Model Integration Approaches for System Design and Analysis

10 September 2019

Michael D. Watson, Ph.D.

Consortium Team
UAH
George Washington University
Iowa State
Texas A&M
Dependable System Technologies, LLC
Multidisciplinary Software Systems
Research Corporation (MSSRC)
Missouri University of S&T
University of Michigan
AFRL Wright Patterson
System Modeling Types

System Relational Modeling (i.e., MBSE)

System Integrating Physics

System Value Models

System Stave Variable Modeling
  - Goal Function Tree (GFT)
  - State Analysis Model (SAM)

System Statistical Modeling

System Dynamics Modeling

Summary
System Models are important to gain understanding of the System

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

System Models convey information at the system level

- Complementary to discipline based engineering models

  - Integrate system functions and relationships within the system context
    - Systems Engineering Principle 3(i): Systems engineering seeks a best balance of functions and interactions within the system budget, schedule, technical, and other expectations and constraints.

  - Provide a technical systems basis for system operations and maintenance functions, approaches, and procedures

  - Provide a relationship of the system capabilities to the stakeholder expectations
System Model Types

- System Modeling is based on a set of system models rather than a single system model

System Modeling Types
- Relational (i.e., MBSE)
- Physics-Based
- State Variable
- System Value
- Statistical
- System Dynamics
System Relational Models
System Relational Models focus on the relationships between system components, system process flows, and requirements

- Model Based Systems Engineering (MBSE) typically refers to SysML based models

- Several vendors provide these tools
  - Cameo Systems Modeler/Magic Draw
  - Innoslate (Lifecycle Modeling Language (LML))
  - Enterprise Architect
  - Rationale Rhapsody
  - Visual Paradigm
  - Modelio SysML Architect
  - Eclipse Papyrus

- Other tools provide better capabilities in some aspects
  - Requirements Management
    - Cradle
    - Doors
    - CORE
  - Visualization
    - Visio
    - Tom Sawyer
System Relational Models Support

- System Block Diagrams and Internal Block Diagrams define relationships between components/assemblies/subsystems of the system
- Provides Requirements traceability including verification support
- Provides Activity Diagrams, State Machine Diagrams to illustrate Use Cases and Process Flows
  
  - Use Cases can provide a structure to feed into an initial Discrete Event Simulation (DES) model to support statistical process flow analysis
System Integrating Physics
Physics provides some help with capture of the system physics relationships to develop a physics based model.

These integration relationships exist in physics but are not often used in engineering design.

These physics based integration relationships are driven by the system type:

- **Thermodynamic Systems**
  - Aircraft
    - Propeller Driven
    - Jet Aircraft
    - Electric
    - Rotorcraft/VTOL
    - Gliders
  - Automobiles
  - Electrical Systems
  - Fluid Systems
  - Launch Vehicles and Spacecraft
    - Robotic
      - Integrated through the bus which is a thermodynamic system

Each instrument may have a different integrating physics but integrates with the bus thermodynamically:

- Crew Modules
  - Integrated by the habitable volume (i.e., ECLSS)

A thermodynamic system:

- Entry, Descent, and Landing (EDL)
  - Integrated by thermodynamics as spacecraft energy is reduced (i.e., destroyed) in EDL
- Power Plants
- Ships

- **Optical Systems**
- **Logical Systems**
  - Data Systems
  - Communication Systems
- **Structural Systems**
- **Biological Systems**

System Integrating Physics provides the engineering basis for the System Model.
Thermodynamics Has Balance Relationships

**Energy Balance (First Law of Thermodynamics)**
- \( E_{\text{in}} - E_{\text{out}} = E_{\text{sys}2} - E_{\text{sys}1} \),
- \( Q - W = m(u_2 - u_1) + \frac{1}{2}m(v_2^2 - v_1^2) + mg(z_2 - z_1) \) for a control volume

**Entropy Balance (Second Law of Thermodynamics)**
- \( S_{\text{in}} - S_{\text{out}} + S_{\text{gen}} = (S_{\text{sys}2} - S_{\text{sys}1}) \), where \( S_{\text{gen}} \geq 0 \)

**Exergy Balance (Integration of First and Second Laws)**
- \( X_{\text{in}} - X_{\text{out}} + X_{\text{des}} = (X_{\text{sys}2} - X_{\text{sys}1}) \), where \( X_{\text{des}} = T_0S_{\text{gen}} \geq 0 \) (\( T_0 \) in Kelvin)
- \( \sum(1 - \frac{T_0}{T_k})Q_k - [W - P_0(Vol_2 - Vol_1)] + X_{\text{des}} = m \left[(h_2 - h_1) + T_0(S_{\text{sys}2} - S_{\text{sys}1}) + \frac{1}{2}(v_2^2 - v_1^2) + g(z_2 - z_1)\right] \) for a control volume

**All relationships maintain mass balance**
- \( m_{\text{in}} - m_{\text{out}} = m_2 - m_1 \)
Exergy Balance Relationship

Exergy Balance for a rocket balances the exergy expended (fluid flow out of the nozzle) with the change in the vehicles kinetic and potential energy

- Mass balance is maintained
- Rockets are control volumes, not control masses
  - Each stage is a constant control volume
  - The vehicle is the integration (addition) of separate control volumes
  - Staging results in the dropping of a control volume (mass drop) but not a change in the individual stage control volumes
  - Entropy and Enthalpy of propellant products assumed negligible (are for LOX, LH2)

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

\[
M_{\text{vehicle,initial}} \frac{V_{\text{vehicle,initial}}^2}{2} + \left( \frac{GM_e M_{\text{vehicle,initial}}}{r_{\text{altitude,initial}}} - \frac{GM_e M_{\text{vehicle,final}}}{r_{\text{altitude,final}}} \right)
\]

- Represents energy expended in gaining velocity and altitude
- Rocket equation can be derived from the exergy balance for a rocket
- Orbital mechanics energy balance is also maintained in the exergy balance equation for a rocket

\[
\eta_{\text{exergy}} = \frac{X_{\text{recovered}}}{X_{\text{expend}}}; \quad \frac{X_{\text{final}} - X_{\text{initial}}}{X_{\text{expend}}} = 1 - \frac{X_{des}}{X_{\text{expend}}} = 1 - \frac{X_{des}}{\Delta m_{prop} (H_{\text{prop}} + \frac{V_e^2}{2})}; \quad \text{Launch Vehicle, Planetary Departure (Accelerating)}
\]

\[
\eta_{\text{exergy}} = \frac{-X_{\text{recovered}}}{X_{\text{expend}}}; \quad \frac{X_{\text{initial}} - X_{\text{final}}}{X_{\text{expend}}} = 1 - \frac{X_{des}}{\Delta m_{prop} (H_{\text{prop}} + \frac{V_e^2}{2})}; \quad \text{Lander, Planetary Arrival (Braking)}
\]
Launch Vehicle System Exergy Efficiency

\[
\Delta m_{prop} \sum_{\text{stage}} \left( h_{\text{prop}} + \frac{v_e^2}{2} \right) - X_{\text{des}} = \left( M_{\text{vehicle,final}} \frac{v_e^2}{2} - M_{\text{vehicle,initial}} \frac{v_e^2}{2} \right) + \left( \frac{GM_{E}}{r_{\text{altitude,initial}}} - \frac{GM_{E}}{r_{\text{altitude,final}}} \right).
\]
Un-crewed Spacecraft Exergy Balance and Optical Transfer Function

Spacecraft Exergy Balance

\[
\Delta m_{\text{propellant,engine}} \left( h_{\text{prop,engine}} + \frac{V_{e,\text{engine}}^2}{2} \right) + \Delta m_{\text{propellant,thruster}} \left( h_{\text{prop,thruster}} + \frac{V_{e,\text{thruster}}^2}{2} \right) + \sum_t \left[ \sigma A e \left( T_{\text{radiator}}^4 - T_{\text{space}}^4 \right) + V_{\text{bus}} I_{\text{bus}} \cos(\theta) \right] \Delta t - X_{\text{des}}
\]

\[
= \left( M_{\text{vehicle,final}} \left( I_{c,\text{final}} \frac{\omega_{\text{vehicle,final}}^2}{2} + \frac{V_{\text{vehicle,final}}^2}{2} \right) - M_{\text{vehicle,initial}} \left( I_{c,\text{initial}} \frac{\omega_{\text{vehicle,initial}}^2}{2} + \frac{V_{\text{vehicle,initial}}^2}{2} \right) \right) + \sum \left( 1 - \frac{T_{\text{fluid}}}{T_{\text{instrument}}} \right) Q_{\text{instrument}} + \sum_{\text{battery}} p_{\text{electric,stored}} + \sum_{\text{instruments}} p_{\text{electric,used}} \Delta t
\]

Optical Transfer Function

\[
\int_{-\infty}^{\infty} \psi_{\text{obj}} s_{\text{r}} \, dx \, dy = \int_{-\infty}^{\infty} \psi_{\text{obj}} (x_0 + \epsilon x, y_0 + \epsilon y) e^{i \left( \frac{k_{\epsilon}}{2R} \right) (x^2 + y^2)} \circ \left( \frac{x + \Delta x + \delta x}{R}, \frac{y + \Delta y + \delta y}{R} \right) \, dx \, dy
\]

Where

\[
\epsilon x = 1.22 \lambda_0 \frac{f_1}{d_0 + \epsilon z + \omega_z \Delta t} + v_x \Delta t + \omega_x \Delta t
\]

\[
\epsilon y = 1.22 \lambda_0 \frac{f_1}{d_0 + \epsilon z + \omega_z \Delta t} + v_y \Delta t + \omega_y \Delta t
\]

\[
f_1 = - \frac{R}{2} = - \sqrt{(x + \Delta x + \delta x)^2 + (y + \Delta y + \delta y)^2 + (z + \Delta z + \delta z)^2} \]

\[
\Delta x = \alpha x \Delta T
\]

\[
\Delta y = \alpha y \Delta T
\]

\[
C^2 > 4Mk \quad \text{Over Damped}
\]

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

\[
C^2 = 4Mk \quad \text{Critically Damped}
\]

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

\[
C^2 < 4Mk \quad \text{Under Damped}
\]

\[
\delta x = c_3 e^{-\left( \frac{C}{2M} \right) t} \cos \left( \sqrt{4Mk - C^2} t - \varphi \right)
\]

\[
\tan(\varphi) = \frac{x'(0)}{k M} x(0) \sqrt{\frac{M}{k}}
\]

\[
c_3^2 = \left( x(0) \right)^2 + \frac{M}{k} x'(0)^2
\]
System State Variables
System Stage Models represent the system as a whole in terms of the hardware and software states that the system transitions through during operation.

Goal Function Tree (GFT) Model
- “Middle Out” model of the system based on the system State Variables
- Shows relationship between system state functions (hardware and software) and system goals
- Does not contain system physical or logical relationships and is not executable

System State Machine Model
- Models the integrated State Transitions of the system as a whole (i.e., hardware states and software states)
- Confirms system functions as expected
  - Checks for system hazardous, system anomalies, inconsistent state progression, missing states, improper state paths (e.g., short circuits in hardware and/or software design)
  - Confirms that the system states progress as stated in the system design
- Executable model of system
- 14% of R12 modeled
- Over 7,200 Transitions in the Vehicle and Software
- Over 3,500 States in the Vehicle
System Value
A System Value Model is a mathematical representation of Stakeholders Preferences (Expectations) for the system
• The basic structure is straightforward
• The sociology/psychology of representing the Preferences can be a challenge

The System Value Model is the Basis of System Validation!!!
• The Requirements and Design Models form the basis of System Verification
• The System Value Model forms the basis of System Validation

Constructing an SLS Value Model to compare to System Validation results
• Can expand to Integrated Stack with input from MPCV and GSDO

System Value model also provides basis for a measure of System Robustness
• How many mission types are supported by the system?
Mapping System Capability to Value

**Will it work?** (Reliability)

- Load Factors
- Shock Loads
- Payload Services
- Injection Accuracy

**What can it carry?**

- Load Factors
- Shock Loads
- Payload Volume
- Payload Services
- Injection Accuracy

**How expensive is it?**

- Production cost
- Launch cost
- etc.

---

<table>
<thead>
<tr>
<th>Mission A</th>
<th>Mission B</th>
<th>Mission C</th>
</tr>
</thead>
<tbody>
<tr>
<td>20,000 m/s dV required</td>
<td>15,000 m/s dV required</td>
<td>32,000 m/s dV required</td>
</tr>
<tr>
<td>Value = $50000 * m</td>
<td>Value = $30000 * m</td>
<td>Value = $80000 * m</td>
</tr>
<tr>
<td>Demand = 25% of total</td>
<td>Demand = 60% of total</td>
<td>Demand = 15% of total</td>
</tr>
</tbody>
</table>

---

**Missions Attempted**

**Missions Succeeded**

**Total Value Delivered by Launch Vehicle**

---

**Percentage of Total Missions**

- LEO: 40%
- GEO: 35%
- Luna: 15%
- NEO: 10%
- Mars: 5%
- Jupiter: 5%

**Payload avg. value delivered vs location**

- LEO: $120,000
- GEO: $110,000
- Luna: $140,000
- NEO: $160,000
- Mars: $180,000
- Jupiter: $200,000
Engineering Statistics
Applying Akaike Information Criteria (AIC) corrected (AICc) to assess sensor coverage for a system

\[
AICc(F) = -2 \left( I^K(F|G) \right) + 2K + \frac{2K(K+1)}{n-K-1}
\]

Two Views of Information Content

- AIC Information
  - Information is viewed as the number of meaningful parameters
    - Parameters with sufficient measurements to be reasonable estimates

- Fisher Information Matrix
  - Defines information as the matrix of partial second derivatives
    - Information is the amount of parameters with non zero values (so provides an indication of structure)
    - This value converges to a maximum as the number of parameters goes to infinity
    - Does not contain an optimum, always increases with added parameters

AIC/AICc has an adjustment factor to penalize sensor arrangements where:
number of sensors < 3x(number of measurements)

Provides an optimization tool for use with System Models
System Dynamics
Tools and Methodologies

• Tools and techniques have been developed using the System Dynamics methodology that make it possible to efficiently decompose complex systems and to quickly set-up and test models of system operation.

• Tools promote understanding through visual diagramming and modeling.
Understanding the systems integrating relationships provides an important advancement in the practice of systems engineering and contribution to the engineering of the system

• Provides a complete understanding of the system functions and interactions
  – Basis to define system GR&A in a way to have a closed set to begin design work
  – Basis of system closure criteria
  – Basis for identifying adjustments to the system function design solutions
  – Basis for determining optimal system performance

Provides a method to quickly compare system configurations and identify best balance result, reducing time necessary for DACs

Provides a method to more completely test software algorithms, reducing amount of real-time software testing

Analysis complements detailed design work done by the Engineering Disciplines
• System Exergy is an integrating relationship
  – Depends on results from each Engineering Discipline
• A positive for systems engineers in conducting system level design
• More difficulty to use (depends on results from each Engineering Discipline) for specific components of subsystems
Summary

◆ System Modeling is composed of several different model types to gain a complete understanding of the system
  • System Relational Modeling (i.e., MBSE)
  • System Integrating Physics
  • System Value Models
  • System Stave Variable Modeling
    – Goal Function Tree (GFT)
    – State Analysis Model (SAM)
  • System Statistical Modeling
  • System Dynamics Modeling

◆ These System Models provide the basic understanding of the system leading to:
  • Reduced development analysis cycle time
  • Reduced system software testing time
  • Better correlation of system capabilities with stakeholder expectations

◆ The results of the research conducted by all Consortium members is available on the NASA Portal
  • https://www.nasa.gov/consortium

    – NASA Technical Publication in work (Due out in October 2019)

  • “Engineering Elegant Systems: The Practice of Systems Engineering”
    – NASA Technical Publication in work (Due out in November 2019)
Backup
Motivation

◆ System Engineering of Complex Systems is not well understood

◆ System Engineering of Complex Systems is Challenging
  • System Engineering can produce elegant solutions in some instances
  • System Engineering can produce embarrassing failures in some instances
  • Within NASA, System Engineering does is frequently unable to maintain complex system designs within budget, schedule, and performance constraints

◆ “How do we Fix System Engineering?”
  • Michael D. Griffin, 61st International Astronautical Congress, Prague, Czech Republic, September 27-October 1, 2010
  • Successful practice in System Engineering is frequently based on the ability of the lead system engineer, rather than on the approach of system engineering in general
  • The rules and properties that govern complex systems are not well defined in order to define system elegance

◆ 4 characteristics of system elegance proposed as:
  • System Effectiveness
  • System Efficiency
  • System Robustness
  • Minimizing Unintended Consequences
Consortium

◆ **Research Process**
  - Multi-disciplinary research group that spans systems engineering areas
  - Selected researchers who are product rather than process focused

◆ **List of Consortium Members**
  - Michael D. Griffin, Ph.D.
  - Air Force Research Laboratory – Wright Patterson, Multidisciplinary Science and Technology Center: Jose A. Camberos, Ph.D., Kirk L. Yerkes, Ph.D.
  - Doty Consulting Services: John Doty, Ph.D.
  - George Washington University: Zoe Szajnfarber, Ph.D.
  - Iowa State University: Christina L. Bloebaum, Ph.D., Michael C. Dorneich, Ph.D.
  - Missouri University of Science & Technology: David Riggins, Ph.D.
  - NASA Langley Research Center: Peter A. Parker, Ph.D.
  - Texas A&M University: Richard Malak, Ph.D.
  - Tri-Vector Corporation: Joey Shelton, Ph.D., Robert S. Ryan, Kenny Mitchell
  - 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., George Nelson, Ph.D.
  - The University of Colorado – Colorado Springs: Stephen B. Johnson, Ph.D.
  - The University of Michigan: Panos Y. Papalambros, Ph.D.
  - The University of Texas, Arlington: Paul Componation, Ph.D.
  - The University of Bergen: Erika Palmer

◆ **Previous Consortium Members**
  - Massachusetts Institute of Technology: Maria C. Yang, Ph.D.
  - Stevens Institute of Technology – Dinesh Verma
  - Spaceworks – John Olds (Cost Modeling Statistics)
  - Alabama A&M – Emeka Dunu (Supply Chain Management)
  - George Mason – John Gero (Agent Based Modeling)
  - Oregon State – Irem Tumer (Electrical Power Grid Robustness)
  - Arkansas – David Jensen (Failure Categorization)

~50 graduate students and 15 undergraduate students supported to date
Definition – System Engineering is the engineering discipline which integrates the system functions, system environment, and the engineering disciplines necessary to produce and/or operate an elegant system.

- Elegant System - A system that is robust in application, fully meeting specified and adumbrated intent, is well structured, and is graceful in operation.

Primary Focus
- System Design and Integration
  - Identify system couplings and interactions
  - Identify system uncertainties and sensitivities
  - Identify emergent properties
  - Manage the effectiveness of the system
- Engineering Discipline Integration
  - Manage flow of information for system development and/or operations
  - Maintain system activities within budget and schedule

Supporting Activities
- Process application and execution
  - Processes organize the engineering
### Systems Engineering Postulates

- **Postulate 1**: Systems engineering is system specific and context dependent in application.
- **Postulate 2**: The Systems Engineering domain consists of subsystems, their interactions among themselves, and their interactions with the system environment.
- **Postulate 3**: The function of Systems Engineering is to integrate engineering disciplines in an elegant manner.
- **Postulate 4**: Systems engineering influences and is influenced by organizational structure and culture.
- **Postulate 5**: Systems engineering influences and is influenced by budget, schedule, policy, and law.
- **Postulate 6**: Systems engineering spans the entire system life-cycle.
- **Postulate 7**: Understanding of the system evolves as the system development or operation progresses.
- **Postulate 7 Corollary**: Understanding of the system degrades during operations if system understanding is not maintained.
Systems Engineering Principles

- **Principle 1:** Systems engineering integrates the system and the disciplines considering the budget and schedule constraints
- **Principle 2:** Complex Systems build Complex Systems

- **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
  - Sub-Principle 3(a): Mission context is defined based on understanding of the stakeholder needs and constraints
  - Sub-Principle 3(b): Requirements and models reflect the understanding of the system
  - Sub-Principle 3(c): Requirements are specific, agreed to preferences by the developing organization
  - Sub-Principle 3(d): Requirements and design are progressively elaborated as the development progresses
  - Sub-Principle 3(e): Hierarchical structures are not sufficient to fully model system interactions and couplings
  - Sub-Principle 3(f): A Product Breakdown Structure (PBS) provides a structure to integrate cost and schedule with system functions
  - Sub-Principle 3(g): As the system progresses through development, a deeper understanding of the organizational relationships needed to develop the system are gained.
  - Sub-Principle 3(h): Systems engineering achieves an understanding of the system’s value to the system stakeholders
  - Sub-Principle 3(i): Systems engineering seeks a best balance of functions and interactions within the system budget, schedule, technical, and other expectations and constraints.
Principle 4: Systems engineering has a critical role through the entire system life-cycle
- Sub-Principle 4(a): Systems engineering obtains an understanding of the system
- Sub-Principle 4(b): Systems engineering defines the mission context (system application)
- Sub-Principle 4(c): Systems engineering models the system
- Sub-Principle 4(d): Systems engineering designs and analyzes the system
- Sub-Principle 4(e): Systems engineering tests the system
- Sub-Principle 4(f): Systems engineering has an essential role in the assembly and manufacturing of the system
- Sub-Principle 4(g): Systems engineering has an essential role during operations, maintenance, and decommissioning

Principle 5: Systems engineering is based on a middle range set of theories
- Sub-Principle 5(a): Systems engineering has a physical/logical basis specific to the system
- Sub-Principle 5(b): Systems engineering has a mathematical basis
- Sub-Principle 5(c): Systems engineering has a sociological basis specific to the organization(s)

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

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

Principle 8: Both Policy and Law must be properly understood to not overly constrain or under constrain the system implementation
| Principle 9: Systems engineering decisions are made under uncertainty accounting for risk |
| Principle 10: Verification is a demonstrated understanding of all the system functions and interactions in the operational environment |
| Principle 11: Validation is a demonstrated understanding of the system’s value to the system stakeholders |
| Principle 12: Systems engineering solutions are constrained based on the decision timeframe for the system need |
| Principle 13: Stakeholder expectations change with advancement in technology and understanding of system application. |
| Principle 14: The real physical system is the perfect model of the system  |
| • Kullback-Liebler Information shows the actual system is the ideal information representation of the system  |
| \[ I(f, g) = \int f(x)\log(f(x)) \, dx - \int f(x)\log(g(x|\theta)) \, dx = 0 \] |
System Engineering 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
  • Hamilton’s Principle shows this for a physical system
    \[-\int_{t_1}^{t_2} (\delta T - \delta V + \delta W) dt = 0\]

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

◆ Hypothesis 3: Key Stakeholders preferences can be accurately represented mathematically
Consortium identified the significance of understanding and using the System Integrating Physics for Systems Engineering

• First Postulate: Systems engineering is system specific and context dependent.
  – Systems are different, and therefore, the integrating physics for the various systems is different

• Second Postulate: The Systems Engineering domain consists of subsystems, their interactions among themselves, and their interactions with the system environment
  – System interactions among properly defined system functions and with the environment are the basis of systems engineering

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

• Sub-Principle (5a): Systems engineering has a physical/logical basis specific to the system
  – The physics of the specific systems defines the integration relationships

• Principle 7: Decision quality depends on system knowledge present in the decision-making process
  – Understanding of system interactions must be included

• Principle 12: Systems engineering solutions are constrained based on the decision timeframe for the system need
  – Understanding the system interactions shortens the development time and opens design space more for a given timeframe
Methods of System Integration

Goal: System Design and Analysis
System Models Contain an Understanding of the System

- Allow systems engineers to:
  - Define system functions based on the system state variables
  - Understand stakeholders expectations on system value (i.e., capabilities)
  - Integrate discipline engineering models into a system level physics based model (e.g., system exergy)
  - Design and Analyze system responses and behaviors at the System level

- Engineering Statistics
- Goal Function Tree (GFT)
- Goals
- System Functions & State Variables
- Discipline Physics Models
- System Integrated Physics Model (System Exergy)
- System State Transition Model
- Multidisciplinary Design Optimization (MDO)

- Tools:
  - MagicDraw Enterprise (SysML)
  - Matlab
  - Matlab StateFlow
  - Microsoft Excel
System Design and Integration

- 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
- System Design and Analysis Models

Operations and Production