30th Annual INCOSE international symposium
Virtual Event
July 20 - 22, 2020

Ryan Noguchi, James Martin, and Marilee Wheaton
The Aerospace Corporation

$(MBSE)^2$: Using MBSE to Architect and Implement the MBSE System

www.incose.org/symp2020

Orientation and Overview

- Bottom Line:
  - System architecting methods and MBSE can aid system development AND system model development
  - Create and use a model of the model we need to build
- Program Managers:
  - Commission the work to architect the models
  - Use models to monitor progress in achieving program maturity
- Systems Engineers:
  - Define appropriate requirements for the models
  - Flow down model requirements to model components to be built
  - Use the models to execute SE processes
- System Modelers:
  - Plan and execute the modeling effort using agile principles
  - Use the models to monitor progress in building the models

Uses for the Approach

- Use system architecting methods and MBSE to improve system model development
  - Create and use a model of the model we need to build
  - We call this the $(MBSE)^2$ System

- Use this $(MBSE)^2$ System to improve MBSE execution, e.g.,
  - Systems engineering reviews and audits
  - Acquisition and source selection processes
  - Defining CDRLs and DIDs for models
  - Change proposal impact assessments
  - Analyses of alternatives and trade studies
  - Enterprise integration of system-level models
  - Independent assessments

The MBSE System

- Implementing MBSE involves integration of many interdependent pieces
  - Descriptive system models
    - Models, modeling tools, model configuration management tools, model visualization tools, model collaboration tools, modeling methodology, etc.
  - Analytical models (M&S), design models
  - Databases and other data stores
  - Product/application lifecycle management tools, requirements management tools, etc.

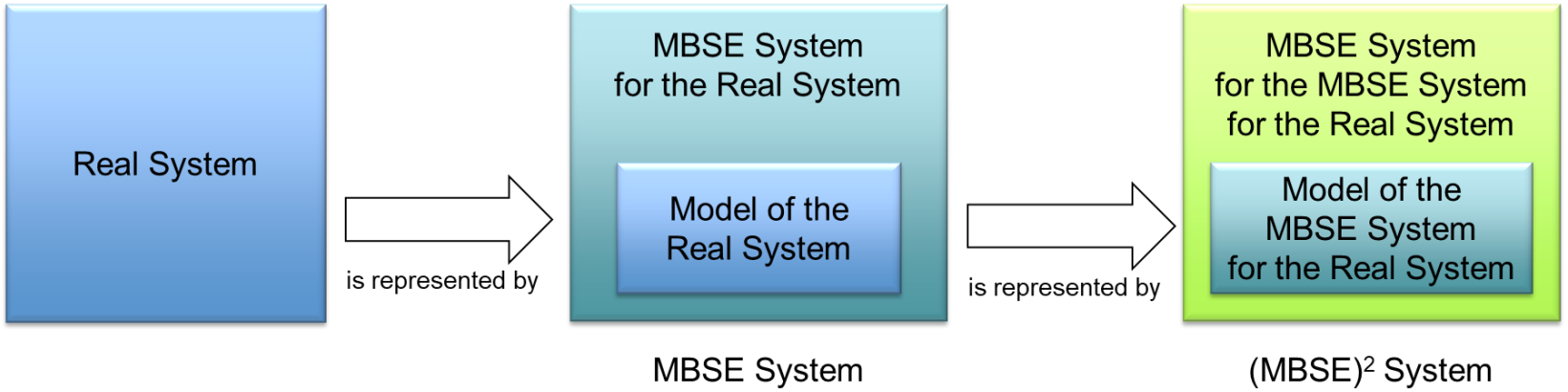
- These pieces should not be developed in isolation
- They should be architected into a coherent **MBSE System**
  - The descriptive models are among the most highly complex components
  - As with any non-trivial system, use systems engineering to architect and implement it

Comparing the MBSE System to the Real System

	The Real System	The MBSE System
Functional Requirements	• What must it do?	• What questions must it answer?
Performance Requirements	• How well must it do it?	• How well must it answer them?
Structure	• What system components comprise it? • How are those components connected?	• What model components comprise it? • How are those components connected?
Requirement Traceability	• Requirements trace to higher requirements	• Questions trace to higher questions and to the decisions they should inform
Requirement Allocation	• Requirements are allocated to sets of system components	• Questions are answered by sets of model components
Requirement Verification	• Test events verify requirements • Test events require components, • facilities, test equipment, etc.	• Model views answer questions for humans • Model views require model components, • tool infrastructure, etc.

The MBSE System and the $(MBSE)^2$ System

- MBSE for the Real System is implemented in an MBSE System
  - The model(s) of the Real System are part of the MBSE System
  - The MBSE System is used to Architect and Implement the Real System

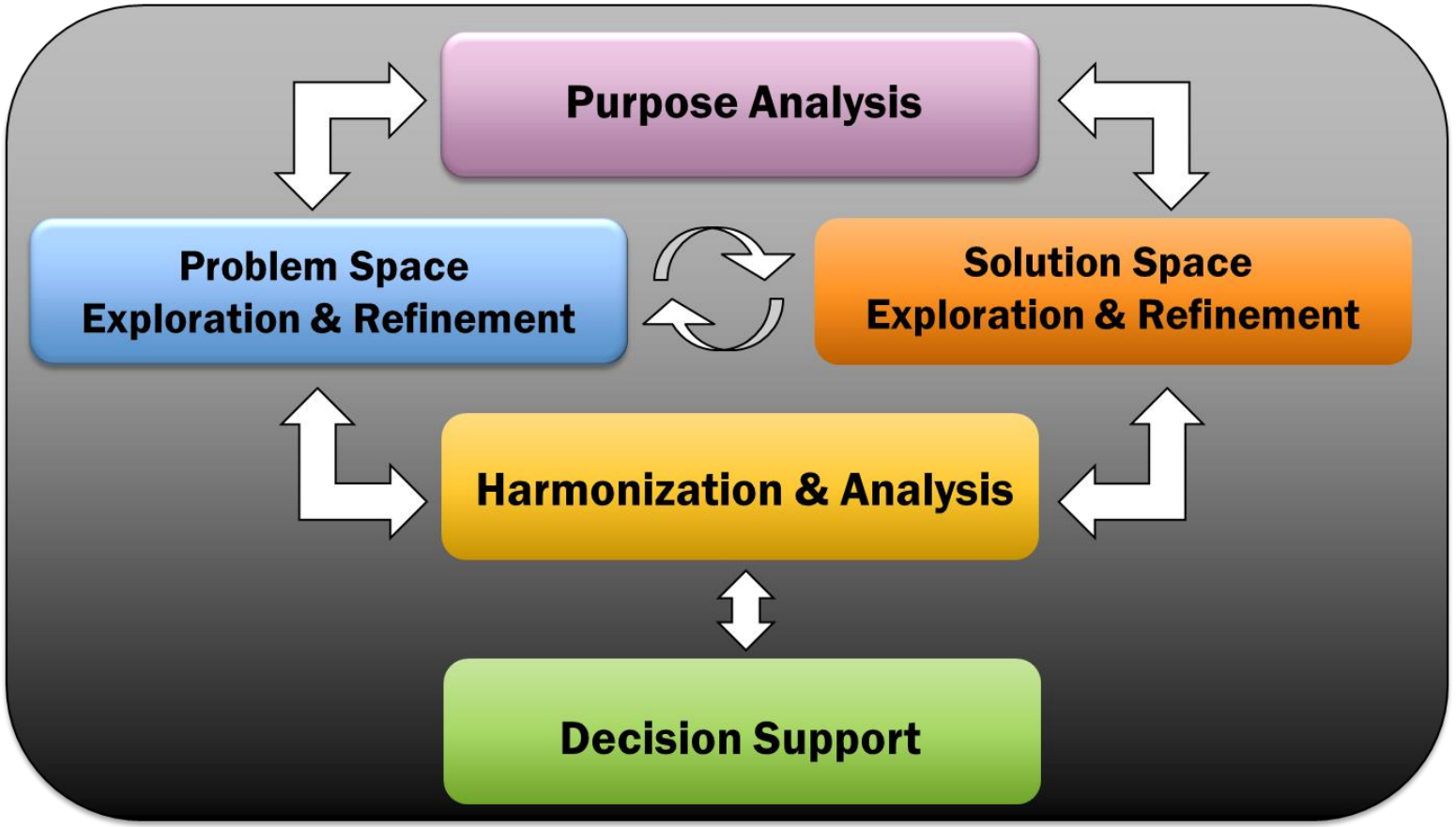


- MBSE for this MBSE System is implemented in an $(MBSE)^2$ System
  - The model(s) of the MBSE System are part of the $(MBSE)^2$ System
  - The $(MBSE)^2$ System is used to Architect, Implement, and Operate the MBSE System

Why Model the MBSE System?

- The modeling process can improve discipline in executing the architecting process
- The model of the MBSE System can help with managing the complex network of dependencies within the model of the Real System
  - The modeling environment enables capture, interrogation, and visualization of dependencies
  - Components of the $(MBSE)^2$ System can be linked to their counterparts in the MBSE System in the same modeling environment
- The experience of modeling the MBSE System can arm acquisition offices with valuable experience in modeling processes, methods, and tools in advance of the formal acquisition
  - And aid in the creation of model-based requirements to support that acquisition

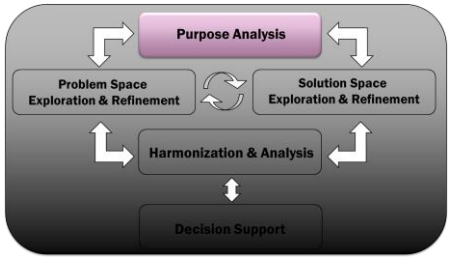
System Architecting Process



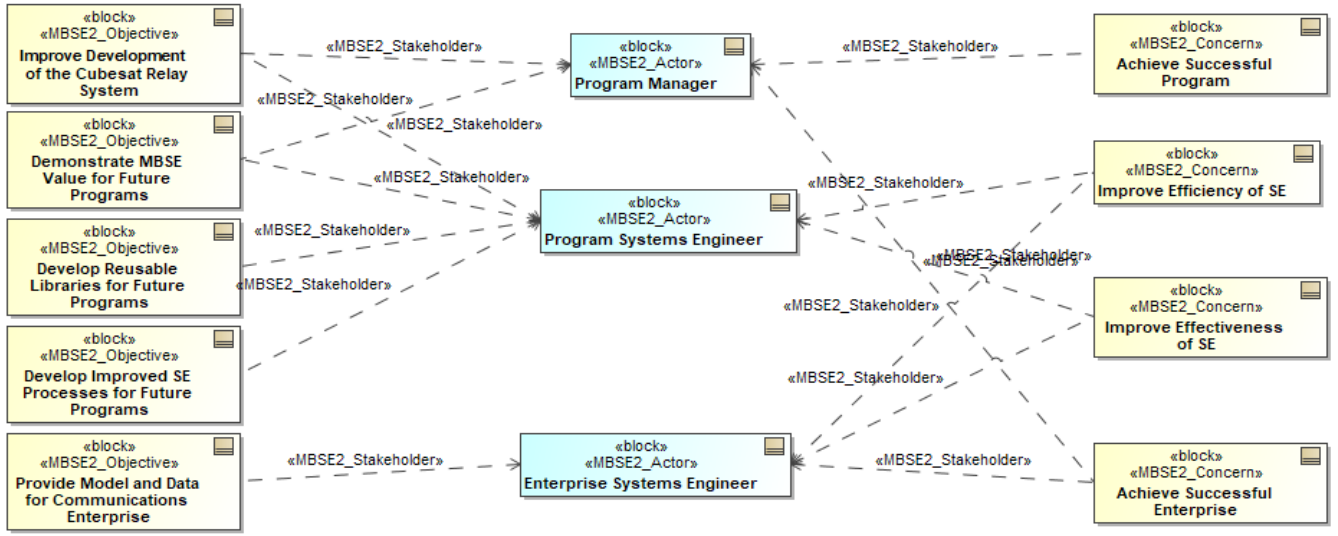
- Exercise both the Problem Space and Solution Space in concert—they are interdependent
- Harmonization is where Problem Space and Solution Space converge
- Objective is not to find the "best" solution
- Rather, the objective is to illuminate tradeoffs among competing objectives and solutions to inform stakeholder decisions

Architecting the MBSE System

Purpose Analysis

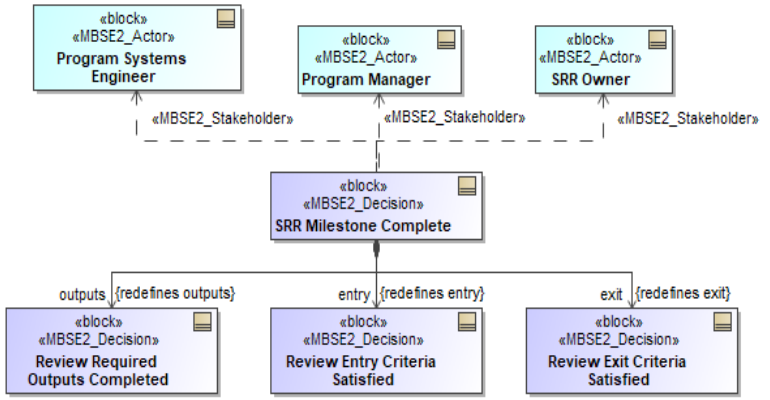


- Define high-level **Objectives** for the MBSE System
  - i.e., What are the reasons for implementing MBSE?
    - Improve development of the Real System?
    - Improve operation of the Real System?
    - Advance the practice of MBSE?



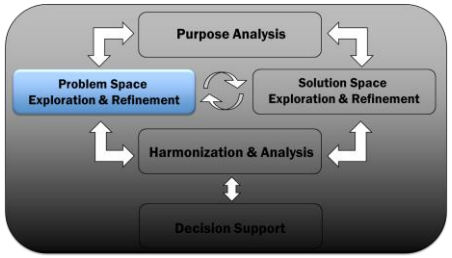
- Identify key **Actors** for the MBSE System
  - Includes all stakeholders for the Real System
  - For each Actor, identify their key **Concerns**

- Each part of the MBSE System should be traceable to important Objectives, Concerns, and Actors

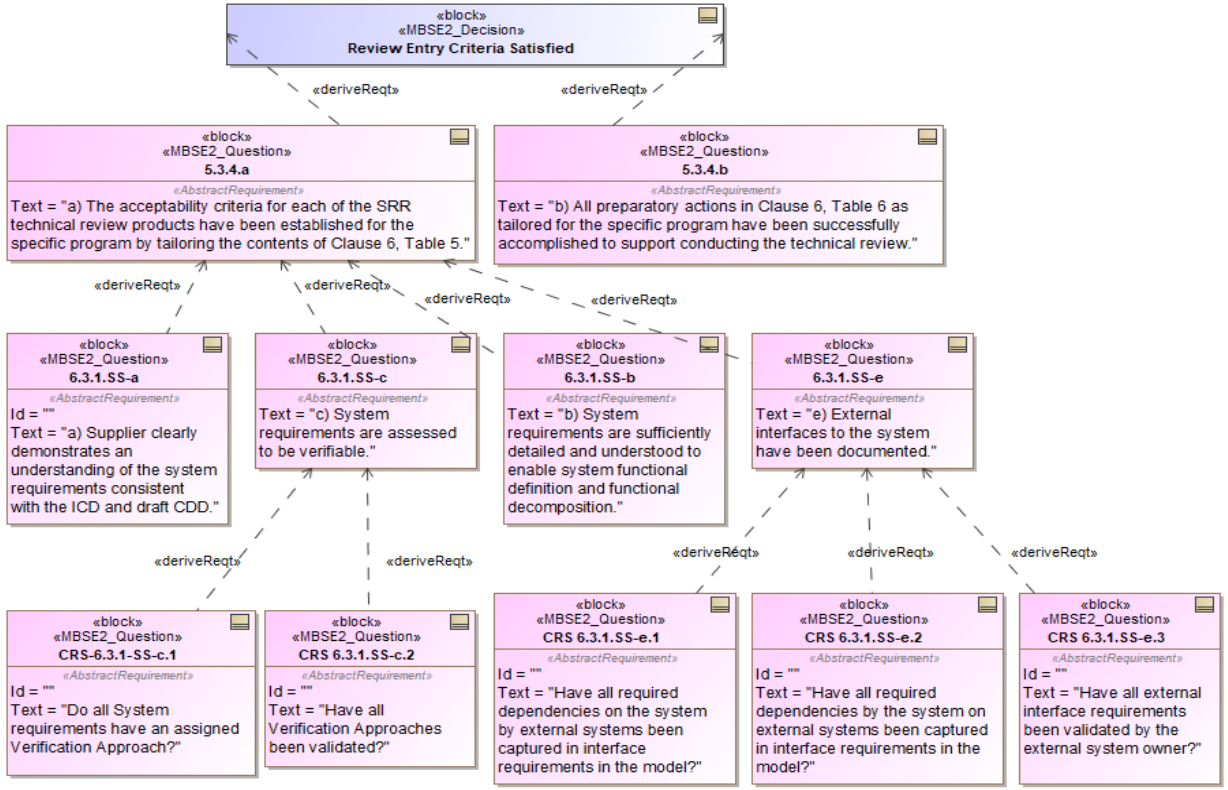


Architecting the MBSE System

Problem Space Exploration & Refinement



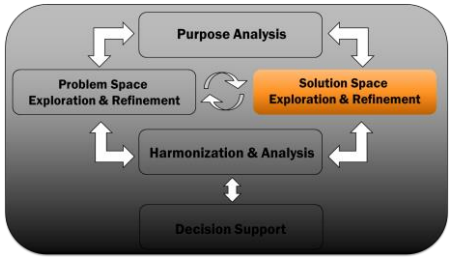
- Identify key **Decisions** about the Real System
  - Key program milestones
  - Key programmatic decisions
  - Key operational decisions
- For each Decision and each Concern, identify the **Questions** that need to be answered
  - Questions may be chained into (or derived from) process workflows
- Every part of the MBSE System should be traceable to (answer) one or more Questions
- **Problem Framing Workshops*** found to be effective for fleshing out the problem space



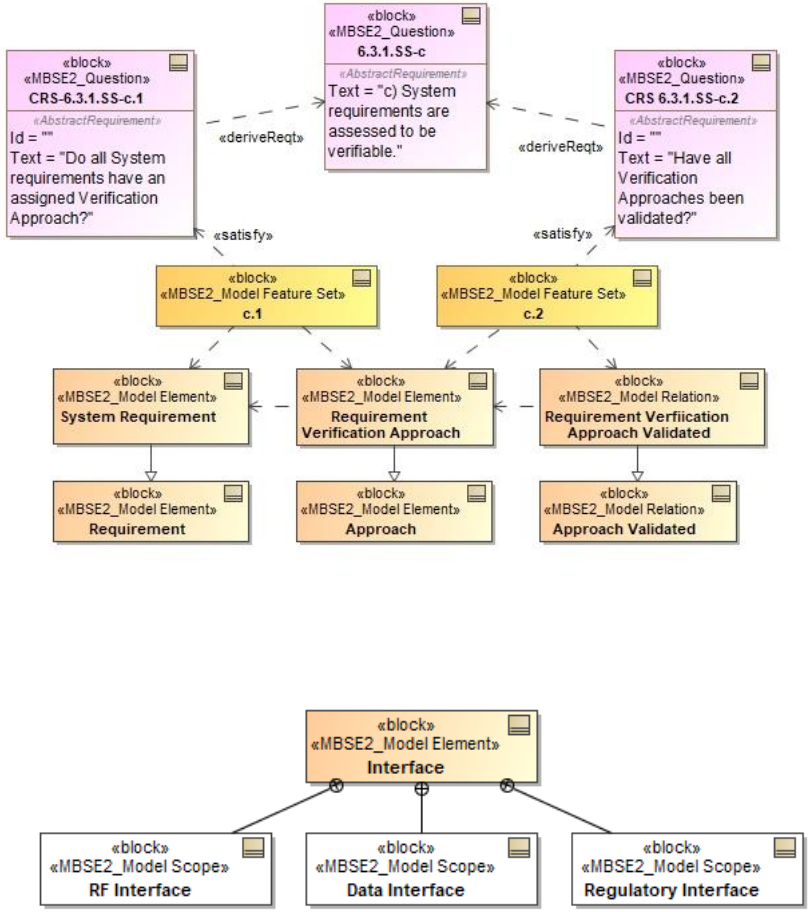
* See James Martin, *Problem Framing - Identifying the Right Models for the Job*, INCOSE International Symposium, Orlando, FL, 2019

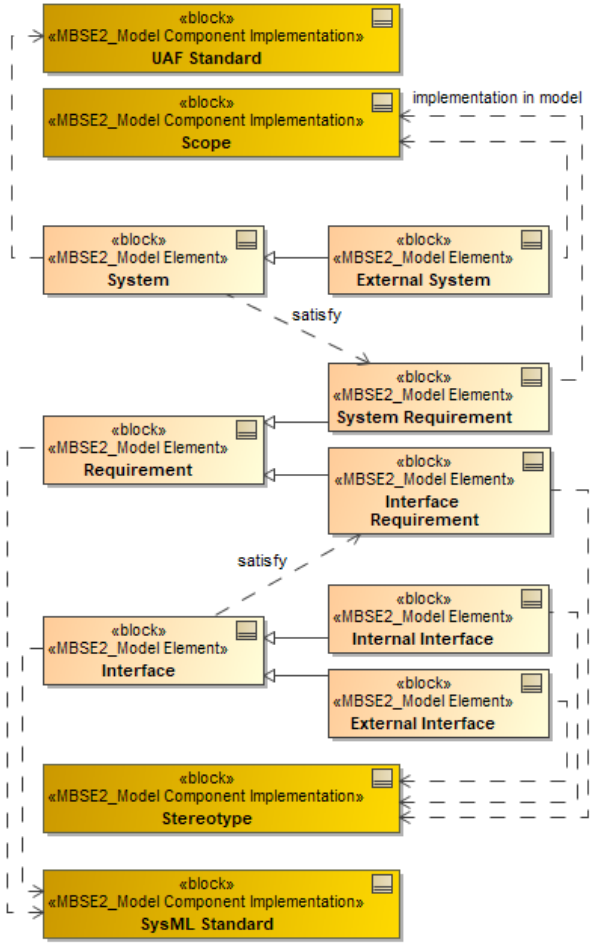
Architecting the MBSE System

Solution Space Exploration & Refinement



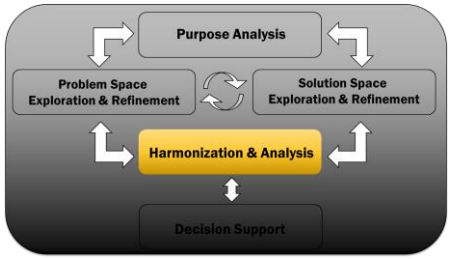
- Identify the complete **Feature Sets** of **Model Components** needed to answer each Question
  - Model elements, relationships, attributes, metadata
- Create a **Conceptual Model** of the Model Components
  - Documents how they should be used together
- Identify useful subsets of the total **Scope** of Model Components
  - Answer Questions incrementally by incrementally implementing scope
    - Subset of elements or relationships
    - Depth of dependency chain



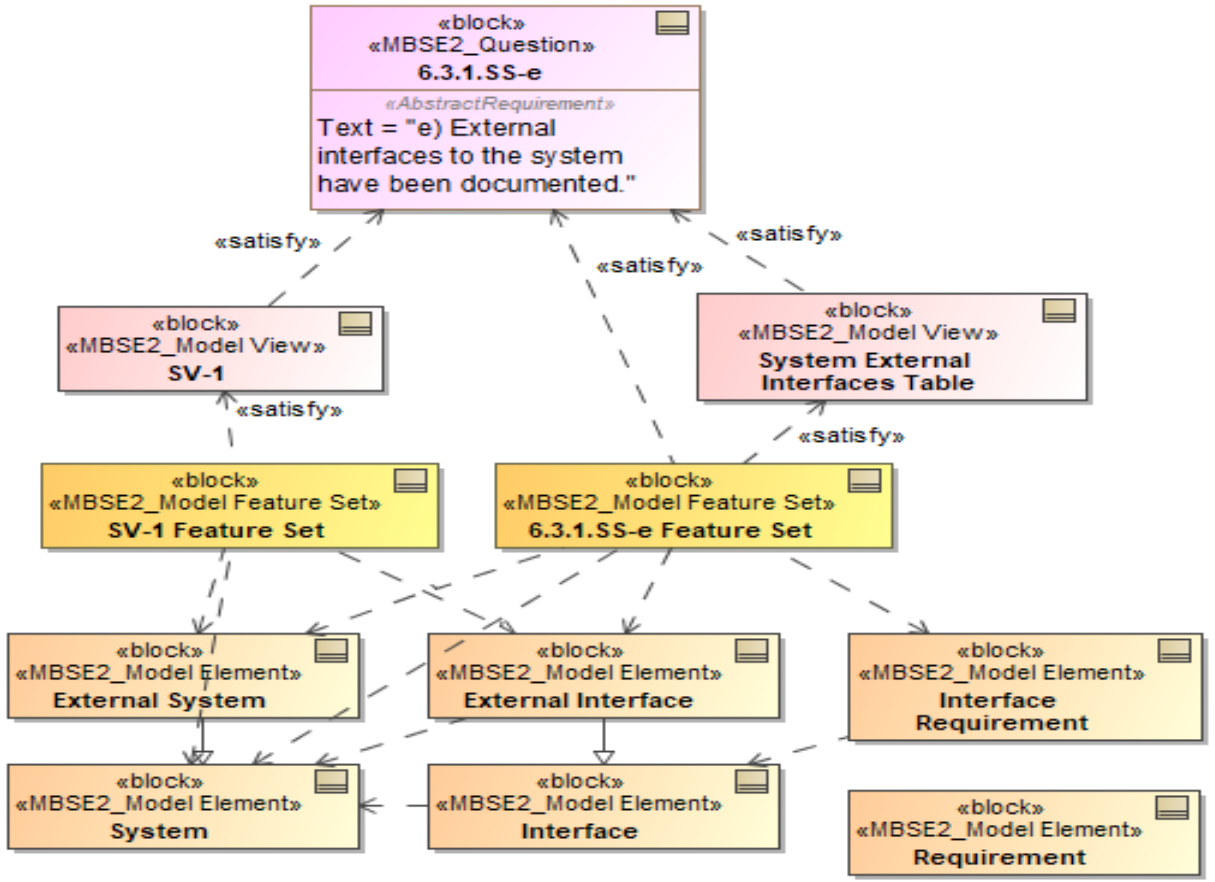


Architecting the MBSE System

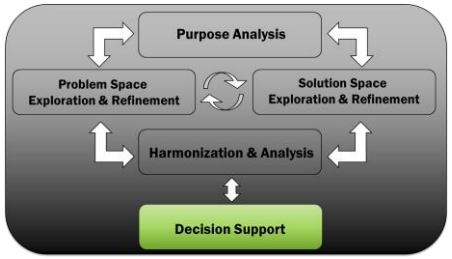
Harmonization and Analysis



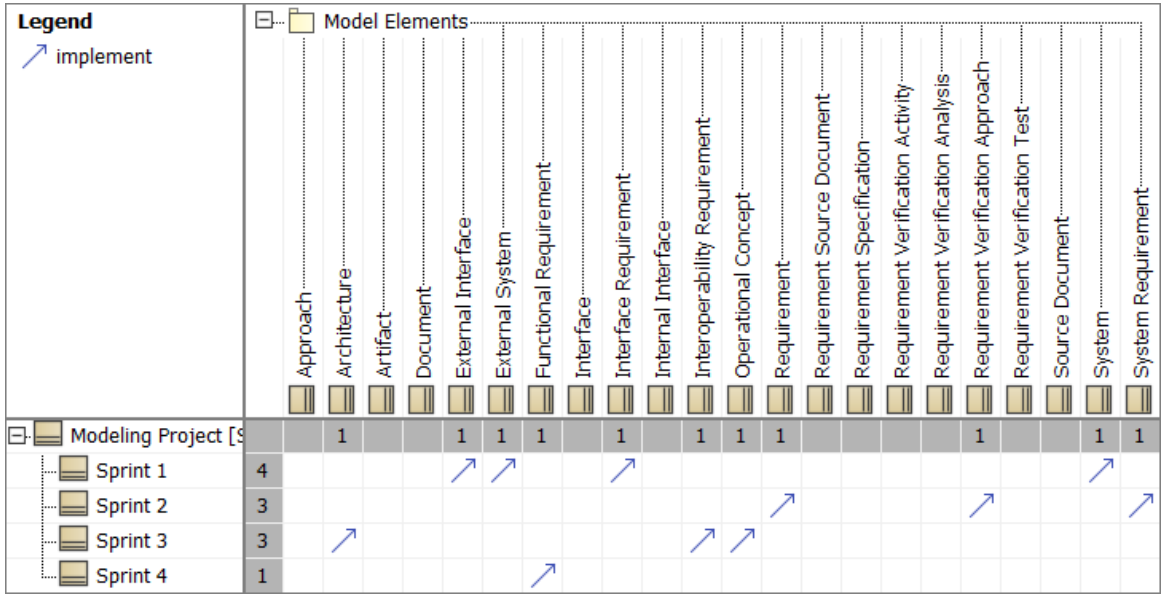
- Identify the **View** that answers each Questio
  - Identify Model Components needed
  - If more than one View is needed to answer a Question, the Question may be at too high a level
- Identify needed **Model View Construction Components** to create views
  - Query language constructs
  - Report templates
  - Metachain or constraint expressions
  - Automation scripts
- Model Components and Model View Construction Components comprise the **Feature Sets** needed to complete a View

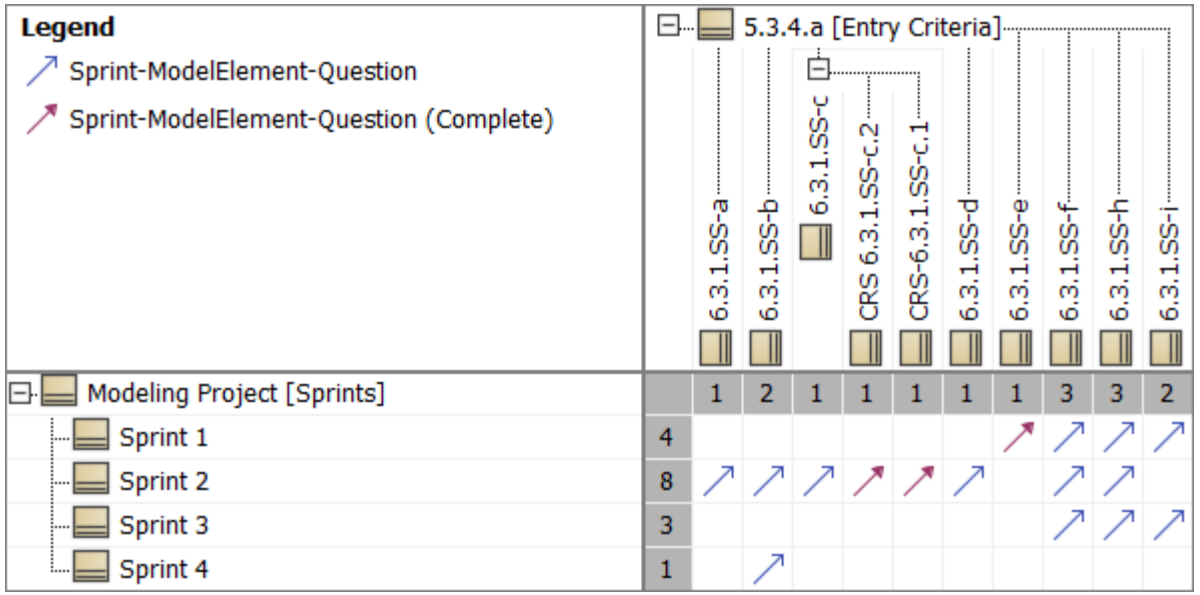


Implementing the MBSE System
Decision Support



- Decide what to Build and Plan the Modeling Effort using Agile development principles
  - Develop an Initial List of Questions, Views, Components, etc. (Backlog)
  - Choose and prioritize Questions to be answered (User Stories)
  - Prioritize Feature Sets to be built to answer the Questions (Features)
  - Decide scope of each sprint—answer one or more Questions per sprint (Sprint planning)

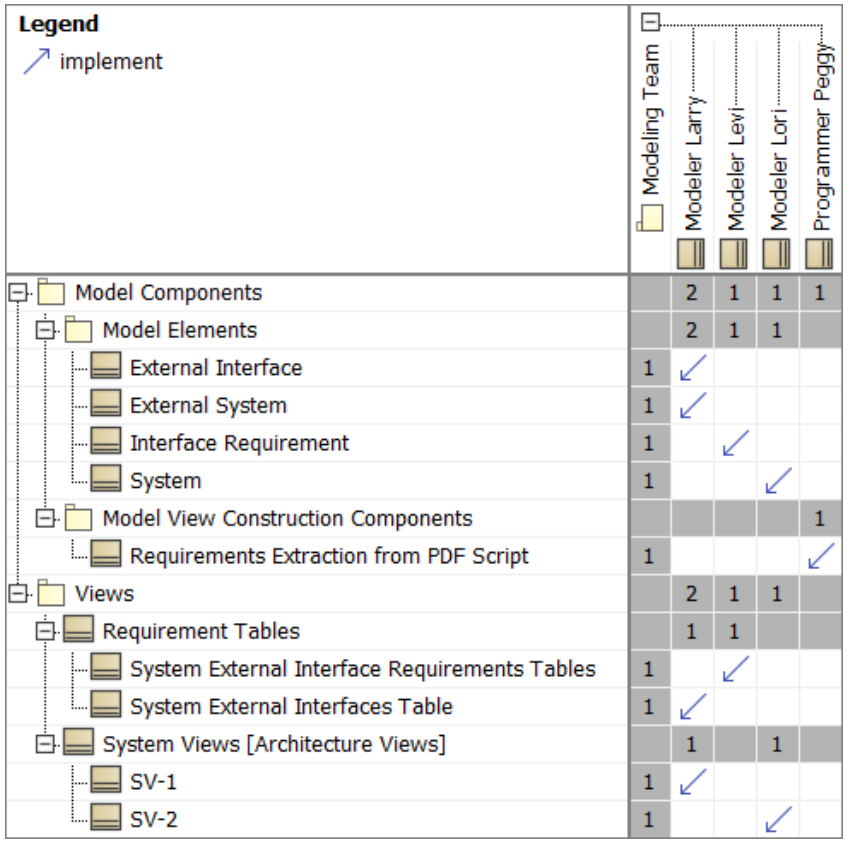




Implementing the MBSE System

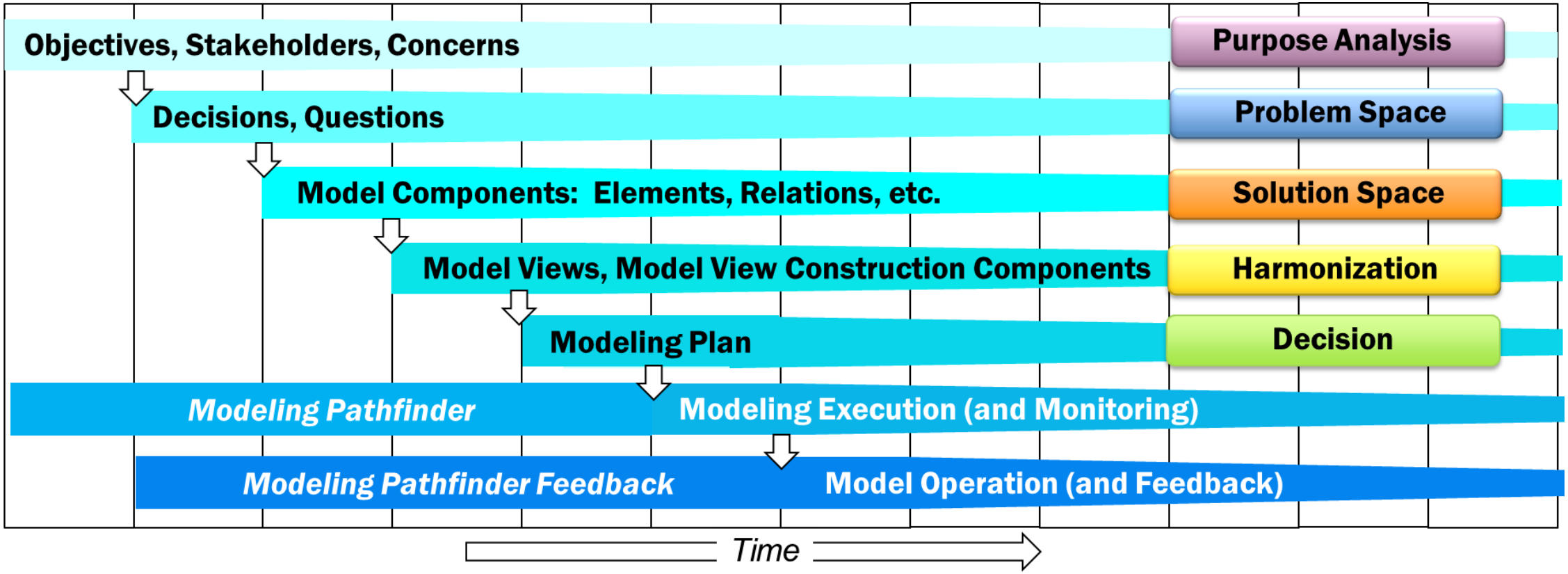
The (MBSE)^2 System as a Development Dashboard

- Execute the plan using Agile development principles
  - Monitor progress using the (MBSE)^2 System
    - Impact analysis, coverage/gap analysis
  - Measure progress by Questions answered
    - Quantity
    - Quality (scope, depth, fidelity)
  - Solicit feedback from Question owners
  - Adjust (MBSE)^2 model based on feedback and learning
    - Update backlog and modeling plan based on evolving knowledge



Implementing the MBSE System

Staggered Streams Enables Agile Development



Stream starts do not need to be synchronous
All streams continue through life of MBSE System
Use Pathfinders to bootstrap the effort and avoid "analysis paralysis"

Operating the $(MBSE)^2$ System

Example Use Case: Model Based Systems Engineering Reviews

Source Material (Requirements, Architecture, CONOPS, Test Plans, etc.)

- Static artifacts go “stale” quickly, sometimes immediately
- Static artifacts are often easily misinterpreted
- Static artifacts grow in inconsistency over time—increasing entropy

- Models are the life-blood of knowledge management—always current
- Models are rigorously defined with far less ambiguity
- Models have automated enforcement of internal consistency

Review Events (e.g., SRR, SDR, PDR, CDR, etc.)

- Discrete review events separated by many months—defects can linger a long time
- Limited to information provided in discrete large data drops
- Changes are implemented offline after events, often requiring delta events
- Success criteria are inconsistently interpreted and addressed

- **Continuous, proactive assessment**, minimizing time to discover defects
- Comprehensive assessment of linked data across the enterprise
- Changes implemented immediately, enabling **quicker impact assessment**
- Success criteria are consistently applied and **proactively drive knowledge/model creation**

Operating the $(MBSE)^2$ System

Example Use Case: Model Based Systems Engineering Reviews

- Objectives
  - Define knowledge to be assessed for maturity at each milestone
    - Provide to contractors as “requirements” for model creation and delivery
  - Facilitate orderly review of knowledge against success criteria
  - Adapt quickly to evolving knowledge
    - Use modeling tools to perform impact and coverage analysis

- Actors (Stakeholders and Beneficiaries)
  - Program Manager
  - Program Systems Engineer
  - Enterprise Systems Engineer
  - Program Review Owner
  - MBSE System Modelmaster(s)
  - Subject matter experts (SMEs)

- Concerns
  - All questions (review success criteria) answered?
  - Risks identified, quantified, and adjudicated?
  - Implications to enterprise identified (and acceptable)?

- Decisions
  - Has the milestone been successfully achieved?
  - Are tradeoffs identified and understood?

Operating the (MBSE)^2 System

Example Use Case: Model Based Systems Engineering Reviews

- Questions
  - Define success criteria for the review
  - Refine criteria until clear how to verify satisfaction

5.3.4 SRR entry criteria

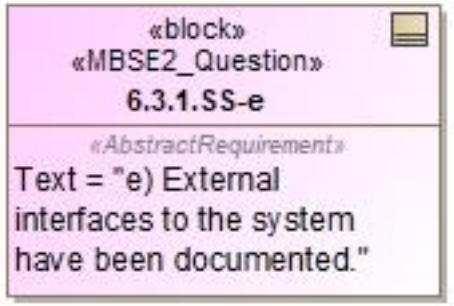
Ref: IEEE 15288.2 Standard

The SRR shall be conducted only after the following events have been successfully completed:

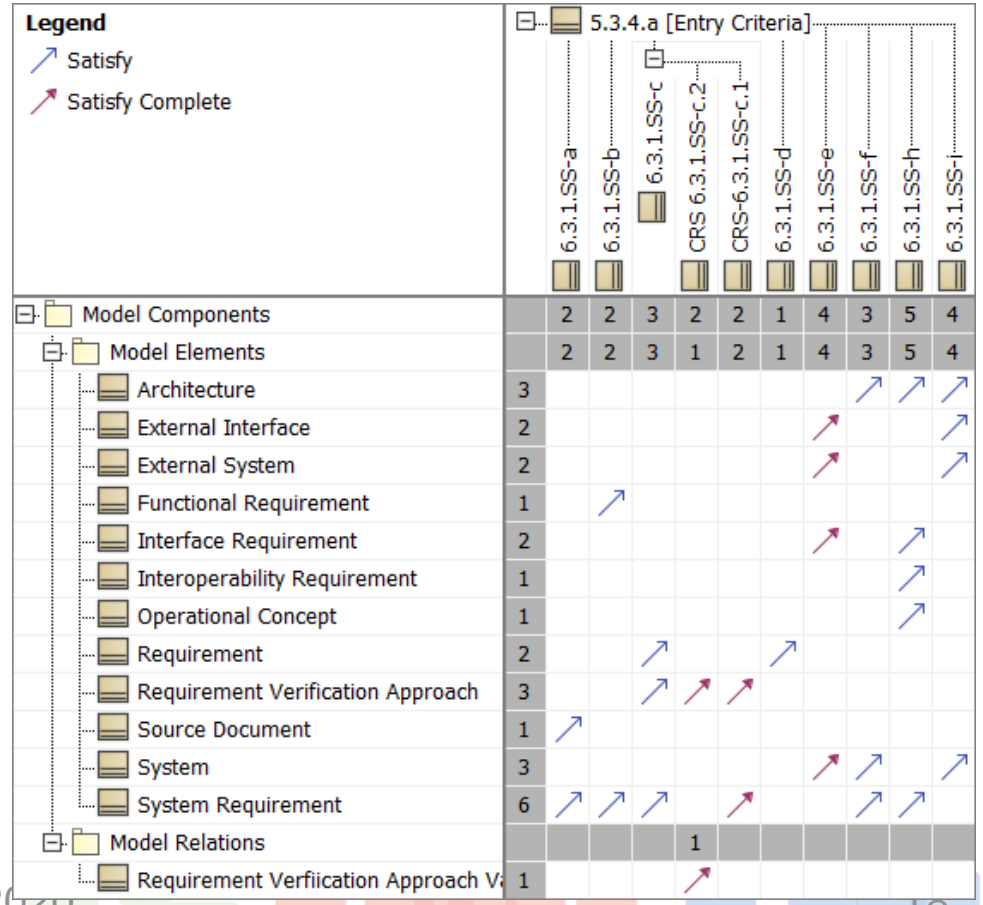
a) The acceptability criteria for each of the SRR technical review products have been established for the specific program by tailoring the contents of Clause 6, Table 5.

Table 5—SRR technical review products acceptability criteria

Product	SRR acceptability criteria
	a) Supplier clearly demonstrates an understanding of the system requirements consistent with the ICD and draft CDD.
	b) System requirements are sufficiently detailed and understood to enable system functional definition and functional decomposition.
	c) System requirements are assessed to be verifiable.
	d) Requirements can be met given the technology maturity.
	e) External interfaces to the system have been documented.
	f) System requirements have been synthesized into conceptual architectures.
	g) System conceptual architecture alternatives are developed and assessed in the context of engineering trade space, technical requirements, system specification, risks (technical, programmatic, schedule, cost), LCC and cost as an independent variable (CAIV).
	h) The preliminary system conceptual architectures support implementation of operational concepts, interoperability objectives, and system and external interface requirements.



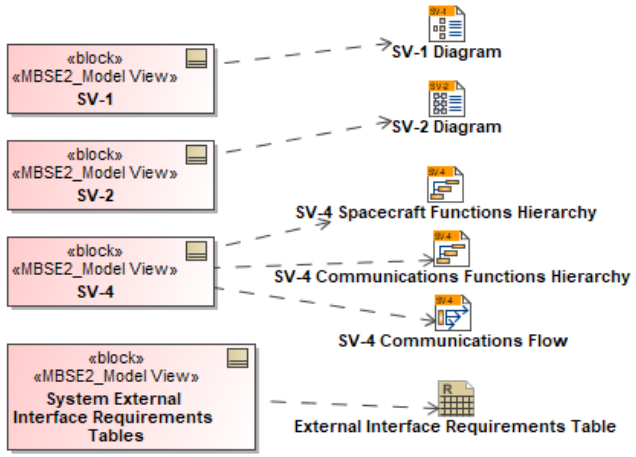
- Model Components
  - Define needed model content to answer each question

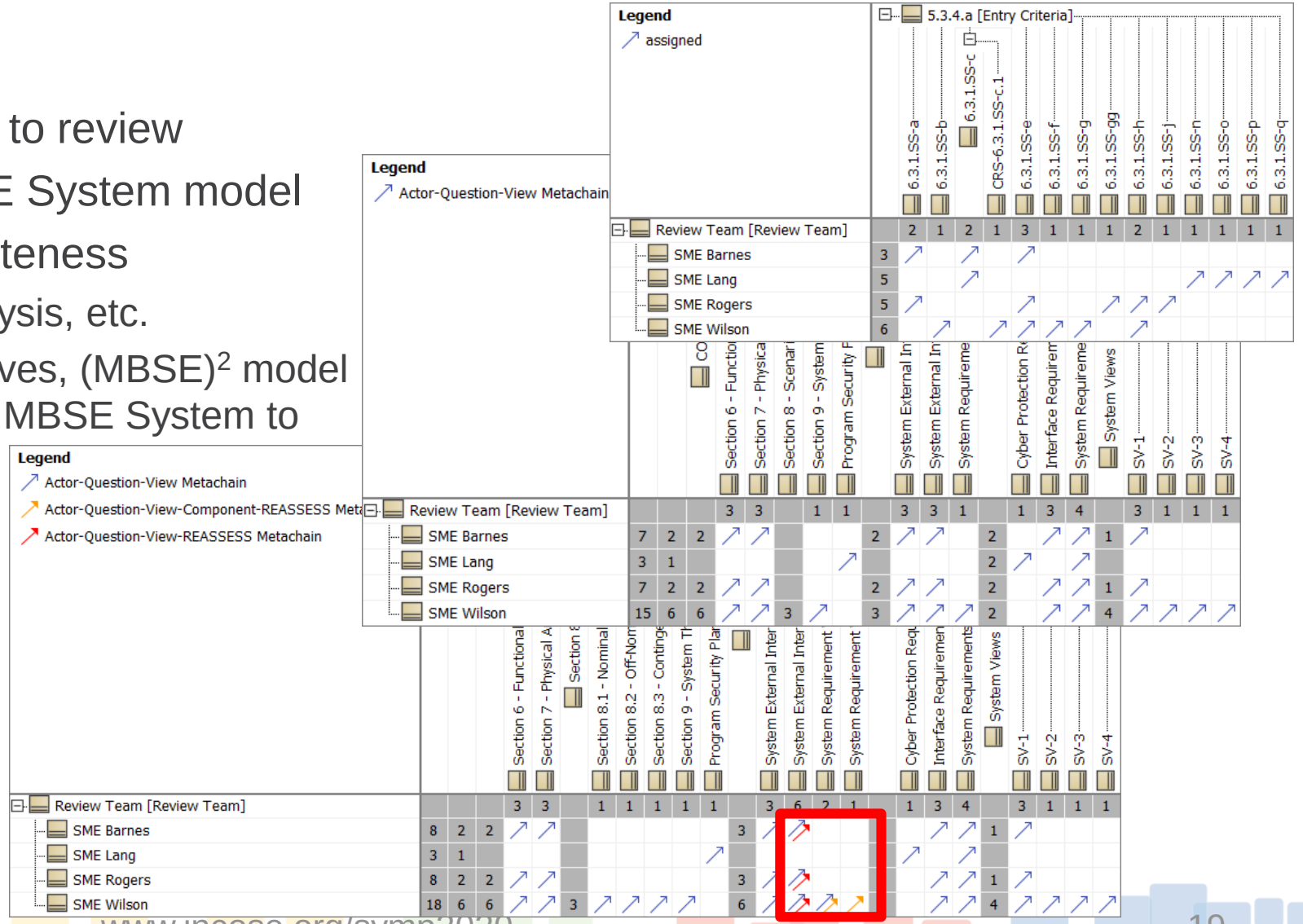


Operating the (MBSE)^2 System

Example Use Case: Model Based Systems Engineering Reviews

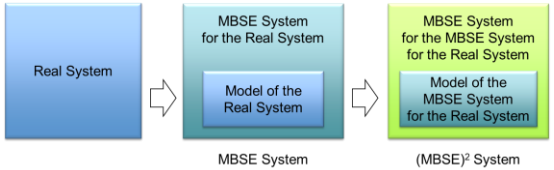
- Assign Questions to reviewers
- Determine Views reviewers need to review
- Link Real System model to MBSE System model
- Assess and track Review completeness
  - Coverage analysis, impact analysis, etc.
  - As the Real System model evolves, (MBSE)^2 model reveals potential impacts to the MBSE System to trigger reassessments

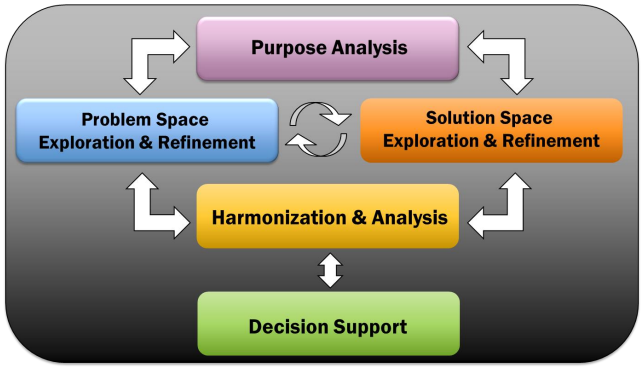


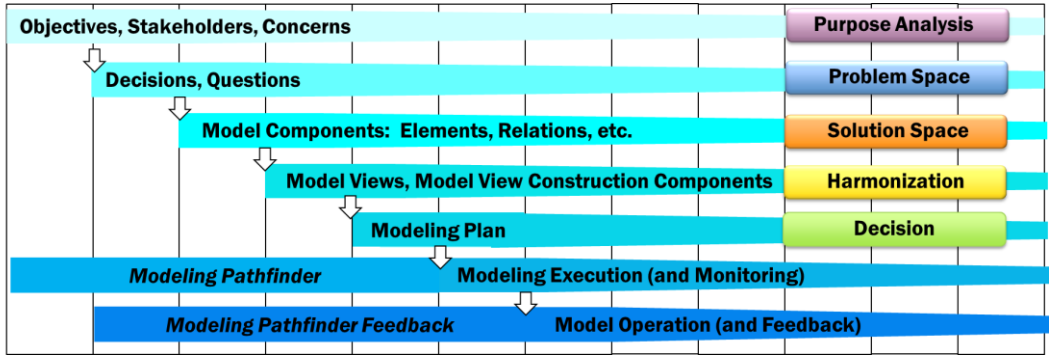


Key Takeaways

- Implementing MBSE is really implementing an **MBSE System**
- Apply system architecting methods to **architect** the MBSE System
- Use an **agile** process to architect, implement the MBSE System
  - Staggered parallel processes facilitate continuous progress, feedback, and evolution
- An $(MBSE)^2$ System model can facilitate management of complexity & traceability
  - Link the MBSE System with the $(MBSE)^2$ System
  - Use the $(MBSE)^2$ System to **manage** the implementation of the MBSE System
  - Use the $(MBSE)^2$ System to **operate** the MBSE System







30th Annual INCOSE international symposium

Virtual Event

July 20 - 22, 2020

www.incose.org/symp2020