

MISSION:

A WORLD OF INNOVATION

From Theory to Reality: *Taking the Fear Out of Model Based Systems Engineering*

Chris Finlay
Raytheon Company
401.842.2691
finlayc@raytheon.com

Stacy Dujardin
Raytheon Company
401.842.4162
Stacy_Dujardin@raytheon.com

10/29/14

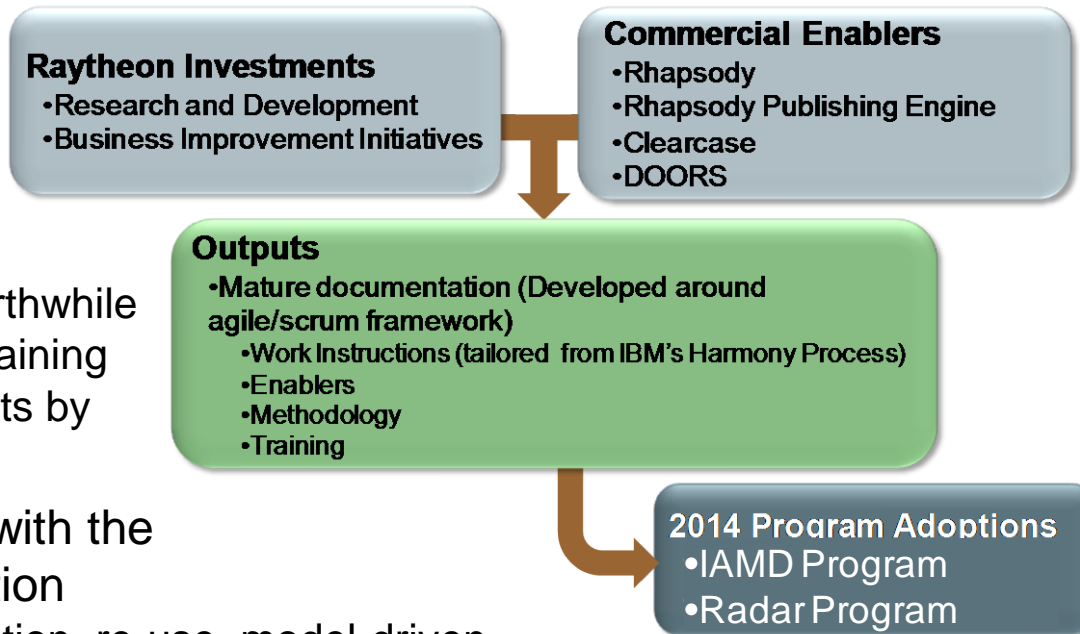


Agenda

- Overview
- Methodology and Process
 - MBSE Context
 - Details
- Program Implementation
 - IAMD Program
 - Radar Program
- Lessons Learned
- MBSE Supplier Innovation

Implementing MBSE Overview

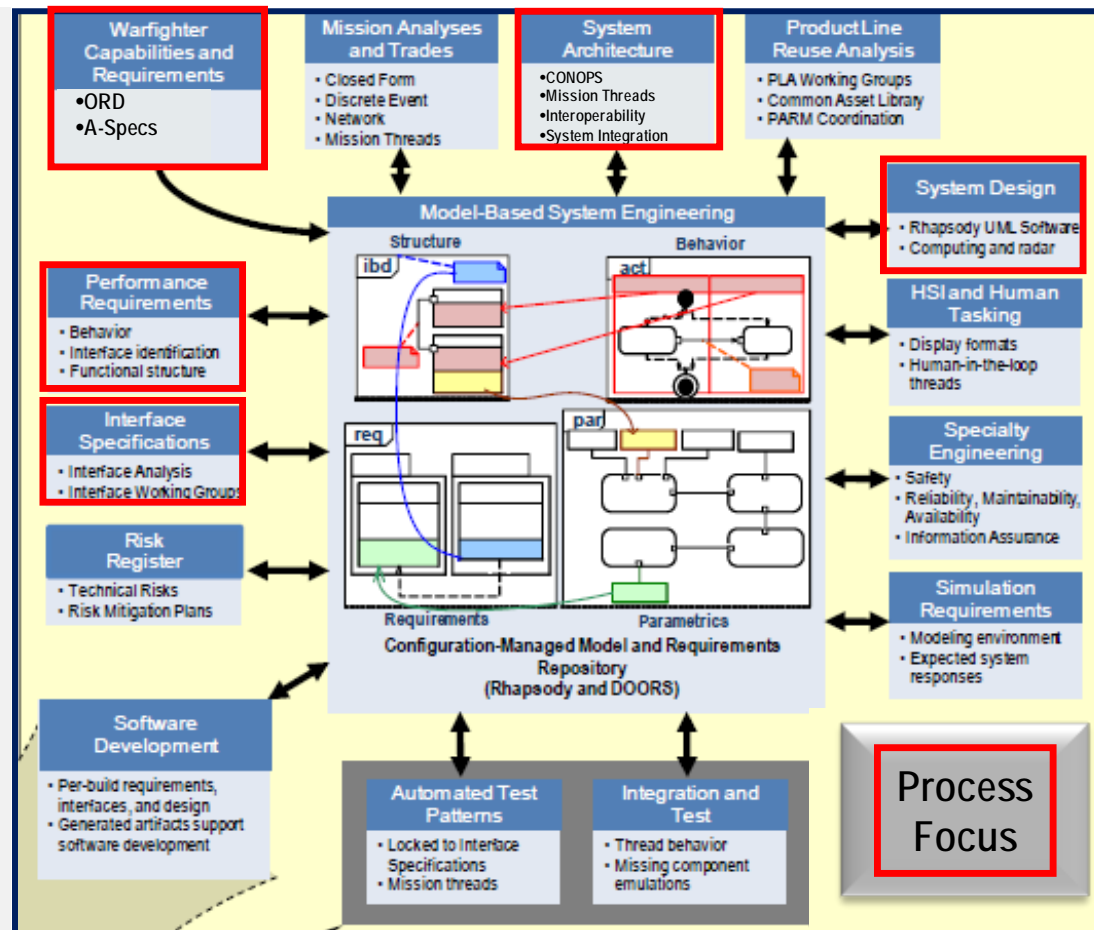
- Programs looking to realize MBSE benefits often don't know how to start
 - Hesitant to adopt MBSE
 - Unsure how to proceed
 - Fearful of whether investment is worthwhile
 - Paralysis prevents programs from gaining productivity and quality improvements by MBSE leverage
- Our approach provides programs with the benefits promised by MBSE adoption
 - Achieved through increased automation, re-use, model-driven documentation and upfront rigor inherent in the process
 - Reduced model variability, makes managing MBSE easier while flattening the learning curve for system engineers as they transition from program-to-program
- At the core is an integrated SysML model that serves as a foundation for the broader Model-Based Engineering environment



Programs thinking of adopting MBSE need a proven, mature, comprehensive process to hit the ground running

Methodology Overview

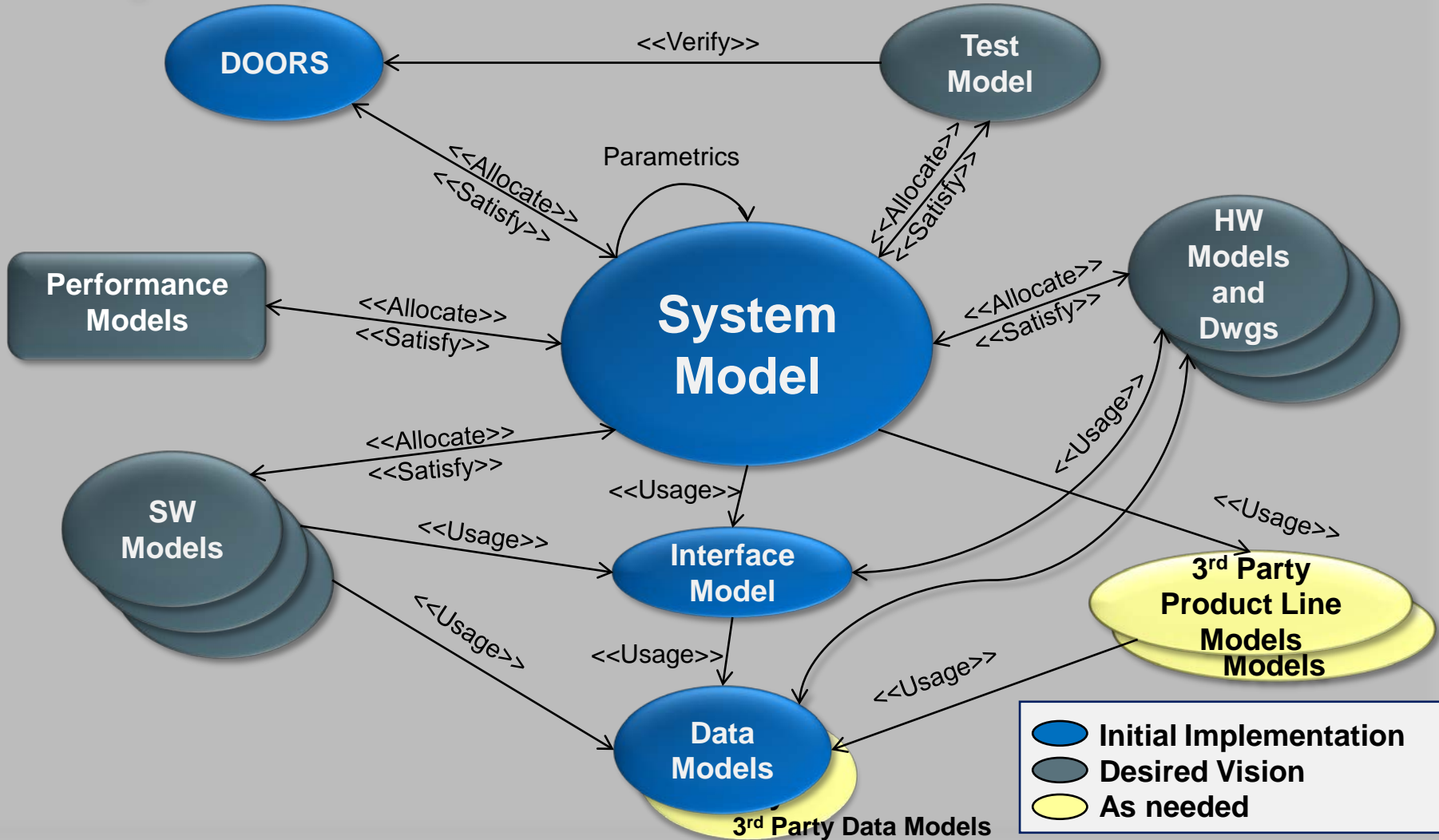
- Analyze system in the context of warfighter mission/capabilities
- Capture architectural decomposition and interaction among elements and components with SysML artifacts
- Use MBSE tools, including Rhapsody[®], ClearCase[®], and DOORS[®]
- Transition from existing mix of document/requirements-based engineering to a self-consistent set of models



A common system model ensures consistency from requirements through detailed design, integration, test and lifecycle support

MBSE Methodology Context

The Integrated System Model – A Model of Models

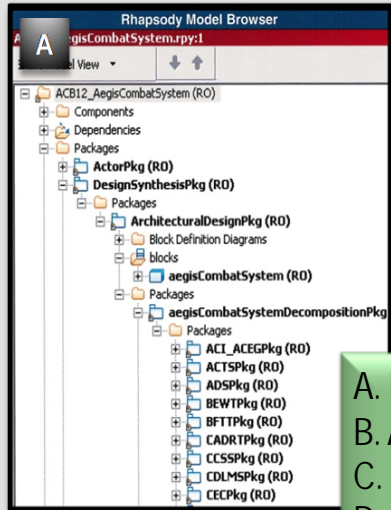


MBSE Process Details

- A tailored version of the IBM® Harmony Process™
 1. Create Use Case Survey (Based on Warfighter Missions and Capabilities)
 2. Define Use Case Model Context (UC Diagrams, Actors, Requirements)
 3. Generate System Activity and Sequence Diagrams and link/generate requirements (end-to-end mission threads)
 4. Decompose and generate Sub-system Activity and Sequence Diagrams
 5. Model Validation Check (Automated, defined set of rules)
 6. Generate Technical Data Package using Rational Publishing Engine® and Final Review
- Requirement linkage to DOORS® via Rhapsody Gateway®
- Reports published using Rhapsody Publishing Engine®
- Automation achieved through Rhapsody® “plug-in” development (leverage Rhapsody® API)

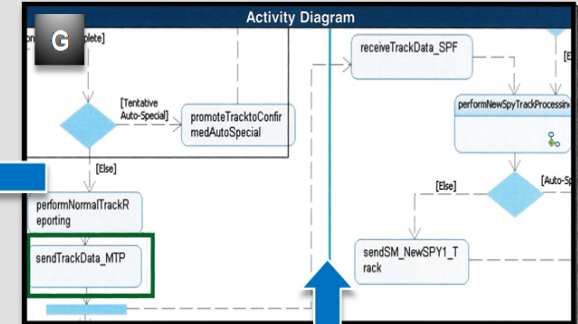
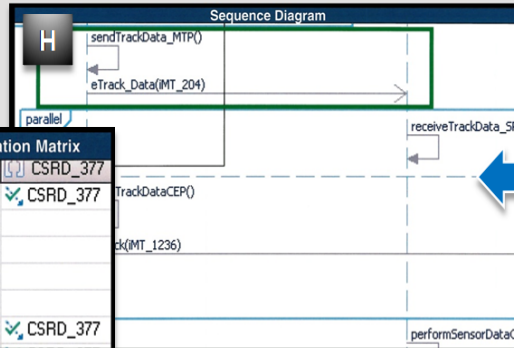
A common process reduces process variability within a program and enables efficient transitions from program-to-program

Workflow: Full Process



I. Allocation Matrix

	CSRD_119	CSRD_372	CSRD_377
END			
SMF			
DIF			
LIF			
EMF			
CEF			
		CSRD_372	CSRD_377



B. Rhapsody Importer Tool

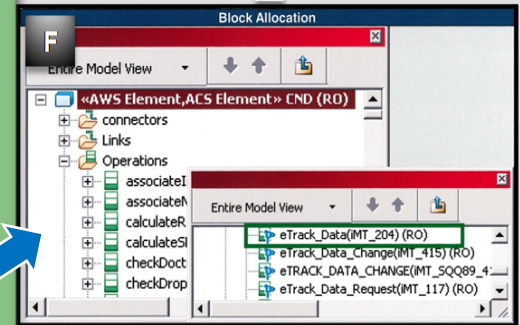
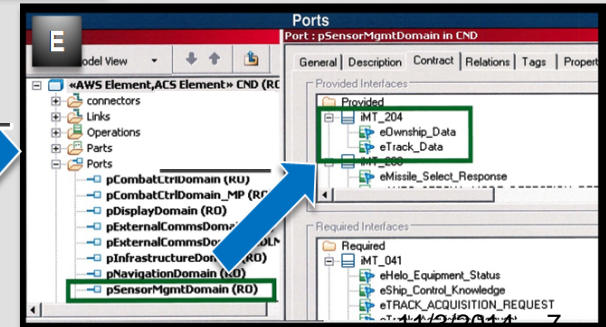
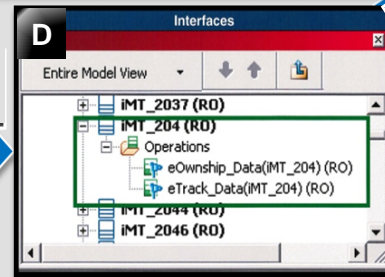
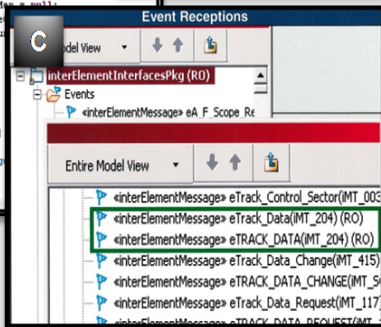
```

import java.io.FileInputStream;

public class InterfaceImporter {

    private static IRPApplication app = null;
    private static IRPProject prj = null;
    private static IRPStereotype interfaceStp = null;
    private static IRPStereotype ieEvStp = null;
    private static IRPStereotype ieFvStp = null;
    private static IRPStereotype connStp = null;
    private static NamingMap namingMap = null;
    private static Integer interfaceId;
    private static Integer recordId;

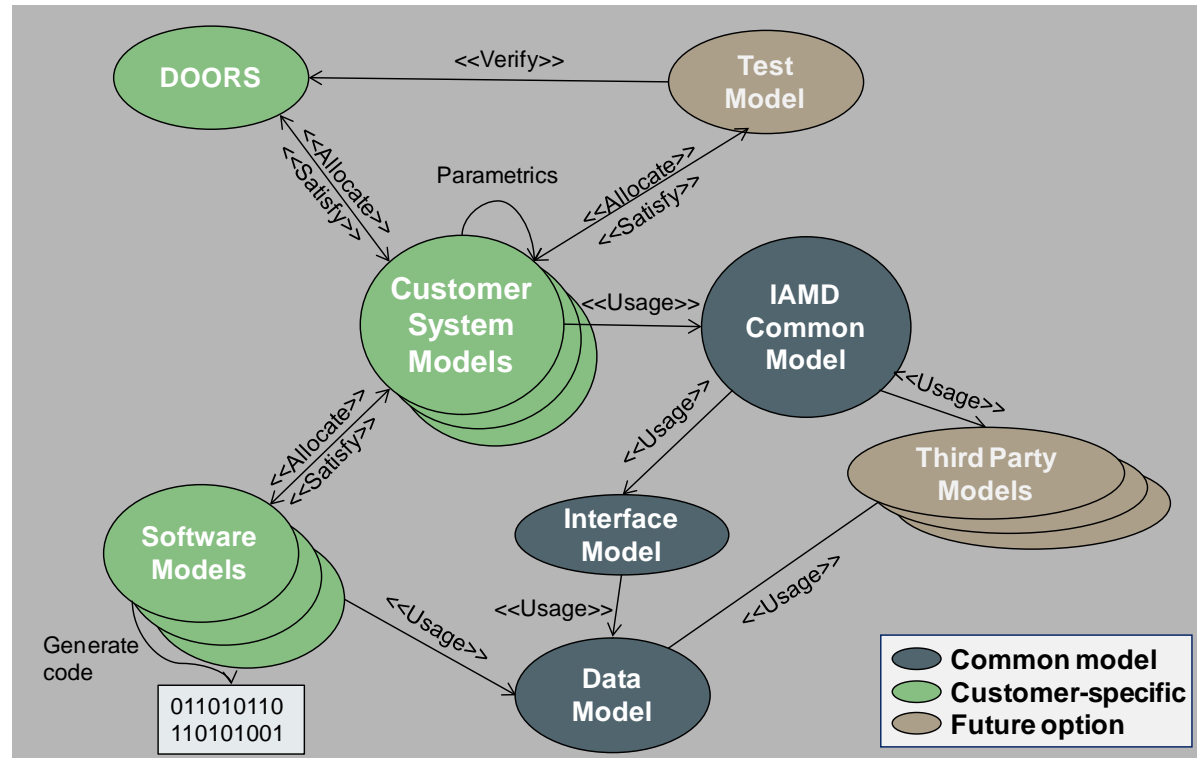
    /**
     * main program
     * @param args
     * @throws IOException
     */
    public static void main(String[] args) {
        if (args.length < 2) {
            System.out.println("usage");
            return;
        }
    }
}
    
```



A. Model browser reflects system architecture hierarchy
 B. Automation used to develop models from existing specifications
 C. Events and interfaces created for all internal and external messages
 D. Each event reception (message) is assigned to an interface
 E. Each interface is assigned to source and destination ports
 F. Event receptions are allocated to appropriate architecture blocks
 G. Operations become actions in an activity diagram
 H. Sequence diagrams are auto-generated from activity diagram
 I. Allocation matrices trace requirements to model elements

IAMD Program MBSE Implementation

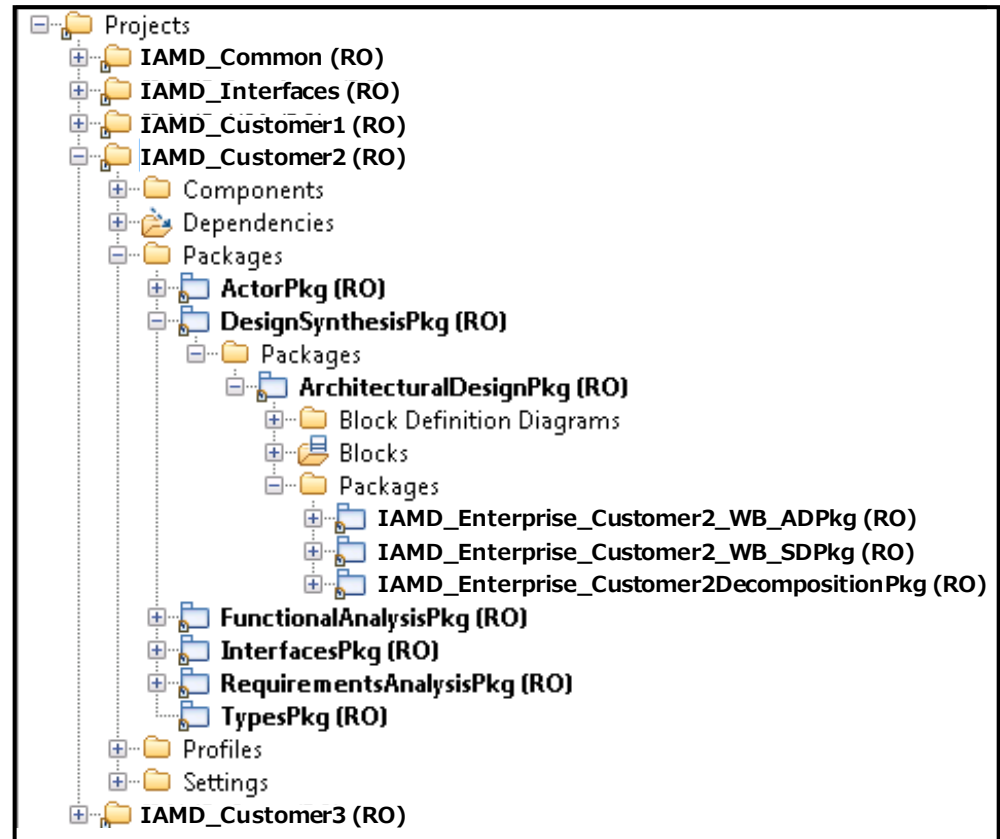
- IAMD Program uses a product line-based model to manage multiple customers
- A common model feeds customer-specific models
 - Reusing common components in individual customer models achieves identity
 - Timeframe to develop new customer specific models decreases due to reuse



Our process is proven for product line modeling and promotes identity

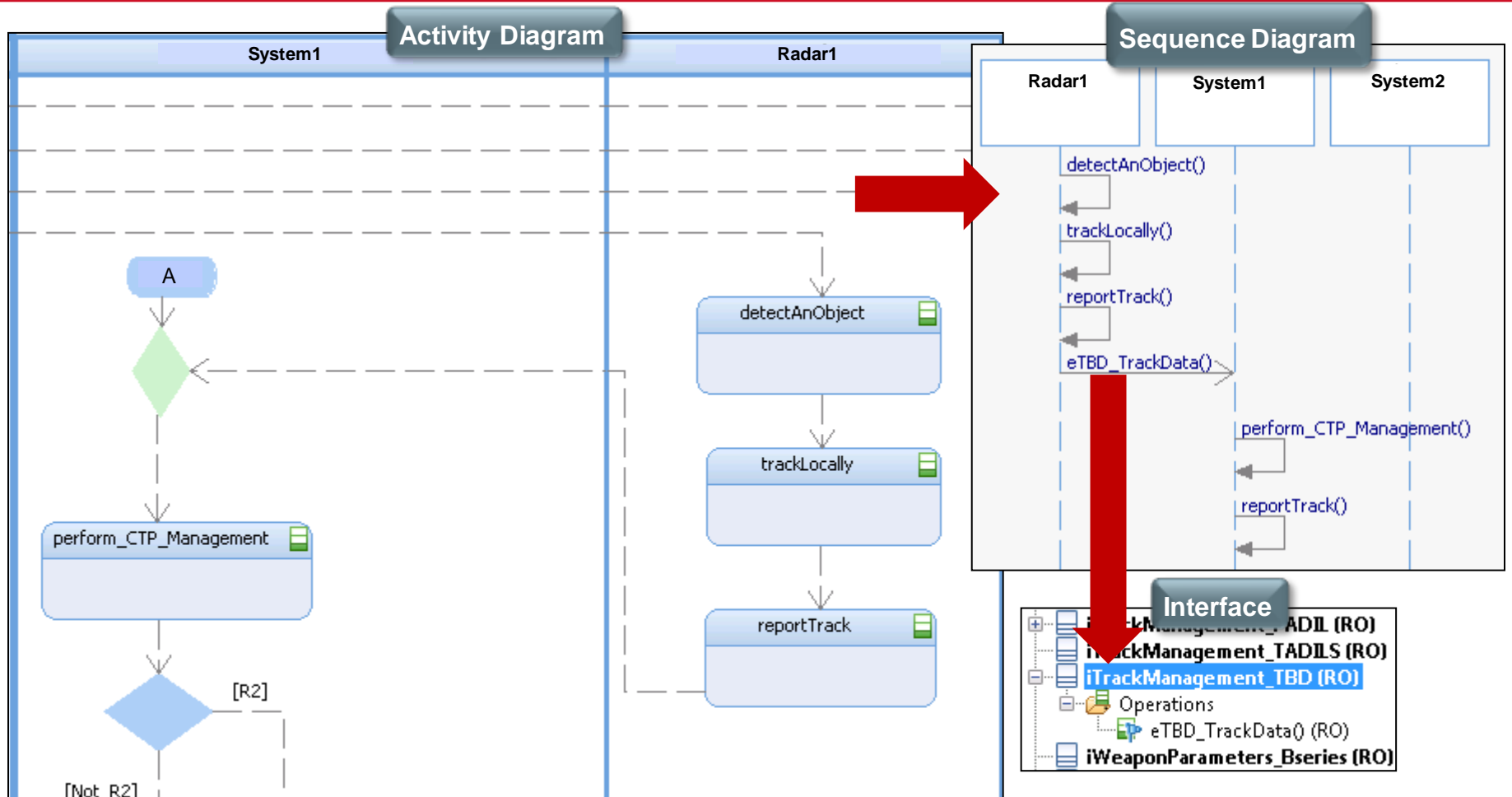
IAMD Program MBSE Implementation cont'd

- Each customer-specific model follows the Harmony™ browser structure
- Each customer model references the Common model and Interface model
 - Identical components from the Common model appear in each Customer model as applicable
 - Components are only modified from within the common model



Consistent structure facilitates navigating between models and re-use of automation

IAMD Program MBSE Implementation cont'd



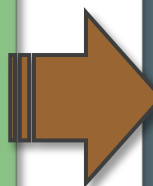
Activity diagram defines the functional flow; sequence diagram follows a path through the activity diagram to define the messages.

Radar Program MBSE Implementation

- Radar Program Need:
 - Ways to improve system engineering and Systems-to-SW transition performance
 - Risk averse: ways to reduce program cost and schedule risks
- MBSE enabled through Mature work instructions and enablers:
 - Successfully implemented, evolved and matured on Naval Combat System Program
 - Currently in use on IAMD Program

Initial Radar Mission Thread Criteria

- **Value added** – fills Radar program requirement gap
- **Limited scope** – balance challenge of learning process, tools and subject matter
- **External interfaces** - Customer looking for leadership in this area
- **Stable Requirements** – enables rapid infrastructure development to facilitate future capability modeling



Initial Mission Threads Modeled

- Perform Calibration
- Perform Sensor Registration
- Perform Health and Status Monitoring

Start small and simple to learn the process, then move on to more complex capabilities

Radar Program MBSE Implementation

- MBSE approach compared to Document based approach
 - Draft Requirements Documents manually created (>100)
 - Included Use cases, activity and sequence diagrams and interfaces
 - Recreated document using MBSE and Rhapsody Publishing Engine
 - Process uncovered holes and ambiguities in manually generated document
 - 24 of 41 system functions were missing requirements from power point
 - Several instances of an unclear order of events (12)
 - Several undefined and/or ambiguous interfaces (4)
 - Several missing functions/operations (8)
 - Identified 44 missing functionality and interface discrepancies that would otherwise have slipped through to SW preliminary design
 - Created comprehensive draft requirement documents
 - Forced key decisions to be made early while impact was minimal
 - Created model framework for future Radar Program modeling

Our MBSE approach enabled Radar Program to identify and resolve issues earlier in the design process

Lessons Learned

Positives

- Easily aligns with agile development environment
- Improved requirements and design quality
 - Our MBSE process yields model artifacts similar to outputs required by manually generated documentation
 - A use case/thread based approach in conjunction with RPE, enabled auto-document generation
 - RPE allowed engineers to focus all their time on engineering rather than document preparation
- Increased consistency and reduced process variability across work products from mature work instructions, automation and doc generation tools
 - Makes management of large MBSE teams easier
 - Enterprise leverage of solutions, automation techniques and document generation templates across programs
- SysML diagrams facilitate reviews while reducing prep time
 - Reviewers immediately see engineer's intent

Observations

- Model diagrams may not be conducive to review on an 8 ½ x11 paper
 - Provide HTML version of model along with documentation
- During initial start-up, where modeling proficiency is low, it is vital to have an MBSE champion co-located with the modelers
 - MBSE champion should operate outside the sprint backlog, as he/she spends the majority of time answering questions and solving problems that are unknown at sprint planning time
- Customer ultimately drives MBSE.
 - Some customers strongly in favor, some strongly against
 - Requirement documents still expected on most DoD contracts
 - Engaging with the customer early on MBSE initiatives can help align expectations

Achieves major MBSE goal where the model drives the documentation because the model is the design

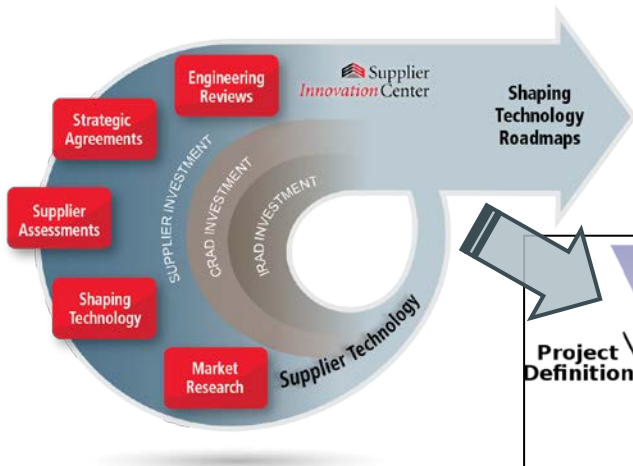
Lessons Learned, cont'd.

Improvements

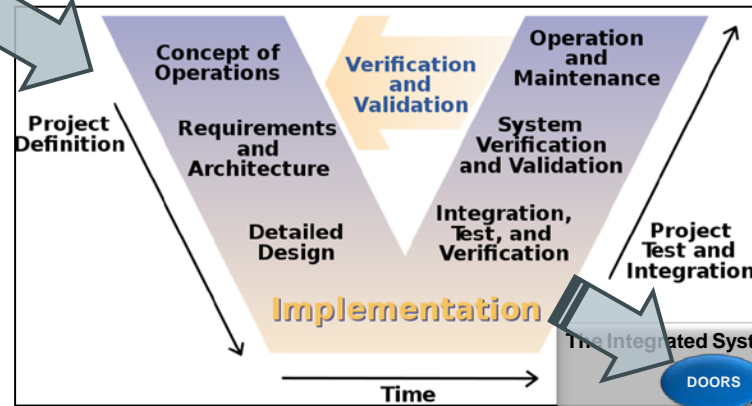
- *“The new venture is not flooded with adequate resources to do the job” **
 - Train the right people
 - Substantially faster and easier to teach Program and Technical SMEs how to model than vice versa
 - Willing MBSE adopters or those with little-to-no experience doing things the “traditional way”
 - Appoint dedicated MBSE product owners (Agile/Scrum environments) who solely focuses on developing story backlogs that foster MBSE and takes into consideration process nuances
- *“Too little fanfare is made about the promise of the new venture. Worst yet, sometimes, management tries to hide it”**
 - Program Leadership Team attitude is pivotal to MBSE culture change and implementation
 - Enthusiastic leadership who embrace MBSE help set a positive and motivating tone beyond the modelers
 - Leadership that ignores or unfavorably reacts to MBSE efforts de-motivate the team, prevent further adoption from the non-modelers and discourages collaboration with the SMEs
- *“They limit the new venture’s autonomy.” **
 - Implementing a new process with a new team unfamiliar with the tools and technology is a challenge
 - Keeping a focused team unencumbered by regular “fire drills” or other tasks during the steep ramp-up phase is critical to any new venture success
 - Allowing MBSE experts to prescribe initial tasking enables a clearer path to success
 - When leaders unfamiliar with the new venture have too much control over its initial direction, it is more likely to end up in a less desirable situation

* Jack Welch on why new ventures fail from “Winning” © 2005

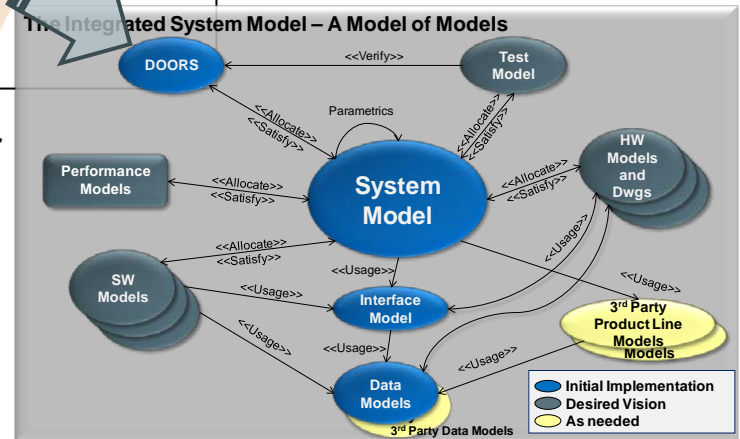
Raytheon MBSE Supplier Innovation



•Leveraging Supplier Innovations with MBSE can improve how we do business on both sides of the Engineering “V”



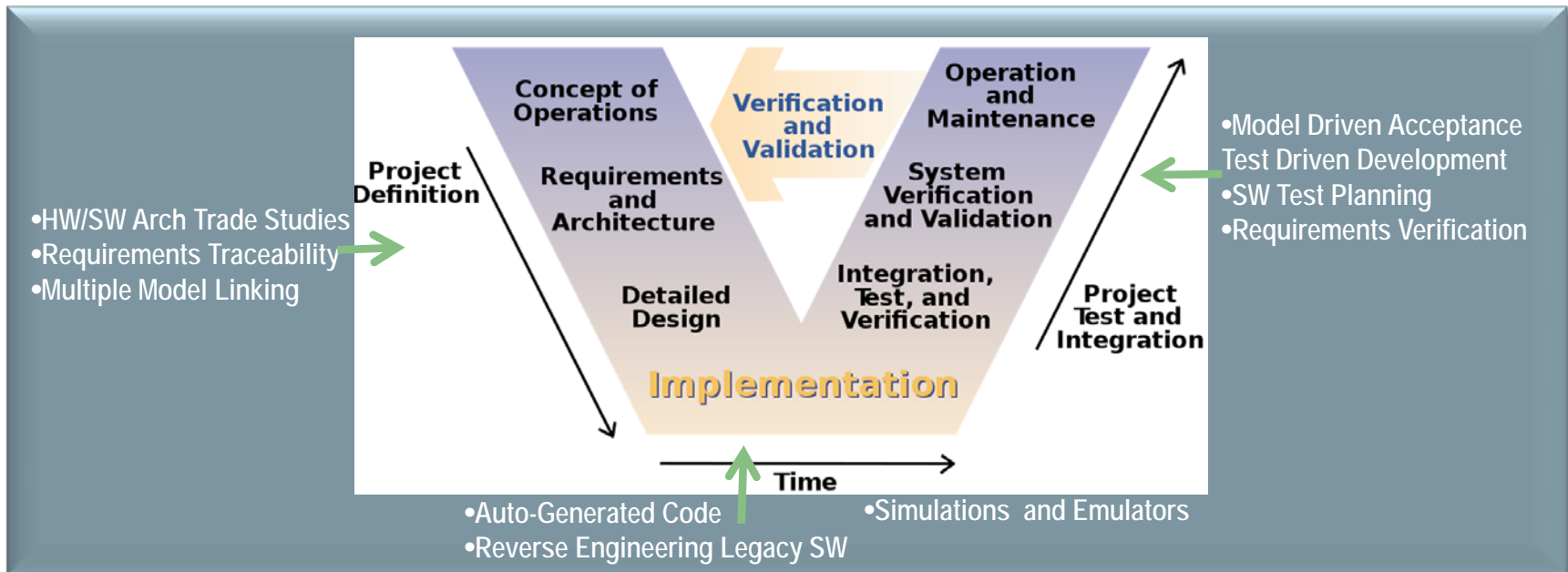
- Raytheon’s Supplier Innovation team aligns supplier technology innovations to achieve **Increased Affordability and Reduced Time to Market**
 - Provide technology to our programs early in research and design phases by working closely with suppliers
 - Connect internal programs to supplier technology



Interdisciplinary team leveraging suppliers’ innovative solutions to provide effective and affordable solutions

Raytheon MBSE Supplier Innovation cont'd

- Improve System engineering performance and System-to-SW transition (SysML-to-UML)
- Develop and mature approaches to link non-SysML models
 - Leverage Open Services Lifecycle Collaboration (OSLC) standard to link data across different engineering modeling environments (e.g., Excel Cost Models, ME/EE models, Performance Sims)
- Create MBE pull through IV&V
 - Use SysML activity and sequence diagrams as baseline for model driven integration and test
 - Regression Test Coverage impact and analysis



MBSE can improve how we do things on both sides of the “V”