

Raytheon

Customer Success Is Our Mission

Air
Land
Sea
Space
Cyberspace

Innovation. In all domains.

Robust Architecture Development: SysML Usage across Industry Tools

Geoff Martin

Raytheon IDS, San Diego
gdmartin@raytheon.com

Rick Steiner

Raytheon IDS, San Diego
fsteiner@raytheon.com

October 25th, 2008

Objectives

- Provide the case that Systems Modeling Language (SysML) is most effective for capturing complex system architectures.
- Demonstrate that the architect needs to consider the interfaces, nuances, and customer demands of the system when selecting modeling tool.
- Provide guidance on how to be master of your model vs. its slave with proper organization and consistency.

Agenda

- The Why's and How's of a Systems Architecture Model
- Consider Your System Needs
- Finding the Right Tool-fit
- System concept – check, Tool – check, Model Organization – huh?
- Maintaining Model Consistency and Allocation

How to express and develop a complex architecture model?

Why Model a System Architecture?	Needs from a Modeling Language	SysML Architecting Capabilities
Segment a complex system into workable and definable elements.	<ul style="list-style-type: none"> - Capture behavior of system - Capture interactions of system elements - Capture containment of system - Capture usage/hierarchy of system - Represent flow between system elements 	<ul style="list-style-type: none"> - Activity Diagrams - Sequence Diagrams - Block Definition Diagrams - Internal Block Diagrams - Links and Flow on IBDs
Rapidly realize and respond to changes in system design.	Provide a architecting method that is adaptable to system change and regeneration of system descriptions documentation.	<ul style="list-style-type: none"> - Swimlanes integrated into Activity Diagrams - New and/or changed requirements are verified and satisfied in system behavior and structure artifacts.
Show system behavior to external triggers.	Identify external elements and constraints they place on system.	<ul style="list-style-type: none"> - Use Cases - State Machines
Flesh out requirements and interface definitions of the system.	Provide construct to capture system make-up, specifications, performance and linkages therein.	<ul style="list-style-type: none"> - Parametric Diagrams - Requirement Diagrams
Provide logical and concise documentation of system.	Create model artifacts that describe system concepts to instantiation.	Provide easy translation into DoDAF artifacts while still providing ability to further analyze architecture.

SysML provides an ideal framework for representing the structure, behavior, and performance of an Architecture.

How to express and develop a complex architecture model?

Why Model a System Architecture?	Needs from a Modeling Language	SysML Architecting Capabilities
Segment a complex system into workable and definable elements.	<ul style="list-style-type: none"> - Capture behavior of system - Capture interactions of system elements - Capture containment of system - Capture usage/hierarchy of system - Represent flow between system elements 	<ul style="list-style-type: none"> - Activity Diagrams - Sequence Diagrams - Block Definition Diagrams - Internal Block Diagrams - Links and Flow on IBDs
Rapidly realize and respond to changes in system design.	Provide a architecting method that is adaptable to system change and regeneration of system descriptions documentation.	<ul style="list-style-type: none"> - Swimlanes integrated into Activity Diagrams - New and/or changed requirements are verified and satisfied in system behavior and structure artifacts.
Show system behavior to external triggers.	Identify external elements and constraints they place on system.	<ul style="list-style-type: none"> - Use Cases - State Machines
Flesh out requirements and interface definitions of the system.	Provide construct to capture system make-up, specifications, performance and linkages therein.	<ul style="list-style-type: none"> - Parametric Diagrams - Requirement Diagrams
Provide logical and concise documentation of system.	Create model artifacts that describe system concepts to instantiation.	Provide easy translation into DoDAF artifacts while still providing ability to further analyze architecture.

SysML provides an ideal framework for representing the structure, behavior, and performance of an Architecture.

How to express and develop a complex architecture model?

Why Model a System Architecture?	Needs from a Modeling Language	SysML Architecting Capabilities
Segment a complex system into workable and definable elements.	<ul style="list-style-type: none"> - Capture behavior of system - Capture interactions of system elements - Capture containment of system - Capture usage/hierarchy of system - Represent flow between system elements 	<ul style="list-style-type: none"> - Activity Diagrams - Sequence Diagrams - Block Definition Diagrams - Internal Block Diagrams - Links and Flow on IBDs
Rapidly realize and respond to changes in system design.	Provide a architecting method that is adaptable to system change and regeneration of system descriptions documentation.	<ul style="list-style-type: none"> - Swimlanes integrated into Activity Diagrams - New and/or changed requirements are verified and satisfied in system behavior and structure artifacts.
Show system behavior to external triggers.	Identify external elements and constraints they place on system.	<ul style="list-style-type: none"> - Use Cases - State Machines
Flesh out requirements and interface definitions of the system.	Provide construct to capture system make-up, specifications, performance and linkages therein.	<ul style="list-style-type: none"> - Parametric Diagrams - Requirement Diagrams
Provide logical and concise documentation of system.	Create model artifacts that describe system concepts to instantiation.	Provide easy translation into DoDAF artifacts while still providing ability to further analyze architecture.

SysML provides an ideal framework for representing the structure, behavior, and performance of an Architecture.

How to express and develop a complex architecture model?

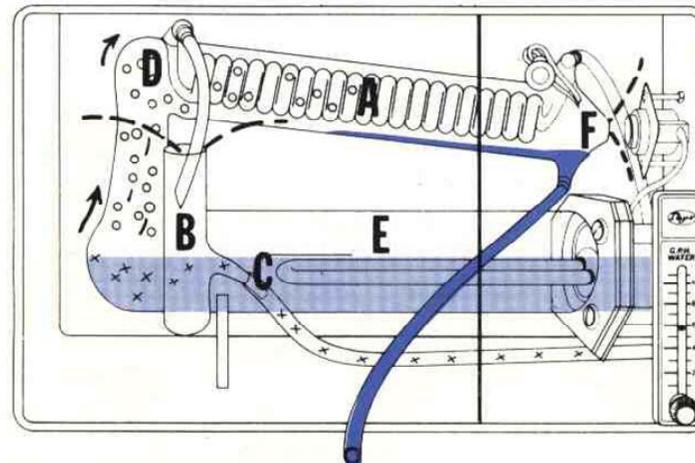
Why Model a System Architecture?	Needs from a Modeling Language	SysML Architecting Capabilities
Segment a complex system into workable and definable elements.	<ul style="list-style-type: none"> - Capture behavior of system - Capture interactions of system elements - Capture containment of system - Capture usage/hierarchy of system - Represent flow between system elements 	<ul style="list-style-type: none"> - Activity Diagrams - Sequence Diagrams - Block Definition Diagrams - Internal Block Diagrams - Links and Flow on IBDs
Rapidly realize and respond to changes in system design.	Provide a architecting method that is adaptable to system change and regeneration of system descriptions documentation.	<ul style="list-style-type: none"> - Swimlanes integrated into Activity Diagrams - New and/or changed requirements are verified and satisfied in system behavior and structure artifacts.
Show system behavior to external triggers.	Identify external elements and constraints they place on system.	<ul style="list-style-type: none"> - Use Cases - State Machines
Flesh out requirements and interface definitions of the system.	Provide construct to capture system make-up, specifications, performance and linkages therein.	<ul style="list-style-type: none"> - Parametric Diagrams - Requirement Diagrams
Provide logical and concise documentation of system.	Create model artifacts that describe system concepts to instantiation.	Provide easy translation into DoDAF artifacts while still providing ability to further analyze architecture.

SysML provides an ideal framework for representing the structure, behavior, and performance of an Architecture.

OMG SysML Tutorial (omgsysml.org)

Water Distiller Example

- Functional Analysis based, not OOA
 - Relies heavily on activity diagrams and functional allocation
- Solution to problem focused on activity modeling, flow allocation, item flows & parametrics
 - Heat balance of distiller relies on properties of water flowing through system
- Traditional UML tools just don't do these things



**Consideration is needed on how to represent the aspects of the system:
Behavior, Item Flows, Performance/Parametrics, Allocation, etc.**

Tool Comparison For Distiller Example

- No tool “fully” implements SysML
- Clearly, each tool has strengths & weaknesses
 - Make sure tool is compatible with your method
- Other tools exist, but not evaluated

	Enterprise Architect ver 7.1	Magic Draw ver 15.1	Rhapsody ver 7.2	RS(X) ver 7.0.5 E+ SysML ver 2.0.5.1
Activity Modeling	full	full	limited	full
Structural Modeling	full	full	full	full
Item Flows	limited	full	full	limited
Ports/Interfaces	full	limited	full	full
Functional Allocation	yes	yes	yes	yes
Flow Allocation	none	yes	yes	yes
Parametrics	full	full	full	full
Code Gen/Animation	none	none	yes	yes
Requirements	full	full	full	full
Distiller Model Source	Steiner	Steiner	Lussier	Steiner
UML4SysML 2.1	most	all	most	most

SysML Diagrams— a Method for Model Integration

- 3 separate hierarchies of Structure, Behavior, and Data
 - Usage (internal connection) is documented with separate diagrams
- These 3 hierarchies maintained at Operational and System level

	Hierarchy	Usage		Cross-Connect
Structure	bdd	ibd		act (swimlane), seq (lifeline, op)
Behavior	bdd	act, stm		ibd (itemFlow), seq (msgType)
Data	bdd	(none)		act (objFlow), seq (msg,op), stm

bdd = Block Definition Diagram (no DoDAF)

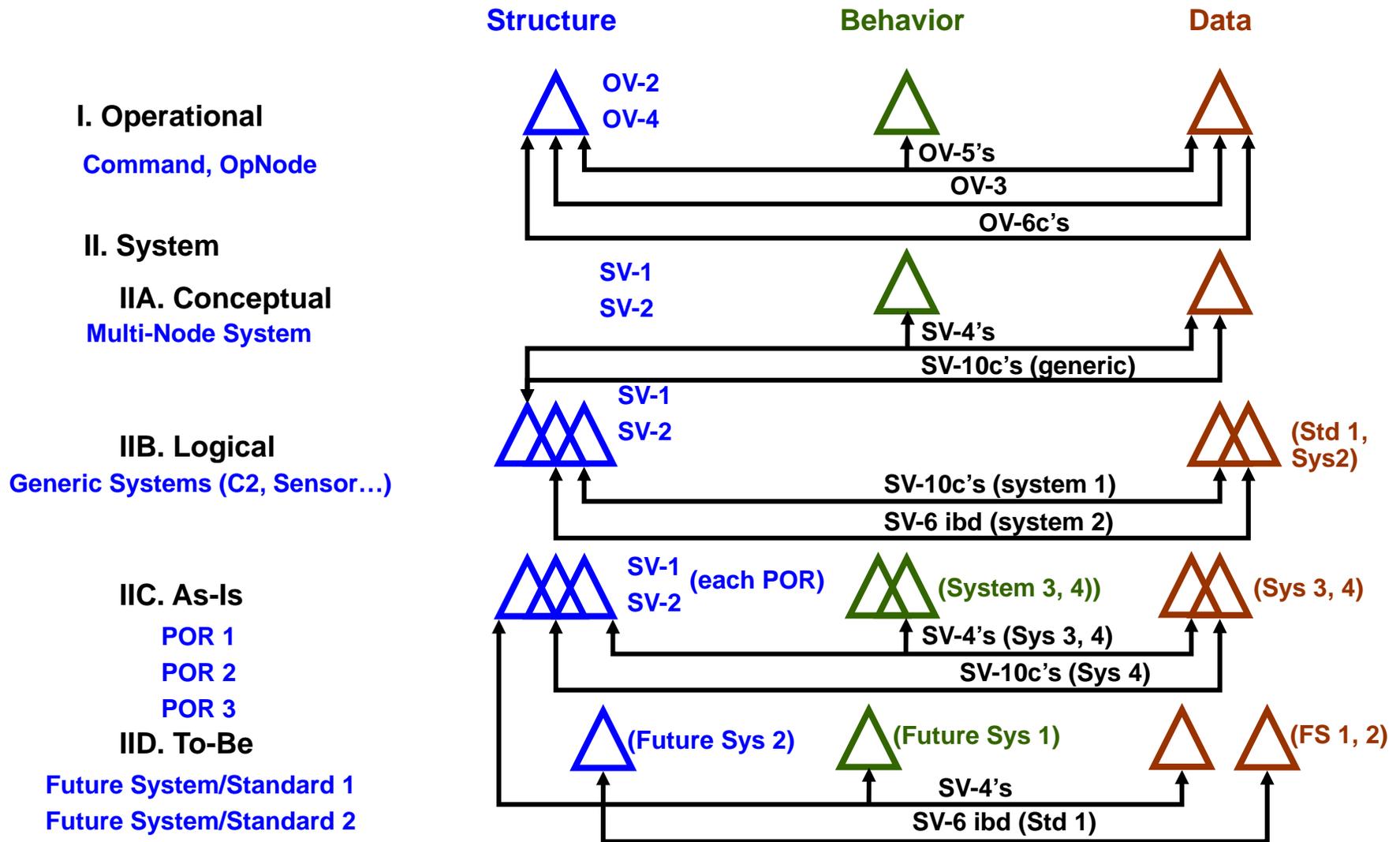
ibd = Internal Block Diagram (OV-2, SV-1, SV-2)

act = Activity Diagram (OV-5, SV-4)

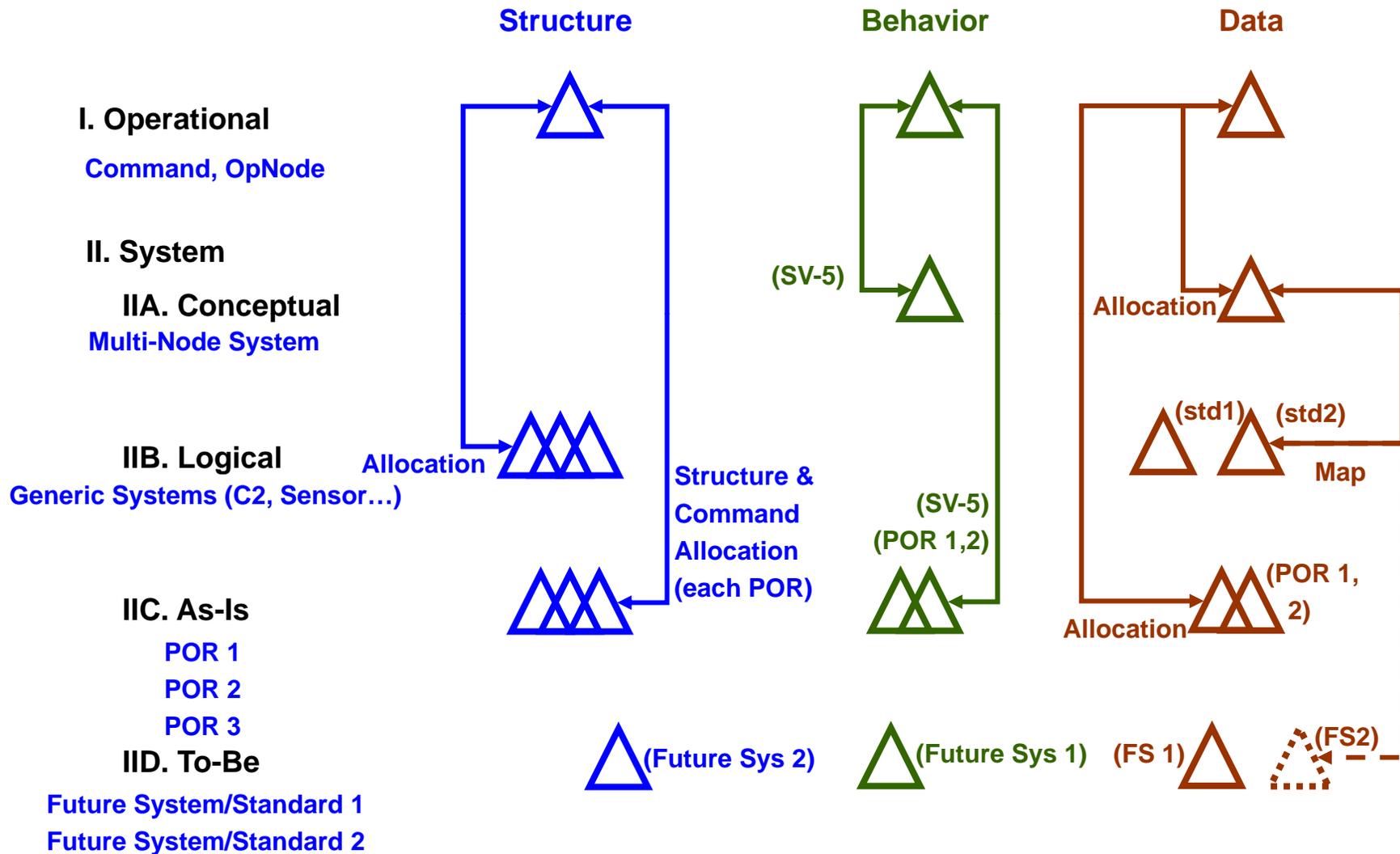
seq = Sequence Diagram (OV-6c, SV-10c)

stm = State Machine Diagram (OV-6b, SV-10b)

DoDAF Views Horizontally Cross-Connecting a Complex SoS Model



Allocation Vertically Cross-Connecting a Complex SoS Model



Triangles represent hierarchy diagrams (no DoDAF equivalent)

Conclusions and some wisdom...

- The variable aspects (interfaces, information exchange, etc.) of MSI can be effectively studied, represented and communicated through a systems architecture model.
- Understanding of the system concept is a key discriminator in modeling tool selection.
- Don't model yourself into a corner...consider how to organize the product (logical, near/far term, classification, etc.).
- Two modelers is company, three's a crowd...keep the peace with configuration management.
- The model is good for the architect, but what does the customer want? That is, make sure the model provides standard outputs for team and customer reviews.