Simulation and Modeling for Systems Engineering

SYS-611: Simulation and Modeling

Paul T. Grogan, Ph.D.
Assistant Professor
School of Systems and Enterprises

Agenda

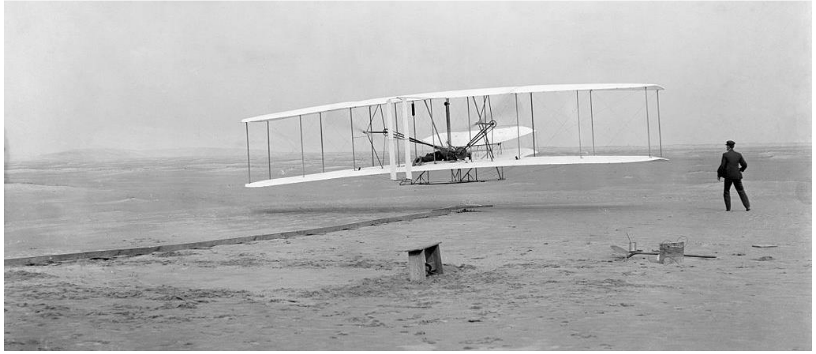
1. Modeling in Systems Engineering
2. Modeling and Simulation Process
3. Example Application: SpaceNet

Reading: J.V. Farr, “Overview of Modeling and Simulation of Complex Systems,” Ch. 1 in *Simulation of Complex Systems and Enterprises,* Stevens Institute of Technology, 2007.

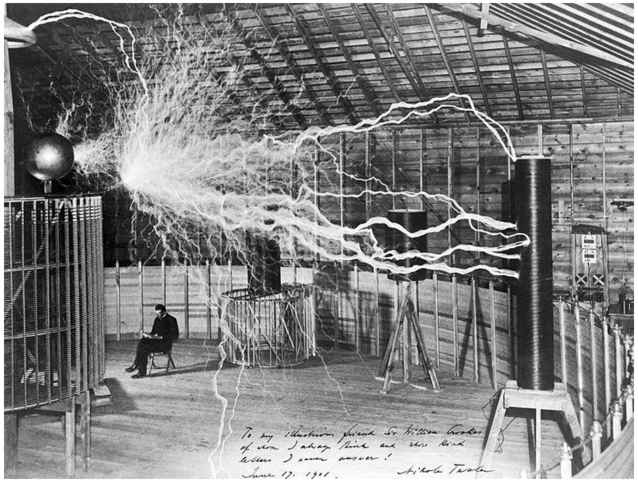
A. Maria, “Introduction to Modeling and Simulation,” in *Proceedings of the 1997 Winter Simulation Conference*, 1997.

Modeling in Systems Engineering

Engineering 100+ Years Ago



[Wright Brothers at Kitty Hawk NC (1902)]



[Tesla in Colorado Springs CO (1899)]

[Bell in New York NY (1892)]

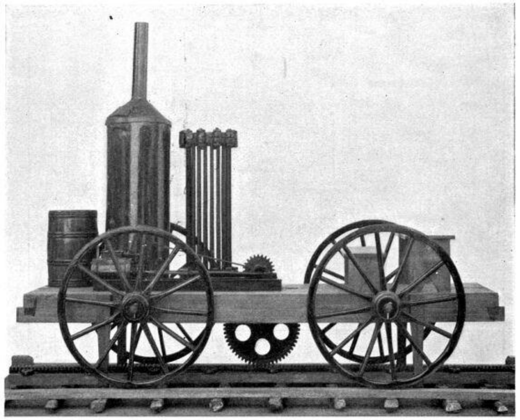


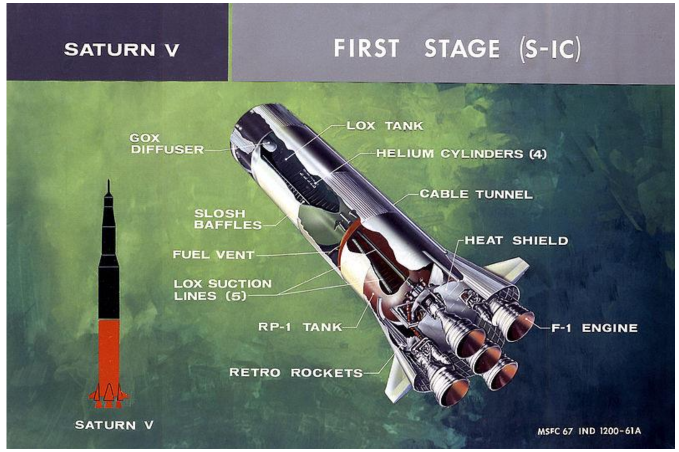
FIG. 1.—STEVENS' RACK RAIL ENGINE, 1825—FIRST ENGINE TO CARRY PASSENGERS ON A TRACK IN THE UNITED STATES.

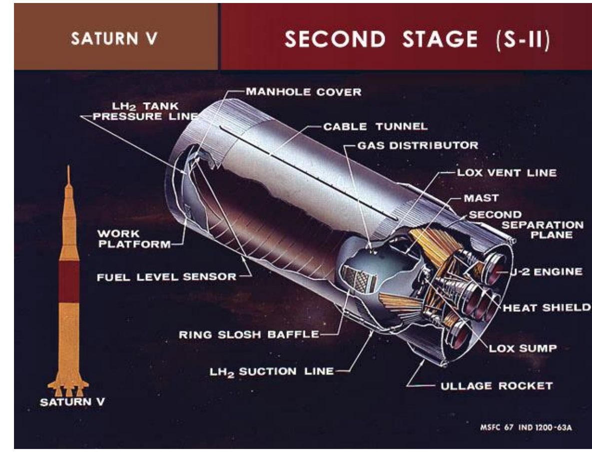
[Stevens Rack Rail Engine (1825)]

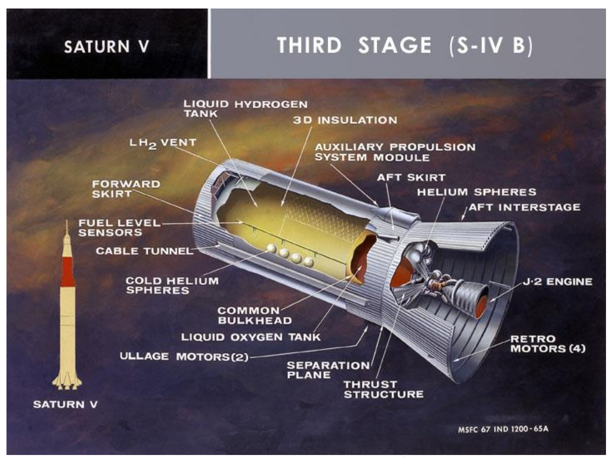
Engineering 50+ Years Ago

[Apollo 11 (1969)]

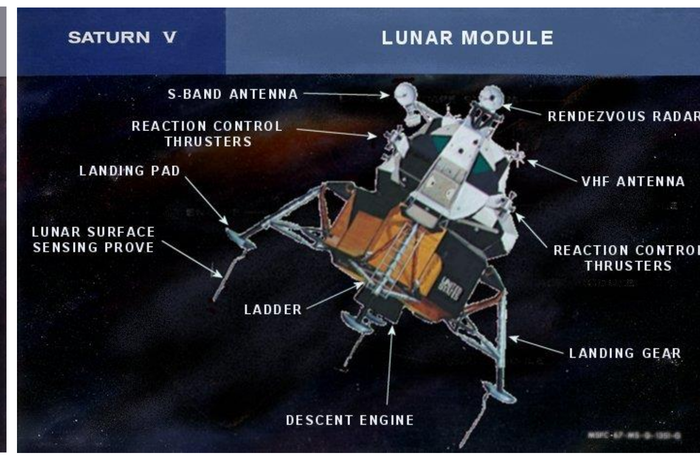
BOEING







DOUGLAS



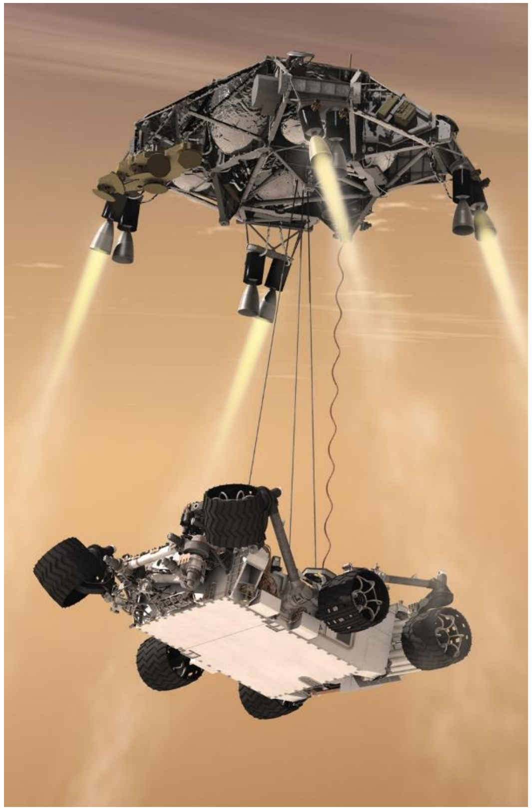
GRUMMAN

Engineering Today

[Formula E Racecar (2013)]

[International Space Station (2011)]

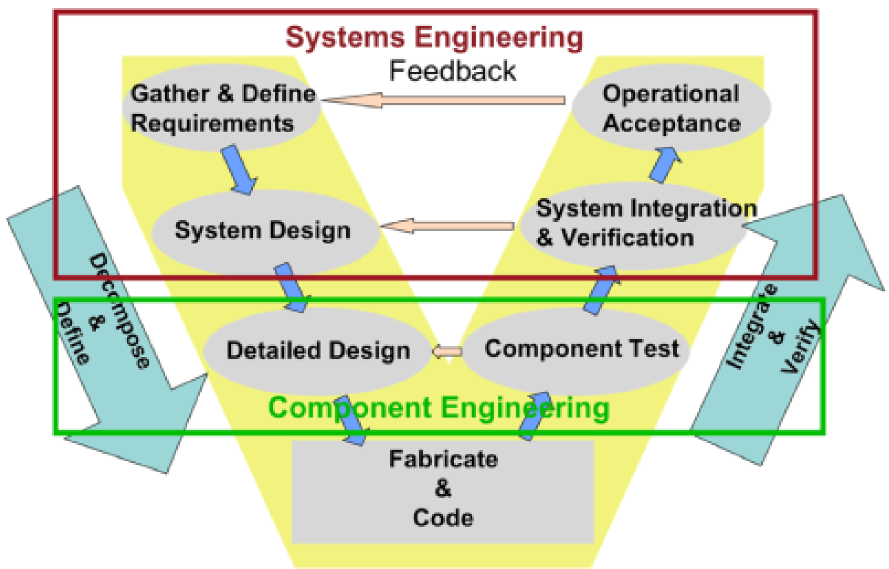
[Wind Turbines (2005)]



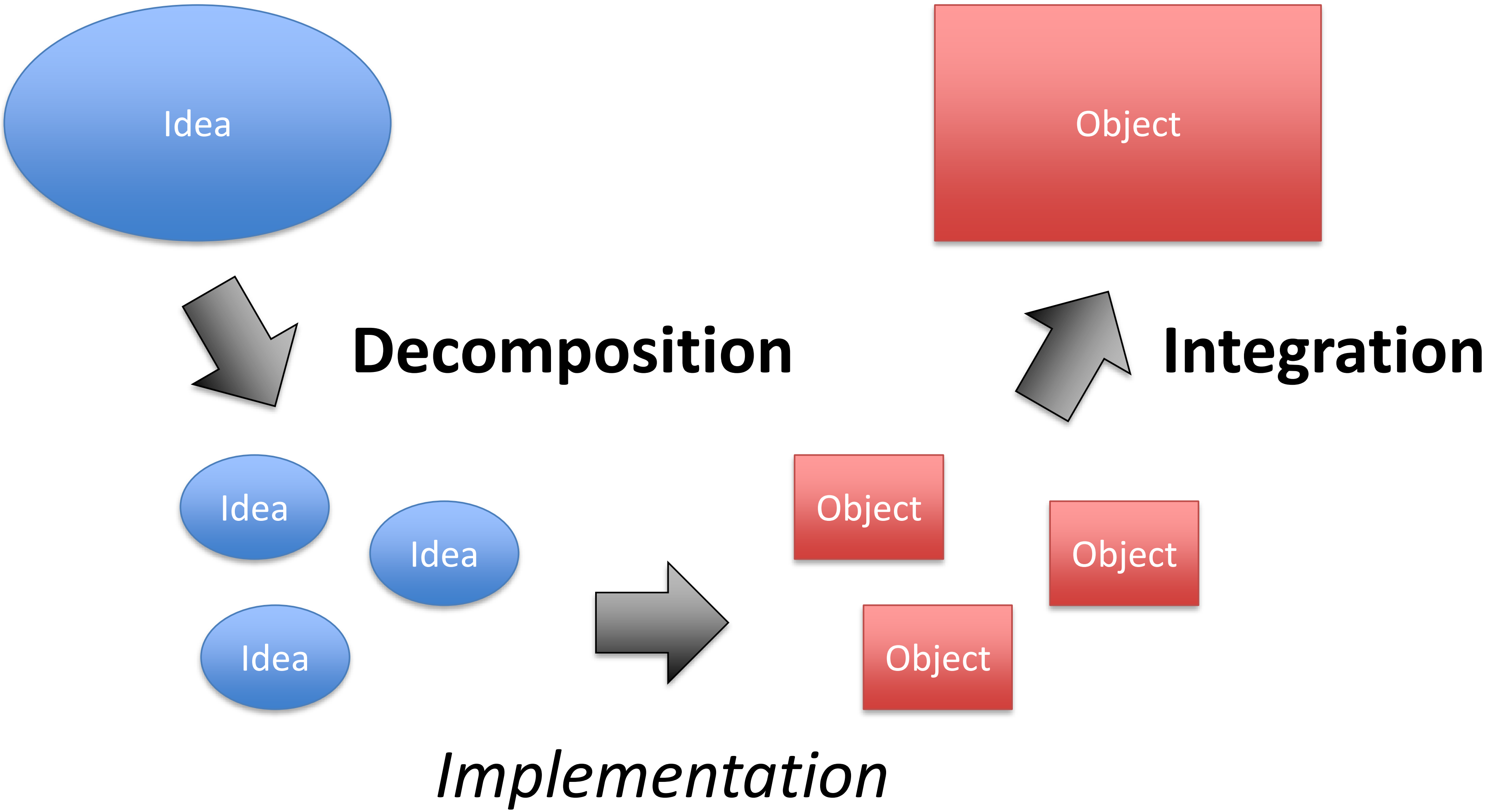
[Curiosity Rover (2014)]

Role of Systems Engineering

"**Systems Engineering** is a methodical, disciplined approach for the design, realization, technical management, operations, and retirement of a system." (NASA SE Handbook 2007:21)

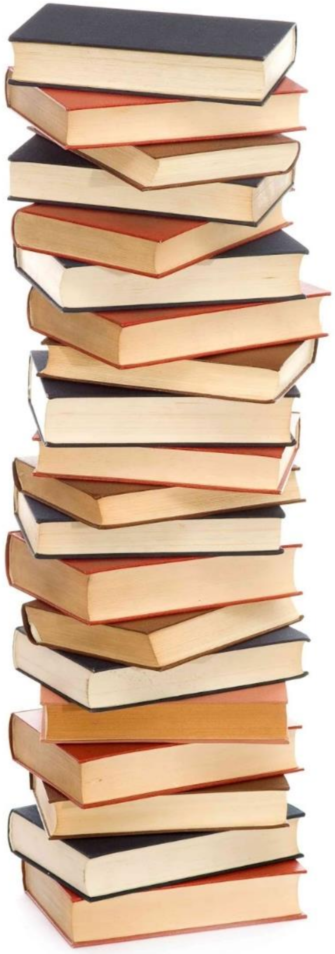


Systems Engineering Activities



Systems Engineering Artifacts

- Top-level Requirements and Expectations
- Concept of Operations
- Technical Requirements
- System Architecture
- System/Subsystem Specifications
- Interface Specifications
- Verification and Validation Plans
- Logistics and Operations Procedures
- End Product Documentation



Model-based Systems Engr.

"**Model-based Systems Engineering (MBSE)** is the formalized application of modeling to support system requirements, design, analysis, verification, and validation activities … throughout development and later lifecycle phases" (INCOSE *Systems Engineering Vision 2020*)

Analogous to the use of CAD/CAM in electrical and mechanical engineering disciplines

Models in Systems Engineering

Process Standards	EIA 632: Processes for Engineering a System	IEEE 1220: Application and Management of the SE Process	ISO 15288: Systems and Software Engineering – Life Cycle Processes	
Architectural Frameworks	DoD Architecture Framework (DoDAF)	MOD Architecture Framework (MODAF)	ISO 42010 Systems And Software Engineering Architecture	
Modeling and Simulation Standards	ICAM Definition for Functional Modeling (IDEF0)	Systems Modeling Language (SysML)	Web Ontology Language (OWL)	High Level Architecture (HLA)
Metamodeling and Data Exchange Standards	Meta-Object Facility (MOF)	XML Metadata Interchange (XMI)	Query View Transform (QVT)	Standards for Exchange of Product Model Data (STEP)

M&S in Systems Engineering

M&S for System Design
(anticipate performance)

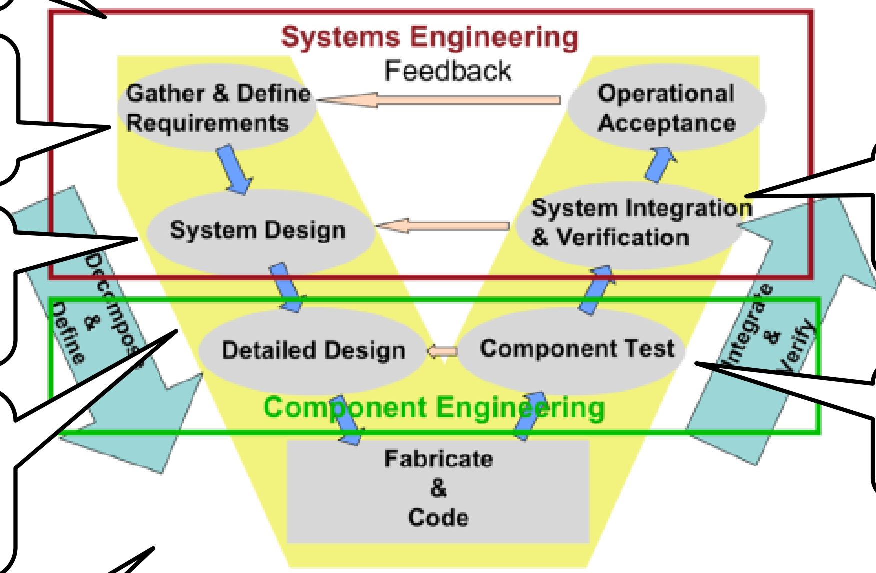
What are the fundamental needs?

What requirements address the needs?

What system design meets requirements?

What are the imposed component requirements?

What component design meets requirements?



M&S for System Analysis
(verify requirements in simulated environment)

Do systems meet requirements?

Do components meet requirements?

Overall goal: anticipate down-stream effects earlier and at lower cost

Advantages of Simulation

1. Natural representation and intuitive form
2. Amenable to complex, real-world situations
3. Rapid calculations on digital computers
4. Lower cost than physical models (sometimes)
5. Virtual analysis processes (what-if, sensitivity)
6. Repeated use of existing models (sometimes)

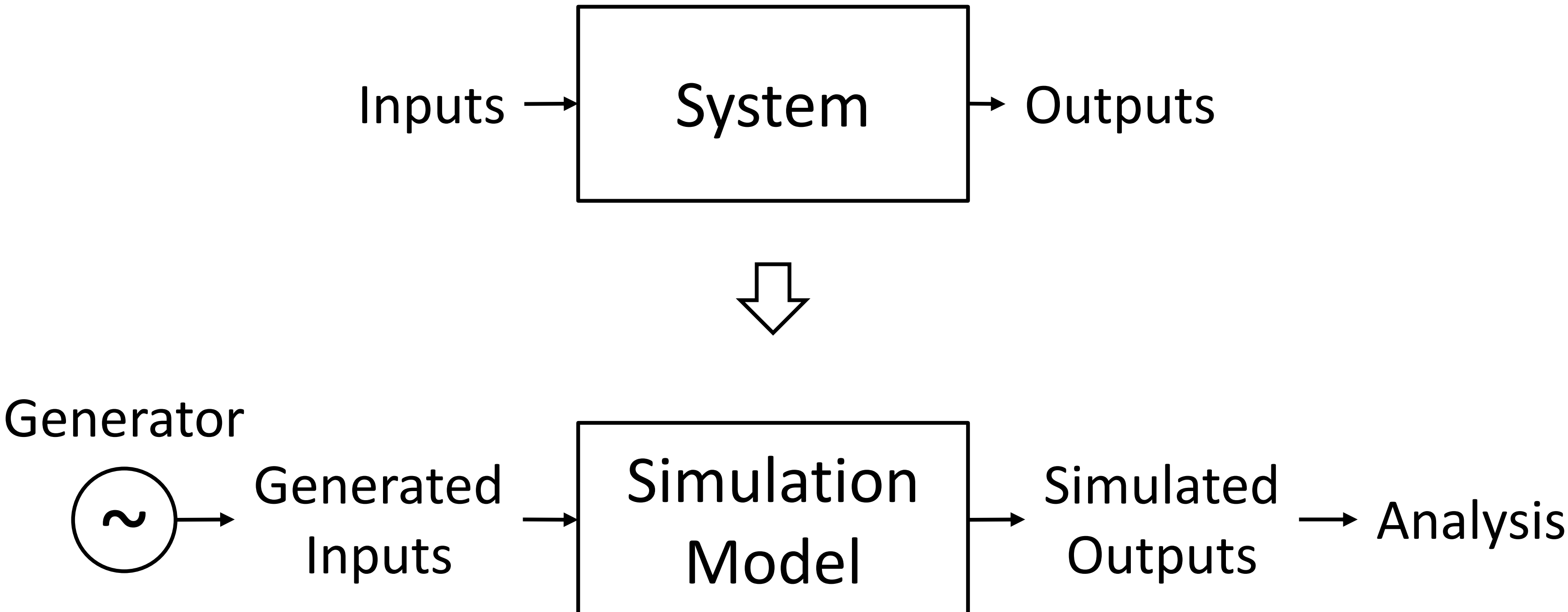
Disadvantages of Simulation

1. Results do not solve problem (no optimization)
2. Does not contribute to fundamental theory
3. Can be abused to avoid physical testing
4. High human costs during development
5. Results limited to underlying assumptions

Based on A. Maria, “Introduction to Modeling and Simulation,” *Proceedings of the 1997 Winter Simulation Conference*, 1997.

Modeling and Simulation Process

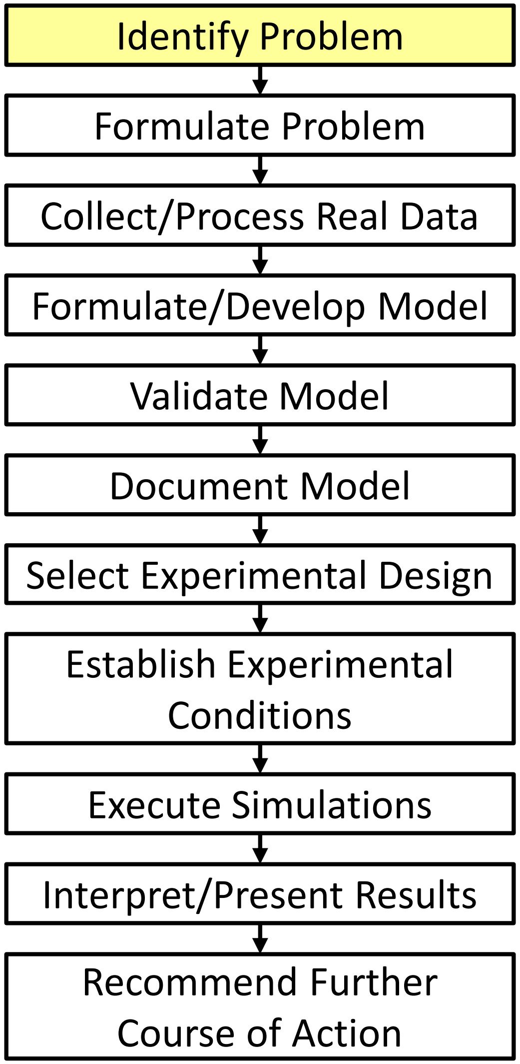
System Simulation Model



- Historical
- Estimated
- Hypothesized
- Random

- Decision
- Experiment
- Hypothesis Test
- Entertainment

1. Identify Problem



- Do not start developing a model without a clear problem
- The problem context is critical to inform assumptions and limitations
- “All models are wrong, but some are useful” (G. Box)

A. Maria, “Introduction to Modeling and Simulation,” *Proceedings of the 1997 Winter Simulation Conference*, 1997.

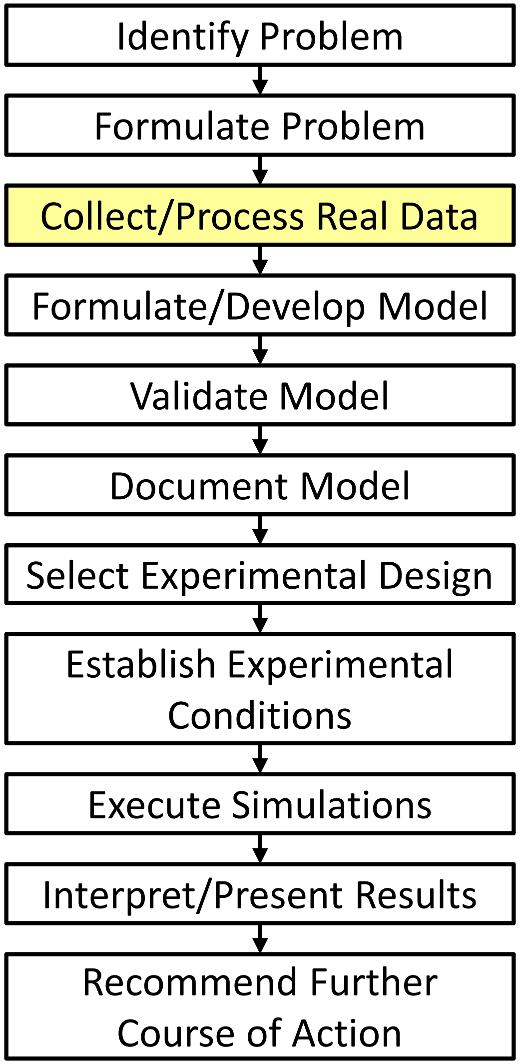
2. Formulate Problem

- Identify Problem
- **Formulate Problem**
- Collect/Process Real Data
- Formulate/Develop Model
- Validate Model
- Document Model
- Select Experimental Design
- Establish Experimental Conditions
- Execute Simulations
- Interpret/Present Results
- Recommend Further Course of Action

- Define the study objectives
- Draw the system boundary
  - Scope and scale (spatial/temporal)
- Identify performance measures
  - Quantitative criteria
  - Hypothesize candidate designs

A. Maria, "Introduction to Modeling and Simulation," *Proceedings of the 1997 Winter Simulation Conference*, 1997.

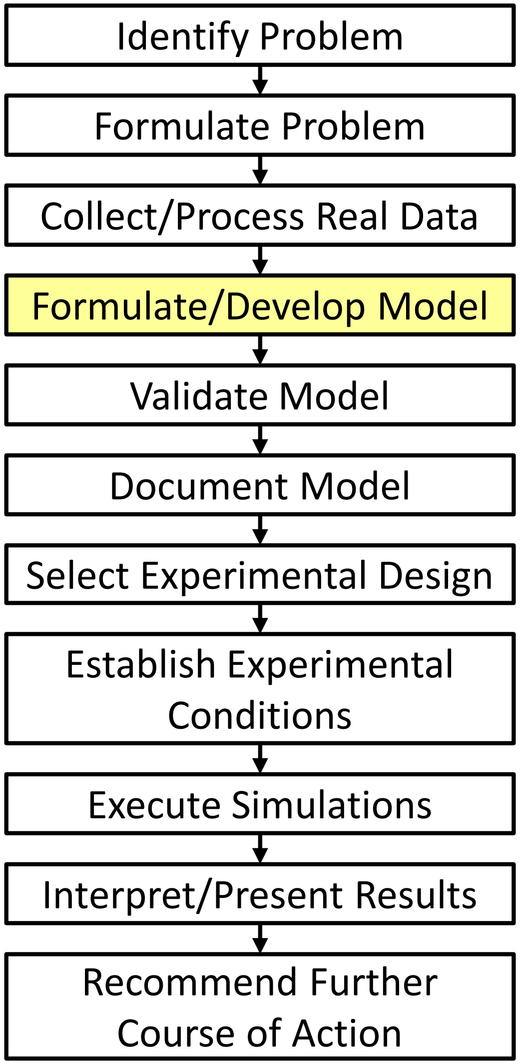
3. Collect Real Data



- Source of future validation
- Challenging for design where, by definition, there is no existing data
- Identify sources of uncertainty
  - Aleatory vs. epistemic
  - Characterize with distributions
  - Perform goodness-of-fit tests

A. Maria, "Introduction to Modeling and Simulation," *Proceedings of the 1997 Winter Simulation Conference*, 1997.

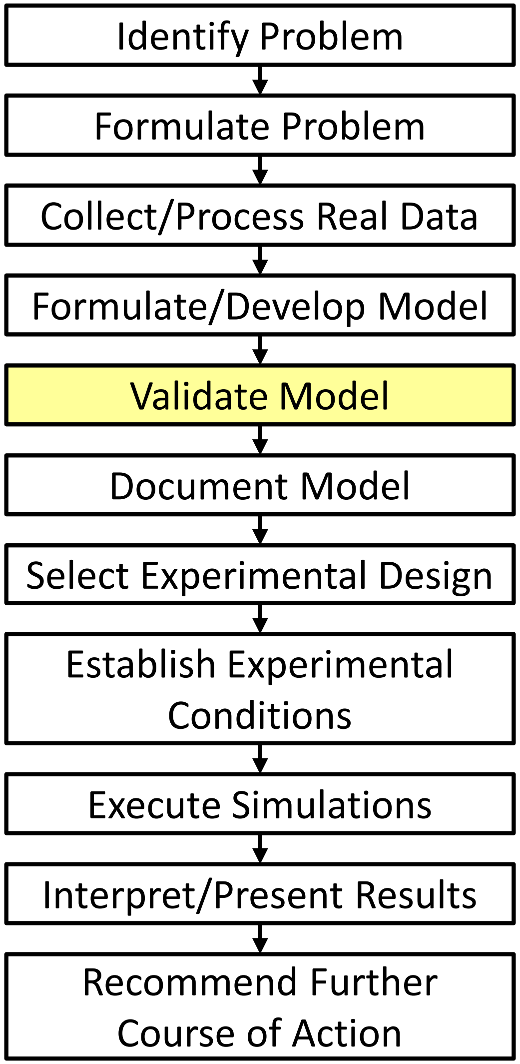
4. Formulate/Develop Model



- Create conceptual models of structure and behavior
- Choose appropriate formalism
- Implement in simulation software
- Verify simulation produces expected results

A. Maria, "Introduction to Modeling and Simulation," *Proceedings of the 1997 Winter Simulation Conference*, 1997.

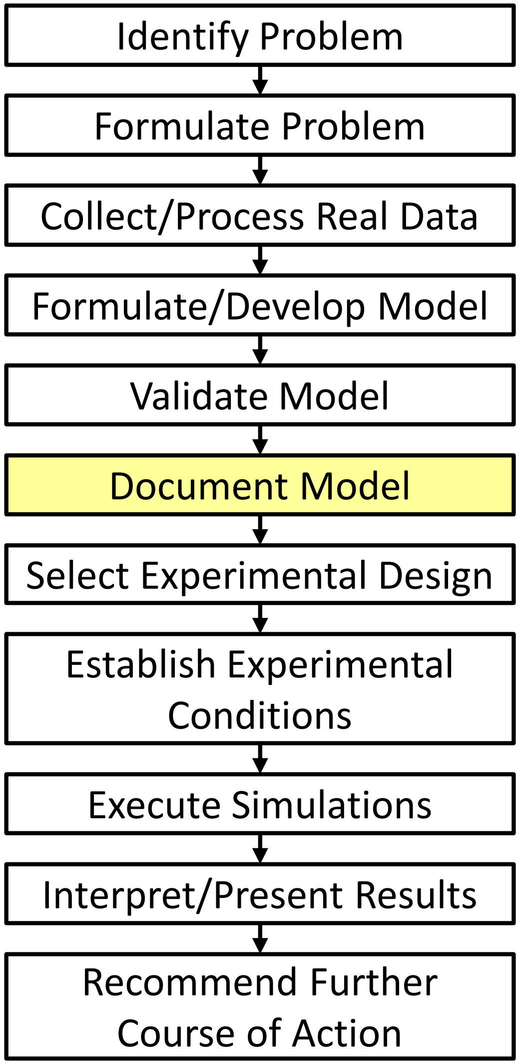
5. Validate Model



- Compare model output under known conditions to real data
- Perform statistical inference tests of sample data sets
- Build credibility with users and decision-makers

A. Maria, “Introduction to Modeling and Simulation,” *Proceedings of the 1997 Winter Simulation Conference*, 1997.

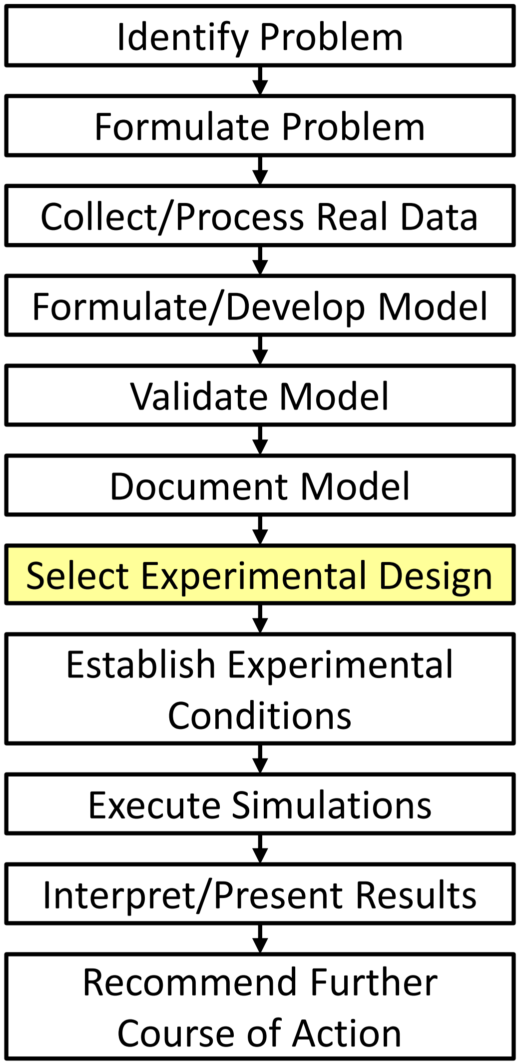
6. Document Model



- Describe all model objectives, assumptions, and limitations
- Consider metadata standards for data types, formats, etc.
  - Be cautious of implicit or tacit knowledge in inputs/outputs
- Provide instructions for execution

A. Maria, "Introduction to Modeling and Simulation," *Proceedings of the 1997 Winter Simulation Conference*, 1997.

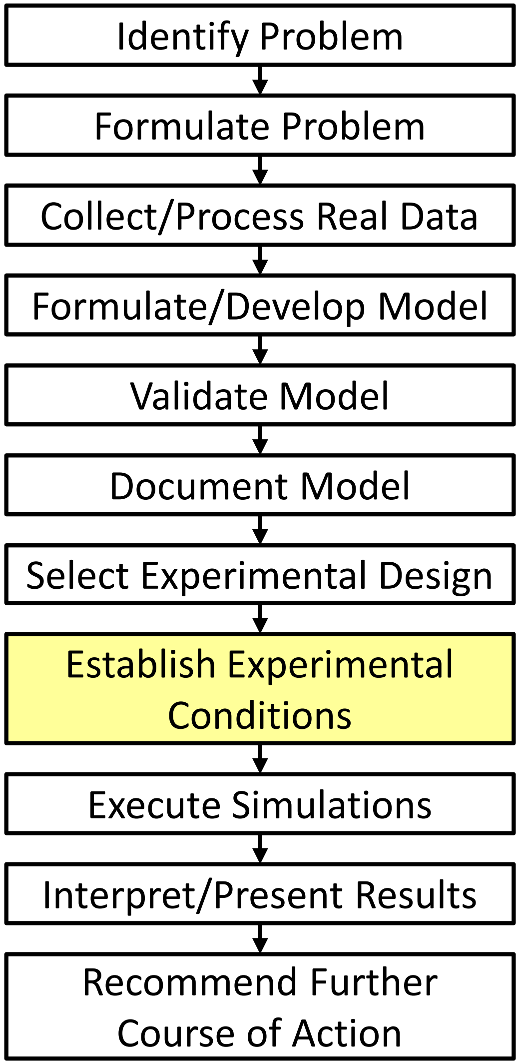
7. Select Experiment Design



- Naïve experimental design will often succumb to combinatorial issues of tractability
- Select variables likely to influence a key performance measure
- Consider sampling and variance reduction techniques

A. Maria, "Introduction to Modeling and Simulation," *Proceedings of the 1997 Winter Simulation Conference*, 1997.

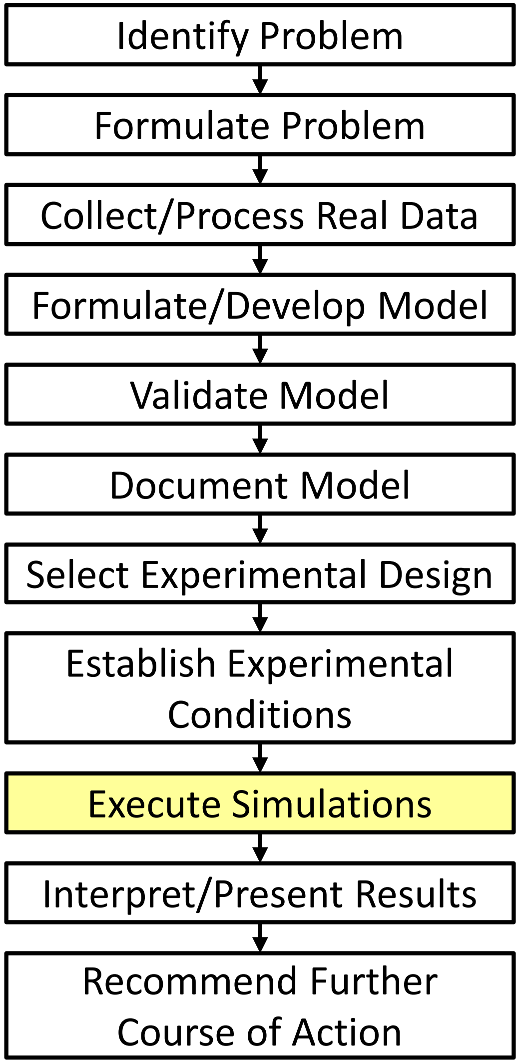
8. Experiment Conditions



- Appropriate starting conditions
  - Warm-up periods
  - Random number streams
- Steady-state vs transient response
- Convergence criteria

A. Maria, "Introduction to Modeling and Simulation," *Proceedings of the 1997 Winter Simulation Conference*, 1997.

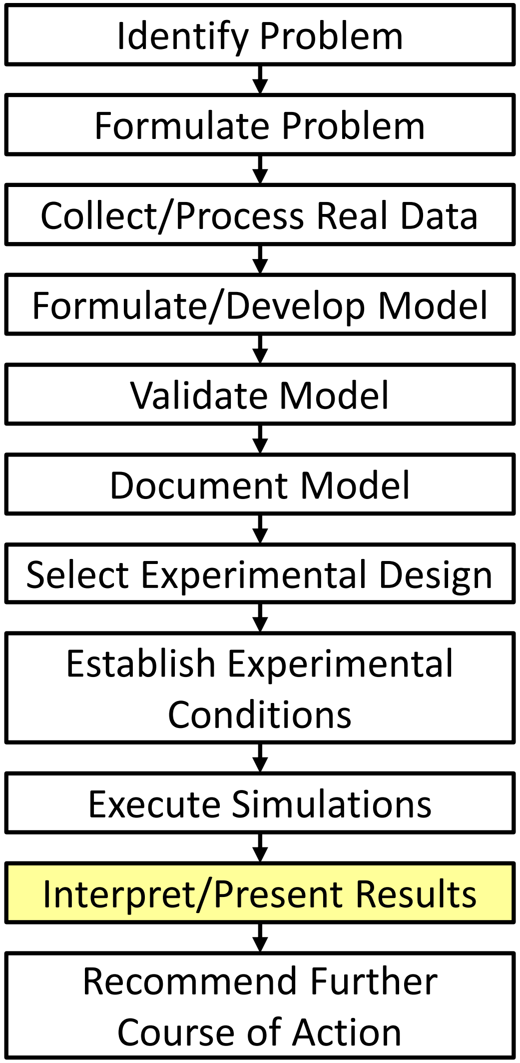
9. Execute Simulations



- Collect and record data for future analysis
- Consider fail-safe mechanisms to preserve data for long-running experiments

A. Maria, "Introduction to Modeling and Simulation," *Proceedings of the 1997 Winter Simulation Conference*, 1997.

10. Interpret/Present Results



- Perform statistical analysis of results using appropriate measures (mean/median/max)
- Report confidence intervals
- Use graphical displays to communicate quantitative results

A. Maria, "Introduction to Modeling and Simulation," *Proceedings of the 1997 Winter Simulation Conference*, 1997.

11. Recommend Next Action

- Identify Problem
- Formulate Problem
- Collect/Process Real Data
- Formulate/Develop Model
- Validate Model
- Document Model
- Select Experimental Design
- Establish Experimental Conditions
- Execute Simulations
- Interpret/Present Results
- Recommend Further Course of Action

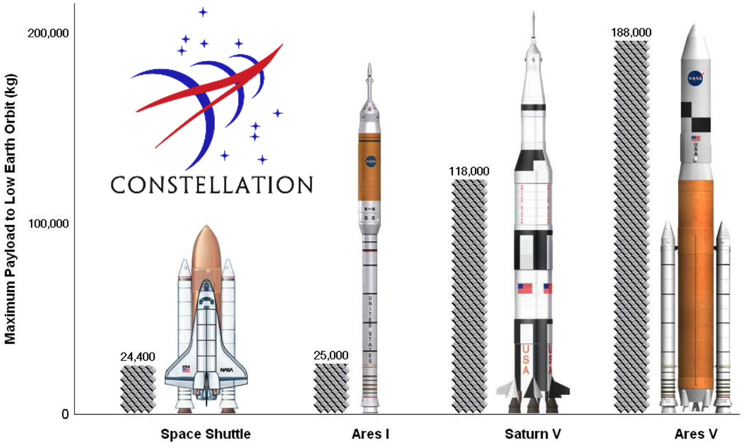
- Additional experiments
- Model refinements/extensions to improve precision or remove bias
- Decisions to address problem

A. Maria, "Introduction to Modeling and Simulation," *Proceedings of the 1997 Winter Simulation Conference*, 1997.

Example Application: SpaceNet

Application Case: SpaceNet

- Developed by MIT Strategic Engineering during NASA's Constellation Program
- Evaluate key performance indicators (KPIs) for human space exploration campaigns
  - Compare alternative campaign architectures
  - Focus on logistics of transporting/utilizing resources
  - Only consider deterministic scenarios
- Create user-friendly graphical interface



http://github.com/ptgrogan/spacenet

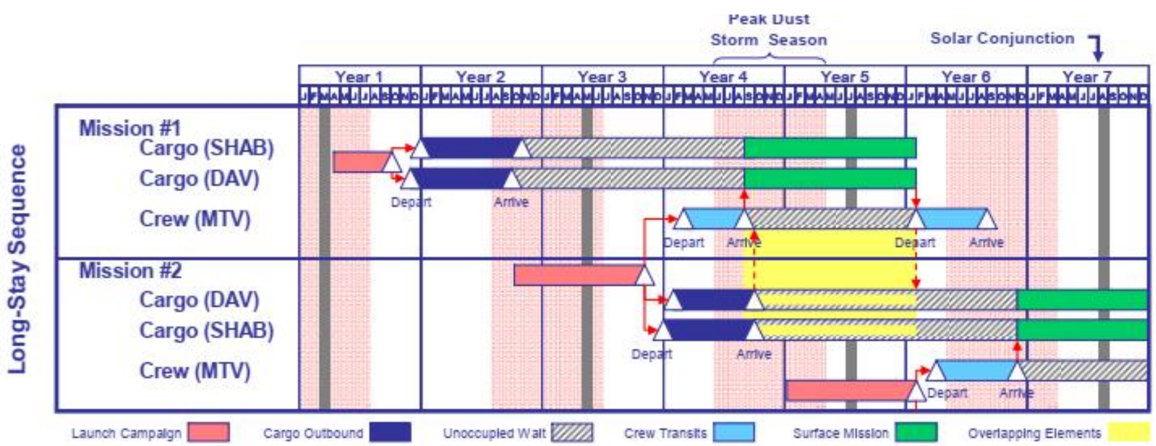
1. Identify Problem

- Challenges in long-duration human spaceflight:
  - Critical need for resources (food, water, oxygen)
  - High cost of supply ($2000-$20000/kg to orbit)
  - Long delivery time (weeks-to-years)
- Ensure realistic planning
- Help manifest cargo flights
- Assess the "value" of a proposed mission early on to compare alternatives

Image Source: NASA/JPL

2. Formulate Problem

- Model movement of people, systems, supplies between major exploration sites
- Integrate resource demands over time (days-to-months timescale)



Mars Design Reference Architecture 5.0

Mission Sequence Timeline (NASA)

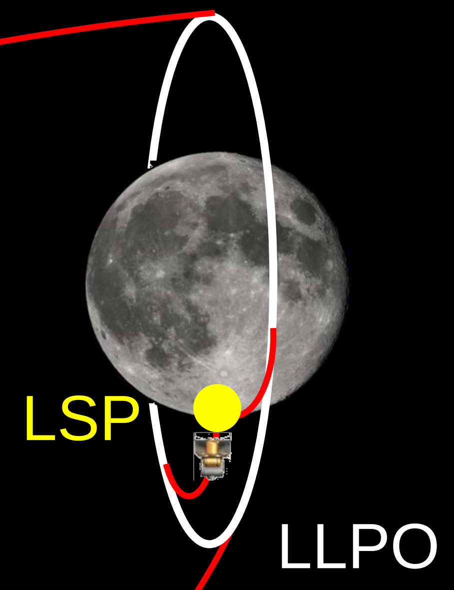
- Key performance measure: exploration capability

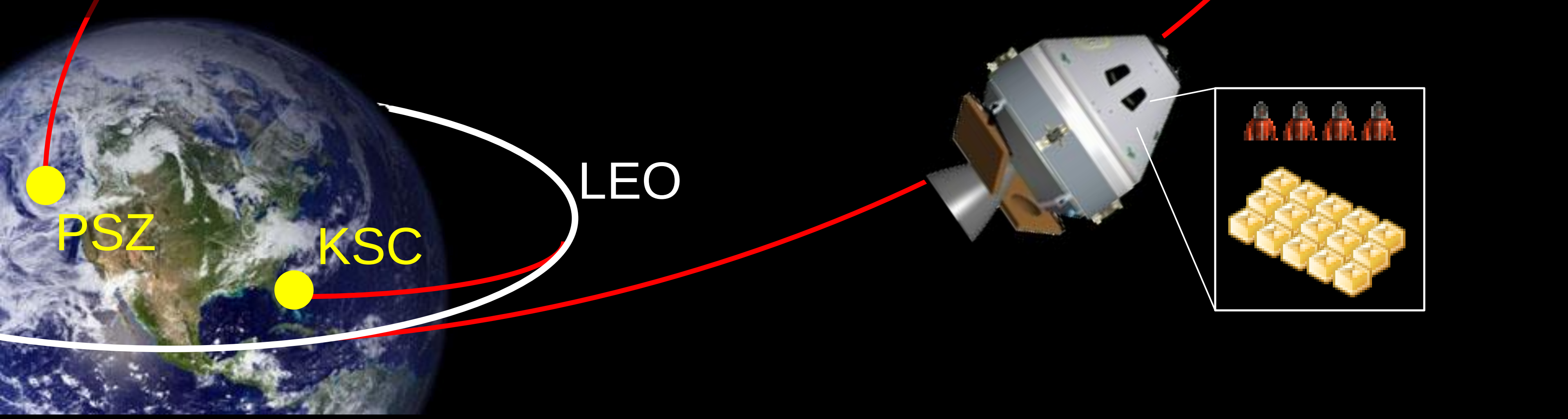
3. Collect Real Data

- Information on past and planned system elements:
  - Exploration sites and trajectories
  - Launch vehicle capacities (fuel, cargo)
  - Entity characteristics (size, mass, cargo, etc.)
  - Demands: crew metabolism, failure rates, etc.
- Information on past and planned missions:
  - Apollo 17
  - Constellation Lunar Sortie
  - Mars Design Reference Architecture 5.0

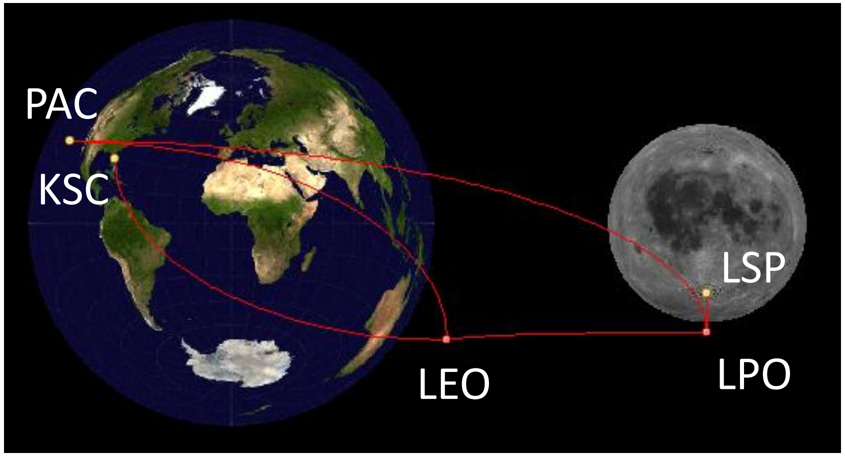
4. Formulate Model

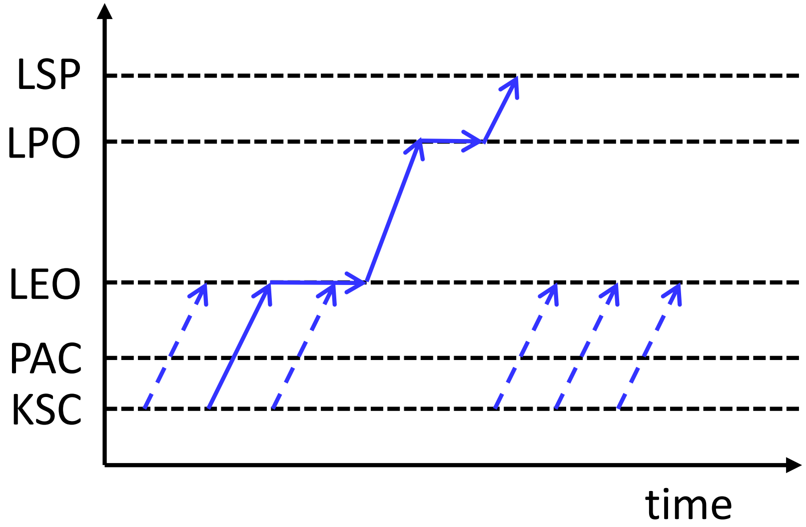
- Network Model (time-invariant)
  - Nodes and edges
- Element and Resource Models
  - 10 functional classes of supply (COS)
  - Element hierarchy with demands
- Simulation Events
  - 7 core events + composite events





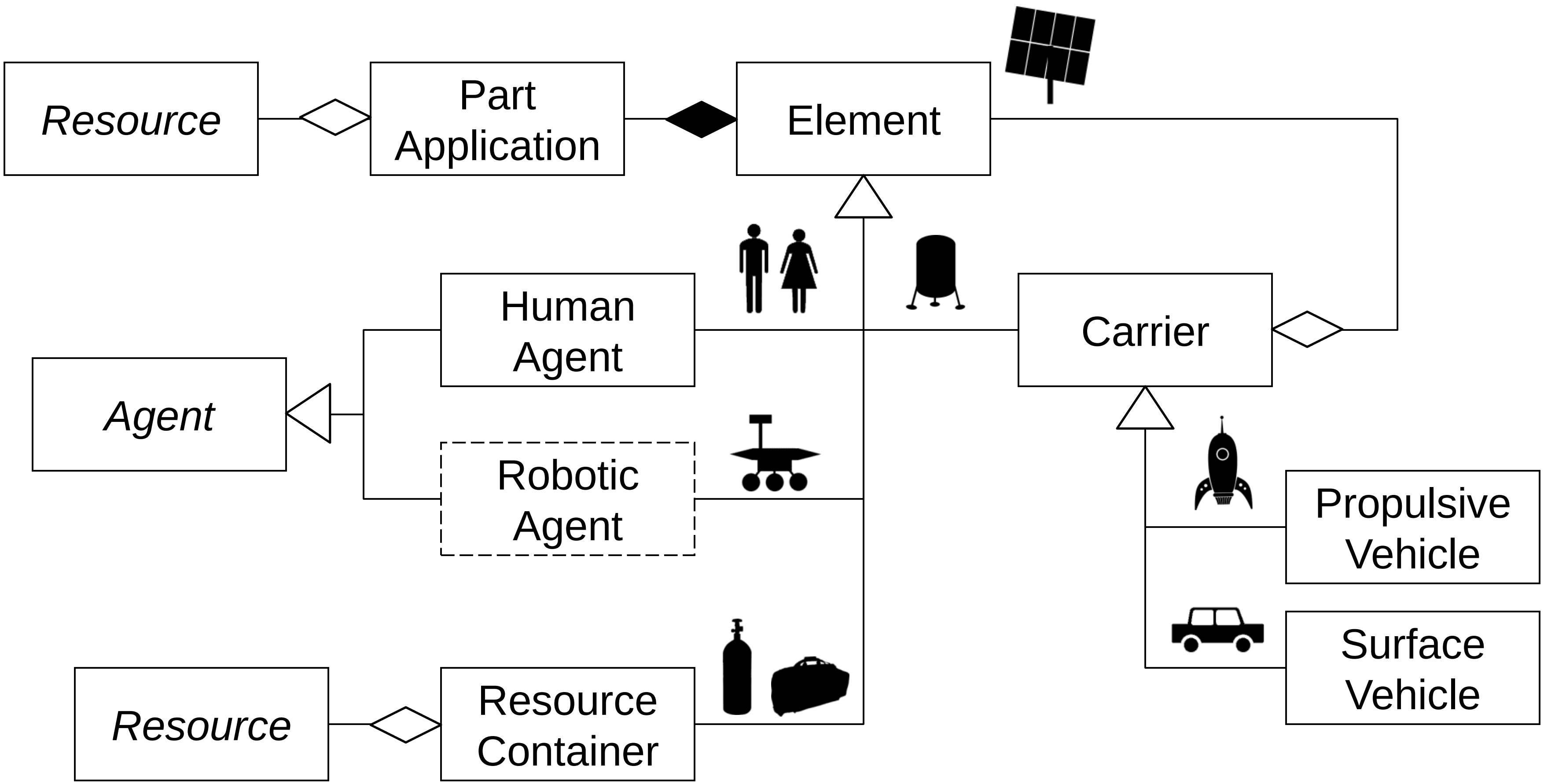
Time-expanded Network



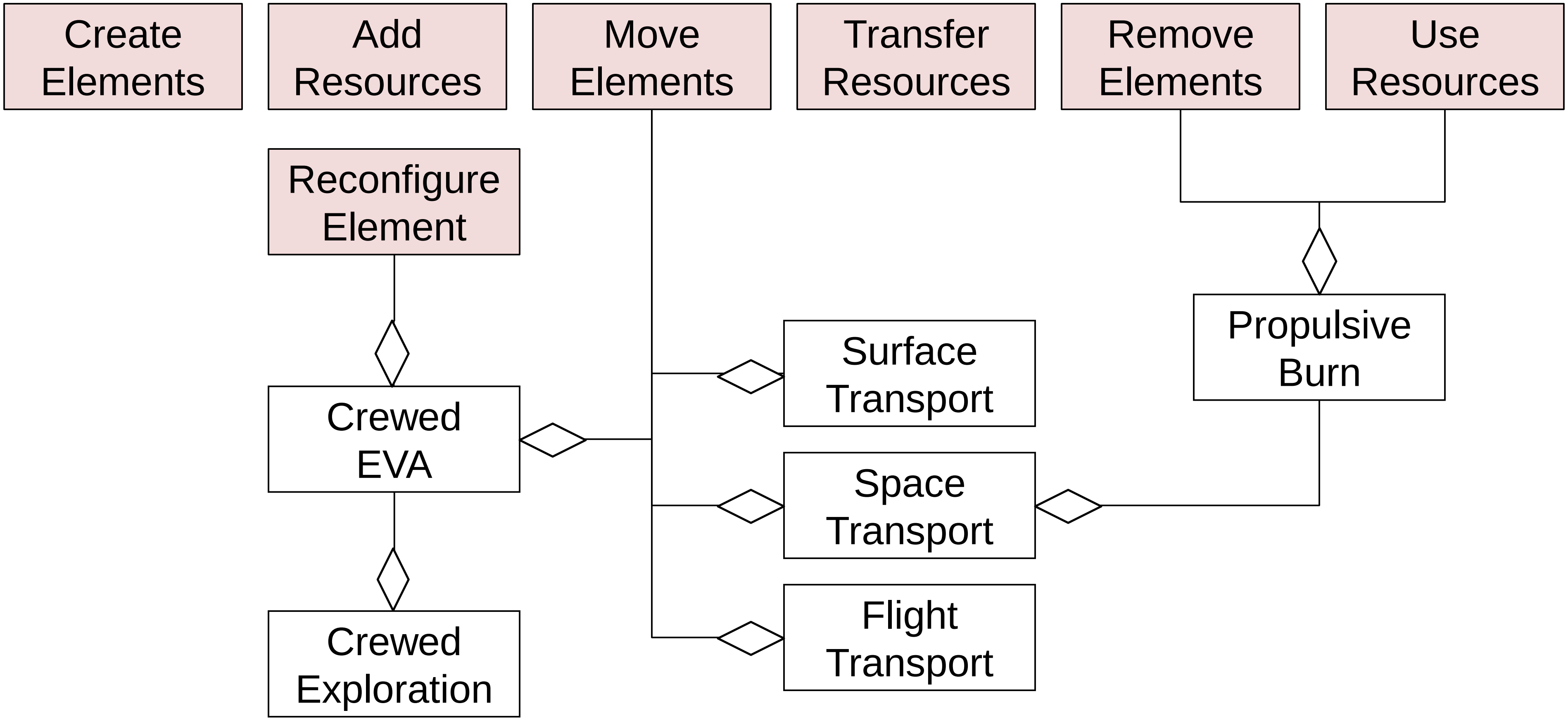


- Identify stable points in space:
  - Surface locations (e.g. KSC: Kennedy Space Center)
  - Stable orbits (e.g. LEO: Low Earth Orbit)
  - Lagrange points
- Represent network in time-expanded notation

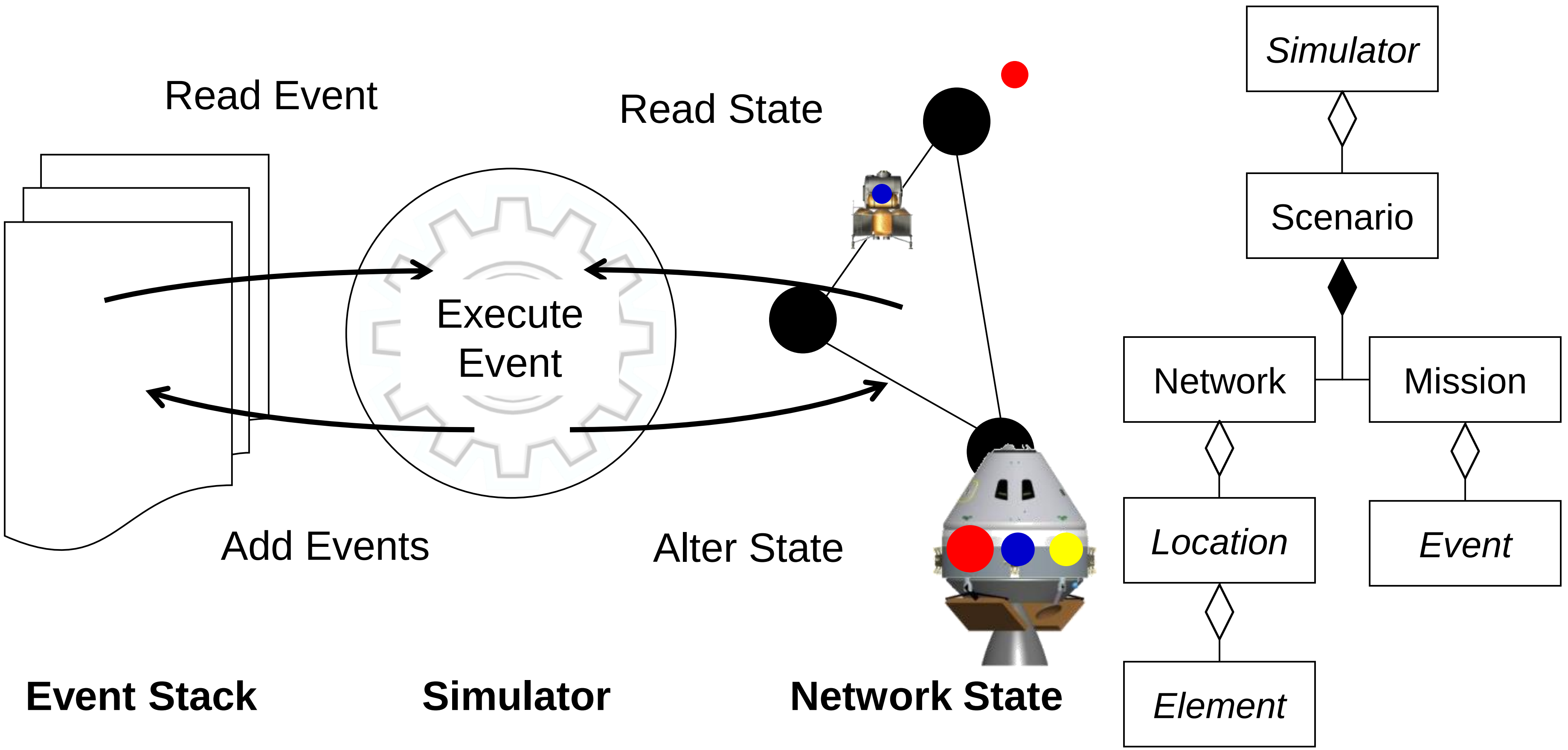
Simulation Element



Simulation Event

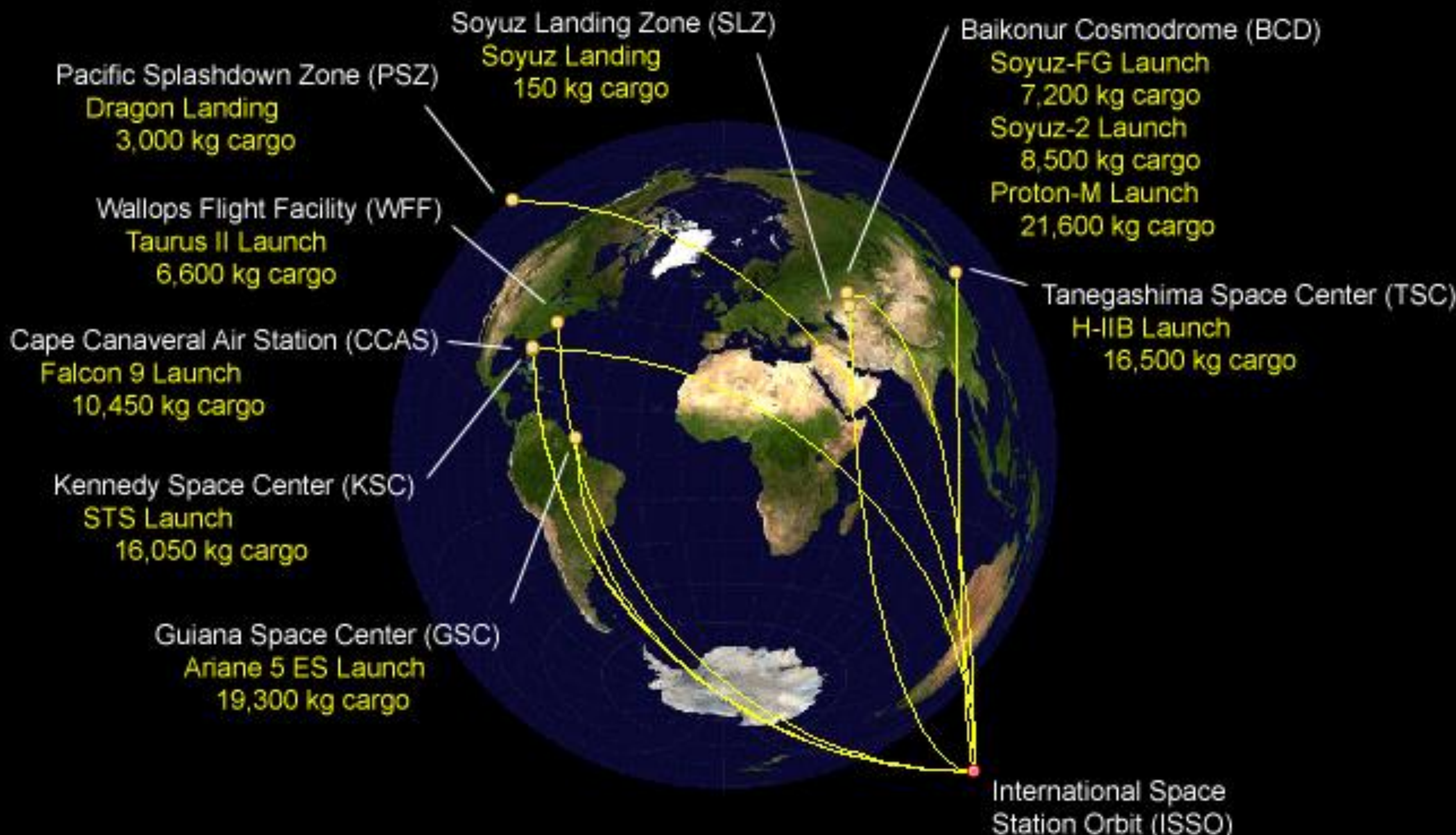


Discrete Event Simulation



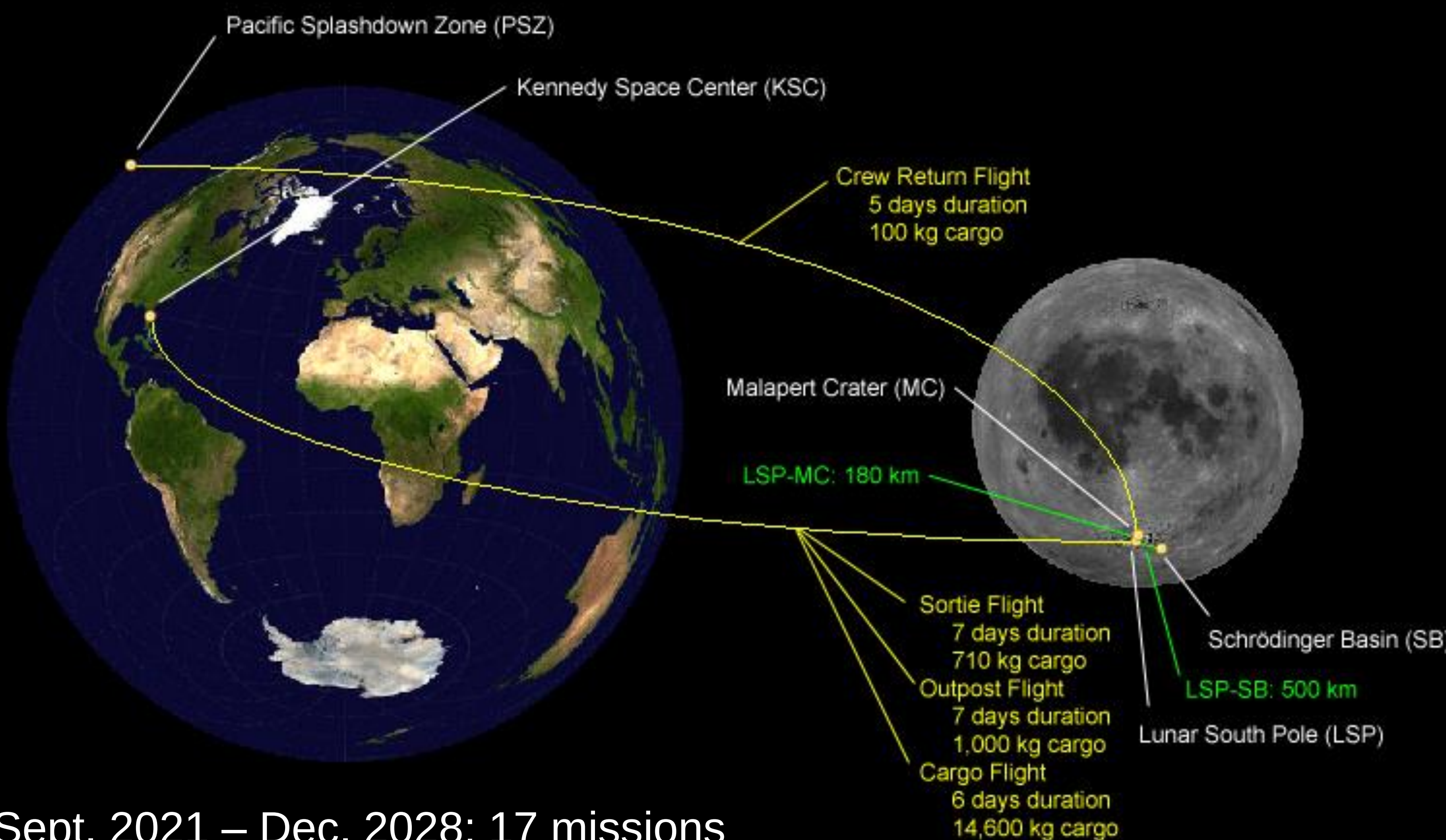
5. Validate Model

- Demonstrate with past/planned missions
  - Apollo 17 Lunar Mission
  - International Space Station Resupply Campaign
  - Constellation-style Lunar Exploration Campaign
  - Near-Earth Object (Asteroid) Mission
  - Design Reference Architecture 5.0 Mars Campaign



Sept. 2010 – Dec. 2015: 77 missions

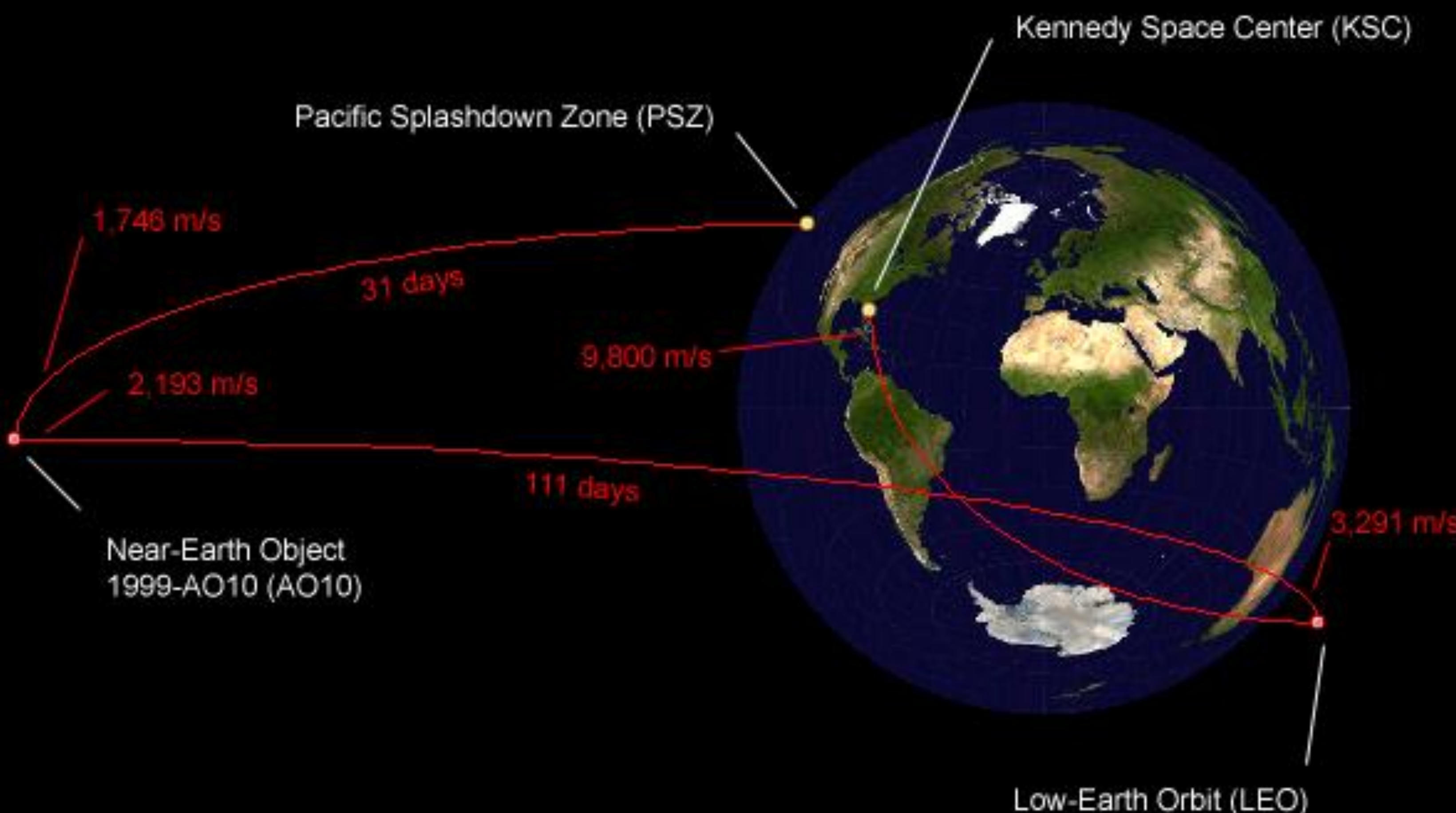
2 STS	22 Progress	22 Soyuz	12 Dragon
8 Cygnus	6 HTV	4 ATV	1 Proton-M



Sept. 2021 – Dec. 2028: 17 missions

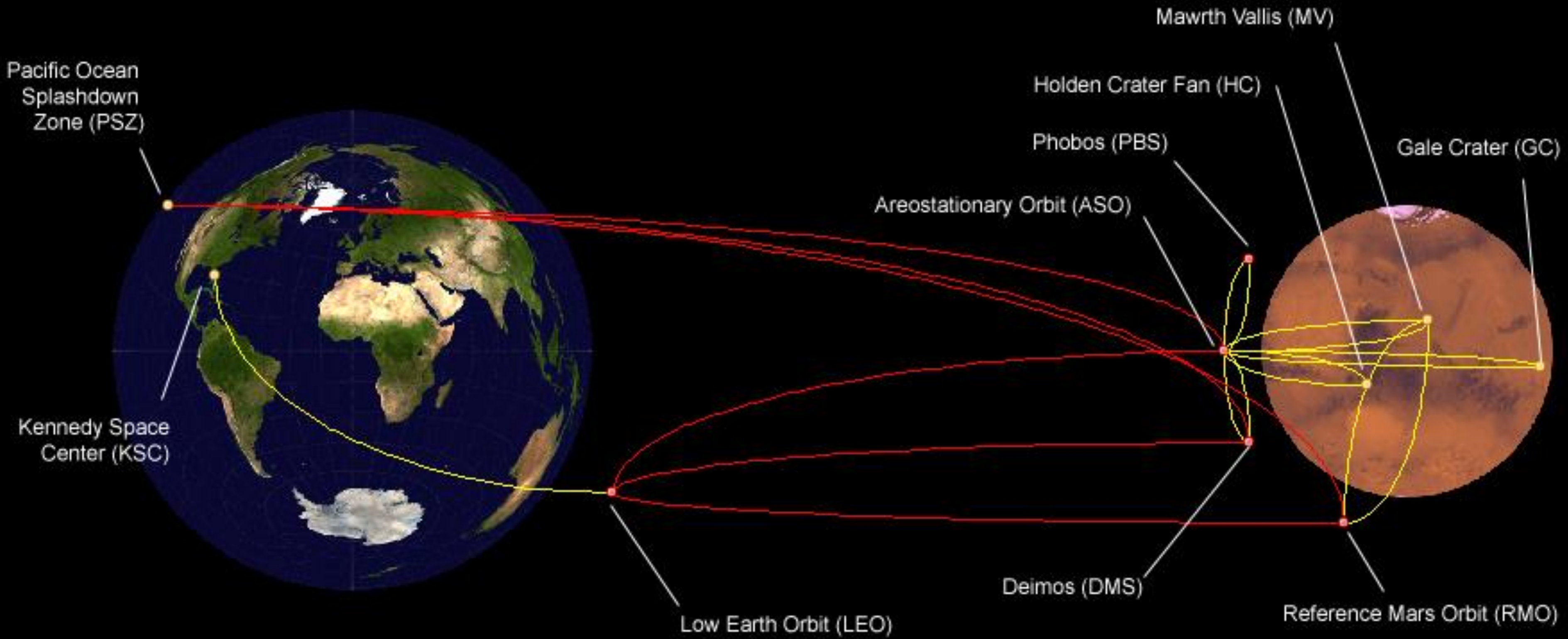
- 2 sortie-style (1 un-crewed)
- 7 outpost-style
- 8 cargo resupply

Excursions to Malapert Crater and Schrödinger Basin



Sept. 2025 – Feb. 2026: 1 mission

- 2 crew members (7.5 kg/person/day demands)
- Upper Stage reused for Earth departure and 1999-AO10 arrival
- 5-day exploration at 1999-AO10



2034-2053: 4 flexible missions

- Mars Tele-exploration Mission (MTM) – 3 kg returned (hoppers)
- Phobos and Deimos Sorties (PDS) – 150 kg returned (*Pirogue*)
- Phobos Exploration Mission (PEM) – 150 kg returned
- Mars Surface Mission (MSM) – 250 kg returned

Lessons Learned I

- Simulation state and state transitions are the most important concepts
  - Element attributes – load from Excel (structure)
  - Event specifications – programmed (behavior)
- Many software “gotchas” during development:
  - Floating-point round-off errors
  - Usability affordances
  - Memory management in Java
  - Serialization/deserialization of model state

Lessons Learned II

- Too much time on user interface (>65%), not enough time on the underlying model
- Models are one piece of an analysis toolchain
  - Avoid monolithic integrated environments
  - Carefully think about input/output interfaces first
  - Command line interface (CLI) useful for automation
  - Graphical user interface (GUI) useful for non-experts
  - Flexibility through interoperability with external tools