Modeling Complex Cross-Systems Software Interfaces using SysML

Sanda Mandutianu, Ron Morillo, Kim Simpson, Otfried Liepack, Kevin Bonanne

Jet Propulsion Laboratory/Caltech

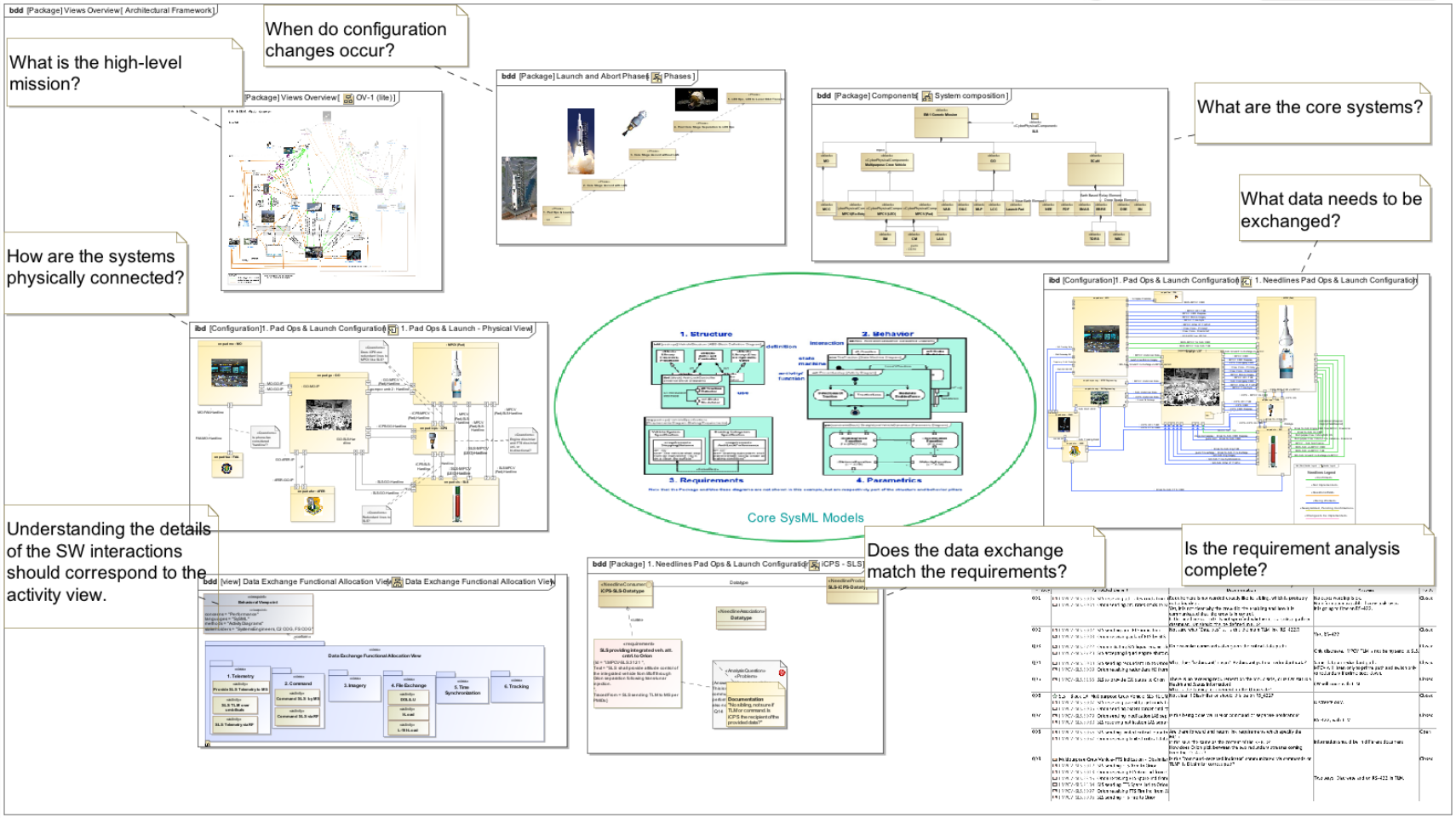
Infotech@AIAA, August 19-22 2013, Boston, MA

AIAA
The World's Forum for Aerospace Leadership

Goals and Objectives

- NESC sponsored task to
  - Identify potential ICD gaps
  - Characterize cross-systems behaviors of the critical MPCV-SLS
  - Demonstrate approach for cross-system integration

Engineering Questions



AIAA
The World's Forum for Aerospace Leadership

Systems Engineering Problems

- Describing and Documenting System Interfaces
- Software Architecture Specifications
- System Modeling

AIAA
The World's Forum for Aerospace Leadership

Modeling Approach

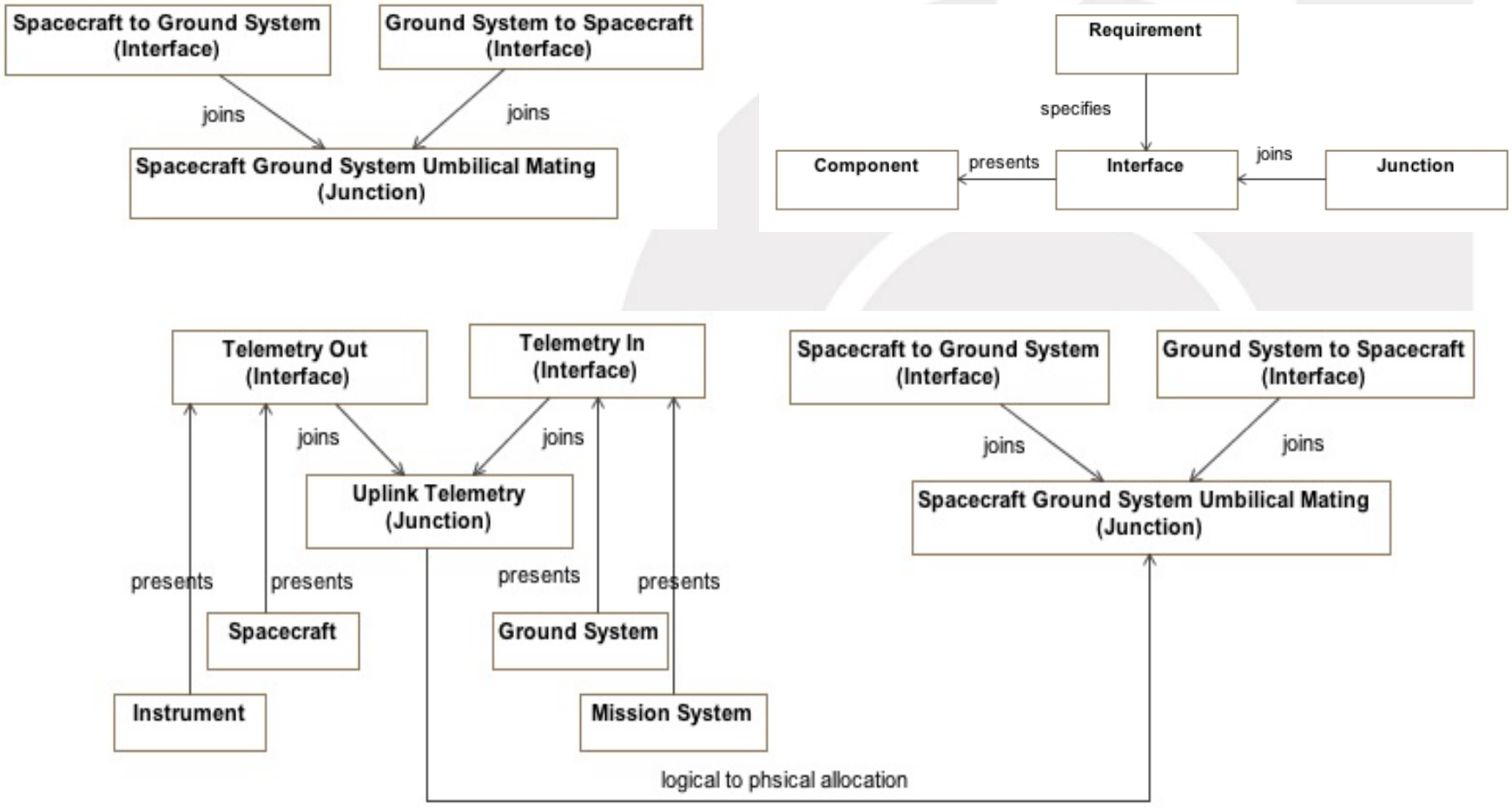
- Leverage:
  - EFT-1 and EM-1 end-to-end architecture models based on the IEEE 1471 and DoDAF
  - JPL ontologies and interface pattern recommendation
- Support MS-SLS software ICD development
- Modeling scope: architecture descriptions of the software interfaces at different levels of abstractions and their behaviors

AIAA
The World's Forum for Aerospace Leadership

Interface Modeling Pattern

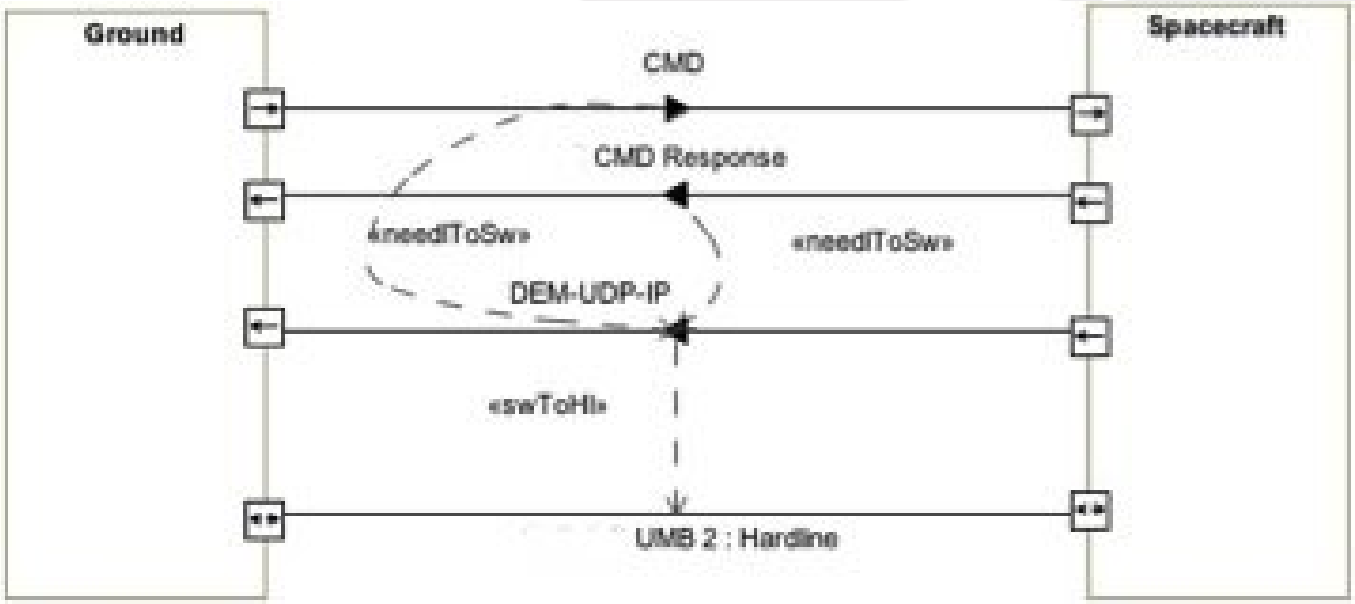
- The interface pattern allows to define an abstract interface without specifying any implementation.
- The interface is an abstraction of multiple interactions
- The intent of interacting is abstracted as an "interface"
  - Specifies characteristics of what is shared
  - May include behavioral features

Interface Modeling Pattern Definition and Usage



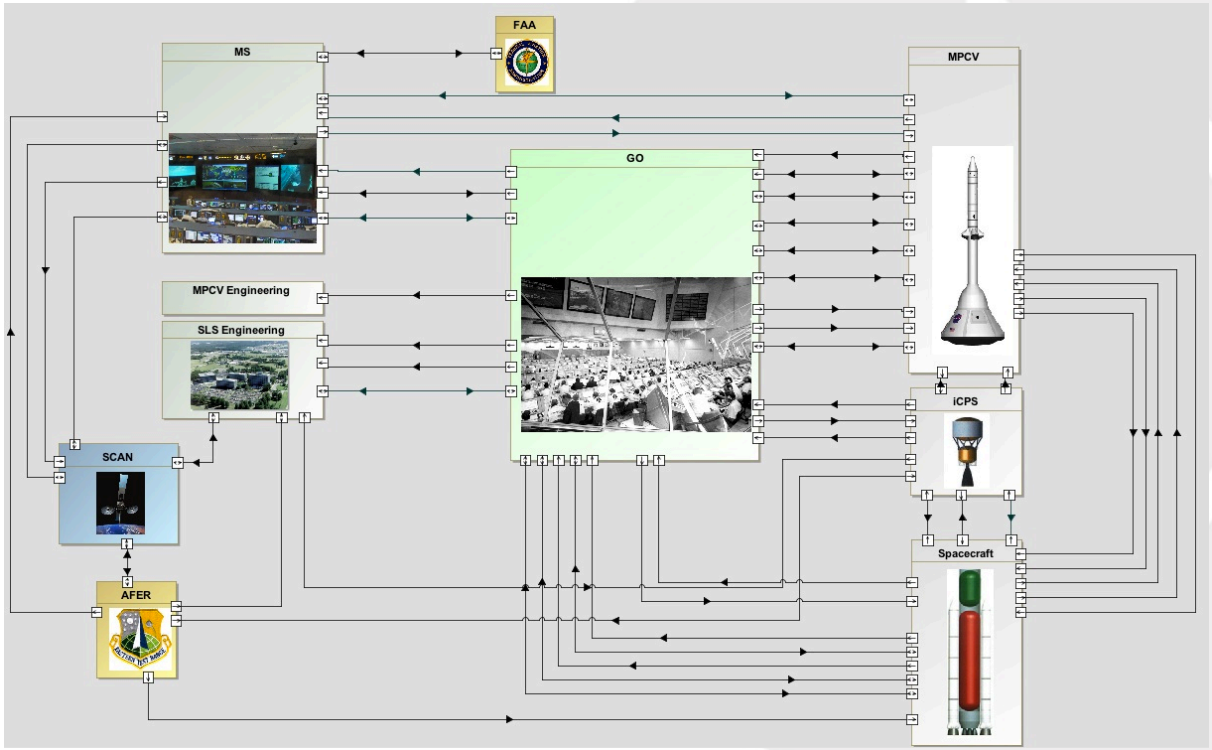
Modeling Software Interfaces

SW interface – particular case of interface
Layered approach – loosely coupled



End-to-end Data Exchange Interfaces

Providing mission context
High-level logical connections



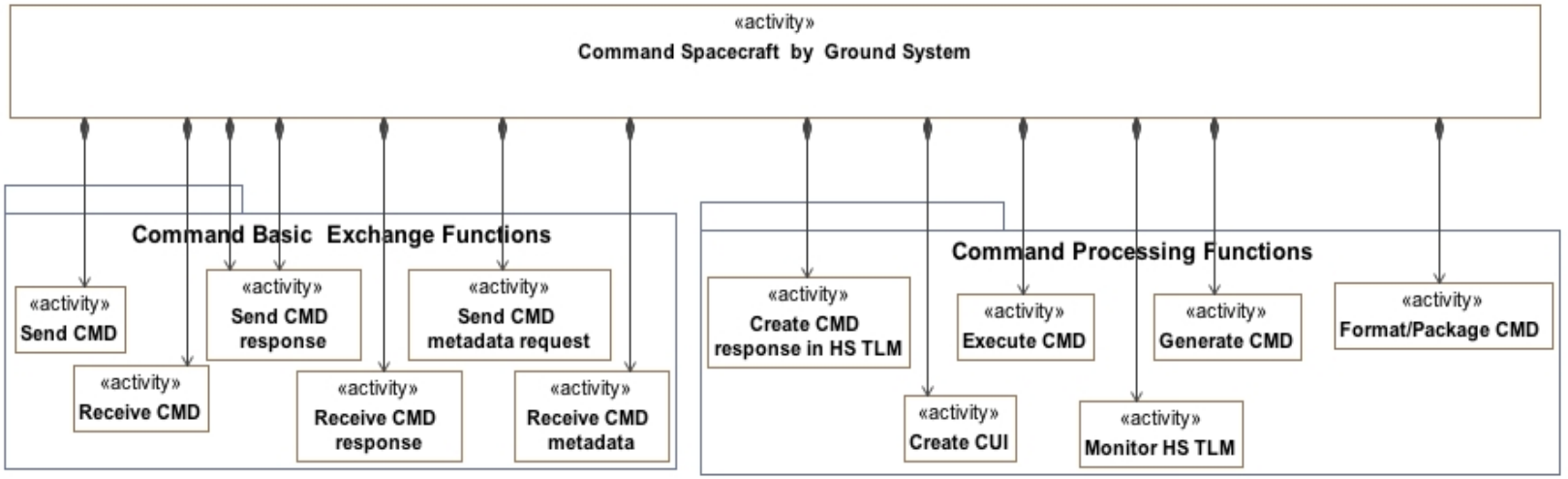
AIAA
The World's Forum for Aerospace Leadership

Modeling Functions and Behaviors

- Understand behavior by decomposing top-level functions derived from functional requirements specification
- Key: functional architecture allocated to performing structural elements
- Two representations of functional architecture
  - Functional hierarchy
  - Behavioral model

Functional Hierarchy

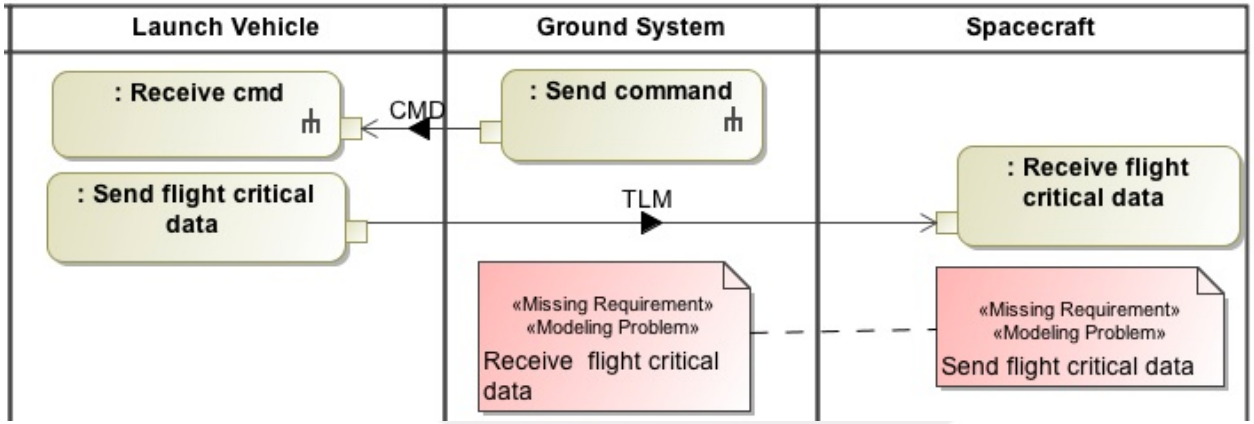
- Conveys the transformation
  interface specifications → interface functions
- Provides traceability
  requirements → products



Behavior Model

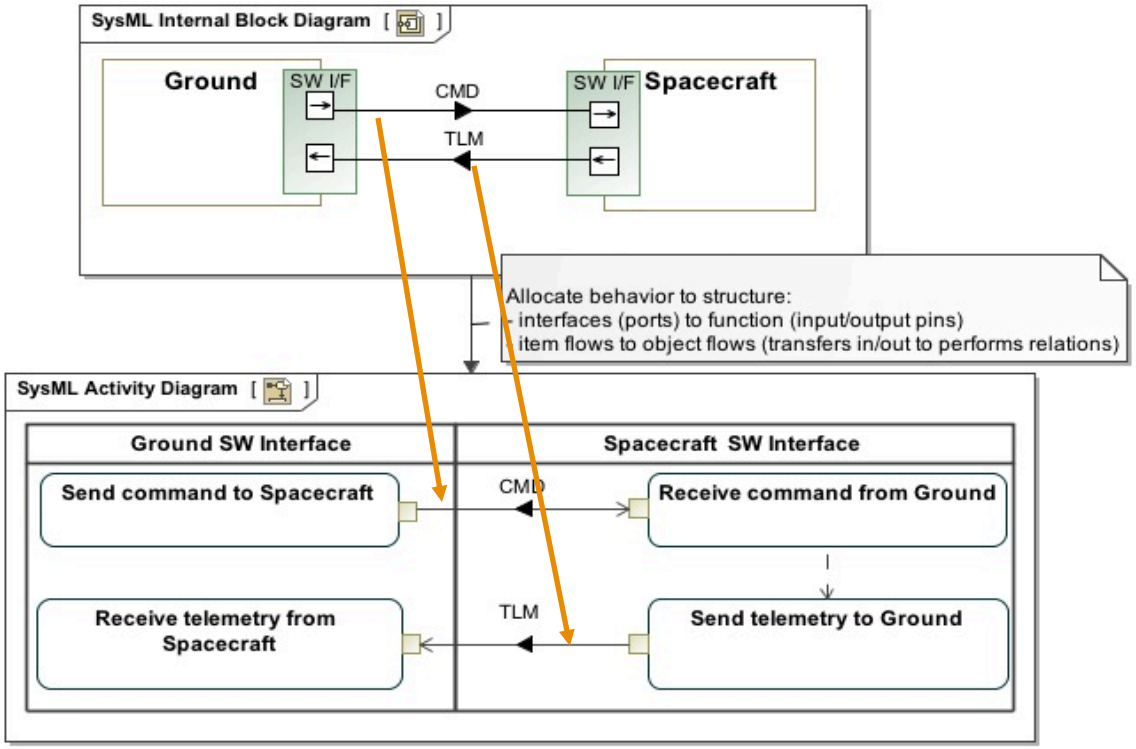
Supports the specification of the behavior of functional components

- Sequence of functions (functional flow)
- Data flow
- Control logic for execution (control flow)
- Resources (mapping to structure)



Mapping behaviors to structure – interface pattern

Explicit mapping of the flows



AIAA
The World's Forum for Aerospace Leadership

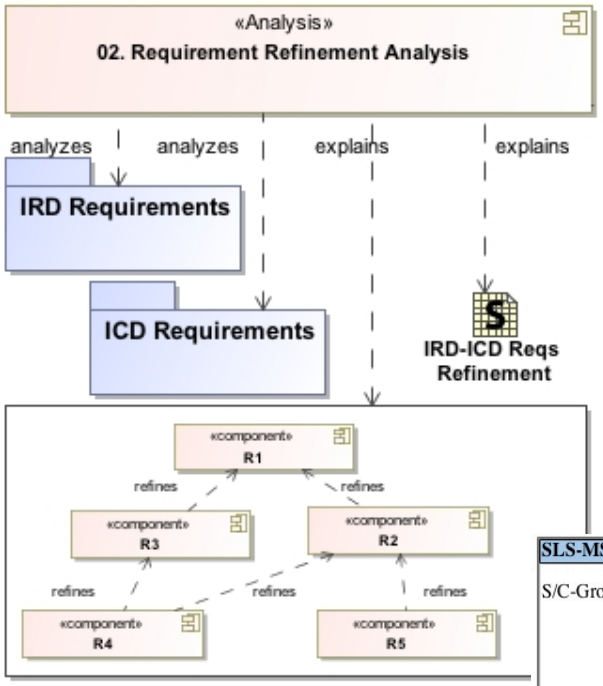
Analysis Modeling Concept

- One of the main purposes of modeling: accurate, reliable, automated analysis
- Analyses – explicitly represented
  - Identify analyses
  - Specify relationships with other model elements
- Conceptually – a justification
- Examples: requirements analysis, trade studies, functional analysis, etc.

Engineering Analyses

Name	Documentation	Owned Diagram
01. Requirements Allocation Analysis	**Purpose**: Find the requirements that are not allocated to a component. **Rules:** 1. All requirements have to be allocated to components 2. All components have to be specified by one or more requirements. **Method:** For all requirements check the specified components. Generate table (or an external representation of it, such as an Excel file) **Reports/Findings:** *Currently:* Dependency table – requirements allocated to components (can be reversed)–model/tool built-in query	Not Allocated Requirements Analysis
02. Requirement Refinement Analysis	**Purpose**: Check for valid requirements refinement graph **Rules**: Except for top-level requirements, every requirement must refine at least one other requirement (rule 1) The requirements refinement graph must not contain cycles (rule 2) **Method**: Construct Built-in dependency tables More built-in sophisticated queries	SLS-MS Requirements IRD-ICD Derive Tabl

Requirements Refinement Analysis



Purpose: Check for valid requirements refinement graph

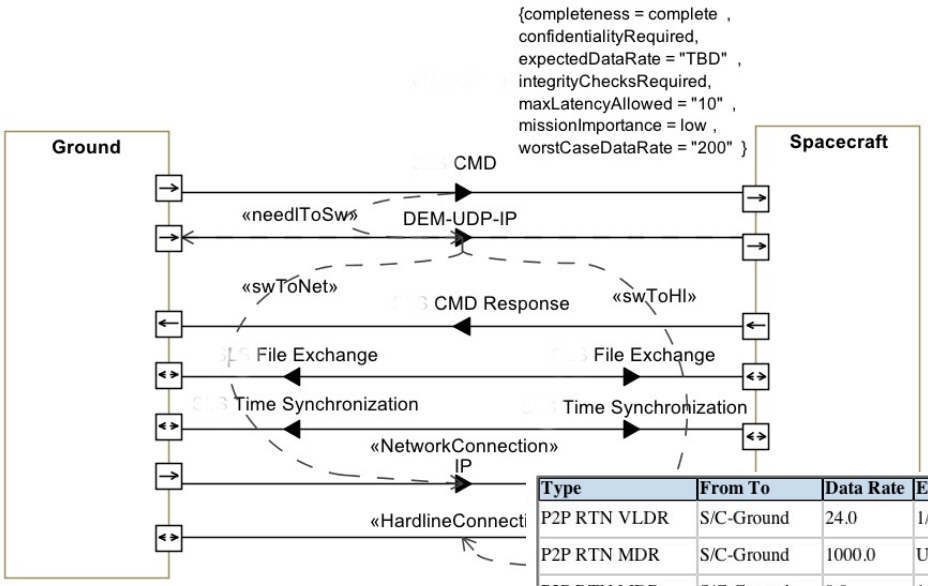
Rules:
Except for top-level requirements, every requirement must refine at least one other requirement (rule 1)
The requirements refinement graph must not contain cycles (rule 2)

Method:
Construct Built-in dependency tables
More built-in sophisticated queries

Results:
Display requirements flow-down
Show gaps

SLS-MS ICD Requirement	MPCV-SLS Traced Requirement
S/C-Ground 3.▷◁1.▷◁1.▷◁1 MS commanding S/C via InterSys	- - I.InterSys-S/C.2722 InterSys initiating S/C lengine shutdown - - I.InterSys-S/C.3004 InterSys sending att. rates cmds to S/C while manual steering off-nominal - - I.InterSys-S/C.3006 InterSys sending ascent target cmd to S/C
S/C-Ground 3.▷◁1.▷◁1.▷◁2 S/C receiving commands from Ground via InterSys	- - I.InterSys-S/C.3005 SLS receiving att. rates cmds from InterSys while manual steering - - I.InterSys-S/C.3007 SLS receiving ascent target cmds from InterSys - - I.InterSys-S/C.3078 SLS receiving notification of SA separation from InterSys - - I.InterSys-S/C.3080 SLS receiving notification LAS separation from InterSys

Configuration Analysis



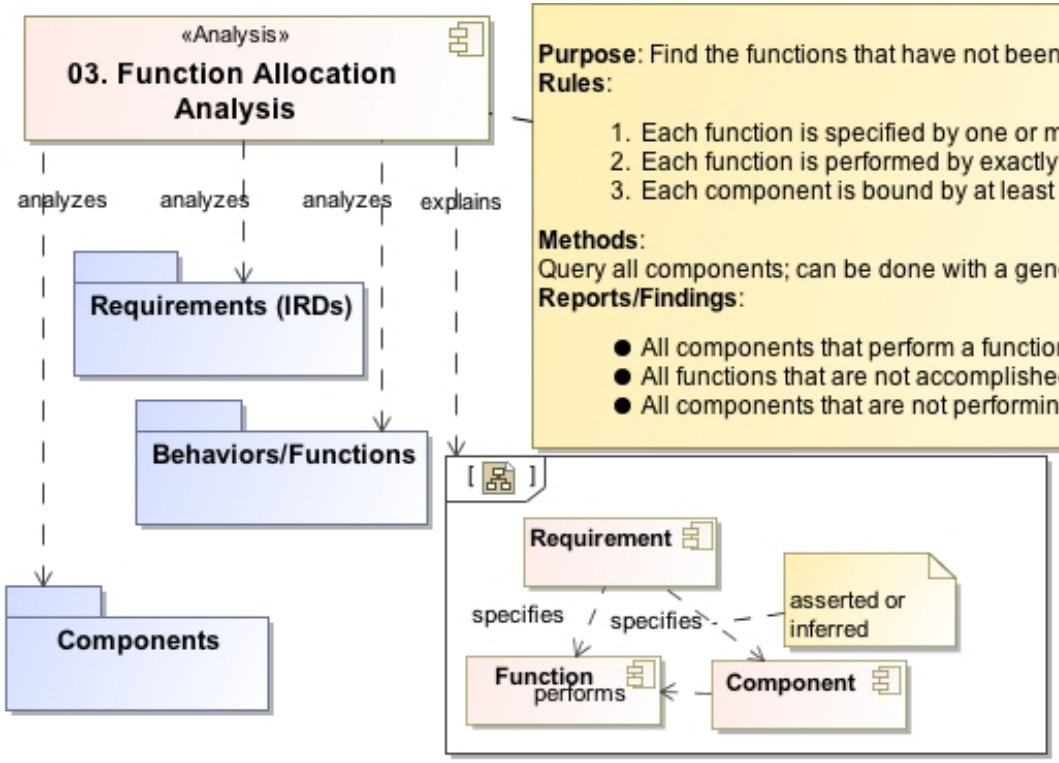
Name	Phase	Type	Net Address
Computer 1 IP addr	Pad Ops & Launch Configuration	IP	CTC2 IP Addr
Computer 2 IP addr	Pad Ops & Launch Configuration	IP	CTC1 IP Addr

Name	Phase	Type	Options
Ground Umbilical 1	Pad Ops & Launch Configuration	Hardline	100Base-TX
Ground Umbilicall 2	Pad Ops & Launch Configuration	Hardline	100Base-TX

Type	From To	Data Rate	Encoding	Modulation	XMIT Freq.	Link Margin	Max Band	BER	Phases
P2P RTN VLDR	S/C-Ground	24.0	1/2 LDPC	SQPSK	421	TBD	TBD	0.0	Lunar Orbit
P2P RTN MDR	S/C-Ground	1000.0	Uncoded	SQPSK	425	TBD	TBD	0.0	Lunar Orbit
P2P RTN MDR	S/C-Ground	0.0	1/2 LDPC	SQPSK	500	TBD	TBD	0.0	Lunar Orbit
P2P RTN LDR	S/C-Ground	0.0	Uncoded	SQPSK	333	TBD	TBD	0.0	Lunar Orbit
P2P FWD VLDR	S/C-Ground	24.0	1/2 LDPC	SQPSK	421	TBD	6	0.0	Lunar Orbit
P2P FWD MDR	S/C-Ground	144.0	Uncoded	SQPSK	425	TBD	6	0.0	Lunar Orbit
P2P FWD MDR	S/C-Ground	500.0	1/2 LDPC	SQPSK	500	TBD	6	0.0	Lunar Orbit
P2P FWD LDR	S/C-Ground	72.0	Uncoded	SQPSK	kbps	TBD	6	0.0	Lunar Orbit

Determine configurations of control paths: connections, characteristics of data exchanges, link options, etc.

Functional Allocation Analysis



Purpose: Find the functions that have not been allocated to a mission component.
Rules:

1. Each function is specified by one or more requirements
2. Each function is performed by exactly one component
3. Each component is bound by at least one requirement

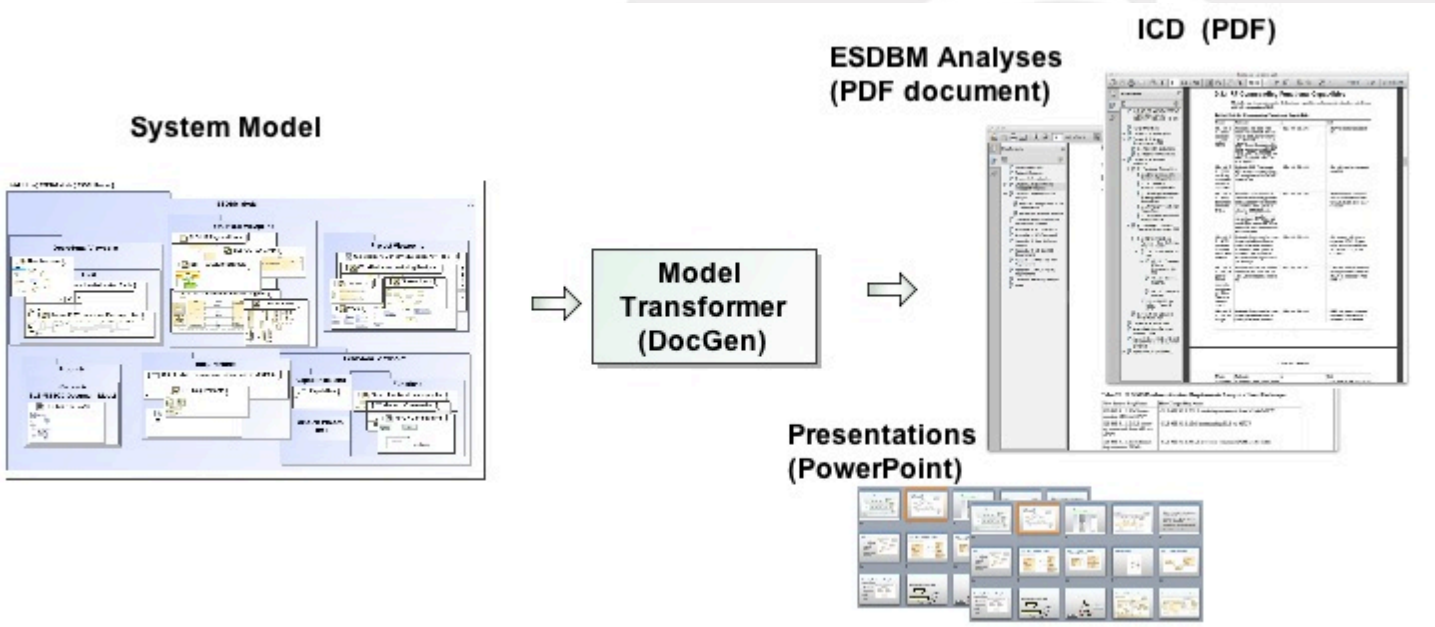
Methods:
Query all components; can be done with a generic table in the first approximation. More sophisticated queries if needed.
Reports/Findings:

- All components that perform a function, by functions accomplished.
- All functions that are not accomplished by any component
- All components that are not performing any functions

Interface	Behavior	Component	Function
S/C to Ground CMD	Receive CMD	Spacecraft	Provide Telemetry
Ground to S/C TLM	Package TLM	Spacecraft	Provide Telemetry
Ground to S/C TLM	Send TLM	Spacecraft	Provide Telemetry

Automated Generation of SE Artifacts

The output of SE is ultimately specification, not implementation: system architecture, IRD, ICD, analysis results, etc.

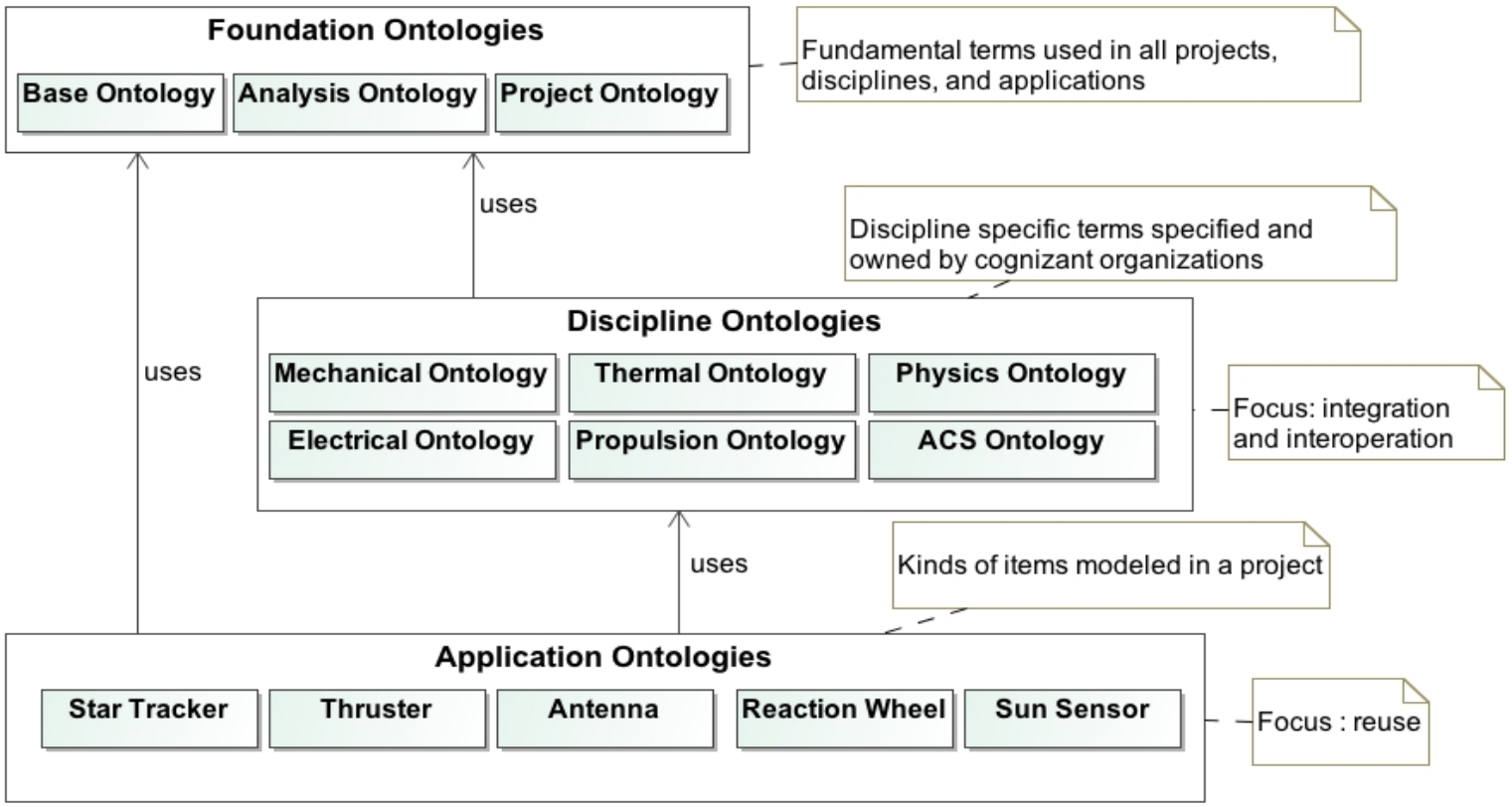


Conclusions and Future Work

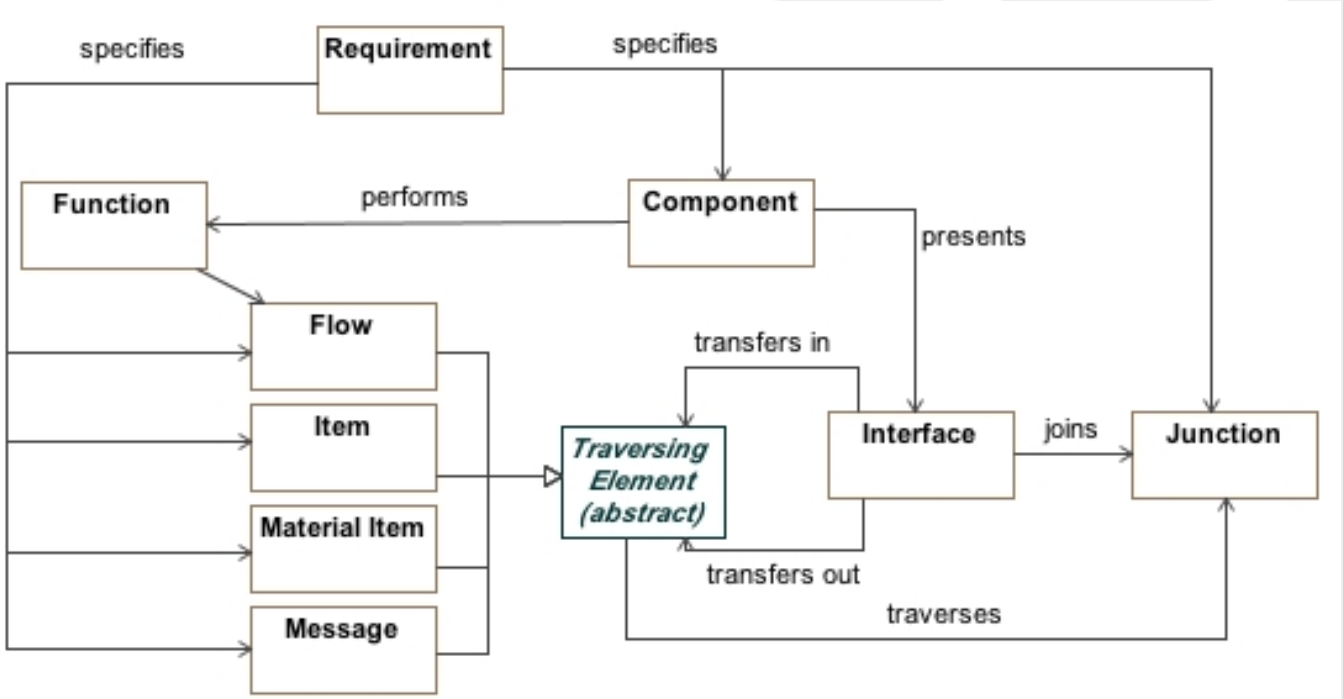
- Observations
  - Communication with stakeholders is critical
  - Product oriented modeling
  - MBSE doesn't replace established SE processes
    - SE rules implemented in the model
  - Graphics are very powerful
- Future work – extend the current capabilities (functional and behavioral), explore more areas

BACKUP

Ontology Hierarchy



Foundation Ontology Fragment



The World's Forum for Aerospace Leadership