_n
m
Cognitive Communication Function, C_n

Figure 45. Board member cognitive processing model.

When all the board members and SMEs are combined, the board meeting can be represented using a cascade filter model. Figure 46 illustrates the control theory model of board operation. In this representation, the information theory model relationship is clearly seen. The addition of the information of the board members and SME during discussion is the channel and noise is injected into the channel from external disturbances.

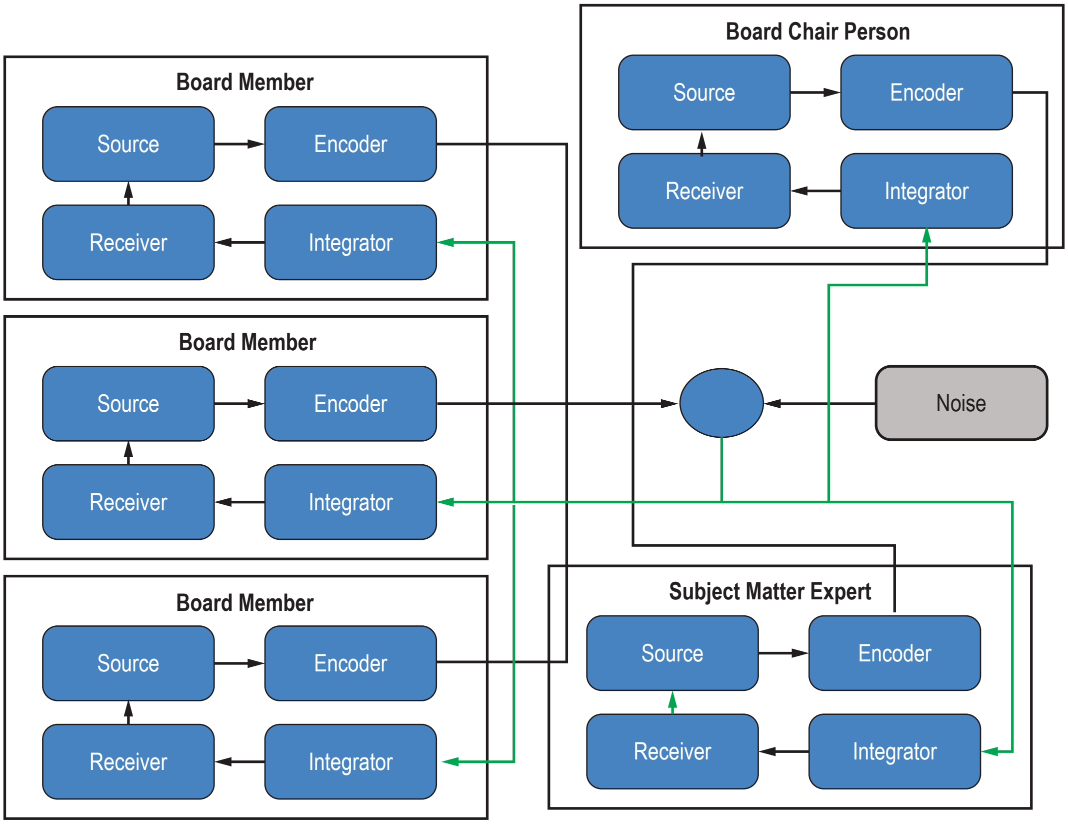


Figure 46. Control theory board model with information theory board member representation.

The board model can be updated with the board member decision-making model where the Encoder is one form of Cognitive Communication Function and the Decoder is another form of Cognitive Communication Function. The Source and Receiver are combined as part of the Cognitive Processing Function, and X_n is contained in memory. Figure 47 illustrates this model.

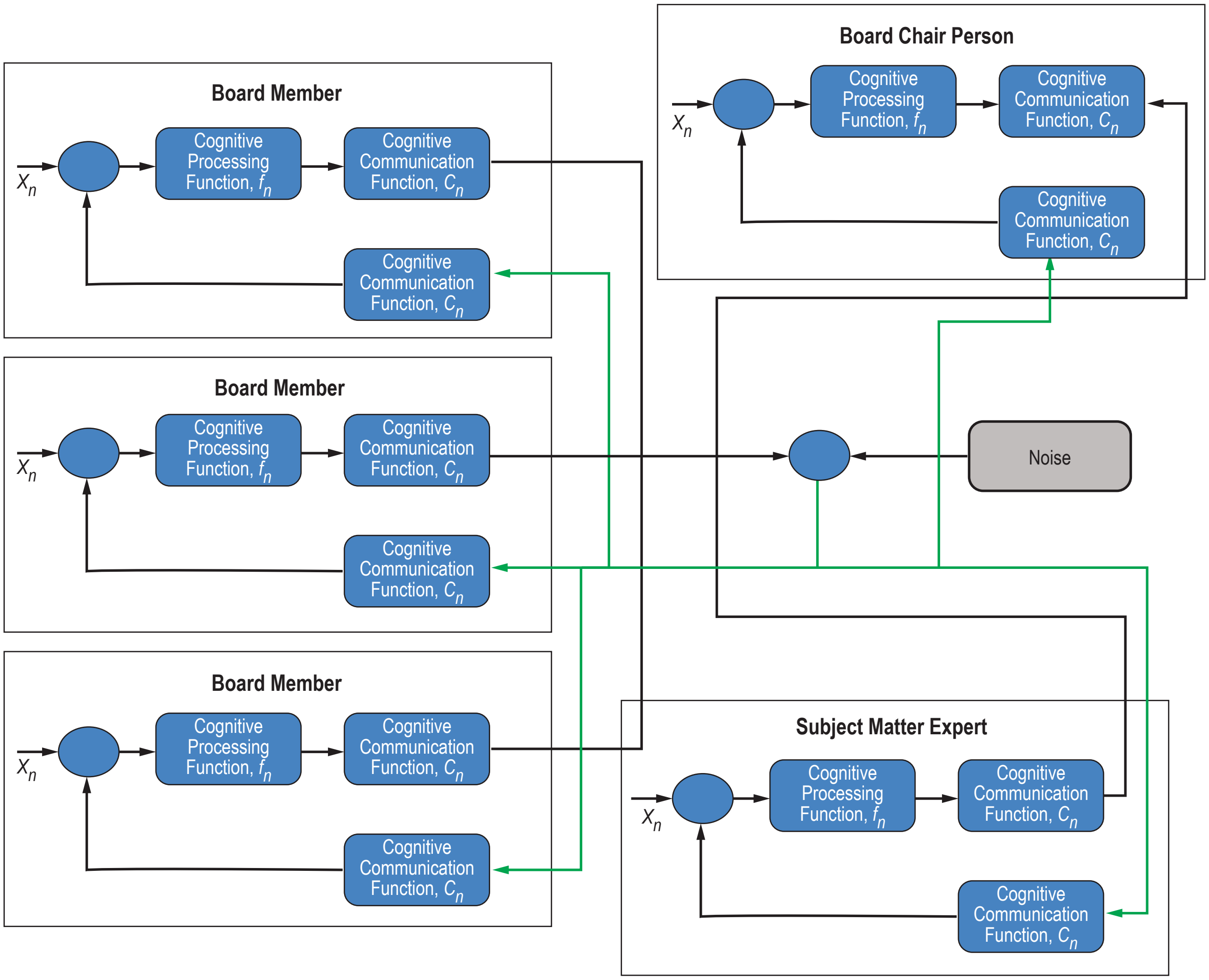

Board Member
X_n
Cognitive Processing Function, f_n
Cognitive Communication Function, C_n
Cognitive Communication Function, C_n
Board Member
X_n
Cognitive Processing Function, f_n
Cognitive Communication Function, C_n
Cognitive Communication Function, C_n
Board Member
X_n
Cognitive Processing Function, f_n
Cognitive Communication Function, C_n
Cognitive Communication Function, C_n
Board Chair Person
X_n
Cognitive Processing Function, f_n
Cognitive Communication Function, C_n
Cognitive Communication Function, C_n
Noise
Subject Matter Expert
X_n
Cognitive Processing Function, f_n
Cognitive Communication Function, C_n
Cognitive Communication Function, C_n

Figure 47. Control theory board model with cognitive functions.

The equation represented in figure 46, can be written as:

$$X_n + \sum_q f_{p,q+1}\left\{C_p\left[f_{p,q}\left(X_n\right)+\sum_{m,\,m\neq p} C_m\left[f_m\left(X_n\right)\right]+\text{noise}\right]\right\}, \qquad (224)$$

where X_n represents a specific piece of information, q is the number of iterations in the board discussion, the subscript, p, represents a specific board member or SME, and the sum over m is the remaining board members or SMEs. Equation (224) then represents the decision reached by the decision-making board member with inputs from the other board members and SMEs. This model assumes all board members and SMEs start with the same basic information, X_n. It allows understanding of the information to vary among the board members represented by the function, $f_{p,q}$. In this model, if a board member or SME has no knowledge of the topic (i.e., ignorance of the subject), $f_{p,1}\left(X_n\right)=0$. Similarly, if the board member withholds information on the topic, $C_p\left[f_{p,q}\left(X_n\right)\right]=0$. The function, $f_{p,q}$

represents the level of understanding on a subject. A decision to not share information is represented by this function as well. This function also encompasses preconceived ideas on the given information, preferences (personal or shared), intuition, deductive reasoning, and inductive reasoning. Clearly, the form of this function is complex and the subject of future application of cognitive science.

Using this model, the transfer function of the board can be represented as the ratio of the initial understanding of the information divided by the final decision as shown in equation (225):

$$H_n = \frac{X_n + f_{p,q}\left(X_n\right)}{X_n + \sum_q f_{p,\,q+1}\left\{C_p\left[f_{p,q}\left(X_n\right) + \sum_{m,\ m\neq p} C_m\left[f_m\left(X_n\right)\right] + \text{noise}\right]\right\}} \ . \tag{225}$$

There are other information sources that can contribute to this model as SME inputs. These include text and e-mails to board members, personal side discussions (which also contribute to noise and affect the intake of other information). Since these inputs do not go to the whole board, but rather to individual members, and the SME (particularly in electronic communication) may not be receiving all the board discussion, they have a function of $C_s\left[f_s\left(X_n\right)\right]$, where *s* represents a specific SME and there is no iteration with the board discussion, *q*. The inputs are single events since SME is not part of the board discussion.

5.3.2.2 Multiple Board Relationships. A question often asked is, what is the most efficient board structure? Will a single board suffice or are multiple boards more efficient? This has been a difficult question to answer. The set theory view of information theory provides an approach to answer this question.

A range that is too small (missing expertise) does not map into the decision domain for the intended outcome of the system. If this range can be mapped, then the missing expertise is not necessary for the decision in the context of the system. This immediately tells us that our board must have the right distribution of expertise for the system context and is therefore system specific.

Information theory provides additional keys to understanding the board structure. Partitioning of information entropy, *H*, can only increase the uncertainty in the system by the relation

$$H\left(p_1,\ p_2,\ \ldots,\ p_n,\ q_1, q_2,\ \ldots q_m\right) \geq H\left(p_1,\ p_2,\ \ldots,\ p_n\right) \ . \tag{226}$$

Thus, as more members are added to the board, more uncertainty is created in the decision process. However, this is balanced by range mapping becoming complete. Thus, the board structure needs to have only those members necessary for the system decisions (satisfying the mapping condition) and no more (minimizing *H*). This applies to a single board structure or a multiple board structure.

Within the set theory view of information theory, the board can be split (or delegated) if the information needed in one board is different than that needed in any other board. Then,

$$I_A \not\subset I_B \text{ and } I_B \not\subset I_A \;, \tag{227}$$

so there is no intersection of the information needed by the board and the board's domain (scope) can be different for each board.

When there is scope overlap, then, $I_A \cap I_B$, and the boards cannot be separated. In this case

$$I_A \subset I_B \text{ and/or } I_B \subset I_A \,. \tag{228}$$

5.3.2.3 Statistical Properties of Boards. Splitting a board into multiple boards where there is significant overlap greatly increases the information uncertainty, *H*, in the board structure. To examine this, we need to start with the characteristics of the uncertainty, or entropy, function. There are four axioms the information entropy must meet:

(1) Continuity:

$$H\left(p_1,\ p_2,\ \ldots,\ p_n\right) \tag{229}$$

is continuous in all p_n. Thus, there are no discontinuities in the information probabilities. This means, as noted earlier, that the range maps completely to the domain within the board. Discontinuities lead to highly uncertain, or in some cases, blind, decisions. A robust board has all relevant disciplines (i.e., affected or contributing parties) represented. This satisfies the range to domain mapping criteria and the related continuity property.

(2) Symmetry:

$$H\left(p_1,\ p_2,\ \ldots,\ p_n\right) = H\left(p_2,\ p_1,\ \ldots,\ p_n\right). \tag{230}$$

The order of uncertainty does not contribute to the decision uncertainty. This must be distinguished from temporal order leading to a momentary information void on a subject until all aspects are explained for understanding. The process of understanding is always assumed to be complete in this model, and symmetry holds for a complete understanding of a subject. The order in which you discuss or think of a subject does not matter if you fully understand the subject.

(3) Extrema:

$$H\left(p_1,\ p_2,\ \ldots,\ p_n\right) = H\left(p_2,\ p_1,\ \ldots,\ p_n\right). \tag{231}$$

The maximum uncertainty results when all decisions are equally uncertain. If any single decision can be distinguished from the others, then the uncertainty to choose or not choose that option is smaller. Similarly, if no options satisfy the decision criteria, then the board has no information on which to base a decision leading to

$$\text{Min}\left[H\left(p_1,\ p_2,\ \ldots,\ p_n\right)\right] = H\left(0,0,\ \ldots,0\right) = 0\,. \tag{232}$$

(4) Additivity. If a probability of occurrence, p_n, can be subdivided into smaller segments, q_k, then the uncertainty can be represented as

$$H\left(p_1,\ p_2,\ \ldots,\ p_{n-1},\ q_1, q_2,\ \ldots q_k\right) = H\left(p_1,\ p_2,\ \ldots,\ p_n\right) + p_n H\left(\frac{q_1}{p_n}, \frac{q_2}{p_n}, \ldots \frac{q_k}{p_n}\right). \qquad (233)$$

Thus, information can be subdivided during discussions if all the information is presented (i.e., all q_k is present in the discussion) without affecting the uncertainty of the decision. Note that this requires all information to be present and communicated. Subdividing boards that segment the information does not meet this criteria and results in higher uncertainty.

These four properties and five equations from information theory provide guidance in the structuring of boards. These relationships indicate that impossible solutions do not affect the information entropy. These solutions do not fit in the domain of the solution and cannot be mapped from the range of the original decision question. In addition, the continuity of H requires all information be present for a decision. While a decision may be made with missing information, the decision is not actually addressing the original question. The question essentially changes when all information is not present and the decision addresses a different question than the one intended.

Information theory assumes a statistical basis of the information. Before we proceed further, we need to establish the statistical nature of boards, not that they are predictable, but that their underlying operations can be represented statistically.

There are four principles that establish the statistical nature of a decision board:

(1) Uncertainty exists in complex decisions. In these cases, simplifying assumptions lead to a lower understanding of the decision intricacies and a higher uncertainty (not always recognized) in the decision process. Interactions among differing factors in complex decisions have dependencies that are not recognized (ignorance) or not well understood. Missing information is not always easily recognized. Factors not considered important in the decision can end up driving the system. Missing information comes from events (physical, chronological, or fiscal) not recognized as relating to the decision, unknown environments in which a system operates, unrecognized dependencies, and cultural biases (e.g., politics).

(2) The uncertainty of which option is best collectively, and in some cases individually, leads to a statistical representation of which answer is best. In a board decision, the board vote is a statistical event with a distribution of yes and no positions. This is tied back to the cognitive functions, which are currently derived from large surveys of people producing a statistical function in the processing of information by an individual. This statistical function is then combined with other statistical functions (i.e., other board members and SMEs) to produce a model of the decision based on the statistical functions.

(3) The potential for misunderstanding (i.e., error) is also statistical. This includes miscommunication—not stating clearly what is meant or not understanding clearly what is stated, and therefore meant. These lead to unintended consequences in the decision-making process. These unintended consequences can be social, physical, chronological, fiscal, or environmental.

(4) Cultural and historical bias lead to suboptimal decisions. Large social population actions form the basis for these biases and the effects on a person's cognitive information processing function, f_n, are statistical in nature.

Decisions can be represented statistically with various distribution functions depending on the individual preferences, biases, knowledge, and experience with the subject as discussed in the control theory model above. The cognitive processing functions, based on the properties of H, should fulfill continuity, symmetry, extrema, and additivity.

5.3.2.4 Information Channel in the Board Context. In the board context, the board discussion forms the information channel. The board members and SMEs are both information sources and sinks as modeled in figure 46. Information theory treats communication as the transmission of symbols. Natural language, where letters form words, words form sentences, and the order of the symbols and words are important in interpretation fits this model perfectly as noted by Shannon and Weaver,[38] and the board discussion is the channel where this information is transmitted between the board members and SMEs.

Information theory models the transfer of information through the board channel very well. A definition of terms is convenient at this point:

$H\left(X_n\right)$ = average information shared by a single board member or SME as defined in equation (224).

$H\left(Y_n\right)$ = average information received by a single board member or SME also following the definition in equation (224).

$H\left(X_n,Y_n\right)$ = joint probability that what was shared by one member and is heard by another (the average uncertainty in the total transmission through the board channel).

$H\left(Y_n \mid X_n\right)$ = probability that one member actually heard what was stated by another. This brings in the effects of noise (and misunderstanding) in the channel. This focus is on the receiver of the information.

$H\left(X_n \mid Y_n\right)$ = equivocation probability that one member actually stated what was heard by another. This brings in the effects of recovery (or proper understanding) of the information sent and is a measure of how well the information is understood by the receiving member.

If the board discussion is clear, and no misunderstanding exists, then the information provided by the speaker is accurately received by the listener (receiver). The information is perfectly transferred and Information theory tells us that,

$$I(X;Y)=H(X,Y)=H(X)=H(Y) \ . \tag{234}$$

Now, if there is complete confusion, then what is stated is not related to what is heard. This is the case where the received information is independent of the transmitted information and,

$$I(X;Y)=0\ . \tag{235}$$

In this case, no information is transmitted through the channel (i.e., discussion). These two extremes, perfect transmission and no transmission, provide bounds on the information sharing in a board meeting. Typically, neither of these conditions is achieved and there is always some noise or misunderstanding during the discussion that limits the amount of information transferred among the board members.

5.3.2.5 Information Theory Representation of a Board. Set theory provides the mathematical basis for information theory which fits the board structure well. Information shared in a board discussion is the sum of all the information provided by the individual board members. This is illustrated in figure 47 for the example board structure used in figure 45.

This picture is somewhat complex in that there are many different areas of shared information. Note that the symbol, |, is read as 'and not' so that $I(S;D,X|Y,Z)$ is the information shared between the SME, S; the Board Decision Maker or Chair, D; Board Member, X; and not Board Member Y or Board Member Z. For a decision to be fully informed, the information for the decision must be contained in the center-most ellipsoid, $I(S;D,X,Y,Z)$. This represents the set of all information shared and received in the board discussion. Other information is shared based on the knowledge of individual board members and the SME, the ability of each to understand the information, and individual distractions. This can lead to board discussions that do not fully incorporate all board member knowledge. All permutations of this case are represented in the figure except for $I(Y{:}D,Z|S,X)$, $I(S{:}X,Z|D,Y)$, $I(S;X,Y|D,Z)$, $I(X{:}D,Z|Y,S)$, and $I(S;X,Z|Y,D)$, which is an artifact of the figure geometry (where nonadjacent sets cannot be shown as excluded).

Information theory represents this as

$$H(S,D,X,Y,Z)\leq H(S)+H(D)+H(X)+H(Y)+H(Z)\ , \tag{236}$$

where $H(S)$, $H(D)$, $H(X)$, $H(Y)$, and $H(Z)$ are how well the board members and SME communicate their information. This indicates that the sum of information can be no more than that provided by each of the members. Noise (distractions, misunderstanding, poorly stated (poor transmission)) and information not shared (intentional, unintentional, missing board member) invokes the inequality in the relationship.

Following the work of Reza, set theory can relate the rules for information.[106] This yields the following relationships:

$$I(X;Y)\ =f(X\cap Y), \tag{237}$$

which is the expected value of mutual information shared in the discussion. In set theory, this is a function of the intersection of the information held by X and Y:

$$H(X,Y)=f(X\cup Y)\ , \tag{238}$$

which is the average uncertainty of the discussion. This is a function of the union of the information available:

$$H(X\,|\,Y)=f(XY')\ , \tag{239}$$

which is the information received by X given the information that Y shared. This is the probability that the board understood the information shared by Y. Note, in set theory, this is a function of the information X has that Y does not:

$$H(Y\,|\,X)=f(Y'X)\ , \tag{240}$$

which is the information shared by Y given the information that X heard. This is the probability that the board understanding is what was shared by Y. Note, in set theory, this is a function of the information Y has that X does not.

From these relationships, then, perfect understanding occurs when $f(\mathrm{X})=f(Y)$ and both parties understand the information fully. When there is, no information shared, $I(X;\,Y)=f(X\cap Y)=0$. Thus, there is no intersection of the information sets and no common understanding. In the board example used above, $I(S;D,X,Y,Z)=f(S\cap D\cap X\cap Y\cap Z)$ and the shared information is represented in figure 48 by the intersection of the five circles representing the knowledge to share for each decision.

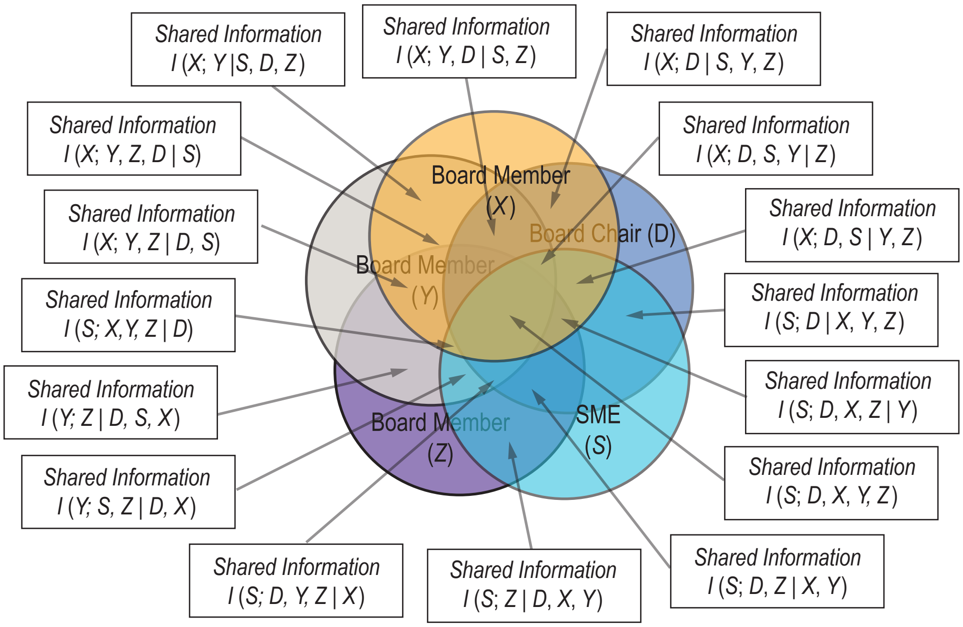

Shared Information I (X; Y |S, D, Z)
Shared Information I (X; Y, D | S, Z)
Shared Information I (X; D | S, Y, Z)
Shared Information I (X; Y, Z, D | S)
Shared Information I (X; D, S, Y | Z)
Shared Information I (X; Y, Z | D, S)
Shared Information I (X; D, S | Y, Z)
Board Member (X)
Board Chair (D)
Board Member (Y)
Shared Information I (S; X,Y, Z | D)
Shared Information I (S; D | X, Y, Z)
Shared Information I (Y; Z | D, S, X)
Shared Information I (S; D, X, Z | Y)
Board Member (Z)
SME (S)
Shared Information I (Y; S, Z | D, X)
Shared Information I (S; D, X, Y, Z)
Shared Information I (S; D, Y, Z | X)
Shared Information I (S; Z | D, X, Y)
Shared Information I (S; D, Z | X, Y)

Figure 48. Set theory representation of board.

In these representations, $H(X)$, etc. represents the uncertainty in the information shared by board member X. This uncertainty stems from the board members' understanding (or knowledge) of the decision requested and the associated decision factors, cultural bias (which indicates if information will be shared or withheld), and personal comfort in sharing specific information or engaging in debate about the information.

Channel capacity (i.e., board capacity) in information theory is:

$$C = \max\left(I\left(X;Y\right)\right) = \max\left(f\left(X \cap Y\right)\right) \ . \tag{241}$$

Channel capacity (i.e., the board capacity) for a decision is defined by the mutual information, or the intersection of information, shared in the board discussion. The maximum board capacity then is based on the intersection of knowledge held by each board participant. The intersection represents the integration of individual board participant's knowledge to form a decision. This indicates that if a board is segmented, and required knowledge for a decision is not present, then the board does not have the information necessary to decide. A decision can be made, but the scope of the decision does not address the actual question being considered. This results in unintended consequences for the system decision because the board does not have all the facts.

Note that a board with a missing member(s) will have a lower capacity since mutual information for the topic will be reduced. Similarly, adding a member that has largely overlapping knowledge can create disjoint relationships where the two members approach the topic differently (based on their differing cognitive functions), do not overlap in their understanding, and $I(S;D,X,Y,Z)=0$, blocking the board decision.

5.4 Cognitive Science Influences

5.4.1 Overview

The success of complex systems design efforts depends on engineers' aptitude for synthesizing knowledge across disciplines and creating shared understanding of technical system components and the interactions between them (postulates 2, 3). Doing this requires a unique blend of technical acumen, cognitive skill, and social capital, collectively referred to as 'engineering systems thinking.' This section describes ongoing research in engineering systems thinking and network analysis that analyzes human and organizational influences on systems engineering, design, and management.

Cognitive and social science principles are being applied by investigation of engineering systems thinking. This capability provides methods for understanding and teaching engineering systems thinking skills to systems engineers. To address organizational influences on system performance, network theory is being explored as technique for modeling and analyzing sociotechnical systems management processes.

5.4.2 Engineering Systems Thinking for Systems Engineering and Design

Several general trends in the study of engineering systems thinking have been identified. First, engineering systems thinking is primarily studied through behavioral observation and defined in behavioral terms in the systems engineering literature. One notable example is a 2008 study conducted by NASA's Office of the Chief Engineer, in which 'highly-valued' systems engineers—those identified by NASA Centers as 'go-to' people with regards to systems engineering—were interviewed, shadowed, and observed. Findings summarizing their behavior were organized and distributed as an interagency report.[107] The study resulted in rich descriptions of individual attitudes, problem solving strategies, technical acumen, and systems thinking capabilities required to do systems engineering at the highest level, and described leadership, communication, and other social skills as equally important for systems thinking and systems engineering. These findings are mirrored in several other studies from industry and academia.[108–112]

Second, practicing engineers and engineering researchers are beginning to acknowledge the need for a more rigorous interdisciplinary framework for engineering systems thinking, to understand the social and psychological underpinnings of engineering systems thinking behaviors.[113] Frank[111,112] describes systems thinking as ‘a high-order thinking skill’ and suggests several cognitive processes that might facilitate systems thinking. Work by Rhodes and colleagues Lamb, Nightingale, and Davidz[108,109] suggests that enabling systems thinking is a critical step in advancing the development of senior systems engineers, and that experiential learning, education, interpersonal interactions, and training in a supportive environment are necessary for enabling systems thinking. A small-scale Consortium-sponsored study conducted at MSFC in July 2016 characterizes the systems thinking strategies of systems engineers along cognitive and social axes, shown in figure 49. These abilities were repeatedly described as necessary for engineering systems thinking and successful systems engineering, consistent with literature results. Additional work sponsored, in part, by the Consortium explores similarities and differences between cognitive strategies of systems engineers working on complex systems and design engineers working on innovative consumer products.[114,115] The goal of this work is to identify theory and methods from disciplines such as design and psychology that might be applied towards improving systems engineering in a meaningful way.

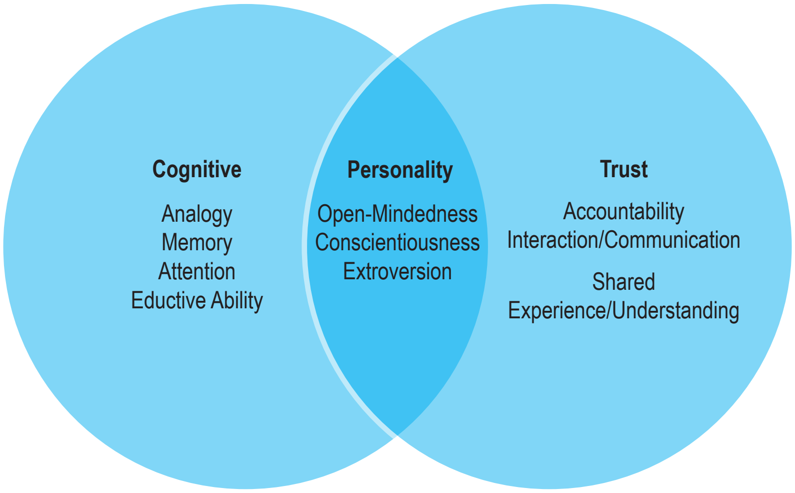


Figure 49. Systems engineering systems thinking strategies.

Some of the systems thinking behaviors described by engineers map directly to concepts from cognitive psychology and could be investigated using psychological paradigms. A list of engineering terminology used to describe systems thinking competencies and their correlates in cognitive psychology are illustrated in table 16. Some competencies have direct mappings to psychological concepts as noted in the table, although the majority do not. It is important to keep in mind that these mappings are not perfect and at this stage only serve as examples of how systems engineering might begin to move from engineering descriptions of cognitive competencies to cognitive descriptions. Higher level factor analysis is required to assess statistical correlations between qualitative descriptions of systems thinking and formal measurements of cognitive processes.

Table 16. Mapping between observed cognitive 'competencies' of successful systems engineers to cognitive processes required for generating these behaviors.

Cognitive Competencies From Frank, 2012[112]	Related Concepts From Cognitive Psychology[116–121]
Understand the whole system and see the big picture	Sensemaking; information integration; mental model formation; generalization
Understand interconnections	Induction; classification; similarity; information integration
Understand system synergy	Deductive inference
Understand the system from multiple perspectives	Perspective taking (direct mapping)
Think creatively	Creativity (direct mapping)
Understand systems without getting stuck on details	Abstraction; subsumption
Understand the implications of proposed change	Hypothetical thinking
Understand a new system/concept immediately upon presentation	Categorization; conceptual learning; inductive learning/inference
Understand analogies and parallelism between systems	Analogical thinking (direct mapping)
Understand limits to growth	Information integration
Ask good (the right) questions	Critical thinking
(Are) innovators, originators, promoters, initiators, curious	Inquisitive thinking
Are able to define boundaries	Functional decomposition
Are able to take into consideration nonengineering factors	Conceptual combination
Are able to 'see' the future	Prospection
Are able to optimize	Logical decision making

Some of the cognitive processes implicated in engineering systems thinking have been studied at length by the product, industrial, and mechanical design communities, and findings may be applicable towards the study of systems engineering and design. These disciplines offer many formalized methodologies for design that include descriptions of theory, processes, and best practices, as well as observations about social behaviors and psychological processes that comprise 'designerly ways of knowing.[122] One such approach—design thinking (DT)—bears strong resemblance to engineering systems thinking, having emerged in parallel as another method for tackling wicked design problems in diverse settings.[115,122–126] Empirical inquiry into design thinking is extensive, with practical application in engineering design,[127–129] management and organization,[130–132] pedagogy,[133,134] and myriad other contexts. Design thinking methodology has also been experimentally applied at NASA Centers to cultivate innovation during rapid conceptual design.[135]

Contemporary applications of the term 'designer thinking' generally refer to the study of 'cognitive processes and strategies employed by human designers working on design problems.[128]

Popular research topics in design cognition include the role of sketching and visual representation, the use of analogies, methods for fostering creativity and ideation, methods for overcoming fixation and blocking, approaches for balancing divergent and convergent thinking, and elucidating strategic differences between expert and novice designers. These processes are explored through case studies, cognitive ethnography, verbal protocol analysis, controlled laboratory experiments, psychometric measurement, and, more recently, through physiological measurement techniques such as functional magnetic resonance imaging (fMRI). A comprehensive review of empirical studies of designer thinking is provided in the paper by Dinar et al.[128]

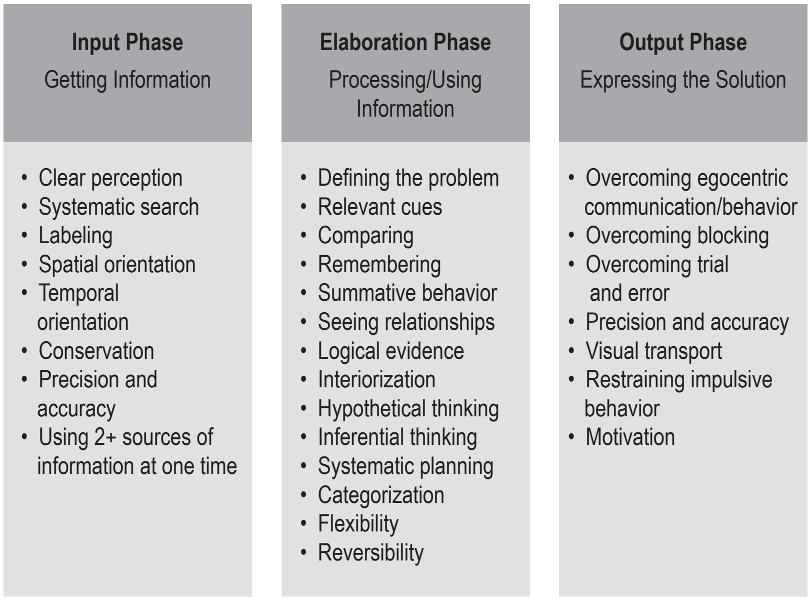
Until relatively recently, design thinking and engineering systems thinking have been explored in isolation from one another. Systems thinking is sometimes described as a component of design thinking,[136] sometimes as separate but related to design thinking,[137] and is sometimes not described alongside design thinking at all. References to design thinking are not often found in the engineering systems thinking literature. Contemporary approaches to engineering systems thinking draw instead from historical perspectives in systems science, operations research, and management science, and have only recently moved to include nontechnical factors (e.g., cognition) in the analysis of engineering systems.

Ongoing theoretical research seeks to make a rigorous, informed comparison of the DT/engineering systems thinking concepts observed in engineering practice today. Analyzing these two concepts relative to one another can help elucidate historical features that make them distinct, while simultaneously identifying areas of contemporary overlap in which one methodology might benefit from insights about the other. This could be a useful first step towards resolving redundancy and discrepancies in the literature and could also help bridge the gap between engineering design research in academia and systems engineering practice in industry and government.

Understanding the foundational cognitive processes required for engineering systems thinking also allows for the development of a prescriptive approach for teaching engineering systems thinking skills in practice. Below, we provide a mapping between some of the cognitive competencies of successful systems engineers as described by Moti Frank and the work of Reuven Feuerstein, a clinical, developmental, and cognitive psychologist renowned for developing an educational method designed to create or correct many of the same cognitive functions that Frank describes.

Feuerstein is recognized for his development of the theory of structural cognitive modifiability and his foundational teaching method, the mediated learning experience (MLE). The theory of structural cognitive modifiability suggests three basic tenets: the brain is plastic and structurally modifiable throughout life, cultural transmission provides an important method for the creation of cognitive structures, and a human mediator may intervene in the mental processes of a learner, creating missing structures or correcting dysfunctional structures in the brain. The essential feature of this approach is that these changes are not simply psychological, but rather of a structural nature that alter the course and direction of cognitive development. The changes that occur ‘are not a fragmented, episodic consequence of exposure to experiences, but rather a type of change that affects the basic structure of behavior.’ Through use of a well-trained mediator, subconscious information processing skills can be brought to conscious awareness and be created, corrected, and improved, ultimately resulting in physiological changes in the brain in addition to the psychological. While these may seem to be rather powerful statements with little substantiation, newer research in neuroscience has provided theoretical support for Feuerstein's work.[138–140]

A pertinent feature is the commonalities between the goals of the MLE and the desired traits of engineering systems thinkers. Figure 50 identifies the cognitive processes targeted by the MLE; processes in the elaboration phase and output phase are quite like the cognitive competencies of individuals with the capacity for engineering systems thinking. To provide an example, 'seeing relationships' is a universal theme in the systems thinking literature, and Feuerstein's method is directed at improving this process and several others. Hypothetical thinking, inferential thinking, and flexibility are also attributes of systems thinkers, and egocentric communication/behavior are barriers to systems thinking; Feuerstein's approach offers a method for addressing these.



Input Phase Getting Information	**Elaboration Phase** Processing/Using Information	**Output Phase** Expressing the Solution
• Clear perception • Systematic search • Labeling • Spatial orientation • Temporal orientation • Conservation • Precision and accuracy • Using 2+ sources of information at one time	• Defining the problem • Relevant cues • Comparing • Remembering • Summative behavior • Seeing relationships • Logical evidence • Interiorization • Hypothetical thinking • Inferential thinking • Systematic planning • Categorization • Flexibility • Reversibility	• Overcoming egocentric communication/behavior • Overcoming blocking • Overcoming trial and error • Precision and accuracy • Visual transport • Restraining impulsive behavior • Motivation

©1976 Feuerstein, R. Hadassah Wizo-Canada
Research Insitute, All Right Reserved

Figure 50. Cognitive processes targeted by Feuerstein's MLE.

While a deeper exploration of Feuerstein's work would be required before claiming that the MLE is a consistent way to develop systems thinking skills in engineers, these early mappings provide a promising direction for future studies. Exploring Feuerstein's work as a complement to contemporary design thinking methodology creates an additional opportunity for future inquiry.

To study systems thinking, an approach was taken using psychometric theory and methods to test, measure, and assess skills and attitudes for systems engineering and systems design. A Likert-scale 'systems design thinking' survey was used consisting of statements about systems engineering and design, and asking participants to select responses ranging from 'strongly disagree' to 'strongly agree' that best characterize how they feel about the statements. Statements describing technical systems, social aspects of design, design management and organization, and individual workflow preferences were asked. Statements were derived from existing research in systems engineering, systems thinking, and engineering design, and drew also from the International Council on Systems Engineering (INCOSE) Systems Engineering Handbook.[141]

5.4.3 Modeling Systems Engineering Processes Using Network Theory and Agent-Based Modeling

Because engineering systems thinking is both an individually and socially oriented process, it is valuable to study the organizational impact of systems thinkers in addition to the individual features that make them unique. Effective verbal communication, for example, is a commonly cited characteristic of engineering systems thinkers, and has been identified as a major contributor to speed and efficacy of problem resolution within organizations.[142] To model individual social attributes and characterize relationships between individuals, a network model of organization is valuable. Technical systems can also be modeled as networks, based on subsystems properties and the interfaces between them. Network models allow analysis at the scales of individual, group, and system. The objective of this analysis is to identify the impact of network structures on observable system behaviors.[143] The relational representations of organization and technical system architecture offered by network modeling and analysis can improve our understanding of the systems engineering process—specifically, the effect of systems engineers and systems thinkers on technical system performance. Systems thinkers are represented by the skills, preferences, and behaviors that comprise systems thinking and are used in systems engineering. These skills, preferences, and behaviors can therefore be integrated into descriptive network models of organization, the designed technical system, and their relationship.

The relationship between organizational and technical systems has been operationalized as measures of sociotechnical performance. Measures include socio-technical congruence,[144] misalignment,[145] and coordination deficit.[146] These make use of matrix- or network-based models of organization and technical architecture. Organizations are represented by the group membership of individuals (e.g., engineering discipline or subsystem expertise) and their interactions through communication or shared work. Technical architectures are similarly represented as subsystems joined through technical interfaces or task dependencies that reflect ordering of tasks to design those subsystems.

Socio-technical congruence is a measure based on the assumption that improved coordination, i.e., the matching of organization to technical system architecture at an interface level, yields improved technical performance. This evokes Conway's Law,[147] which hypothesizes that communication across technical interfaces is what creates technical interfaces, and the 'mirroring' hypothesis that organizational networks and technical system networks have the same global structural properties for successful projects.[148] Misalignment describes an 'unmatched' organizational interface, where there is a technical interface but no communication, or an unmatched technical interface, where there is observed communication but no technical interaction.[145] Misalignment extends the concept of sociotechnical congruence by proposing a possible source of incongruence: cross-organizational boundaries and technical interfaces between modular subsystems are particularly prone to misalignment.[145] This work suggests implicitly that misalignment is an indicator of decreased technical performance, and should be the focus of organizational mediation.

An assumption of causality between organization structure and system architecture is not being made, nor is an assumption made of interface-level matches between organization and technical networks being directly correlated to technical success. Instead, we seek to identify the network properties, including attributes of individuals, communication between individuals, subsystems, or technical interfaces between subsystems, that facilitate technical success. This also includes characteristics of individuals and subsystems, e.g., node degree, the number of individuals one communicates with regularly, or local network motifs, regularly repeating patterns within the network structure.[149]

Skills, preferences, and behaviors of systems thinkers and systems engineers used for the task of coordinating distributed work were identified from the interview data described in the previous section. Two complementary behaviors are firstly setting formal plans and communication to structure one's own and others' work; and secondly proactive communication to tailor interactions with individuals, listen, and ask questions. These behaviors do not solely reside in systems engineers, but may be found in management roles or other engineers as well. Our analysis aims to explicitly include the roles of project managers, technical managers, and systems engineers, in contrast to previous work that generally excludes those not directly engaged in technical design work. This analysis can be done through a multilevel, multimodal network modeling approach, deliberately including multiple organizational roles and both technical and social attributes of design activity.[150,151]

Survey data obtained from 44 small student design teams throughout characterizes communication and task breakdowns within teams: without prompting, individuals adopted different roles characterized by communication patterns and task breakdown preferences.[152] Communication behavior ranges from proactively initiating conversation with all peers to primarily receiving communication, and task breakdowns range from generalist contributing to all tasks to specialist on a single task. Teams were comprised of members with complementary communication preferences and task breakdowns: a mix of communication initiators and receivers, and a mix of generalists and specialists. These complementary behaviors are similar to the structured and proactive communication identified from interviews of professionals. Preliminary work has simulated the interactions of proactive communicators, such as systems engineers or communication intermediaries within an organizational network. Results suggest that the impact of these intermediaries in technical design work, though perhaps indirect (e.g., through a managerial role), may have a significant impact on technical performance as estimated by congruence.[153] The directionality of communication moderates the benefit of intermediary communicators.[154]

An agent-based model, briefly described in section 5.6.2, can simulate actors within an organization to test hypotheses regarding the positive impacts of systems engineers and systems thinkers within a design organization. In such a model, agents represent individuals within the organization with roles including designers, managers, and systems engineers. Communication can be restricted to follow network edges; the network's structure representing an organizational communication network or an organization's hierarchy. Multiple agent types can be created, each with different preferences for communication and task focus. Agents critically are decision makers, and thus can be connected to not only the organization, but also the decisions that are part of system design. Simulated organizations can be evaluated based on their ability to corroborate organizational behaviors derived from qualitative and quantitative analysis of interview and survey data like that described above. By introducing network structure into an agent-based model, these artificial organizations can be used to test the impact of network structure and the distribution of different roles or individuals with specific cognitive and communicative skills within the organization network on organizational performance as well as the decisions made by that organization, resulting in a designed system.

5.5 Policy and Law Influences

Postulate 5 states that Policy and Law have a substantial influence on systems engineering. A failure to properly understand the significance of policy and law can drive systems down unnecessary paths. Some policies are loose and the response to these can be adjusted with a good solution for the system. Other policies are rigid and cannot be changed, leading to strict constraints on the system solution. Failing to understand this can lead to unnecessary constraints or a complete system failure in violating a strict policy. Understanding the subtleties of politics is extremely important. Being too explicit with political statements (which can have unexpressed ambiguity behind the author's statement) can also drive systems toward inefficient or unsatisfactory solutions. The general approach is to define a system solution based on the mission, systems integrating physics efficiency, system value, and system cost, before interacting with political stakeholders. Then bring this back for a review by the political interests. If there are sensitive points, these will then be more clearly identifiable and the system design can be adjusted, and if not, the solution is free to proceed with needed support. These types of influences need to be understood by the systems engineer.

5.5.1 Policy and Law at the Program Execution Level Oversight

At the program execution level, policy and law are most commonly associated with contract oversight. Oversight is a necessary part of any government acquisition, enabling the government to evaluate the performance of contractors or program offices.[155] Oversight policies are driven by rules and procedures including the Federal Acquisition Regulations, agency-specific acquisition rules, and specific contract structures. These oversight policies are enacted through monitoring activities. These monitoring activities include audits, meetings, reports, and other required activities that provide the government with information necessary to evaluate the cost, schedule, and performance of a project.[156] Some of these activities are explicitly required by a particular regulation, policy, or as part of the contract, such as milestone review meetings or monthly reports. Other monitoring activities do not require an explicit deliverable, rather they occur as part of the oversight relationship in order to enable the government to understand the reasoning behind why certain decisions have been made, such as answering questions or explaining why work was done in a certain way. While the former has been referred to as oversight and the latter as insight,[157] both are examples of monitoring activities. In addition, while monitoring activities can be considered direct interactions between the government and its contractors, oversight can also be thought to include second order effects such as the suite of internal processes, procedures, and business systems that organizations have built to facilitate their relationship with the government.

While monitoring activities are necessary parts of a contract, they can result in additional program costs. These activities can be extremely detail oriented, require a significant amount of time to complete, involve many stakeholders, and can include activities that a contractor or program office would do as part of their normal processes.[158] For example, a report for a technical milestone in a procurement may require one or more presentations at different levels of the acquisition chain of command to follow all necessary steps of the acquisition process. But, each presentation may also be followed by multiple, sometimes extraneous, requests for additional information. Stories similar to the aforementioned example are widespread, perpetuating a belief that oversight monitoring activities are burdensome and result in additional costs and increases in program schedules.[159]

In recent years, the appropriate level of oversight has been a frequent subject of debate, both in the media and in Congress. On one hand, proponents of oversight argue that these activities are a necessary part of the process for a relatively inexpensive price. This view is represented historically by Stephen B. Johnson in the book, *The Secret of Apollo*, where speaking of the Atlas, Titan, and Corporal missile projects he notes, "Reliability improvement programs—that is, systems management processes—improved reliability into the 60%–80% range during the 1950s and early 1960s and into the 85%–95% range thereafter. ... Therefore, systems management could easily have added more than 50% to each missile's cost and still been cost-effective."[160] Previous assessments of the burden of oversight range from 2% of a product's total cost to a factor of 5 times the price of a commercially available product.[161,162] The wide range of estimates of this burden can be attributed to measurement challenges and the phenomenon being measured. Previous estimates of oversight's burden used subjective, retrospective estimates of time spent on monitoring activities.[163–165] These types of measurements, however, tend to overestimate or underestimate positive memories, and the extent of oversight's burden could be misrepresented.[166,167] Additionally, each study used its own implicit definition of oversight, each emphasizing a different part of the oversight process. As a result, the scope of oversight's burden is vastly different across the studies.[168]

An empirically valid estimate of the time spent on monitoring activities at a major aerospace contractor provides a basis for understanding the value of these activities.[169] A 6-month time allocation study using a minimally invasive, real-time sampling technique was performed to establish the empirical estimate. This showed that when the definition of oversight monitoring activities was limited to nonvalue-added, government-requested monitoring, the extent of the burden was on the order of 6% of total work performed. When the definition was broadened to include both external, government-requested monitoring and internal government-support tasks, the burden ranged from a factor of 1.2 to 1.6 times. Moreover, this also showed that there exists a mismatch between the actual and perceived sources of oversight's burden.

The difference between actual and perceived sources of oversight burden can be attributed to how oversight makes engineers feel. Objectively, monitoring activities require engineers to spend time performing certain tasks. Subjectively, they develop their own opinions about whether these activities are effective or not.[170,171] While a group of engineers could view monitoring activities as burdensome, the government officials requiring the monitoring activities might find the information extremely important and useful and require the engineers to do the tasks anyway.[172] Research by public administration scholars has shown that when individuals feel burdened by rules and regulations (like oversight), they experience feelings of administrative delay and work alienation.[173,174] In this vein, we contend that the feelings engineers have associated with burdensome oversight requests creates this disconnect between actual and perceived sources of oversight's burden.

5.6 Discipline Integration Modeling

The information flow through the organization can be modeled using several approaches. System dynamics models provide a way to show not only the structural flow of information but also the short cuts and gaps that exist in the information flow. ABM allows the cultural aspects to be modeled to determine how different social norms and biases may be affecting the organizational information flow. Discrete event simulations (DESs) provide a means to study statistical variations in flow of information and material through an organization and through organizational processes (e.g., manufacturing flows, maintenance flows).

5.6.1 System Dynamics

There are systems engineering tools that have the potential to facilitate the development of products in a multidisciplinary environment, requiring the integration of social systems. One such tool is system dynamics. This section provides a brief history of system dynamics and an outline of system dynamics concepts and methods.

5.6.1.1 Background. System dynamics is a relatively new discipline that saw its formation in the mid-20th century and began to spread with the publication of Industrial Dynamics by Jay Forrester.[127] System dynamics is used when analyzing a domain as a system to understand the feedback within the system to develop solutions to inherent problems versus symptoms. The methodology was originally developed at the Massachusetts Institute of Technology (MIT).[175] System dynamics is an iterative, interdisciplinary process, which views problems holistically. Essentially, using system dynamics involves identifying elements, subsystems, and the systems' context, boundaries and properties of the system under investigation. System dynamics is both systematic and systemic in that there are systematic processes, and it is rooted in systemic thinking to recognize and solve complex problems by seeing the whole instead of only the parts.

5.6.1.2 Feedback Theory. A fundamental concept in system dynamics is feedback theory. In the evaluation of the relationships between elements in a system, there are often feedback loops operating in a system.[176] A feedback loop is the interconnection of variables in a system that feeds back into itself. This is a closed-loop system. Open-loop systems do not have a feedback loops, and often the policy goal in these systems is to close the loops, especially in environmental management systems. Open-loop systems have exogenous variables that influence the system structure from outside the system to generate the system behavior. Closed-loop systems have endogenous variables, where forces within the system influence the behavior. An example of this is climate change variables. When modeling societal collapse in history (e.g., the classic Mayans), climate change (drought) influenced societal collapse. Climate change is exogenous in this example because the population was not causing the drought.

A causal loop diagram (CLD) shows the relationships between elements in a system (the feedback loop), which can be either positive or negative. A positive relationship means the elements develop in the same direction (when one increases, so does the other), and a negative relationship means the elements develop in opposite directions (when one increases, the other decreases). A balancing feedback loop means that the relationships between the elements keep the accumulated elements (stocks) at equilibrium. In addition to the balancing feedback loop, there is also a reinforcing feedback loop, where the behavior of the stock does not find an equilibrium and continues to increase or decrease over time.

5.6.1.3 System Dynamics Modeling. There are both qualitative and quantitative modeling techniques in system dynamics. The qualitative system dynamics modeling usually takes the form of CLDs as described in the previous section. CLDs are a simplified form of the system structure and are usually used in conjunction with quantitative modeling. The main modeling technique is stock and flow modeling represented as stock and flow diagrams (SFDs) (see figure 50). Stock and flow models are often termed 'system dynamics models,' which are ordinary differential equation models. System dynamics models consist of stocks, flows, and variables in an SFD. Stocks are an accumulation of flows over time, and flows represent addition and subtraction to the stock over time. Variables in stock and flow models are elements that affect the inflows and outflows. The variables are linked to other variables and flows through instantaneous causal links. The accumulated causal behavior in the stock is affected by the flows, which are in turn affected by the variables, shown in figure 51.

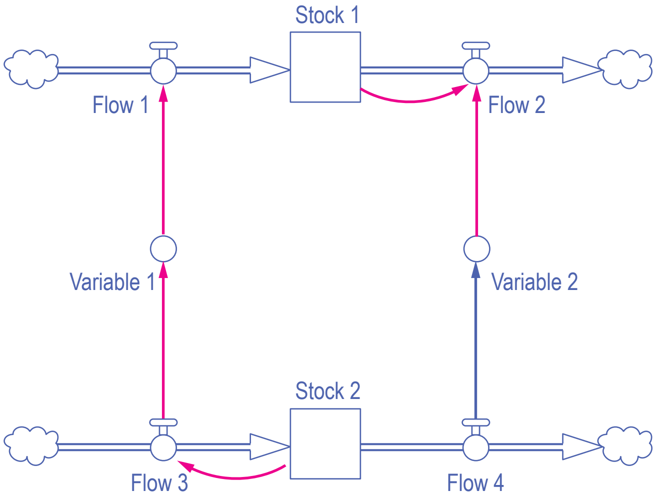


Figure 51. Example of a system dynamics model as an SFD.

System dynamics modeling is aided using software. Popular programs include Powersim Studio™, Vensim, and Stella Architect; the examples in figures 51 and 52 are from Stella Architect. The structure of a system yields the behavior over time (accumulated in stocks), and the goal is to discover all the elements and relationships in a system and reproduce the observable reference mode behavior (actual system behavior, shown in figure 52). In system dynamics models, there are endogenous and exogenous elements. Endogenous elements are incorporated in the model structure in relation to other structural elements. Exogenous elements are variables that contain data that are directly imported into the model structure. One of the major goals of system dynamics is to understand the structure of the system, shown in figure 51, that results in the observable behavior, shown in figure 52.

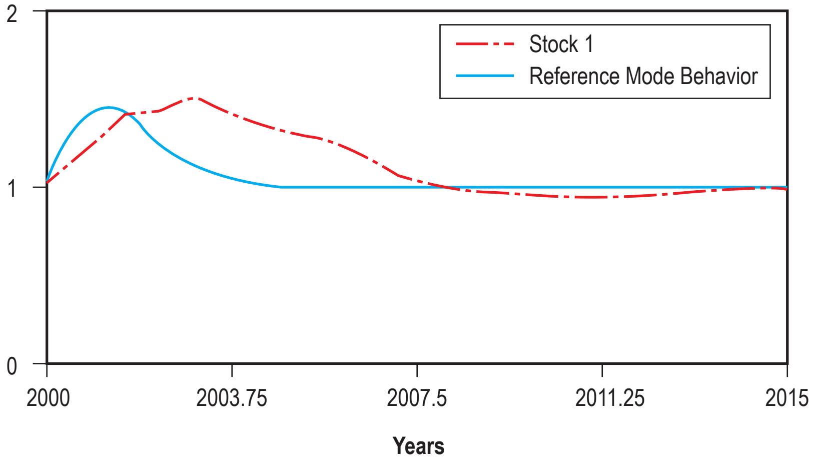


Figure 52. System behavior: Simulated (stock 1) versus actual (reference model behavior).

Originally, system dynamics modeling was applied in industrial engineering, but over time, its application broadened to a wide variety of research areas, such as supply chain management, economics, sustainable development, health, and more recently, social systems. System dynamics is especially useful for exploring the underlying structure of a complex, dynamic problem with the objective to eliminate undesirable dynamics and to strengthen the desirable dynamics. In addition to this, system dynamics modeling is interdisciplinary, with experts on specific model sectors giving input into how the system operates.[177] Because of this, system dynamics modeling can be very helpful in the analysis of problems arising in a multidisciplinary environment.

5.6.1.4 System Dynamics as a Tool During Operations. System dynamics has a long history of successful application in organizational and management science.[178] One reason for this success is system dynamics' ability to model the interaction between different functional systems. An important part of system dynamics modeling is 'walking the line.[179] Here, we define each discipline as a social system, and in modeling social system integration, system dynamics modeling methods require modelers to investigate or 'walk the line' in each social system to build the system mathematically. Each system dynamics evaluation begins with a problem[180] (a problematic behavior over time, shown in figure 53). The modeler then works backwards from the undesired behavior to uncover the structural elements influencing the problematic behavior (developing the dynamic hypothesis). The process of modeling backwards involves interviewing the experts in each functional system.

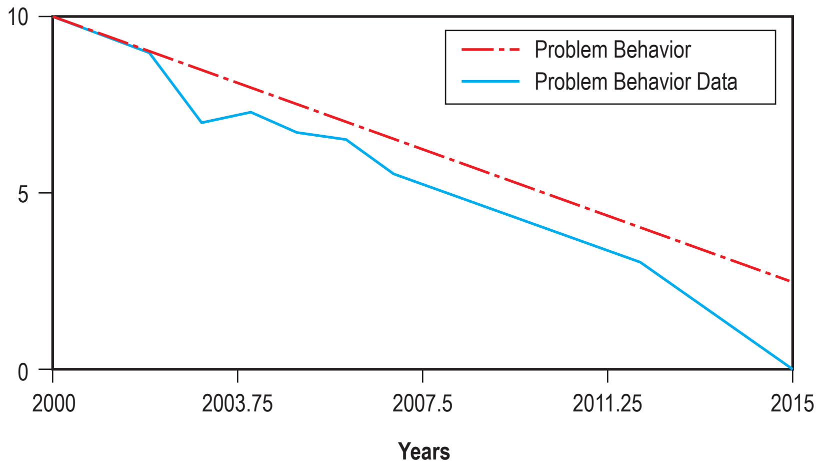


Figure 53. Problematic behavior—*X* is decreasing over time in discipline 1.

To illustrate this, figure 54 shows the simple modeling of three example disciplines (disciplines 1, 2, and 3). We start with the problem behavior shown in figure 53 and develop system structure as shown in figure 54. In discipline 1, there is a problem behavior uncovered during operations. The problem behavior is directly influenced by discipline 2 with the execution of their normal operations. Normal operations in this example encompass everything that the subsystem does, part of which influences other disciplines. The problem behavior itself influences what discipline 3 needs in its operations. The output of discipline 3 then influences the output of discipline 2, creating a feedback loop. This feedback loop is represented as a CLD shown in figure 55. Figure 55 shows a negative (decreasing) reinforcing feedback loop: the problem in discipline 1 decreases the input and, hence, output in discipline 3. This then decreases the output of discipline 2, which decreases the problem behavior in discipline 1. To validate the model, we compare the simulated system behavior with the actual system behavior, shown in figure 53.

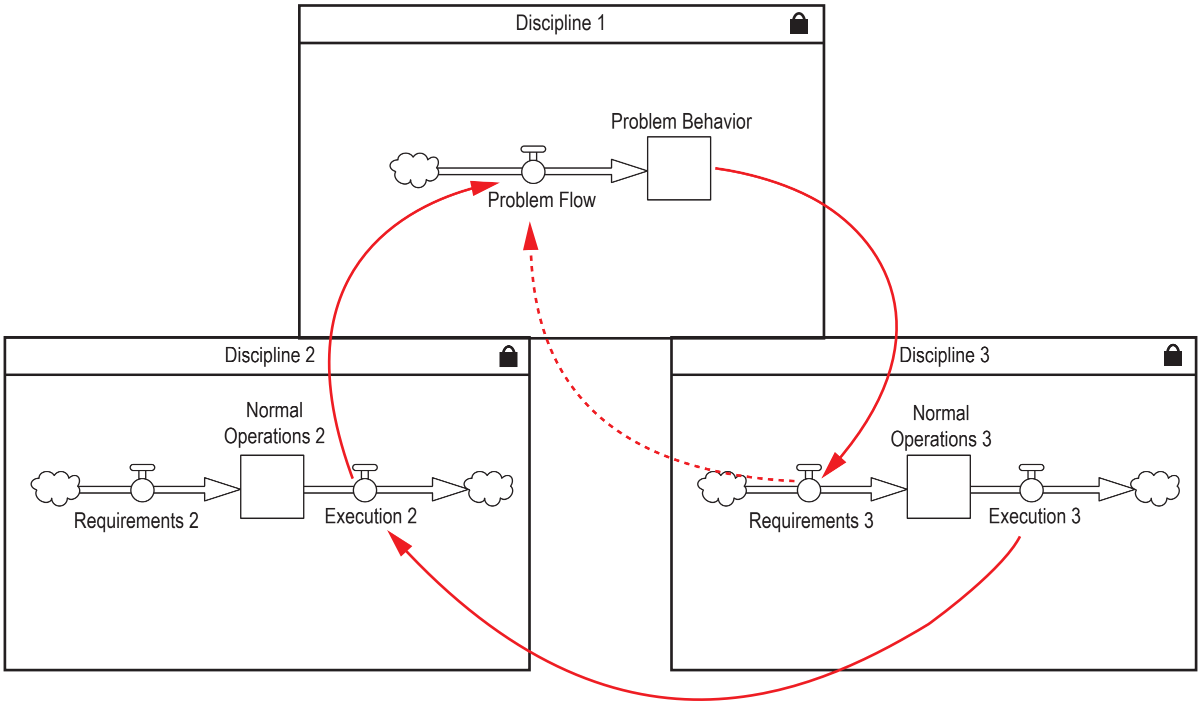


Figure 54. Three disciplines interacting to cause a problematic behavior.

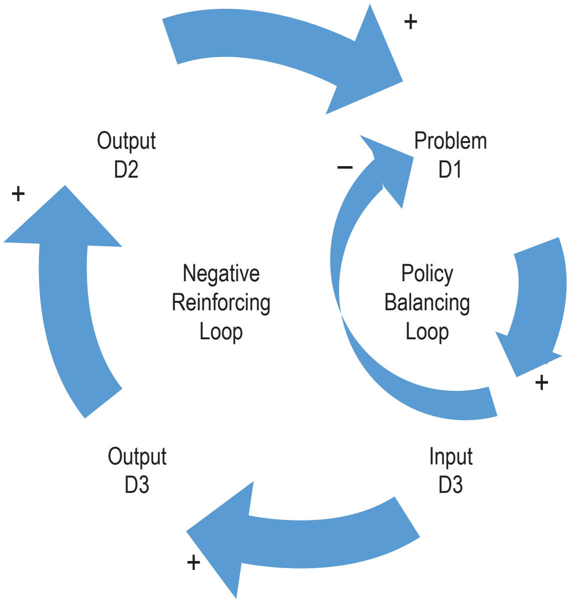


Figure 55. CLD-casual loop diagram showing a negative reinforcing loop.

Once a system dynamics model is built and validated, a policy structure can be developed to correct the problematic behavior. In figure 54, the policy solution is to create a feedback loop between discipline 1 and discipline 2. This is a balancing feedback loop to counteract the negative reinforcing loop in the system structure, shown in figure 55. This means that the policy structure causes the input to discipline 3 to increase the input to the problem behavior. The result of such a policy is shown in figure 56.

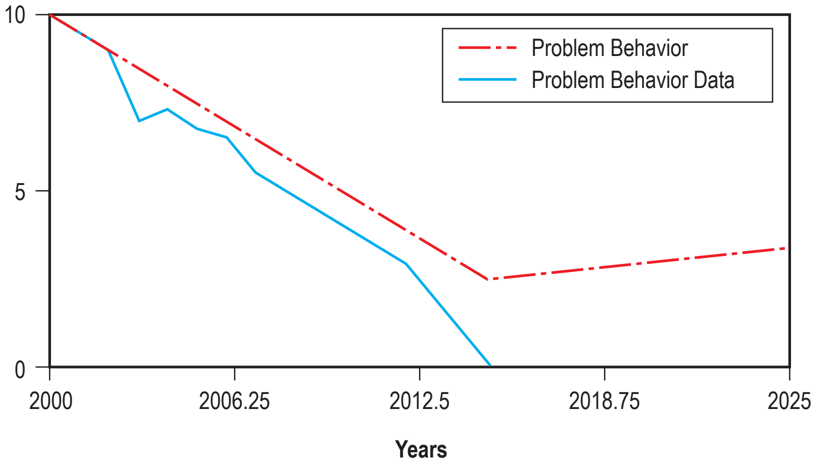


Figure 56. Problem behavior in discipline 1 with a policy introduced in 2015.

System dynamics modeling can give rough predictions of the effect of introducing policy. This is an 'all else being equal' scenario (meaning everything else in the model remains constant over time), and although sensitivity analysis can be conducted, the predictions should be taken as an estimate.

5.6.2 Agent-Based Modeling

ABMs represent humans with simple heuristics to produce behaviors. The behaviors are a consequence of the interactions of agents. The rules that the agents follow can range from simple logic (always turn right when colliding with someone) to more complex game theory (have the agent use their probabilistically defined beliefs and preferences to determine the Nash equilibrium of the collision situation). Uncertainty is typically a part of the ABM driving the use of Monte Carlo simulations in order to determine a probabilistic distribution of the outcome. The ABM is most prevalent in work that tries to capture the impact of humans and their interactions on a system without having to do a human study. This can be seen in examples such as evacuees interacting with planes or buildings during an egress.[181–183] ABMs enable an examination that would take years, decades, or centuries, to be performed in a matter of minutes. ABMs will become an important tool as systems become increasingly complex, enabling systems engineers to understand the interactions with stakeholders rapidly, further enabling rapid optimization frameworks. An important topic that must be addressed in any ABM application is the identification of the bounds of the work and the relationship between the model and the related real-world situation.

5.6.3 Discrete Event Simulation

DESs provide an operational flow-modeling capability to examine manufacturing, maintenance, and operations process flows. The DES provides statistical variations and can be used in the design phase to analyze the system manufacturing, transportation, operations, and maintenance flows to identify choke points, gaps, and duplications in flow paths. This model and data provide the systems engineer the ability to examine the plans for these activities and provide guidance on improvements to meet system readiness and system availability needs. These models also provide early identification of flow problems that can lead to cost and schedule over-runs. Supply chain management maps are important for the manufacturing flows and should be developed. These SCM maps can be used in the DES to determine component availability issues.

5.7 Discipline Integration Summary

Integrating the disciplines through the organizational structure and ensuring clear and complete communication between the disciplines brings in several aspects of sociology, organizational theory, information theory, and cognitive science. Each of these elements is a system itself with its own functional purpose. The applications for each of these are typically different between development organizations and operational organizations. This section discusses the integration of the disciplines in each of these lifecycle phases and how system modeling can be used to facilitate the integration.

5.7.1 Engineering an Elegant Organization During Development

Development organizations are intended to generate specific design information, coordinate this with other disciplines within the organization, and integrate this into an elegant system.

Understanding sociological systems theory principles can facilitate the process of engineering organizational systems. The organization is a polycontextural entity. It is composed of many different social systems, with each system having its own functional purpose (i.e., functional differentiation). These systems have functional logics that overlap, which require the systems to interact. The interaction between these systems makes the organization polyphonic. Each system has its own wants and needs, which shapes the way the systems interact. The polyphony influences the evolution of the subsystems and the greater organization itself. As shown in section 5.1, this interaction/communication between systems can lead to conflict and undesirable behavior. Systems engineers should be aware of the tools available, such as system dynamics modeling, to help manage this conflict.

Sociology provides many resources to address functions that exist within the organization as illustrated in figure 57. Opportunity structures provide an opportunity for the disciplines to mature their ideas and resolve questions and unexpected responses prior to carrying these through the decision board process. The systems engineer provides for these in the organizational structure and information flow process through the formation of informal status meetings, task teams, working groups, communities of practice, etc. as appropriate for the organizational culture and specific system development.

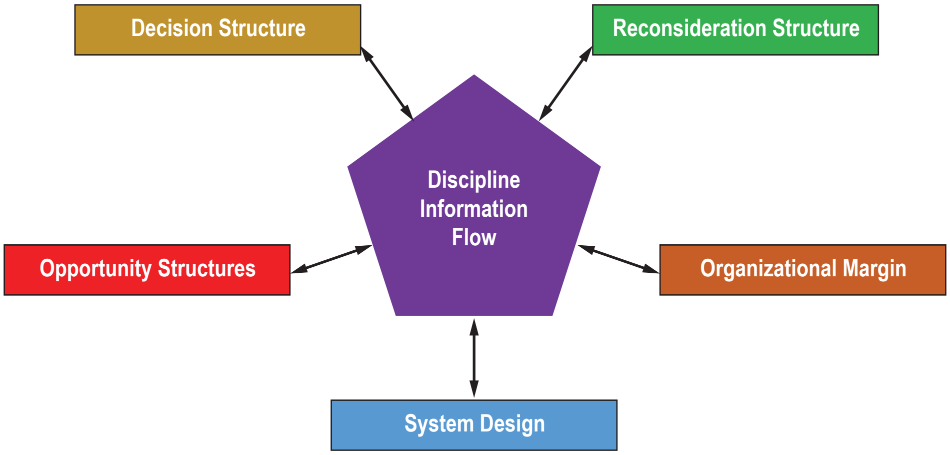


Figure 57. Discipline integration functions.

The decision-making process makes use of these opportunity structures. Different opportunity structures can be used for different decisions. The key is in having the correct knowledge involved in the informal and formal decision making so that a quality decision is made. Information theory shows the importance of the proper knowledge to decide on a specific subject or question. Information theory also provides important guidance in the establishment of decision boards, the membership of these boards, and the relationship of delegated boards. The key is to establish a system that allows information to flow through the decision-making process with minimal uncertainty of the topics, discussions, and results in the organization.

Reconsideration paths (reclama paths) are a key sociological mitigation for those within the organization who encounter a social ambivalence on a specific system decision or topic. These paths should not be through participants in the development organizational culture but should include those who are able to discern the sociological as well as the technical forces contributing to the perceived conflict in the system. This helps mitigate social responses that can lead to system design activities or decision moving outside the organizational structure or attempting to bypass certain decision-making steps.

The systems engineer should be aware that information may reside in the organization, unidentifiable in the design. A design that does not close (i.e., cannot be shown to achieve all intended results) may be due to margin in the design but only identifiable by a discipline. Before action is taken to significantly change the system, the systems engineer should engage in biased information sharing negotiation with the affected disciplines to determine if there are factors (e.g., margin) that the discipline organization can change to support an improved design. The basic approach is to ask questions about conservative estimates, uncertainty margins, and actual factors of safety which may indicate design margin not otherwise identified. As discussed in section 5.2, this may be an iterative negotiation process where disciplines slowly release margin as their confidence in the system design grows.

Cognitive science is important in the consideration of both information flow quantity and in teaching systems thinking to engineers. Information flow quantity is based on the amount of information an individual can deal with at one time. This will place a limit on the number of decisions that can be made about the system in each board meeting and over a defined period of time.

Mediated learning provides an approach to train engineers in how to consider designs at a systems level. The techniques indicated by this can help in developing trade study teams, system integration teams, and systems engineers in general.

Policy and law understanding is essential to the systems engineers. Decisions on the system design must be made with a proper understanding of the intent behind organizational policy, government policy, and laws. Misunderstanding these intents can lead to overconstrained or underconstrained systems detracting from system elegance.

5.7.2 Engineering an Elegant Organization During Operations

Operations organizations are intended to coordinate specific actions about a system based on a detailed understanding of the system's behaviors and responses defined during the system development. The discipline integration approaches are like those discussed with development organizations.

Operational organizations take on many forms. Customer service operations organizations need the ability to matrix into the engineering teams for defined questions or issues. High reliability organizations (such as those encountered in electrical plant operations, ship crews, or spacecraft operations teams can have a more hierarchical structure that flattens during critical operations events to address high risk factors in a timely and successful manner. Systems engineering must recognize how the engineering team integrates with these operations organizations, and how to ensure the correct information flows to the operations teams.

6. SYSTEMS ENGINEERING PROCESSES

The approach to conducting systems engineering is dependent on the specific system being developed or operated. In the development phase, these processes aid the systems engineer in a logical sequence of events to achieve the system. The execution of these processes do not accomplish the engineering of the system but rather provide a schema in coordinating design activities between the various system design disciplines. NASA/SP-2016-6105, Rev. 2, NASA Systems Engineering Handbook[184] provides a good description of the potential processes to be applied.

Systems engineering during the operations phases is also critically important. Processes for these phases (i.e., operations, decommissioning, and disposal) are much less defined within the practicing body of systems engineers. These processes depend very heavily on the specific system and its operations and utilization characteristics. Maintenance, logistics, and obsolescence management are some of the activities that are necessary in these operations phases. A basic summary of these can be found in the INCOSE Systems Engineering Handbook, 4th Edition.[141]

In looking at the systems engineering processes, it is important to recognize that all engineering disciplines have processes by which they accomplish their designs and analysis. These disciplines focus on the engineering equations and solutions that achieve the goals of the component being designed, using their processes to ensure a consistent approach. The disciplines build models based on their specific physics or logic. They use the models to determine performance, and then based on that, create a design, then update the models, update the analyses, then based on those, update the design, etc. in a spiral process of model and design maturation. Systems engineering should work in a similar manner to engineer the system. Systems engineering processes are important and necessary to achieve a consistent system design approach within an organization; but, they are not sufficient to ensure that an elegant system is the end result. A successful, elegant system is achieved by engineering the system to ensure the physical aspects and social aspects meet the system intents.

The specific systems engineering processes implemented for a system by an organization are communicated to all participating disciplines, ensuring a well-coordinated development or operation. The SEMP serves as an excellent communication mechanism to describe systems engineering processes in context of the system. Both handbooks mentioned above discuss development of a SEMP and the NASA handbook provides a template for a development phase SEMP.

6.1 Systems Engineering Management Plan Technical Content

A SEMP is used to establish the technical content of the engineering work early in the formulation phase for each project and is updated as needed throughout the project lifecycle. The SEMP provides the specifics of the technical effort and describes what technical processes will be used, how the processes will be applied using appropriate activities, how the project will be organized to accomplish the activities, information flow within the organization, its decision-making structure, and the resources required for accomplishing the activities. The process activities are driven by the critical events during any phase of a lifecycle (including operations) that set the objectives and work product outputs of the processes and how the processes are integrated. The SEMP provides the communication bridge between the project management team and the engineering discipline teams. It also facilitates effective communication within the discipline teams. The SEMP provides the framework to realize the appropriate work products that meet the entry and exit criteria of the applicable project lifecycle phases to provide management with necessary information for assessing technical progress.

6.2 Systems Engineering Management Plan Technical Approach

The role of the SEMP is to document and communicate the technical approach. This includes the application of the common technical processes, resources to be used, system analysis to be conducted, approach to understanding system sensitivities and uncertainties, system model development and model integration, system testing and V&V, and other key technical tasks, activities, and events along with their metrics and success criteria as required by the specific system. The SEMP communicates the technical effort that will be performed by the assigned discipline team to the team itself, managers, customers, and other stakeholders. The SEMP should cover the systems engineering plans for all relevant lifecycle phases.

7. SUMMARY

This Technical Publication was developed to capture the emerging picture of elegant, product-focused systems engineering as an engineering discipline. The volume contains a theoretical representation of the systems engineering discipline that leads into practical guidance for systems engineers. It captures the primary concepts of the discipline while constructing an integrated view of the approach.

This volume details the characteristics of elegant systems and provides a basic framework for the discipline of systems engineering. Guiding postulates and hypotheses are described along with supporting evidence and implications of each. These postulates and hypotheses are used to articulate basic principles that are used as a guide for systems engineering.

Systems engineering encompasses analysis tools and techniques that are specific to the function of the system. These analysis tools and techniques are broken down into elements of systems integrating physics, system value, system state variables, system relations, system statistics, organizational structure, and information flow which are explained in theoretical terms. Understanding of each of these elements is of equal importance to the discipline of systems engineering. The underlying theory provides a foundation for an understanding of the practice of systems engineering which is explained in the companion volume titled, "Engineering Elegant Systems: The Practice of Systems Engineering," NASA/TP–20205003646.[185]

Processes related to systems engineering must be established within the context of the system being developed or operated. Systems engineers need to be aware of process limitations and not to rely completely on process. The SEMP serves as the communication mechanism to describe systems engineering processes in the context of the system. Important elements of this plan are the technical content and the technical approach.

APPENDIX A—DERIVATION OF SPACE RELATIONSHIPS FROM EXERGY BALANCE EQUATION

This appendix contains the detailed derivation of important rocket and spacecraft relationships from the exergy balance equation. These derivations show that the exergy balance equation contains the important relationships in rocketry and spaceflight.

A.1 Derivation of the Rocket Equation From the Exergy Balance Equation

Objective: Determine the relationship between the exergy balance equation and the rocket equation.

This derivation starts with the exergy balance equation for a rocket. The derivation makes use of the limiting assumptions contained in the rocket equation:

(1) The rocket equation considers only mass of the vehicle (M_{vehicle}), mass of the propellant (m_p), velocity of the vehicle (V_{vehicle}), change in the velocity of the vehicle ($\Delta V_{\text{vehicle}}$), and velocity of the exhaust gas (V_e).

(2) In the rocket equation derivation, V_e is considered independent of Δt and changes in vehicle mass (i.e., $\Delta M_{\text{vehicle}}$ and $\Delta m_{\text{propellant}}$). V_e is measured as the distance from the combustion chamber to the nozzle exit. This velocity does not change over course of the trajectory flight time and therefore can be considered constant over the flight time interval. This ignores startup and shutdown transients (i.e., modeled as step functions), and assumes no throttling effects on V_e (throttling effects $\Delta m_{\text{propellant}}$ only).

(3) Engine changes can occur with staging (i.e., different engines and different V_es can exist on different stages) which are handled by breaking the flight trajectory into segments for each stage. V_e can vary between segments but remains constant within the segment.

(4) Note assumptions (2) and (3) make V_e a less reliable variable to differentiate the exergy balance equation with since, in some cases, it would differentiate with respect to a constant. Therefore, differentiation with respect to vehicle velocity is a better choice based on the rocket equation limiting assumptions on V_e.

(5) The propellant mass is fully exhausted at the nozzle exit (i.e., 100% efficiency in propellant burning).

(6) Losses are not considered, including drag forces, aero thermal heating, gravity, etc.

(7) The trajectory path is not considered.

The exergy balance equation for a rocket can be written as:

$$\Delta m_{\text{propellant}}\left(h_{\text{prop}}+\frac{V_e^2}{2}\right)-X_{\text{des}}=\Delta KE_{\text{vehicle}}+\Delta PE_{\text{vehicle}}\,. \tag{242}$$

Expanding the kinetic energy and potential energy terms yields a more explicit form in terms of velocity:

$$\Delta m_{\text{propellant}}\left(h_{\text{prop}}+\frac{V_e^2}{2}\right)-X_{\text{des}}=\left(M_{\text{vehicle, final}}\frac{V^2_{\text{vehicle, final}}}{2}-M_{\text{vehicle, initial}}\frac{V^2_{\text{vehicle, initial}}}{2}\right)$$

$$+\left(\frac{GxM_E\,M_{\text{vehicle, initial}}}{r_{\text{altitude, initial}}}-\frac{GxM_E\,M_{\text{vehicle, final}}}{r_{\text{altitude, final}}}\right)\,. \tag{243}$$

Note that potential exergy is a negative term. Factoring the negative side through reverses the final and initial terms in equation (243). Making use of the fact that Exergy is a work relationship and that the derivative of work yields a force relationship, differentiate with respect to the vehicle velocity, V_{vehicle}:

$$\frac{d}{dV_{\text{vehicle}}}\left[\Delta m_{\text{propellant}}\left(h_{\text{prop}}+\frac{V_e^2}{2}\right)-X_{\text{des}}\right]$$

$$=\frac{d}{dV_{\text{vehicle}}}\left[\left(M_{\text{vehicle, final}}\frac{V^2_{\text{vehicle, final}}}{2}-M_{\text{vehicle, initial}}\frac{V^2_{\text{vehicle, initial}}}{2}\right)\right.$$

$$\left.+\left(\frac{GxM_E\,M_{\text{vehicle, initial}}}{r_{\text{altitude, initial}}}-\frac{GxM_E\,M_{\text{vehicle, final}}}{r_{\text{altitude, final}}}\right)\right]\,. \tag{244}$$

This yields the following with the potential energy term being represented by:

$$\frac{dPE}{dV_{\text{vehicle}}}=\frac{d}{dV_{\text{vehicle}}}\left(\frac{GM_E\,M_{\text{vehicle, initial}}}{r_{\text{altitude, initial}}}-\frac{GM_E\,M_{\text{vehicle, final}}}{r_{\text{altitude, final}}}\right) \tag{245}$$

and

$$\left[\Delta m_{\text{propellant}}V_e\frac{dV_e}{dV_{\text{vehicle}}}-\frac{d}{dV_{\text{ve hicle}}}X_{\text{des}}\right]\text{ve hicle}=\left[\left(M_{\text{vehicle, final}}V_{\text{vehicle, final}}\right)+\left(\frac{dPE}{dV_{\text{vehicle}}}\right)\right], \tag{246}$$

where $\dfrac{dV_{\text{vehicle, initial}}}{dV_{\text{vehicle}}} = 0,$ since the initial velocity is a constant (fixed starting point).

Now, differentiate the rocket equation to find the differential relationship (constant *C*) for $\dfrac{dV_{\text{e}}}{dV_{\text{vehicle}}}.$ So,

$$\frac{dV_e}{dV_{\text{vehicle}}} = C = \frac{1 - \dfrac{M_{\text{vehicle, final}}}{M_{\text{vehicle, initial}}}}{\ln\left(\dfrac{M_{\text{vehicle, initial}}}{M_{\text{vehicle, final}}}\right)} \; . \tag{247}$$

Therefore, equation (247) becomes

$$\left[\Delta m_{\text{propellant}} V_e C - \frac{d}{dV_{\text{vehicle}}} X_{\text{de}s}\right] = M_{\text{vehicle, final}} V_{\text{vehicle, final}} + \left(\frac{dPE}{dV_{\text{vehicle}}}\right) \; . \tag{248}$$

Now, the rocket equation does not contain loss terms (assumption (4) above) and therefore assumes that $\dfrac{d}{dV_{\text{vehicle}}} X_{\text{des}} = 0$ and $\dfrac{dPE}{dV_{\text{vehicle}}} = 0$.

Applying these assumptions yields

$$\Delta m_{\text{propellant}} V_e C = M_{\text{vehicle, final}} V_{\text{vehicle, final}} \; . \tag{249}$$

Now, taking the limit as $\Delta t \geq 0$ yields,

$$dm_{\text{propellant}} V_e = M_{\text{vehicle,final}} \, dV_{\text{vehicle}} \; , \tag{250}$$

where

$$\lim_{\Delta t \to 0} V_{\text{vehicle, final}} = dV_{\text{vehicle, final}} = dV_{\text{vehicle, initial}} + dV_{\text{vehicle}} = dV_{\text{vehicle}} \; . \tag{251}$$

Since $dV_{\text{vehicle, initial}} = 0$ as $V_{\text{vehicle, initial}}$ is a constant, and

$$\lim_{\Delta t \to 0} C = \lim_{\Delta t \to 0} \frac{1 - \dfrac{M_{\text{vehicle, final}}}{M_{\text{vehicle, initial}}}}{\ln\left(\dfrac{M_{\text{vehicle, initial}}}{M_{\text{vehicle, final}}}\right)} = 1 \; . \tag{252}$$

The result of equation (252) is obtained by taking L'Hopitals Rule with the derivative of the top and bottom with respect to time (*t*).

Now, using

$$M_{\text{vehicle, final}} = M_{\text{vehicle, initial}} - \Delta m_{\text{propellant}} \; . \tag{253}$$

Then,

$$dM_{\text{vehicle}} = dM_{\text{vehicle, iniital}} - dm_{\text{propellant}} = -dm_{\text{propellant}} \; . \tag{254}$$

Since, $dM_{\text{vehicle, initial}} = 0$ as $M_{\text{vehicle, initial}}$ is constant.

Therefore,

$$-dM_{\text{vehicle}} V_e = M_{\text{vehicle, final}} dV_{\text{vehicle}} \; . \tag{255}$$

Grouping terms and integrating

$$-V_e \int \frac{1}{M_{\text{vehicle}}} dM_{\text{vehicle}} = \int dV_{\text{vehicle}} \; , \tag{256}$$

which results in the rocket equation,

$$V_e \ln\left(\frac{M_{\text{vehicle, initial}}}{M_{\text{vehicle, final}}}\right) = \Delta V_{\text{vehicle}} \; . \tag{257}$$

This result indicates that the exergy balance relationship for a rocket represents the integration of the rocket equation over the change in vehicle velocity. Thus, the exergy balance relationship defines the integration constants of the rocket equation.

A.2 Derivation of the Orbital Mechanics Energy Relationship From the Exergy Balance Equation

Objective: Determine the orbital mechanics energy relationship defined by the exergy balance equation.

If the spacecraft is thrusting, then the exergy balance equation directly contains the relationship between spacecraft energy and thrust. This is seen directly in equation (258):

$$\Delta m_{\text{propellant}}\left(h_{\text{prop}}+\frac{V_e^2}{2}\right)-X_{\text{des}}=\left(M_{\text{vehicle, final}}\frac{V^2_{\text{vehicle, final}}}{2}\right.$$

$$\left.-M_{\text{vehicle, initial}}\frac{V^2{}_{\text{vehicle, initial}}}{2}\;+\;\frac{GxM_E M_{\text{vehicle, initial}}}{r_{\text{altitude, initial}}}-\frac{GxM_E M_{\text{vehicle, final}}}{r_{\text{altitude, final}}}\right)\quad. \qquad (258)$$

Now, for a vehicle coasting in orbit around a body (i.e., planet, Moon, or Sun), the propulsion components are zero and equation (258) reduces to:

$$0-X_{\text{des}}=\left(M_{\text{vehicle, final}}\frac{V^2_{\text{vehicle, final}}}{2}-M_{\text{vehicle, initial}}\frac{V^2_{\text{vehicle, initial}}}{2}\right)$$

$$+\left(\frac{GxM_E M_{\text{vehicle, initial}}}{r_{\text{altitude, initial}}}-\frac{GxM_E M_{\text{vehicle, final}}}{r_{\text{altitude, final}}}\right)\;. \qquad (259)$$

Combining terms on the right-hand side of the equation yields:

$$X_{\text{des}}=\left(M_{\text{vehicle, final}}\frac{V^2_{\text{vehicle, final}}}{2}-\frac{GxM_E M_{\text{vehicle, final}}}{r_{\text{altitude, final}}}\right)$$

$$-\left(-\frac{GxM_E M_{\text{vehicle, initial}}}{r_{\text{altitude, initial}}}+M_{\text{vehicle, initial}}\frac{V^2_{\text{vehicle, initial}}}{2}\right)\;. \qquad (260)$$

Now, the orbital energy for a spacecraft is:

$$E_{\text{vehicle}}=\left(M_{\text{vehicle}}\frac{V^2_{\text{vehicle}}}{2}-\frac{GxM_E M_{\text{vehicle}}}{r_{\text{altitude}}}\right)\;. \qquad (261)$$

Using this relationship in equation (260) yields

$$X_{\text{des}}=\left(E_{\text{vehicle, final}}\right)-\left(E_{\text{vehicle, initial}}\right)=\Delta E_{\text{vehicle}}\;. \qquad (262)$$

Now, X_{des} is zero for a vehicle that is not thrusting and has no other active sources. Treating the spacecraft as a static mass,

$$\Delta E_{\text{vehicle}} = 0 \ , \tag{263}$$

which is as expected for a spacecraft in orbit where kinetic and potential energy changes are balanced as the spacecraft orbits the body. Note that for a vehicle with an active system to stabilization or station keeping (e.g., control moment gyroscope, thrusters, spin stabilization), these systems would be added to the left-hand side of equation (260) and the vehicle orbital energy change would be related to the change induced by the stabilization or station keeping system.

A.3 Derivation of Multibody Effects in Planetary Exergy Balance

The effects of multiple planetary bodies (i.e., planets, moons, asteroids, comets, Sun) are important to understand the total exergy imparted to a spacecraft in interplanetary transfer. These effects define the energy provided by a planet to a spacecraft in critical fly-by or slingshot maneuvers. Multibody effects can be considered from nearby planets in the solar reference frame. A body's effect on the vehicle in the solar reference frame provides additional forces on the vehicle due to gravitational effects. Essentially, the body is expending exergy to change the vehicle's exergy.

A.3.1 Vehicle Effect on the Body

Starting from an unaffected body, the body's initial potential energy is simply the potential energy the body has around the Sun,

$$PE_{\text{body, initial}} = -\frac{GM_{\text{Sun}}M_B}{r_{SB}} \ , \tag{264}$$

where r_{SB} is the position of the body from the Sun.

As the vehicle approaches, the effect on the body potential energy is given as

$$PE_{\text{body, final}} = -\frac{GM_{\text{Sun}}M_B}{r_{SB}} - -\frac{GM_B\,M_{\text{vehicle, initial}}}{r_{\text{vehicle body}}} \ . \tag{265}$$

The change in potential energy induced on the body by the vehicle is then,

$$\Delta PE_{\text{body}} = PE_{\text{body, final}} - PE_{\text{body, initial}} = -\frac{GM_{\text{Sun}}M_B}{r_{SB}} - -\frac{GM_B\,M_{\text{vehicle, initial}}}{r_{\text{vehicle body}}} - -\frac{GM_{\text{Sun}}M_B}{r_{SB}} \ , \tag{266}$$

which reduces to,

$$\Delta PE_{\text{body}} = \frac{GM_B\, M_{\text{vehicle, initial}}}{r_{\text{vehicle body}}} \quad . \tag{267}$$

A.3.2 Body Effect on the Vehicle

Similarly, the change in vehicle potential energy due to the body is

$$PE_{\text{vehicle, body, initial}} = -\frac{GM_{\text{Sun}}\, M_{\text{vehicle, initial}}}{r_{\text{vehicle, initial}}} \quad . \tag{268}$$

As the vehicle approaches, the effect on the vehicle by the body is given as,

$$PE_{\text{vehicle, body, final}} = -\frac{GM_{\text{Sun}}\, M_{\text{vehicle, initial}}}{r_{\text{vehicle, initial}}} - -\frac{GM_B\, M_{\text{vehicle, initial}}}{r_{\text{vehicle, body}}} \quad . \tag{269}$$

The change in potential energy induced on the vehicle by the body is then,

$$\Delta PE_{\text{vehicle, body}} = PE_{\text{vehicle, body, final}} - PE_{\text{vehicle, body, initial}} = -\frac{GM_{\text{Sun}}\, M_{\text{vehicle, initial}}}{r_{\text{vehicle, initial}}}$$

$$-\frac{GM_B\, M_{\text{vehicle, initial}}}{r_{\text{vehicle, body}}} - -\frac{GM_{\text{Sun}}\, M_{\text{vehicle, initial}}}{r_{\text{vehicle, initial}}} \quad , \tag{270}$$

which reduces to

$$\Delta PE_{\text{vehicle, body}} = \frac{GM_B\, M_{\text{vehicle, initial}}}{r_{\text{vehicle, body}}} \quad . \tag{271}$$

A.3.3 Force Balance

Now, the force on the planet is opposite the force on the vehicle. Thus, the potential energy change is also negative such that the potential energy gained by the vehicle is lost by the body and vice versa. This is seen by

$$F_{\text{vehicle, body}} = -F_{\text{body, vehicle}} \quad , \tag{272}$$

which is,

$$F_{\text{vehicle, body}} = -\frac{GM_{\text{vehicle}}\, M_B}{r_{BV}^2}, \tag{273}$$

and

$$F_{\text{body,vehicle}} = + \frac{GM_{\text{vehicle}}\ M_B}{r_{BV}^2}. \tag{274}$$

A.3.4 Potential Energy Balance

Integrating the force terms in equations (273) and (274) over *dr* yields,

$$PE_{\text{vehicle, body}} = - \frac{GM_{\text{vehicle}}\ M_B}{r_{BV}} \tag{275}$$

and

$$PE_{\text{body, vehicle}} = + \frac{GM_{\text{vehicle}}\ M_B}{r_{BV}}\ , \tag{276}$$

so,

$$\Delta PE_{\text{body, vehicle}} = - \Delta PE_{\text{vehicle, body}}\ . \tag{277}$$

A.3.5 Exergy Balance

The exergy balance can now be defined using the potential energy relationships. Adding the terms from equation (267) and (271) into equation (24) and accounting for the sign change indicated by equation (272) yields

$$\Delta m_{\text{propellant}} \left(\left(h_{\text{prop}} - h_0 \right) - T_0 \left(s_{\text{prop}} - s_0 \right) + \frac{v_e^2}{2} + \frac{GM_{\text{Sun}}}{r_{\text{fluid, initial}}} \right) + \frac{GM_B\ M_{\text{vehicle, initial}}}{r_{\text{vehicle, body}}} - X_{\text{des}}$$

$$= \left(M_{\text{vehicle, final}} \frac{v_{\text{vehicle, final}}^2}{2} - M_{\text{vehicle, initial}} \frac{v_{\text{vehicle, initial}}^2}{2} \right) + \left(\frac{GM_{\text{Sun}} M_{\text{vehicle, initial}}}{r_{\text{vehicle, initial}}} \right.$$

$$\left. - \frac{GM_{\text{Sun}} M_{\text{vehicle, final}}}{r_{\text{vehicle, final}}} \right) - \frac{GM_{\text{B}} M_{\text{vehicle, initial}}}{r_{\text{vehicle, body}}}\ . \tag{278}$$

Combining like terms,

$$\Delta m_{\text{propellant}}\left(\left(h_{\text{prop}}-h_0\right)-T_0\left(s_{\text{prop}}-s_0\right)+\frac{v_e^2}{2}+\frac{GM_{\text{Sun}}}{r_{\text{fluid, initial}}}\right)+\frac{2GM_B\, M_{\text{vehicle, initial}}}{r_{\text{vehicle, body}}}-X_{\text{des}}$$

$$=\left(M_{\text{vehicle, final}}\ \frac{v_{\text{vehicle, final}}^2}{2}-M_{\text{vehicle, initial}}\ \frac{v_{\text{vehicle, initial}}^2}{2}\right)$$

$$+\left(\frac{GM_{\text{Sun}}M_{\text{vehicle, initial}}}{r_{\text{vehicle, initial}}}-\frac{GM_{\text{Sun}}M_{\text{vehicle, final}}}{r_{\text{vehicle, final}}}\right). \tag{279}$$

Thus, the effect of each body can be accounted for by adding the resultant term, $\frac{2GM_B\, M_{\text{vehicle, initial}}}{r_{\text{vehicle, body}}}$.

Let us consider a five-body system consisting of the vehicle, Sun, Earth, moon 1, and Mars. Equation (279) then becomes

$$\Delta m_{\text{propellant}}\left(\left(h_{\text{prop}}-h_0\right)-T_0\left(s_{\text{prop}}-s_0\right)+\frac{v_e^2}{2}+\frac{GM_{\text{Sun}}}{r_{\text{fluid, initial}}}\right)+\frac{2GM_{\text{Earth}}M_{\text{vehicle, initial}}}{r_{\text{vehicle, Earth}}}$$

$$+\frac{2GM_{\text{Moon}}M_{\text{vehicle, initial}}}{r_{\text{vehicle, Moon}}}+\frac{2GM_{\text{Mars}}\, M_{\text{vehicle, initial}}}{r_{\text{vehicle, Mars}}}-X_{\text{des}}$$

$$=\left(M_{\text{vehicle, final}}\ \frac{v_{\text{vehicle, final}}^2}{2}-M_{\text{vehicle, initial}}\ \frac{v_{\text{vehicle, initial}}^2}{2}\right)$$

$$+\left(\frac{GM_{\text{Sun}}M_{\text{vehicle, initial}}}{r_{\text{vehicle, initial}}}-\frac{GM_{\text{Sun}}M_{\text{vehicle, final}}}{r_{\text{vehicle, final}}}\right). \tag{280}$$

Thus, the effect of the planets on the vehicle exergy corresponds to twice the potential energy change induced between each planet and the body.

APPENDIX B—PROPERTIES OF COMPLEX SYSTEMS

One key issue that systems engineers must deal with is system complexity. While there are many definitions of complexity, the NASA Systems Engineering Consortium has considered the following definition for the design of large-scale systems. System complexity is defined as a measure of a system's intricacy and comprehensibleness in interactions within itself and with its environment. This definition points to two factors in complexity: (1) Physical/logical intricacy and (2) human cognitive comprehension. Properties of complex systems are listed in table 17.[186]

Table 17. Complex system properties.

Aggregation
Complex systems are aggregations of less complex systems
Emergence
Complex systems have a propensity to exhibit unexpected performance of intended function
Complex systems exhibit properties not present in the individual subsystems but present in the integration of subsystems (emergent property)
Interaction
Complex system interactions form networks within the system and with the system environments Complex system interactions can be understood through control theory
Complex systems can be analyzed using two concepts: Laws (rules of interaction) states (current state and prior history)
Nonlinearity
Complex systems exhibit nonlinear responses to system stimuli Complex systems are difficult to predict
Optimality
Complex systems have local optimums (organizational efficiency determines ability to achieve local optimum)

These properties illustrate several important characteristics of complex systems and the importance of engineering the system interactions. Aggregation is perhaps the most important property in terms of system design and analysis. This property indicates that complex systems can be split into smaller systems based on engineering discipline, system function, or both. Thus, the systems engineer can allocate the system design and system analysis by subsystem or function and then recombine the results for a complete system representation. Consideration of the recombination is essential to the systems engineer. The presence of the emergent properties indicates the system responses and interactions are not the sum of the parts. They include additional responses and functions not observed by considering individual subsystems, functions, or disciplines. Recombination and analysis must be conducted on the integrated system to evaluate all the complex system responses and functions. The recombination of functions typically results in nonlinear responses. Indeed, many system responses are nonlinear functions of the subsystem responses.

As stated in systems engineering postulate 5, all systems have constraints. Global optimums are typically not a practical engineering result. Complex systems generally have local optimums. These optimums are complex functions of all the system responses and can be difficult to define. This property will be the basis of much research in systems engineering in the future.

The INCOSE Systems Engineering Complexity Primer provides an expansion on these concepts.[187] The primer provides a good basic discussion on system complexity and the characteristics of complex systems. These characteristics have been further elevated using apprecative inquiry methods and applied to the assessment of complex systems.[188] This paper provides improved insight and understanding of complex systems.

REFERENCES

1. Watson, M.D.; and Kelley, G.W.: "Application of System Operational Effectiveness Methodology to Space Launch Vehicle Development and Operations," Paper presented at AIAA Space Operations Conference, Stockholm, Sweden, June 11–15, 2012.

2. *Webster's New World Dictionary, Third College Edition*, V. Neufeldt and G.B. Guralnik (eds.): Simon and Schuster, NY, p. 1359, 1988.

3. von Bertalanffy, L.: *General System Theory: Foundations, Development, Applications*, Braziller, NY, 329 pp., 1968.

4. Ashby, W.R.: "General Systems Theory as a New Discipline," in *General systems*: *yearbook of the Society for the Advancement of General Systems Theor*y, Vol. 3, L. von Bertalanffy (ed.), Society for the Advancement of General Systems Theory, pp. 1–6, 1958.

5. Forrester, J.W.: *Principles of System*s, Productivity Press, Portland, OR, 392 pp., 1968.

6. Rosen, R.: *Dynamical System Theory in Biology, Volume 1: Stability Theory and its Applications*, Wiley-Interscience, New York, NY, 316 pp., 1970.

7. Merton, R.K.: "On Sociological Theories of the Middle Ranges" in *On Social Structure and Science*, P. Sztompka (ed.), Univ. of Chicago Press, Chicago, IL, pp. 41–50, 1996.

8. Frosch, R.A.: "A Classic Look at Systems Engineering," in *Readings in Systems Engineering*, F.T. Hoban and W.M. Lawbaugh (eds.), NASA SP–6102, Washington, DC, 1993.

9. *Webster's New World Dictionary, Third College Edition*, V. Neufeldt and D.B. Guralnik (eds.), Simon and Schuster, NY, p. 438, 1988.

10. Griffin, M.D.: "How Do We Fix System Engineering?" Paper Presented at 61st International Astronautical Congress, Prague, Czech Republic, September 21–October 1, 2010.

11. Lee, G.: "Systems Engineering Advancement Seminar Series," <https://surp.jpl.nasa.gov/videos/index.cfm?fuseaction=ShowVideo&VNum=10&MainVNum=3>, 2005.

12. *Webster's New World Dictionary, Third College Edition*, V. Neufeldt, and D.B. Guralnik (eds.), Simon and Schuster, New York, NY, pp. 1055, 1988.

13. Componation, P.J.; Dorneich, M.C.; Hu, G.; et al.: "Systems Engineering and Project Success in Government and Commercial Organizations," Paper Presented at the American Society for Engineering Management International Annual Conference, Bloomington, MN, October 2–5, 2013.

14. Pool, R.: *Beyond Engineering*: *How Society Shapes Technology*, Oxford University Press, New York, NY, 368 pp., 1997.

15. Wang, Y.: "The Importance of Politics in Engineering: Lessons from a Hands-on Experience," *Leadership and Management in Engineering*, Vol. 2, No. 3, pp. 27–32, 2002.

16. Finley, K.: "Is This a Real Job? An Engineer Who Hacks Policy, Not Code," in *Wired Magazine*, June 5, 2015, <https://www.wired.com/2015/06/policy-engineering/>, June 18, 2019.

17. Denny, F.I.; and Robinson, R.L.: "The Engineers Role in Public Policy," in *Proc. of the 2003 American Society of Engineering Education Annual Conference and Exposition*, American Society of Engineering Education, June 22–25, 2003, Nashville, TN, pp. 8.1130.1–8.1130.10, 2003.

18. Vincenti, W.G.: What Engineers Know and How They Know It, The John Hopkins University Press, Baltimore, MD, 1993.

19. Brennan, J.F.: The Elegant Solution, First edition, Van Nostrand, Princeton, NJ, 183 pp., 1967.

20. Boeing, "DC-3 Commercial Transport," <https://www.boeing.com/history/products/dc-3.page>, 2019.

21. Boeing, "B-52 Stratofortress," <https://www.boeing.com/history/products/b-52-stratofortress.page>, 2019.

22. Burnham, K.P.; and Anderson, D.R.: *Model Selection and Multimodal Inference: A Practical Information-Theoretic Approach*, 2nd ed., Springer-Verlad, New York, NY, pp. 60–64, 2002.

23. von Neumann, J.; and Morgenstern, O.: *Theory of Games and Economic Behavior*, Princeton University Press, NJ, 1944.

24. Vaughan, D.: The Challenger Launch Decision: Risky Technology, Culture, and Deviance at NASA, 1st Edition, University of Chicago Press, Chicago, IL, 1996.

25. Ginsberg, J.H.: *Advanced Engineering Dynamics*, Cambridge University Press, Cambridge, UK, 1998.

26. Watson, M.D.: "System Exergy: System Integrating Physics of Launch Vehicles and Spacecraft," *AIAA Journal of Spacecraft and Rockets*, Vol. 55, No. 2, pp. 451–461, March 2018.

27. Duignan, B.: "Occam's Razor," <https://www.britannica.com/topic/Occams-razor>, 1998.

28. Riggins, D.W.; Moorhouse, D.J.; and Camberos, J.A.: "Characterization of Aerospace Vehicle Performance and Mission Analysis Using Exergy," in Exergy Analysis and Design Optimization for Aerospace Vehicles and Systems, J.A. Camberos and D.J. Moorhouse (eds.), AIAA, Reston, VA, pp. 275–285, 2011, <https://doi.org/10.2514/5.9781600868405.0275.0307>, Published online, August 24, 2012.

29. Sack, W.; Kishimoto, K.; Ogawara, A.; et. al: “The MB-60 Cryogenic Upper Stage Engine - A World Class Propulsion System,” <https://www.alternatewars.com/BBOW/Space_Engines/MB-60.pdf>, 2009.

30. Powell, R.W.; Striepe, S.A.; Desai, P.N.; et al.: *Program to Optimize Simulated Trajectories* (*POST*)*, Volume II*: *Utilization Manual*, Version 5.2, October 1997.

31. McCarter, J.W.: “Simulating Flights of Future Launch Vehicles and Spacecraft,” in NASA Tech Briefs, pp. 15, February 2007.

32. Owen, S.T.; Watson, M.D.; and Rodriguez, M.A.: “Exergy Efficiency of Interplanetary Transfer Vehicles,” 2018 Conference on Systems Engineering Research, Charlottesville, VA, May 8–9, 2018, <https://ntrs.nasa.gov/search.jsp?R=20180005161 2020-02-04T16:47:05+00:00Z>, 2018.

33. Curtis, H.D.: *Orbital Mechanics for Engineering Student*s, 3rd ed., Elsevier, Boston, MA, 768 pp., 2014.

34. Wieland, P.O.: “Living Together in Space: The Design and Operation of the Life Support Systems on the International Space Station,” NASA/TM—1998–206956-Vol. 1, NASA Marshall Space Flight Center, Huntsville, AL, 304 pp., January 1998.

35. Ley, W.; Merkle, F.; and Block, J.: “Subsystems of Spacecraft,” in *Handbook of Space Technology*, W. Ley, K. Wittmann, and W. Hallmann (eds.), John Wiley and Sons, West Sussex, UK, pp. 253–254, February 2011.

36. Watson, M.D.; and Pryor, J.E.: “Systems Engineering of Photonic Systems for Space Applications,” in *Proc. of SPIE: Nanophotonics and Macrophotonics and Space Environments VIII*, Vol. 9226, 8 pp., October 2014.

37. Kreyszig, E.: Advanced Engineering Mathematics, 8th Edition, John Wiley & Sons, NY, pp. 83–89, 1999.

38. Shannon, C.E.; and Weaver, W.: *The Mathematical Theory of Communication*, University of Illinois Press, Urbana, IL, 141 pp., 1948, 1998.

39. Fisher, R.A.: “On the Mathematical Foundations of Theoretical Physics,” *Philosophical Translations* (*Series A*), Royal Society of London, pp. 309–368, 1922.

40. Akaike, H.: “Information Theory as an Extension of the Maximum Likelihood Principle,” in *Proc. of the Second International Symposium on Information Theory*, B.N. Petrov and F. Csaki (eds.), Budapest, Hungary, pp. 267–281, 1973.

41. Burnham, K.P.; and Anderson, D.R.: *Model Selection and Multimodal Inference: A Practical Information-Theoretic Approach*, 2nd ed., Springer, New York, NY, pp. 150–153, 2002.

42. Burnham, K.P.; and Anderson, D.R.: *Model Selection and Multimodal Inference: A Practical Information-Theoretic Approach*, 2nd ed., Springer, New York, NY, pp. 65–66, 2002.

43. Johnson, S.B.; and Breckinridge, J.T.: "Goal-Function Tree Modeling for Systems Engineering and Fault Management," Paper Presented at AIAA InfoTech@Aerospace, Boston, MA, August 19–22, 2013.

44. Berg, P.; Johnson, S.B.; and Trevino, L.: "State Analysis Modeling for Space Launch System Mission and Fault Management Analysis," in *Proc*., AIAA Infotech@Aerospace Conference, San Diego, CA, January 4–8, 2016.

45. Watson, M.D.; and Johnson, S.B.: "A Theory of Vehicle Management Systems," Paper presented at IEEE Aerospace, Big Sky, MT, March 3–10, 2007; in *Proc. IEEE Press*, 8 pp., 2007.

46. Watson, M.D.; Johnson, S.B.; and Trevino, L.: "System Engineering of Autonomous Space Vehicles," Paper Presented at 2014 IEEE International Conference on Prognostics Health Management, Spokane, WA, 28 pp., June 2014.

47. Martins, J.R.R.A.; and Lambe, A.B.: "Multidisciplinary Design Optimization: A Survey of Architectures," *AIAA Journal*, Vol. 51, No. 9, pp. 2049–2075, August 2013.

48. English, K.W.: "Coupling suspension complex system optimization," Ph.D. Dissertation, State University of New York at Buffalo, Buffalo, New York, 2001.

49. English, K.W.: "Development of multiple cycle coupling suspension in multidisciplinary design optimization," M.S. Thesis, State University of New York at Buffalo, Buffalo, NY, 1998.

50. Bloebaum, C.L.: "Coupling strength-based system reduction for complex engineering design," *Structural Optimizatio*n, Vol. 10, pp. 113–121, 1995.

51. Miller E.: "Coupling Suspension and Elimination in Multidisciplinary Design Optimization," M.S. Thesis, State University of New York at Buffalo, Buffalo, NY, 1995.

52. English K.W.; Miller E.; and Bloebaum C.L.: "Total derivative-based coupling suspension for system reduction in complex design," Paper Presented at 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Bellevue, WA, September 4–6, 1996.

53. English, K.W.; and Bloebaum C.L.: "Enhanced total derivative based coupling suspension in complex system optimization," Paper Presented at 39th Aerospace Sciences Meeting and Exhibit, Reno, NV, January 8–11, 2001.

54. English, K.W.; Bloebaum, C.L.; and Miller E.: "Development of multiple cycle coupling suspension in the optimization of complex systems," *Structural and Multidisciplinary Optimization*, Vol. 22, No. 4, pp. 268–283, 2001.

55. Cheung, J.; Scanlan, J.; Wong, J.; et al.: "Application of Value-Driven Design to Commercial AeroEngine Systems," Paper Presented at 10th AIAA Aviation Technology, Integration and Operations Conference, Fort Worth, TX, September 13–15, 2010.

56. Dyas, J.E.; Clerkin, J.H.; and Mesmer, B.A.: "A Congress-Based Value Model for NASA Funding Allocation," Paper Presented at AIAA SPACE and Astronautics Forum and Exposition, Orlando, FL, 2017, September 12–14, 2017.

57. Topcu, T.; and Mesmer, B.: "Incorporating end-user models and associated uncertainties to investigate multiple stakeholder preferences in system design," *Research in Engineering Design*, Vol. 29, No. 3, pp. 411–431, July 2018.

58. Bhatia, G.; Mesmer. B; and Weger, K.: "Mathematical Representation of Stakeholder Preferences for the SPORT Small Satellite Project," Paper Presented at 2018 AIAA SciTech, Orlando, FL, January 8–12, 2018.

59. Millard, D.; Mesmer, B.; Gholston, S.; and Kuhn, S.: "Optimization of Nurse Staffing under Varying Preferences," *Journal of Management and Engineering Integration*, Vol. 11, No. 1, pp. 11–19, 2018.

60. Palma, G.E.; and Mesmer, B.: "A Preliminary Content Analysis of NASA's Nextstep-2 Habitat Documentation for Preference Representation," Paper Presented at 2018 AIAA Aerospace Sciences Meeting, Kissimmee, FL, January 8–12, 2018.

61. Clerkin, J.; and Mesmer, B.: "A Review of Value Modeling in the NASA Systems Engineering Research Consortium," Paper Presented at 16th Conference on Systems Engineering Research, Charlottesville, VA, May 8–9, 2018.

62. Bhatia, G.; Dyas, J.; Clerkin, J.; and Mesmer, B.: "Formation of Preliminary Questionnaires for the Solicitation of Stakeholder Preferences," Paper Presented at ASEM 2017 International Annual Conference, Huntsville, AL, October 18–21, 2017.

63. Systems Engineering Body of Knowledge: "System Resilience," <https://www.sebokwiki.org/wiki/System_Resilience>, accessed June 10, 2019.

64. Jones, K.H.: "Antifragile Systems: An Enabler for System Engineering of Elegant Systems," Paper Presented at 2nd International Workshop on Computational Antifragility and Antifragile Engineering (Antifragile-2015) and 6th International Conference on Ambient Systems, Networks and Technologies (ANT-2015), London, UK, June 2–5, 2015.

65. Taguchi, G.: *Introduction to Quality Engineering: Designing Quality into Products and Processes*, Tokyo, Asian Productivity Organisation, 1986.

66. Kitano, H.: "Towards a theory of biological robustness," *Molecular Systems Biology*, Vol. 3, No. 137, September 18, 2007.

67. Malak, R.J.; Baxter, B.; and Hsiao, C.: "A Decision-based Perspective on Assessing System Robustness," in Proc. of the 2015 Conference on Systems Engineering Research, March 17–19, 2015, J. Wade and Robert Cloutier (eds.), Elsevier Procedia, Procedia Computer Science, Vol. 44, Hoboken, NJ, pp. 619–629.

68. Zumbado, J.R.: "Human System Integration (HIS) Practitioner's Guide," NASA/SP-2015-3709, NASA Johnson Space Center, Houston, TX, 154 pp., 2015.

69. Mac Lane, S.M.: *Categories for the Working Mathematician*, 2nd ed., S. Axler, F.W. Gehring, and K.A. Ribet (eds.), Springer Science and Business Media, New York, NY, 262 pp., 1971.

70. Eilenberg, S.; and Lane, S.M: "General Theory of Natural Equivalences," *Transactions of the American Mathematical Society*, Vol. 58, No. 2, pp. 231–294, 1945.

71. Winona State University Computer Science: "CS 440 Theory of Algorithms / CS 468 Algorithms in Bioinformatics: Dynamic Programming, Part II," <https://cs.winona.edu/lin/cs440/ch08-2.pdf>, 2007.

72. Darmouth College: "Mathematics Across the Curriculum," <http://www.dartmouth.edu/~matc/DiscreteMath/>, 2018.

73. Brown, R.: *Topology and Groupoids: A Geometric Account of General Topology, Homotopy Types and the Fundamental Groupoid*, 2nd ed., <www.groupoids.org.uk>, Deganwy, UK, 539 pp., 2006.

74. Contributors: Bell, J.L.; Ulrich, B.; Cantini, A.; et al.: *One Hundred Years of Russell's Paradox*: *Mathematics, Logic, Philosophy (De Gruyter Series in Logic and Its Applications, Book 6)*, G. Link (ed.), Walter de Gruyter, Berlin, Germany, 671 pp., 2004.

75. Suppes, P.: *Axiomatic Set Theory*, Dover Publications, New York, NY, 288 pp., 2012.

76. Bemer, R.W.: "Survey of coded character representation," *Communications of the ACM*, Vol. 12, No. 3, pp. 639–642, December 1960.

77. Encyclopedia Britannica: "Niels Henrik Abel," <https://www.britannica.com/biography/Niels-Henrik-Abel>, 2018.

78. Wolfram Research: "Abel, Niels Henrik," <http://scienceworld.wolfram.com/biography/Abel.html>, 2018.

79. Hoffman, K.; and Kunze, R.: *Linear Algebra*, 2nd ed., Prentice-Hall, Inc., New Jersey, 416 pp., 1971.

80. Stoll, R.R.: *Linear Algebra and Matrix Theory*, Dover Publications, Mineola, NY, 290 pp., 1952.

81. Strang, G.: *Introduction to Linear Algebra*, 5th ed., Wellesley-Cambridge Press, Wellesley, MA, 584 pp., 2016.

82. Watson, M.D.; Andrews, J.G.; and Larsen, J.A.: "Principles of Sociology in Systems Engineering," Paper Presented at the 2017 International Annual Conference of the American Society for Engineering Management, Annapolis, MD, E-H. Ng, B. Nepal, and E. Schott (eds.), October 18–21, 2017.

83. Watson, M.D.; Andrews, J.G.; Eckley, J.C.; and Culver, M.L.: "Practical Application of Sociology in Systems Engineering," Paper Presented at the 2017 International Annual Conference of the American Society for Engineering Management, Annapolis, MD, E-H. Ng, B. Nepal, and E. Schott (eds.), October 18–21, 2017.

84. Merton, R.K.: "Opportunity Structure," in *On Social Structure and Science*, P. Sztompka (ed.), University of Chicago Press, Chicago, IL, pp. 153–161, 1996.

85. Merton, R.K.: "On Sociological Theories of the Middle Range," and "The Role-Set," in *Social Structure and Science*, P. Sztompka (ed.), University of Chicago Press, Chicago, IL, pp. 44, 113–122, 1996.

86. Merton, R.K.: "Social Dysfunctions," in *Social Structure and Science*, P. Sztompka (ed.), University of Chicago Press, Chicago, IL, pp. 96–100, 1996.

87. Christensen, C.M.: *The Innovators Dilemma: The Revolutionary Book that Will Change the Way You Do Business*, Harper Collins, New York, NY, 2003.

88. Merton, R.K.: "Sociological Ambivalence," in *Social Structure and Science*, P. Sztompka (ed.), University of Chicago Press, Chicago, IL, pp. 123–131, 1996.

89. Roberto, M.A., Know What You Don't Know: How Great Leaders Prevent Problems Before They Happen, Prentice Hall, NJ, 2009.

90. Merton, R.K.: "Social Structure and Anomie," in *Social Structure and Science*, P. Sztompka (ed.), University of Chicago Press, Chicago, IL, pp. 132–152, 1996.

91. Merton, R.K.: "The Self-Fulfilling Prophecy," in *Social Structure and Science*, P. Sztompka (ed.), University of Chicago Press, Chicago, IL, pp. 183–201, 1996.

92. Merton, R.K.: "Specified Ignorance," in *Social Structure and Science*, P. Sztompka (ed.), University of Chicago Press, Chicago, IL, pp. 51–56, 1996.

93. Merton, R.K.: "Socially Expected Durations," in *Social Structure and Science*, P. Sztompka (ed.), University of Chicago Press, Chicago, IL, pp. 162–169, 1996.

94. Merton, R.K.: "Paradigm for the Sociology of Knowledge (1945)", in *Social Structure and Science*, P. Sztompka (ed.), University of Chicago Press, Chicago, IL, pp. 205–222, 1996.

95. Merton, R.K.: "Insiders and Outsiders," in *Social Structure and Science*, P. Sztompka (ed.), University of Chicago Press, pp. 241–263, 1996.

96. Merton, R.K.: "The Unanticipated Consequences of Purposive Social Action," *American Sociological Review*, Vol. 1, No. 6, pp. 894–904, 1936.

97. Columbia Accident Investigation Board: Columbia Accident Investigation Board Report, Vol. 1, NASA, Washington, DC, 248 pp., August 2003.

98. Rogers Commission: Report by the Presidential Commission on the Space Shuttle Challenger Accident, NASA, Washington, DC, pp. 261, 1986.

99. Luhmann, N.: *Social Systems, Writing Science*, Stanford University Press, Stanford, CA, 627 pp., 1995.

100. Knudsen, M.; and Vogd W.: *Systems Theory and the Sociology of Health and Illness*, Taylor and Francis, Oxford, UK, 240 pp., 2015.

101. Leydesdorff, L.: *A Sociological Theory of Communication*: *The Self-Organization of the Knowledge-Based Society*, 2nd ed., Universal Publishers, 359 pp., 2001.

102. Austin-Breneman, J.; Yu, B.Y.; and Yang, M.C.: "Biased Information Passing Between Sub-Systems Over Time in Complex System Design," Paper Presented at the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Biennial International Conference on Dynamics for Design, and 26th International Conference on Design Theory Methodology, Buffalo, NY, August 17–20, 2014.

103. Austin-Breneman, J.; Yu, B.; and Yang, M.C.: "Biased Information Passing Between Subsystems Over Time in Complex System Design," *ASME Journal of Mechanical Design*, Vol. 138, No. 1, November 2015.

104. Watson, M.D.: "Information Theory Applied to Decision Making Structures," Paper Presented at 2018 Conference on Systems Engineering Research, Charlottesville, VA, May 8–9, 2018.

105. Merriam-Webster Dictionary: "information theory," <https://www.merriam-webster.com/dictionary/information%20theory>, 2019.

106. Reza, M.R.: *An Introduction to Information Theory,* Dover, New York, NY, 528 pp., 1994.

107. Williams, C.; and Derro, M.: "NASA Systems Engineering Behavior Study," NASA Office of the Chief Engineer, Washington, DC, 31 pp., 2008.

108. Rhodes, D.H.; Lamb, C.T.; and Nightingale, D.J.: "Empirical Research on Systems Thinking and Practice in the Engineering Enterprise," Paper Presented at 2nd Annual IEEE Systems Conference, Montreal, CA, April 2–8, 2008.

109. Davidz, H.L.; Nightingale, D.; and Rhodes, D.H.: "Enablers and Barriers to Systems Thinking Development: Results of a Qualitative and Quantitative Study," Paper Presented at 2005 Conference on Systems Engineering Research, Hoboken, NJ, March 23–25, 2005.

110. Frank, M.: "Engineering Systems Thinking and Systems Thinking," *Journal of Systems Engineering*, Vol. 3, No. 3, pp. 163–168, 2000.

111. Frank, M.: "Knowledge, abilities, cognitive characteristics and behavioral competencies of engineers with high capacity for engineering systems thinking (CEST)," *Journal of Systems Engineerin*g, Vol. 9, No. 2, pp. 91–103, February 2006.

112. Frank, M.: "Engineering Systems Thinking: Cognitive Competencies of Successful Systems Engineers," *Procedia Computer Science*, Vol. 8, pp. 273–278, 2012.

113. Greene, M.T.; and Papalambros, P.Y.: "A cognitive framework for engineering systems thinking," Paper Presented at 14th Annual Conference on Systems Engineering Research, Huntsville, AL, March 22–24, 2016.

114. Brown, T.: "Design Thinking," Harvard Business Review, Reprint R0806E, 10 pp., June 2008.

115. Greene, M.T.; Gonzalez, R.; Papalambros, P.Y.; and McGowan, A.-M.: "Design Thinking vs. Systems Thinking for Engineering Design: What's the Difference?" Paper presented at 21st International Conference on Engineering Design, The University of British Columbia, Vancouver, Canada, August 21–25, 2017.

116. Doyle, J.K.: "The cognitive psychology of systems thinking," *System Dynamics Review*, Vol. 13, No. 3, pp. 253–265, 1997.

117. Johnson-Laird, P.N.: *Human and Machine Thinking*, Lawrence Erlbaum Associates, Hillsdale, NJ, 200 pp., 1994.

118. Cummins, R.: *The Nature of Psychological Explanation*, MIT Press, Cambridge, MA, 232 pp., 1983.

119. Kelly, G.A.: *The Psychology of Personal Constructs*, W.W. Norton and Company, Oxford, UK, 1955.

120. Dawson, M.R.W.; and Medler, D.A.: "Dictionary of Cognitive Science," University of Alaberta,<http://www.bcp.psych.ualberta.ca/~mike/Pearl_Street/Dictionary/dictionaryhtml>, 2009.

121. *The MIT Encyclopedia of the Cognitive Sciences*, R.A. Wilson and F.C. Kell (eds.), MIT Press, ISBN 10:0262232006/ISBN 13:9780262232005, 2005.

122. Cross, N.: "Designerly Ways of Knowing: Design Discipline versus Design Science," *Design Studies*, Vol. 17, No. 3 pp. 49–55, Summer 2001.

123. Rittel, H.J.W.; and Webber, M.M.: "Dilemmas in a General Theory of Planning," *Policy Sciences*, Vol. 4, No. 2, pp. 155–169, 1973.

124. Checkland, P.: *Systems Thinking, Systems Practice*, John Wiley & Sons, West Sussex, England, UK, 330 pp., 1981.

125. Clarke, S.: *Human Centered Methods in Information Systems: Current Research and Practice*, Idea Group Publishing, Hershey, PA, 241 pp., 2000.

126. Plattner, H.; Meinel, C.; and Leifer, L. (eds.): *Design Thinking Research: Building Innovation Eco-Systems*, Springer Switzerland, 2014.

127. Forrester, J.W.: *Industrial Dynamics*, Pegasus Communications, Martino Fine Books, Waltham, MA, 482 pp., 2013.

128. Dinar, M.; Shah, J.J.; Cagan, J.; et al.: "Empirical Studies of Designer Thinking: Past, Present, and Future," *Journal of Mechanical Design*, Paper No. MD-13-1523, Vol. 137, No. 2, 13 pp., February 2015.

129. Watson, M.; Griffin, M.; Farrington, P.A.; et al.: "Building a Path to Elegant Design," in *Proc. of the American Society for Engineering Managemen*t, 2014 International Annual Conference, Virginia Beach, VA, October 15–28, 2014, 11 pp., 2014.

130. Senge, P.: *The Fifth Discipline: The Art and Practice of the Learning Organization*, Doubleday, New York, NY, 1990.

131. Checkland, P.; and Scholes, J.: *Soft Systems Methodology in Action*, John Wiley and Sons, West Sussex, UK, 418 pp., 1999.

132. Oliver, J.; Vesty, G.; and Brooks, A.: "Conceptualising Integrated Thinking in Practice," *Managerial Auditing Journal*, Vol. 31, No. 2, pp. 228–248, 2016.

133. Dym, C.L.; Agogino, A.M.; Eris, O.; et al.: "Engineering Design Thinking, Teaching, and Learning," *Journal of Engineering Education*, Vol. 94, No. 1, pp. 103–120, 2005.

134. Lammi, M.; and Becker, K.: "Engineering Design Thinking," *Journal of Technology Education*, Vol. 24, No. 2, pp. 55–77, 2013.

135. McGowan, A.M.; Bakula, C.; and Castner, R.: "Lessons Learned from Applying Design Thinking in a NASA Rapid Design Study in Aeronautics," Paper Presented at AIAA Sci-Tech 2017, Grapevine, TX, January 9–13, 2017.

136. Long, C.: "Teach Your Students to Fail Better with Design Thinking," *Learning and Leading with Technology*, Vol. 39, No. 5, pp. 16–20, February 2012.

137. Patel, S.; and Mehta, K.: "Systems, Design, and Entrepreneurial Thinking: Comparative Frameworks," *Systemic Practice and Action Research*, Vol. 30, No. 5, pp. 515–533, November 2017.

138. Lebeer, J.: "Modifiability and mediated learning in the light of neuroscientific evidence of ecological plasticity," *Transylvanian Journal of Psychology*, 2014 Special Issue, pp. 51–79, 2014.

139. Li, S.C.: "Neuromodulation and developmental contextual influences on neural and cognitive plasticity across the lifespan," *Neuroscience and Biobehavioral Reviews*, Vol. 37, No. 9 Pt. B, pp. 2201–2208, August 2013.

140. Lovden, M.; Lindenberger, U.; Sabine, S.; et al.: "A theoretical framework for the study of adult cognitive plasticity," *Psychological Bulletin*, Vol. 136, No. 4, pp. 659–676, 2010.

141. Wiley, I.: *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, 4th ed., Wiley and Sons, Inc., New York, NY, 304 pp., 2015.

142. Herbsleb, J.D.; and Mockus, A.: "An empirical study of speed and communication in globally distributed software development," *IEEE Transactions on Software Engineering*, Vol. 29, No. 6, pp. 481–494, June 2003.

143. Wasserman, S.; and Faust, K.: *Social Network Analysis: Methods and Applications*, Cambridge University Press, Cambridge, UK, 857 pp., 1994.

144. Cataldo, M.; Wagstrom, P.A.; Herbsleb, J.D.; and Carley, K.M.: "Identification of Coordination Requirements: Implications for the Design of Collaboration and Awareness Tools," in *Proc. of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work*, November 4–8, 2006, ACM, New York, NY, pp. 353–362, 2006.

145. Sosa, M.E.; Eppinger, S.D.; and Rowles, C.M.: "The Misalignment of Product Architecture and Organizational Structure in Complex Product Development," *Management Science*, Vol. 50, No. 12, pp. 1674–1689, 2004.

146. Gokpinar, B.; Hopp, W.J.; and Iravani, S.M.R.: "The Impact of Misalignment of Organizational Structure and Product Architecture on Quality in Complex Product Development," *Management Science*, Vol. 56, No. 3, pp. 468–484, January 2010.

147. Conway, M.E.: "How do Committees Invent?" *Datamation*, Vol. 14, pp. 28–31, April 1968.

148. MacCormack, A.D.; Rusnak, J.; and Baldwin, C.Y.: "Exploring the Duality Between Product and Organizational Architectures: A Test of the "Mirroring Hypothesis," *Research Policy*, Vol. 41, No. 8, pp. 1309–1324, October 2012.

149. Milo, R.; Shen-Orr, S.; Itzkovitz, S.; et al.: "Network Motifs: Simple Building Blocks of Complex Networks," *Science*, Vol. 298, No. 5594, pp. 824–827, October 2002.

150. Parraguez, P.; and Maier, A.: "Using Network Science to Support Design Research: From Counting to Connecting," in Experimental Design Research, P. Cash, T. Stanković, and M. Štorga (eds.), Springer International Publishing, Dordrecht, Switzerland, pp. 153–172, May 2016.

151. Collopy, A.X.; Adar, E.; and Papalambros, P.Y.: "Multi-level network modeling for an integrative analysis of coordination in systems design," Paper Presented at 15th Annual Conference on Systems Engineering Research, Los Angeles, CA, March 23–25, 2017.

152. Collopy, A.X.; Li, C.; Liu, T.; et al.: "Coordination in Design Teams as a Lens to Identifying Team Roles," in *Proc. of 22nd International Conference on Engineering Design*, August 4–8, 2019, Delft, The Netherlands, 2019.

153. Collopy, A.X.; Greene, M.T.; Adar, E.; and Papalambros, P.Y.: "Estimating the Impact of Systems Engineers on Systems Design Processes," in *Proc. of the 21st International Conference on Engineering Design*, August 21–25, 2017, Vancouver, CA, 2017.

154. Sosa, M.E.; Gargiulo, M.; and Rowles, C.: "Can Informal Communication Networks Disrupt Coordination in New Product Development Projects?" *Organization Science*, Vol. 26, No. 4, pp. 1059–1078, 2015.

155. Wilson, J.Q.: *Bureaucracy: What Government Agencies Do and Why They Do It*, Basic Books, New York, NY, 464 pp., 1991.

156. Jensen, M.C.; and Meckling, W.H.: "Theory of the firm: Managerial behavior, agency costs and ownership structure," *Journal of Financial Economics*, Vol. 3, No. 4, pp. 305–360, October 1976.

157. Hale, W.; and Bauer, F.: "Government Insight/Oversight for Commercial Crew Transportation," Rev. N, NASA Headquarters, Washington, DC, pp. 14, March 10, 2010.

158. Chaplain, C.: "Space Acquisitions: Challenges Facing DOD as It Changes Approaches to Space Acquisitions," Government Accountability Office, Washington, DC, March 9, 2016.

159. Drezner, J.A.; Blickstein, I.; Raman, R.; et al.: Measuring the Statutory and Regulatory Constraints on Department of Defense Acquisition: An Empirical Analysis, RAND Corporation, Santa Monica, CA, 107 pp., 2007.

160. Johnson, S.B.: *The Secret of Apollo*, Johns Hopkins University Press, Baltimore, MD, pp. 223–224, 2002.

161. Gavrieli, D.: "Regulatory Burden: A Systems Analysis of Weapons Acquisition in the U.S. Department of Defense," Ph.D. Dissertation, University of Southern California, Los Angeles, CA, 24 pp., 2008.

162. SpaceRef, Testimony of Scott Pace, National Security Space Launch Programs, Hearing of the Senate Committee on Defense Appropriations, March 5, 2014.

163. Coopers and Lybrand: "The DoD Regulatory Cost Premium: A Quantitative Assessment," Defense Technical Information Center, Washington, DC, 141 pp., 1994.

164. Lorell, M.; and Graser, J.C.: An Overview of Acquisition Reform Cost Savings Estimates, RAND, Santa Monica, CA, 180 pp., 2001.

165. Drezner, J.A.; Raman, R.; Blickstein, I.; et al.: Measuring the Statutory and Regulatory Constraints on Department of Defense Acquisition: An Empirical Analysis, RAND, Santa Monica, CA, 131 pp., 2007.

166. Bradburn, N.; Rips, L.; and Shevell, S.: "Answering Autobiographical Questions: The Impact of Memory and Inference on Surveys," *Science*, Vol. 236, No. 4798, pp. 157–61, doi:10.1126/science.3563494, April 1987.

167. Tversky, A.; and Kahneman, D.: "Judgment under Uncertainty: Heuristics and Biases," *Science*, Vol. 185, No. 4157, 1124–1131, 1974.

168. Gavrieli, D.: "Regulatory Burden: A Systems Analysis of Weapons Acquisition in the U.S. Department of Defense," Ph.D. Dissertation, University of Southern California, Los Angeles, CA, 173 pp., March 2008.

169. Brainard, S.; and Szanjnfarber, Z.: "Understanding the Burden of Government Oversight on Engineering Work: Adding Empirical Data to the Debate," *Space Policy*, Vol. 42, pp. 70–80, doi: 10.1016/j.spacepol.2017/07.001, August 2017.

170. Gouldner, A.: "Red tape as a social problem," in *Reader in Bureaucracy*, R. Merton (ed.), Free Press, New York, 1952.

171. Bozeman, B.; and Feeney, M.K.: *Rules and Red Tape: A Prism for Public Administration Theory and Research*, M.E. Sharpe (ed.), New York, NY, 216 pp., 2011.

172. Kaufman, H.: *Red Tape, Its Origins, Uses, and Abuses, Brookings Institution Press*, Washington, DC, 100 pp., 1977.

173. Pandey, S.K.; and Welch, E.W.: "Beyond Stereotypes: A Multistage Model of Managerial Perceptions of Red Tape," *Journal of Administration and Society*, Vol. 37, No. 5, pp. 542–575, doi:10.1177/0095399705278594, November 1, 2005.

174. Welch, E.W.; and Pandey, S.K.: "E-Government and Bureaucracy: Toward a Better Understanding of Intranet Implementation and Its Effect on Red Tape," *Journal of Administration and Societ*y, Vol. 17, No. 3, pp. 379–404, October 11, 2006.

175. Meadows, D.: *Thinking in Systems: A Primer*, D. Wright (ed.), Chelsea Green Publishing, White River Junction, VT, 240 pp., 2008.

176. Lane, D.: "Social theory and system dynamics practice," *European Journal of Operational Research*, Vol. 113, pp. 501–527, March 1999.

177. Barlas, Y.: "System dynamics: systemic feedback modeling for policy analysis," *System*, Vol. 1, 29 pp., January 2007.

178. Lyneis, J.M.; and Ford, D.N.: "System dynamics applied to project management: a survey, assessment, and directions for future research," *System Dynamics Review*, Vol. 23, 157–189, 2007.

179. Sterman, J.D.: "System dynamics modeling: Tools for Learning in a Complex World," *California Management Review*, Vol. 43, No. 4, pp. 8–25, 2001.

180. Sterman, J.: *Business Dynamics: Systems Thinking and Modeling for a Complex World*, McGraw-Hill, New York, NY, 1008 pp., 2000.

181. Mesmer, B.; and Bloebaum, C.L.: "Modeling Decision and Game Theory Based Pedestrian Velocity Vector Decisions with Interacting Individuals," *Safety Science*, Vol. 87, p. 116–130, August 2016.

182. Mesmer, B.; and Bloebaum, C.L.: "An End-User Decision Model with Information Representation for Improved Performance and Robustness in Complex System Design," *Research in Engineering Design*, Vol. 26, No. 3, pp. 235–251, July 2015.

183. Mesmer, B.; and Bloebaum, C.L.: "Incorporation of Decision, Game, and Bayesian Game Theory in an Emergency Evacuation Exit Decision Model," *Fire Safety Journal*, Vol. 67, pp. 121–134, July 2014.

184. "NASA Systems Engineering Handbook," NASA/SP-2016-6105, Rev. 2, NASA Headquarters, Washington, DC, https://www.nasa.gov/sites/default/files/atoms/files/nasa_systems_engineering_handbook_0.pdf, 2016.

185. Watson, M.D.: “Engineering Elegant Systems: The Practice of Systems Engineering,” NASA/TP–20205003646, NASA Marshall Space Flight Center, Huntsville, AL, 212 pp., June 2020.

186. Burns, L.: “Complex systems characteristics in the context of Systems Engineering,” Paper presented to NASA Systems Engineering Research Consortium, The University of Alabama in Huntsville, Huntsville, AL, February 2016.

187. McEver, J.: “A Complexity Primer for Systems Engineers,” International Council on Systems Engineering, November 2015.

188. Watson, M.D.; Anway, R.; McKinney, D.; et al.: “Appreciative Methods Applied to the Assessment of Complex Systems,” in *Proceedings*, INCOSE International Symposium, Orlando, FL, Vol. 29, Issue 1, pp. 448–477, July 2019.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
01–06–2020

2. REPORT TYPE
Technical Publication

3. DATES COVERED *(From - To)*

4. TITLE AND SUBTITLE
Engineering Elegant Systems: Theory of Systems Engineering

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
M.D. Watson, B.L. Mesmer,* and P.A. Farrington*

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
George C. Marshall Space Flight Center
Huntsville, AL 35812

8. PERFORMING ORGANIZATION REPORT NUMBER
M–1509

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546–0001

10. SPONSORING/MONITOR'S ACRONYM(S)
NASA

11. SPONSORING/MONITORING REPORT NUMBER
NASA/TP–20205003644

12. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified-Unlimited
Subject Category 31
Availability: NASA STI Information Desk (757–864–9658)

13. SUPPLEMENTARY NOTES
Prepared by the Technical Management Office, Engineering Directorate
*The University of Alabama in Huntsville

14. ABSTRACT

This Technical Publication describes a theoretical basis for systems engineering. A frame work for the theoretical basis of systems engineering is defined indicating systems engineering has two main focuses: system design and integration, and discipline integration. System engineering processes provide the organization of the system engineering efforts. A set of postulates, principles, and hypotheses are defined to guide the application of systems engineering processes and approaches. System design and integration includes the application of important concepts for systems design, analysis, modeling, and integration including system integrating physics, system state variables, engineering statistics, system value models, and multidisciplinary design optimization. Discipline integration incorporates aspects of sociology in managing the flow of information on the system through the development and operations organization(s). Decision-making structures and flows are discussed as well as cognitive science influences. Social forces influencing the system development and operation include policy and law are also addressed. Different modeling types for capturing discipline integration connections and information flows are identified.

15. SUBJECT TERMS

systems engineering, systems engineering principles, system design, system integration, system integrating physics, system state variables, engineering statistics, discipline integration, systems engineering sociology, information theory, decision-making structures

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			STI Help Desk at email: help@sti.nasa.gov
U	U	U	UU	278	**19b. TELEPHONE NUMBER** *(Include area code)* STI Help Desk at: 757–864–9658

National Aeronautics and
Space Administration
IS02
George C. Marshall Space Flight Center
Huntsville, Alabama 35812