Towards a Framework for Modeling Space Systems Architectures

25 May 2006

Peter Shames & Joseph Skipper
NASA Jet Propulsion Laboratory, California Institute of Technology

Topics

- Statement of the problem
  - Space system architecture is complex
  - Existing terrestrial approaches must be adapted for space
  - Need a common architecture methodology and information model
  - Need appropriate set of viewpoints

- Requirements on a space systems model

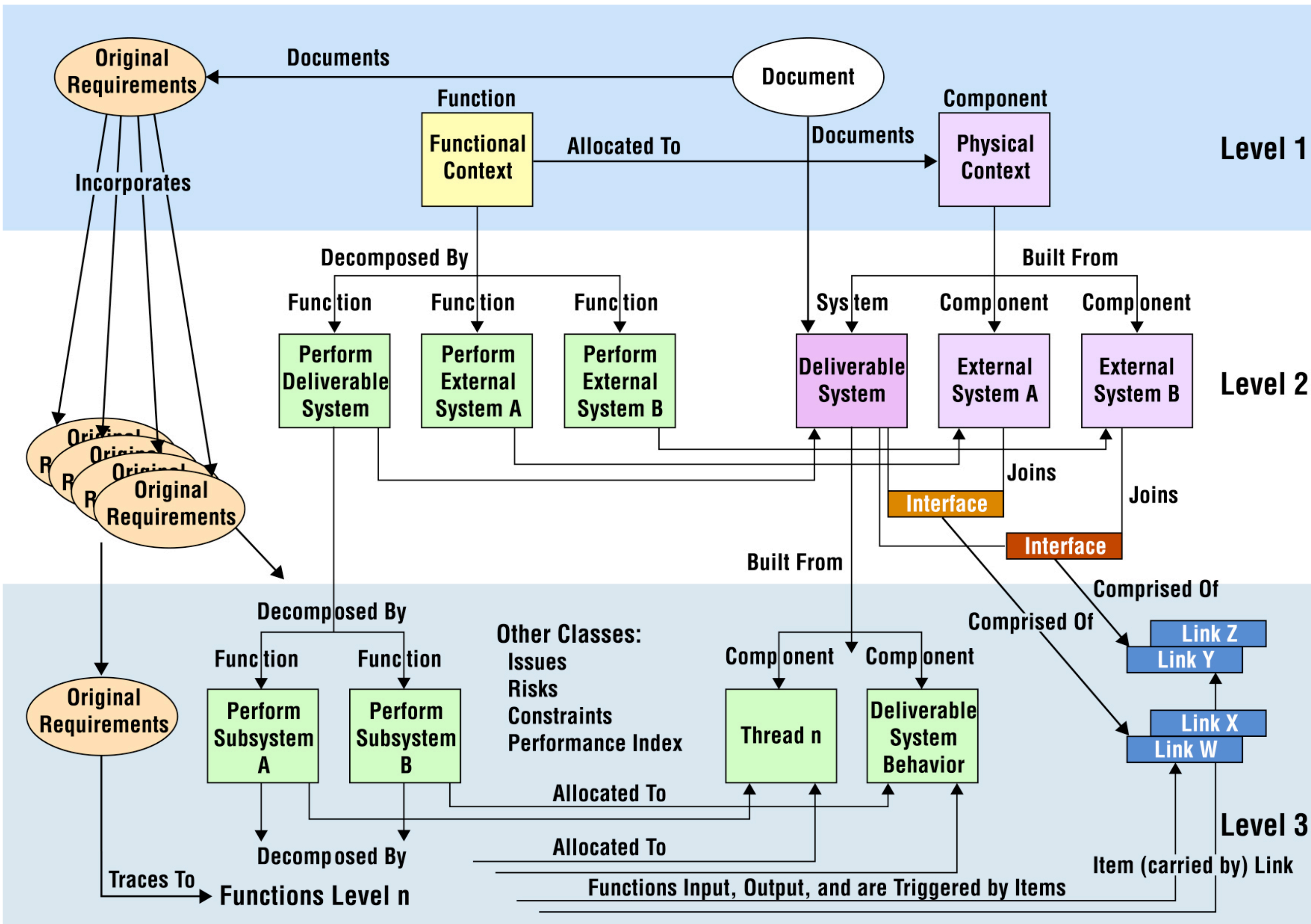
- Model Based Engineering and Design (MBED) project
  - Evaluated different methods
  - Adapted and utilized RASDS & RM-ODP
  - Identified useful set of viewpoints
  - Did actual model exchanges among selected subset of tools

- Lessons learned & future vision

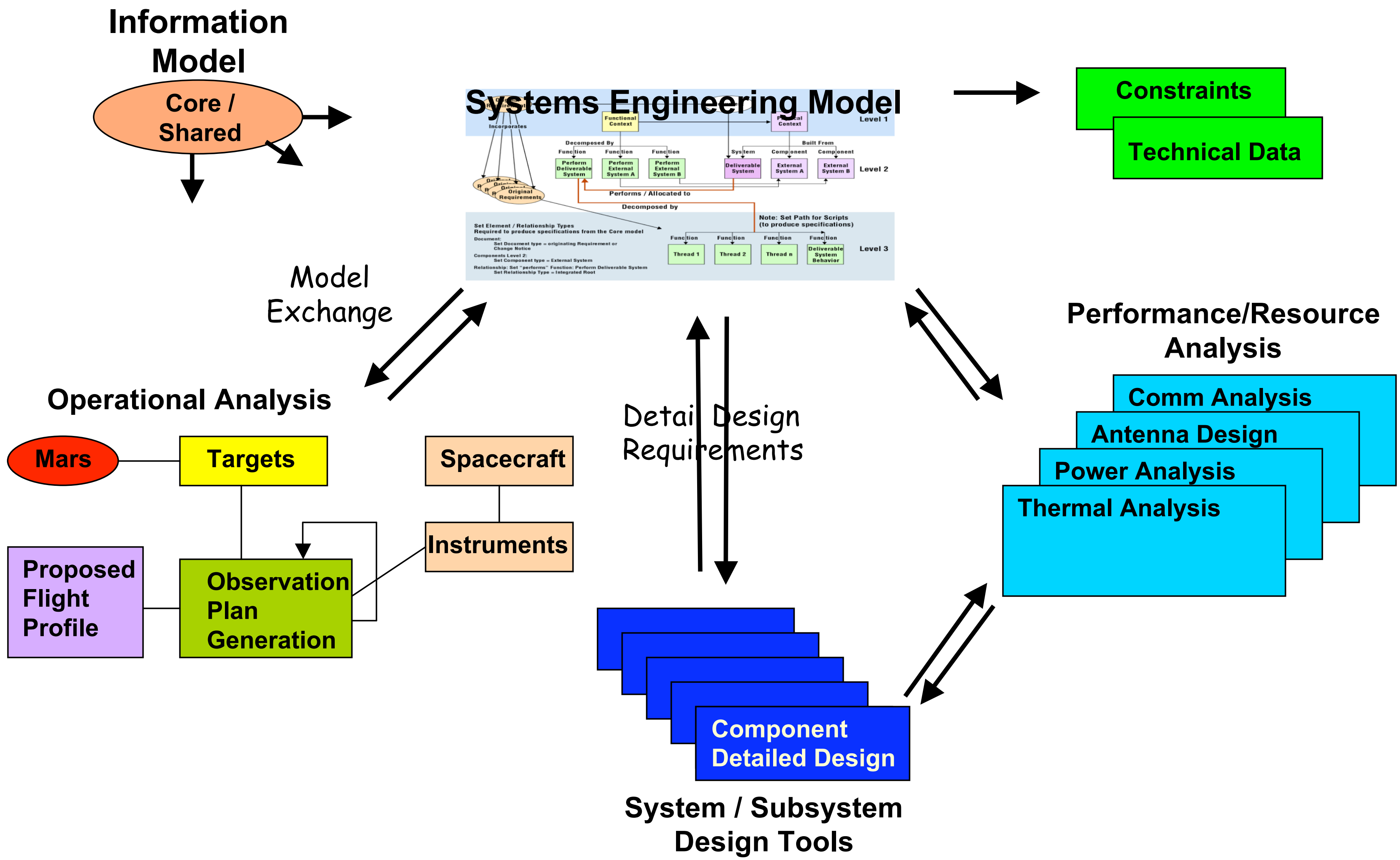
Architecting and Engineering Space Systems is Hard

- **Many Stakeholders**
  - Organizations (NASA, international partners, contractors)
  - Competing requirements (cost, schedule, risk, science, technology, survivability, maintainability, buildability)
- **Many different system aspects**
  - Logical (functionality, information, control)
  - Physical (hardware, software, environment) ***…motion***
  - Interoperability and cross support
  - Science & operational capabilities
  - Autonomous and human mediate operations
- **Long and complex system (of systems) lifecycle**
  - Development phases
    - Requirements, design, implement, I&T, V&V
    - Operations and sustaining
  - Cradle to grave lifecycle

Multi-level Systems Engineering Model



Integration of Architecture Views through Common Shared Models



System Architecture Model Objectives

- Provide a clear unambiguous views of the design
- Show relationship of design to requirements and driving scenarios
- Separate design concerns in the model to maintain degrees of freedom to do trades
- Detail the model views to the level appropriate for further systems engineering
- Provide executable models of the interactions
- Enable concurrent design of spacecraft, ground systems, science operations, control systems, and components
- Establish system engineering (SE) controls over the allocations and interfaces

Existing System Architecture Methods Inadequate

- Existing methods assume modeled objects are fixed in space and are usually in continuous and instantaneous communication
  - DoDAF, RM-ODP, TOGAF, FEAF, …
- Space systems tend to violate those assumptions
- Therefore, any of these modeling methodologies must be adapted to describe space systems
- Viewpoints must accommodate complex logical and physical interactions
  - UML, SysML are about design diagrams, do not directly support the needed viewpoints
- Specific engineering discipline modeling tools must be able to interact with core model, and extend it within their own domains

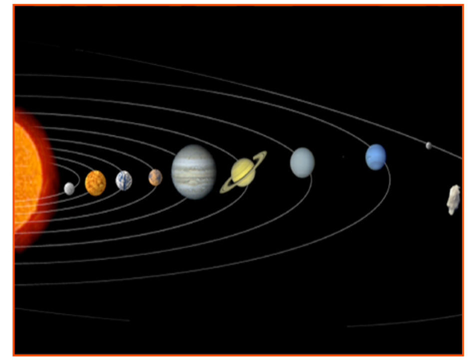
MBED Approach for Developing Architectural Model

- Identify commonalities, overlaps & gaps in different architectural and system engineering methods
- Determine that RM-ODP, and RASDS extensions, are a suitable starting point
- Define needed extensions to RM-ODP & RASDS for modeling space systems, beyond just the data systems elements
  - Physical Viewpoint extensions
  - Engineering and Technical Viewpoints
- Use these model concepts to develop system model and drive information model and tool integration

RM-ODP & RASDS Characteristics

- The specification of a complete system in terms of ***viewpoints***.
- The use of a ***common object model*** for the specification of the system from every viewpoint.
- The use of ***views*** to tailor user or domain specific analyses of the system.
- The definition of a ***modeling infrastructure*** that provides support services for system applications, hiding the complexity and problems of defining mission specific models.
- The definition of a set of ***common transformation functions*** that provide general services needed during the design and development of space systems.
- A ***framework*** for the evaluation of conformance of models and designs based on conformance points.

Extended RASDS Space System Domain Architectural Viewpoints



Enterprise — Business Concerns; Organizational perspective

Information — Data Concerns; Relationships and transformations

Functional — Computational Concerns; Functional composition

Physical — Physical Concerns; Component, Connector & external elements

Engineering — System Design Concerns; Allocation, methods, performance

Technology — Technology & Protocol Concerns; Framework, tools, standards perspective

Derived from: CCSDS RASDS, RM-ODP, ISO 10746 and compliant with IEEE 1471

Extended RASDS Semantic Information Model Derivation

RASDS as Architectural Framework *

Engineering Viewpoint
- System Design & Construction
- Functional allocation
- Distribution of functions and trade-offs
- Development
- Validation & verification

• Based on RMODP**

Enterprise Viewpoint
- Organizations
- People
- Use Case-Scenarios
- Contracts/Agreements

Augment to Capture:
- Mission Design & Drivers
- Requirements
- Cost
- Enterprise Risks

Physical Viewpoint
- Connectivity
- Components & connectors
- Physics of Motion
- End to End View
- External Forces
- Performance

Augment to Capture:
- Structure
- Power
- Mass
- Thermal
- Orbit
- Propulsion

Functional Viewpoint
- Functional Structure
- Functional Behavior & interfaces
- End to End View
- Cross Support Service

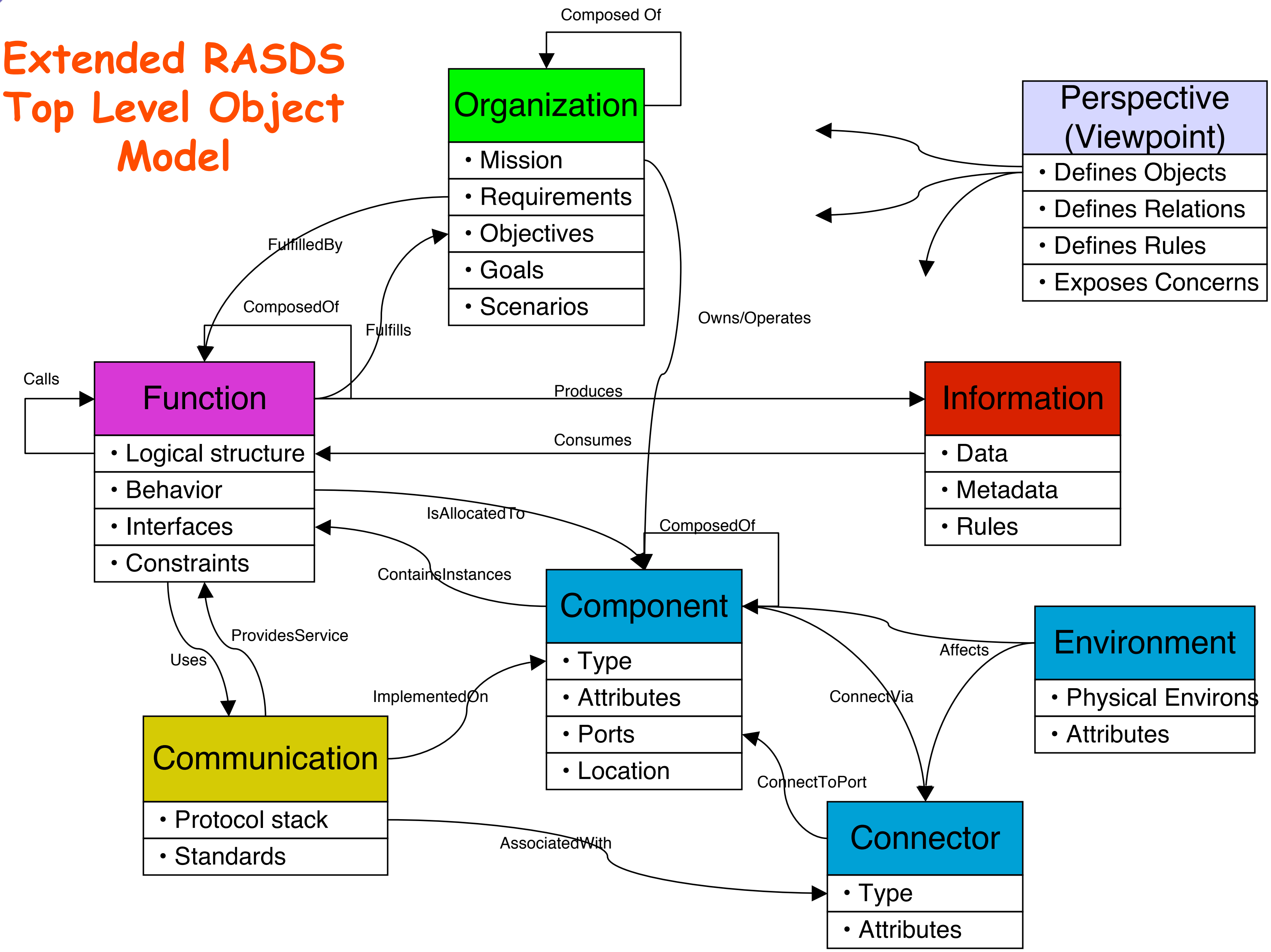
Information Viewpoint
- Information & information management
- Scenarios
- End to End View

Technology Viewpoint
- Protocols & comm standards
- End to end Information Transfer Mechanisms
- Cross Support Services

* Reference Architecture for Space Data Systems (RASDS)

** Reference Model Open Distributed Processing (RMODP, ISO 10746 spec)

Extended RASDS Top Level Object Model



Viewpoint Elements - Functional Example

- **Stakeholders:** system engineers, acquirers, developers, users, and maintainers
- **Concerns:** the functions that are required for the system to meet its requirements and execute its scenarios
- **Modeling Language:** functional objects and relationships, interfaces, behaviors, constraints
- **Consistency & Completeness Methods:** every requirement maps to at least one function, no requirement is not mapped to a function, no function is not mapped to a requirement, and there is structural data and control flow consistency

Typical Functional Views

- **Functional Dataflow** view – An abstract view that describes the functional elements in the system, their interactions, behavior, provided services, constraints and data flows among them. Defines which functions the system is capable of performing, regardless of how these functions are actually implemented.

- **Functional Control** view – Describes the control flows and interactions among functional elements within the system. Includes overall system control interactions, interactions between control elements and sensor / effector elements and management interactions.

Viewpoint Elements - Physical Example

- **Stakeholders:** system engineers, sub-system engineers, acquirers, developers, operators, users, and maintainers
- **Concerns:** the physical structures of the system, their connections, and how they interact with the environment
- **Modeling Language:** physical objects (components) and their connections, physical behavior, motion and interactions, the environment, constraints
- **Consistency & Completeness Methods:** every functional element maps to at least one physical element, no functional element is not mapped, no physical element is not mapped to a function, and there is structural integrity and consistency

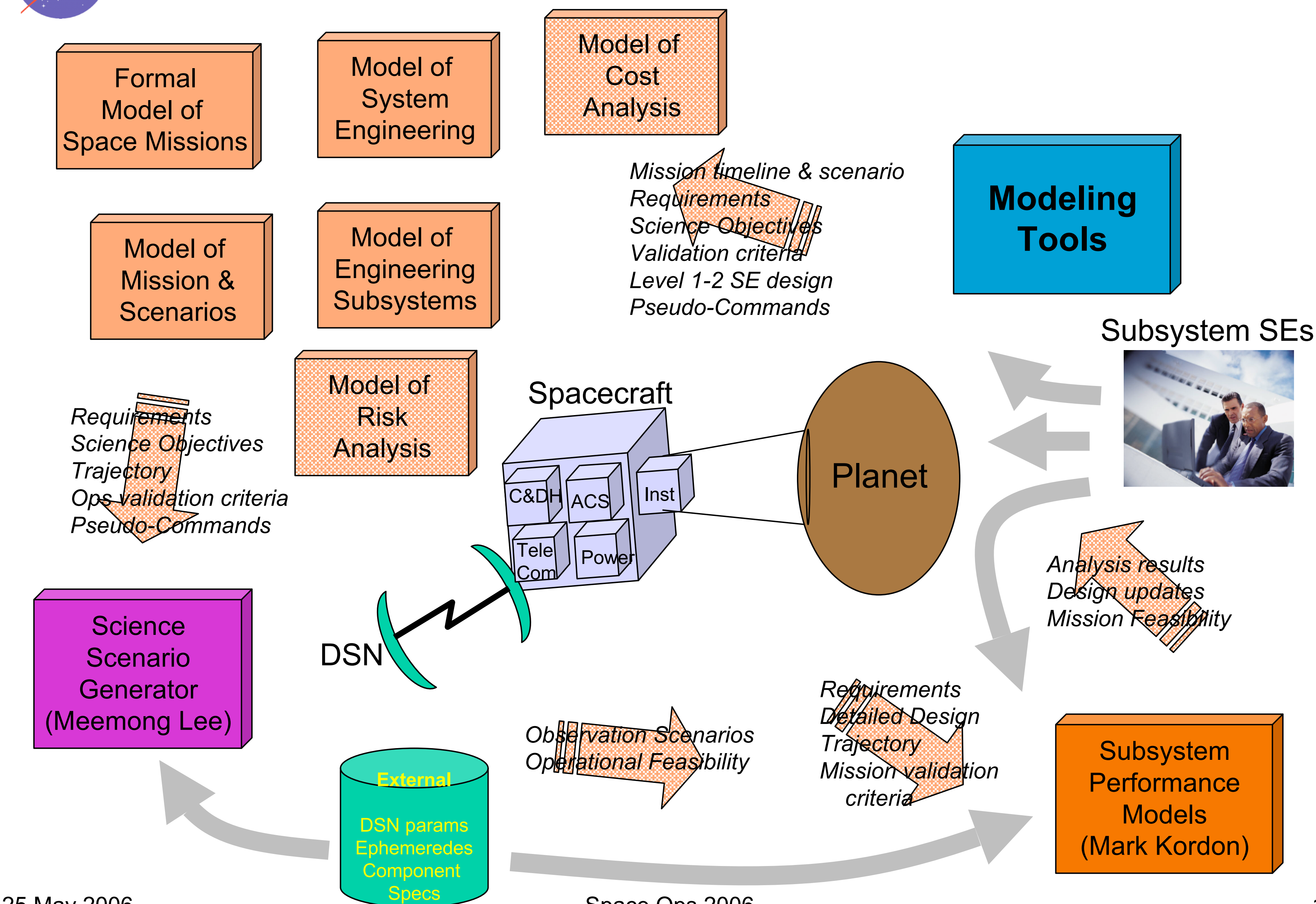
Typical Physical Views

- **Data System** view – Describes instruments, computers, and data storage components, their data system attributes and the communications connectors (busses, networks, point to point links) that are used in the system.
- **Telecomm** view – Describes the telecomm components (antenna, transceiver), their attributes and their connectors (RF or optical links).
- **Navigation** view – Describes the motion of the major elements of the system (trajectory, path, orbit), including their interaction with external elements and forces that are outside of the control of the system, but that must be modeled with it to understand system behavior (planets, asteroids, solar pressure, gravity)
- **Structural** view – Describes the structural components in the system (s/c bus, struts, panels, articulation), their physical attributes and connectors, along with the relevant structural aspects of other components (mass, stiffness, attachment)
- **Thermal** view – Describes the active and passive thermal components in the system (radiators, coolers, vents) and their connectors (physical and free space radiation) and attributes, along with the thermal properties of other components (i.e. instruments as thermal sources (or sinks), antennas or solar panels as sun shade)
- **Power** view – Describes the active and passive power components in the system (solar panels, batteries, RTGs) within the system and their connectors, along with the power properties of other components (data system and propulsion elements as power sinks and structural panels as grounding plane)
- **Propulsion** view – Describes the active and passive propulsion components in the system (thrusters, gyros, motors, wheels) within the system and their connectors, along with the propulsive properties of other components

Physical Viewpoint - Component / Connector Examples

- **Data System**
  - Components (CPU, instruments, SSR)
  - Connectors (network, data bus, serial lines, backplane)
- **Telecomm**
  - Components (transmitter, receiver, antenna)
  - Connectors (RF link, optical link, waveguide)
- **Power**
  - Components (solar panel, battery, RTG, switches, power attrib of other components)
  - Connectors (power bus)
- **Thermal**
  - Components (cooler, heater, thermal attrib of other components)
  - Connectors (heat pipe, duct, free space radiation)
- **Structural**
  - Components (S/C bus, physical link, arm, struct attrib of other components)
  - Connectors (joint, bolt (incl explosive), weld)
- **Propulsion**
  - Components (motor, wheel, thruster)
  - Connectors (contact patch, gravity)

Future Modeling Environment



MBED Lessons Learned

- Space system architectures must be described from multiple views to meet different stakeholder concerns
- Existing architectural methods must be adapted to describe space systems
- A variety of tools for system architecture and engineering must be able to produce vendor independent models
- All tools are required to integrate into the common information model
- Tool integration requires an agreed method of information exchange, preferably XML based
- Performing and verifying model interchange is not (yet) a simple task because of semantic and syntactic issues
- System behavioral / performance modeling is not yet within reach and still very much a challenge

Next Steps

- Define extended space system model within a formalized modeling environment
  - Use UML/SysML to develop a “Space System Profile”
  - Can leverage existing UML4ODP and DoDAF profiling efforts
  - Requires real organizational commitment of resources
- Evaluate method on a suitable project
  - Spacecraft or ground data system
  - DSMS / DSN evaluation underway …