

Object-Oriented Systems Engineering Method (OOSEM)

Overview

Instructor Introduction



- Sanford Friedenthal (safriedenthal@gmail.com)
 - Career in Systems Engineering
 - Phases: proposals, conceptual design, development and production
 - Applications: missile systems, avionic systems, and information systems
 - Roles: project systems engineer, department manager, lead of organizational initiatives (e.g., model-based systems development)
 - Retired from Lockheed Martin at end of 2010
 - Independent Consultant in MBSE
 - INCOSE Fellow and Past Chair of INCOSE MBSE Initiative
 - A Leader of the OMG Systems Modeling Language (SysML)
 - Co-led the effort to develop SysML v1 and SysML v2
 - Co-author: 'A Practical Guide to SysML', and 'Architecting Spacecraft with SysML'

Acknowledgements

- The material from this course acknowledges the following contributions:
 - Content from “A Practical Guide to SysML” by Friedenthal, Moore, and Steiner (3rd Edition - Chapter 17)
 - Tutorial material developed by Joe Wolfrom and reviewed by the INCOSE OOSEM Working Group

OOSEM Topics

- OOSEM Overview
- Method
 - Setup Model
 - Analyze Stakeholder Needs
 - Analyze System Requirements
 - Define Logical Architecture
 - Synthesize Candidate Physical Architectures
 - Optimize and Evaluate Alternatives
 - Manage Requirements Traceability
 - Integrate and Verify System
- Summary

OOSEM Overview

Module Objectives

- After completion of this module, student should understand
 - Key elements of the OOSEM approach
 - Identify the OOSEM activities
 - Considerations for tailoring OOSEM

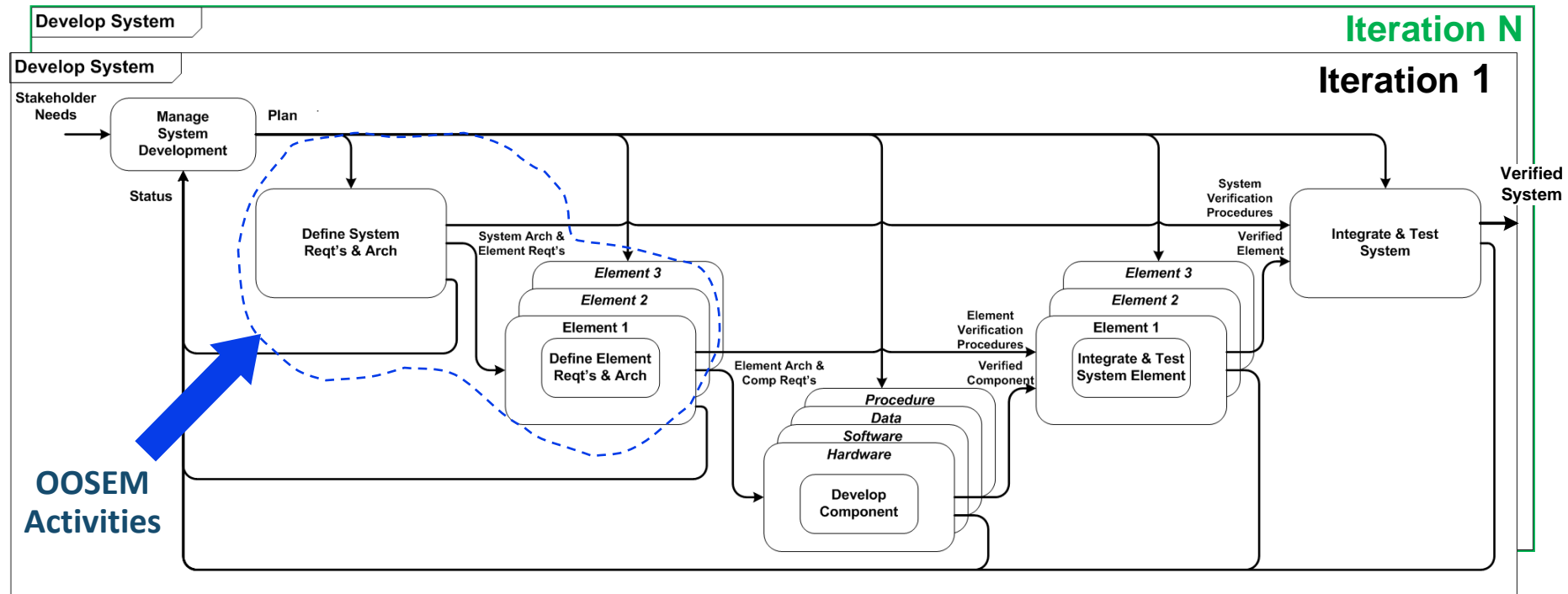
OOSEM Introduction

- Motivation
 - Provide a SE method that leverages object-oriented concepts, traditional top-down SE approach, and SysML to facilitate:
 - Capture and analysis of requirements and design information
 - Integration with software, hardware and test processes
 - Flexibility to accommodate changing requirements and design evolution
- Approach
 - Top-down decomposition coupled with bottoms-up design
 - Scenario-driven process
 - Separation of concerns to deal with complexity
 - Build system model to analyze, specify, design, and verify system

What is object-oriented about “OO”SEM?

- Leverages OO concepts that are supported by SysML
 - Blocks based on classes with attributes and operations
 - Generalization/Specialization
 - Encapsulation using ports
- OO concepts provide useful abstraction techniques for systems engineering
- Some object-oriented concepts used for software may be de-emphasized (e.g., invocation of operations) or reinterpreted for systems engineering (e.g., function vs. operation)

System Development Process



Integrated Product Development (IPD) is essential to improve communications

A recursive Vee process

Applied iteratively throughout the system life cycle

System Development Process Overview

- Includes:
 - Management Process
 - System Specification and Design
 - ‘Next-level’ Development Processes
 - System Integration and Verification
- Can be applied recursively to multiple levels of the systems hierarchy
 - Similar to the SE ‘Vee’
- Course focus is on the ‘Specify and Design System’ activity

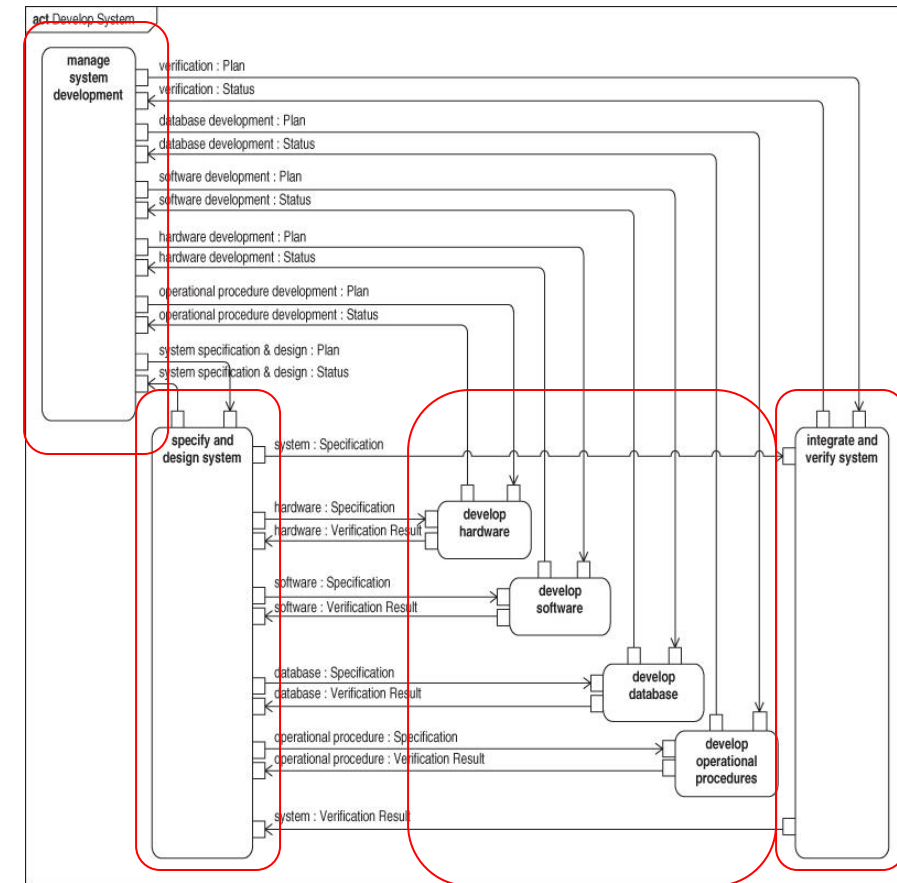
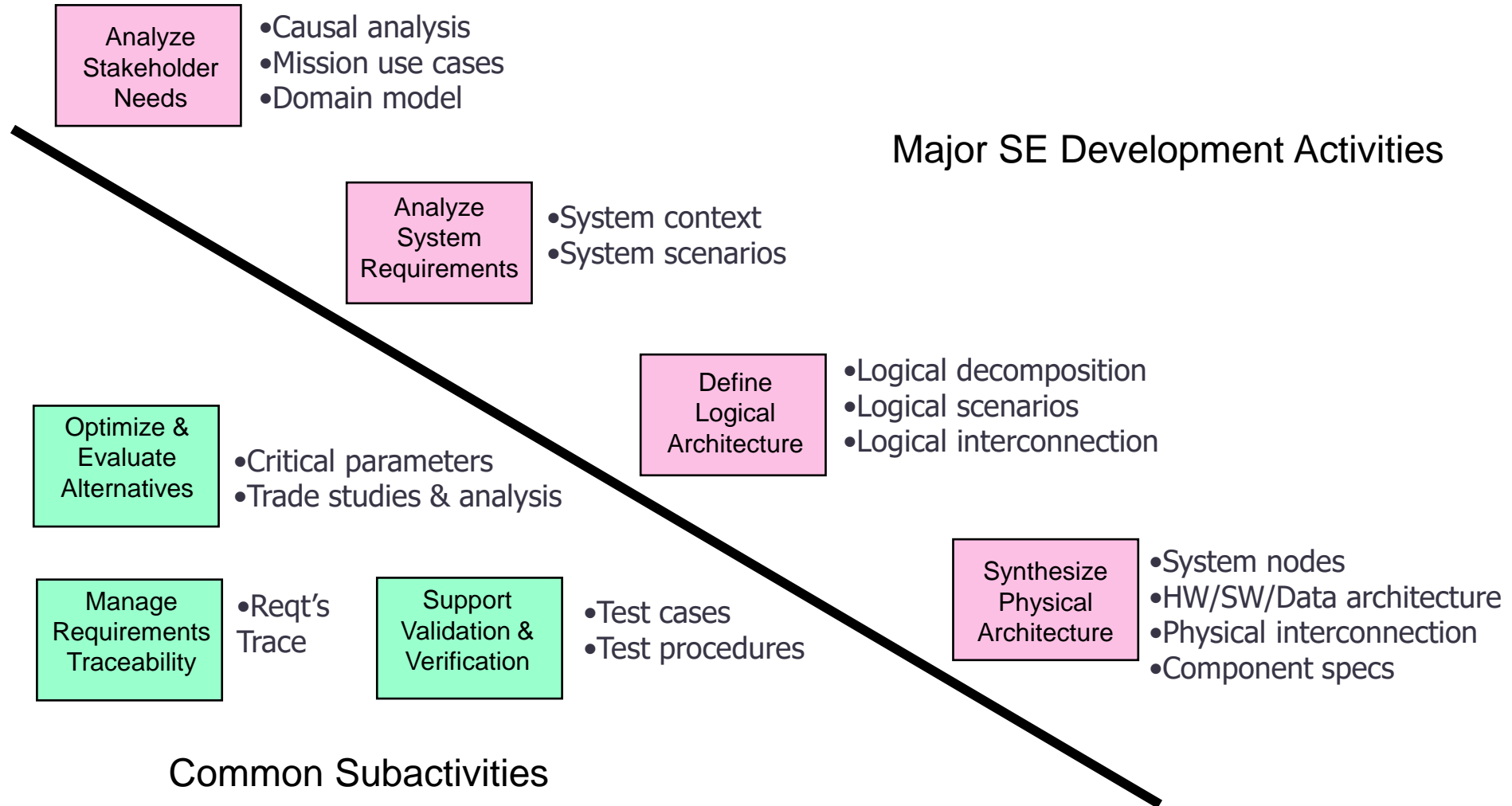
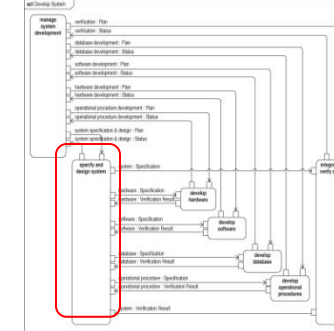


Figure 17.1

OOSEM Specify & Design Activity & Selected Modeling Artifacts





Specify and Design System

- Purpose:
 - Process for analyzing needs and requirements, architecting system, and specifying system elements
- Includes activities to:
 - Analyze requirements
 - Define the architecture
 - Analyze alternative designs
 - Manage traceability
- Can be applied recursively to multiple levels of design

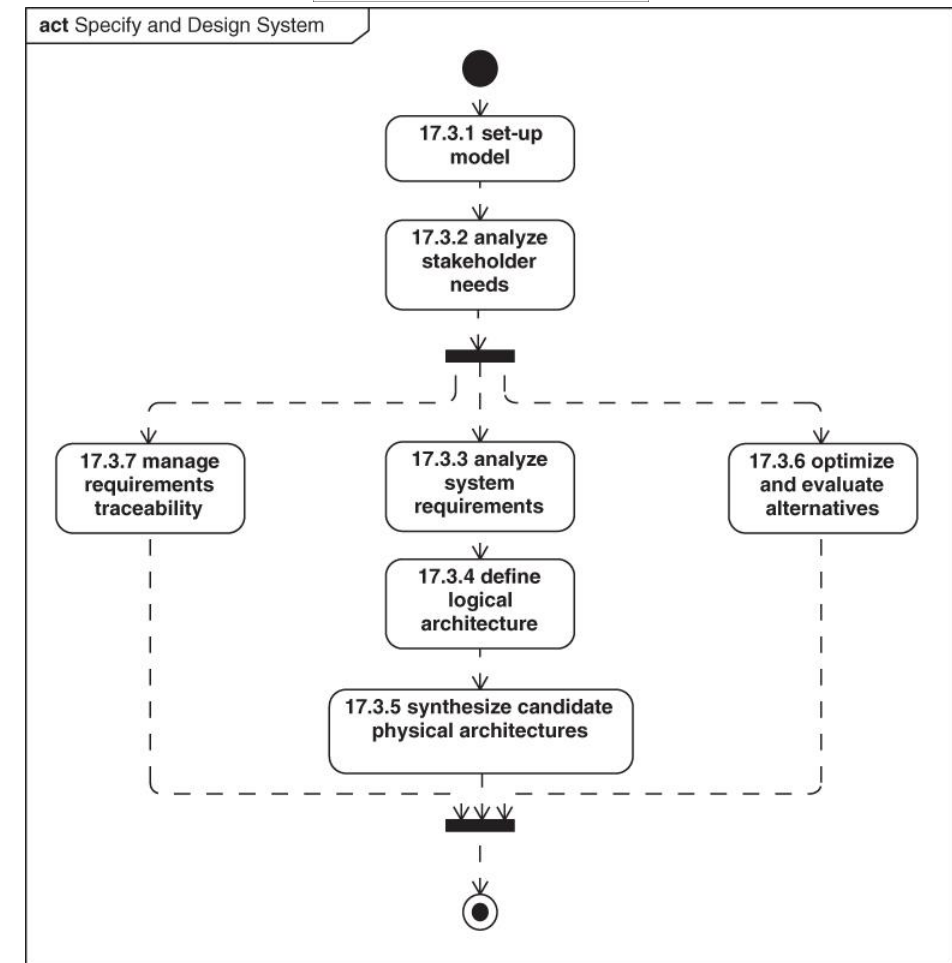
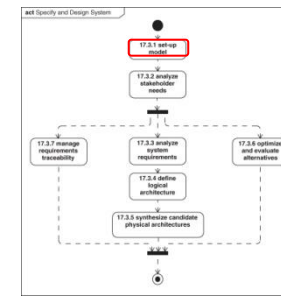


Figure 17.3



Set Up Model

- Modeling conventions and standards facilitate consistent representation and style across the model
- Model organization is essential for model management

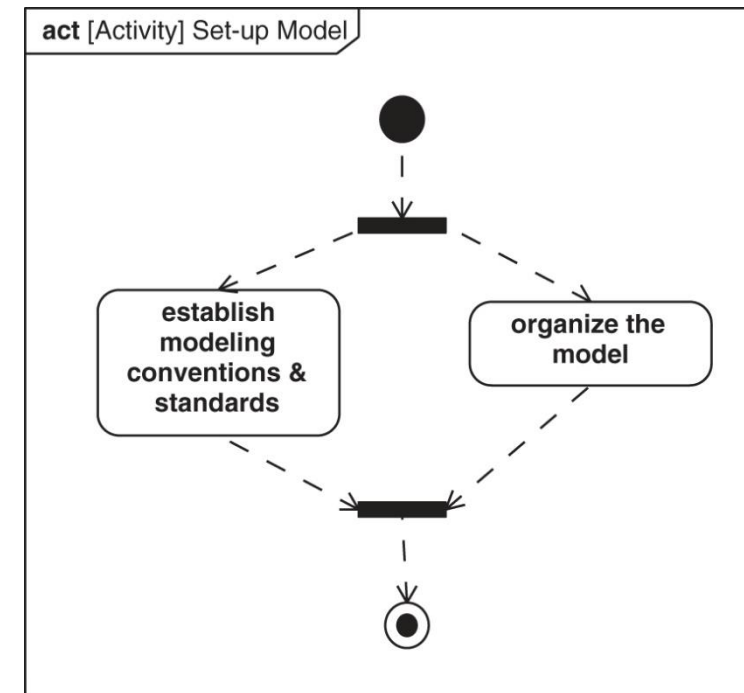
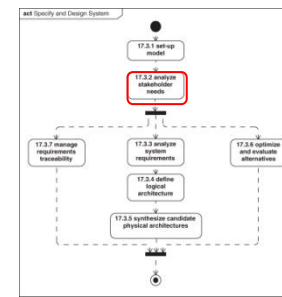


Figure 17.4



Analyze Stakeholder Needs

- Characterize as-is system and enterprise and performing causal analysis
 - Capabilities and limitations
 - Potential for reuse
- Define to-be system and enterprise
 - Mission requirements
 - Top level context
 - Enterprise use cases
 - Measures of effectiveness

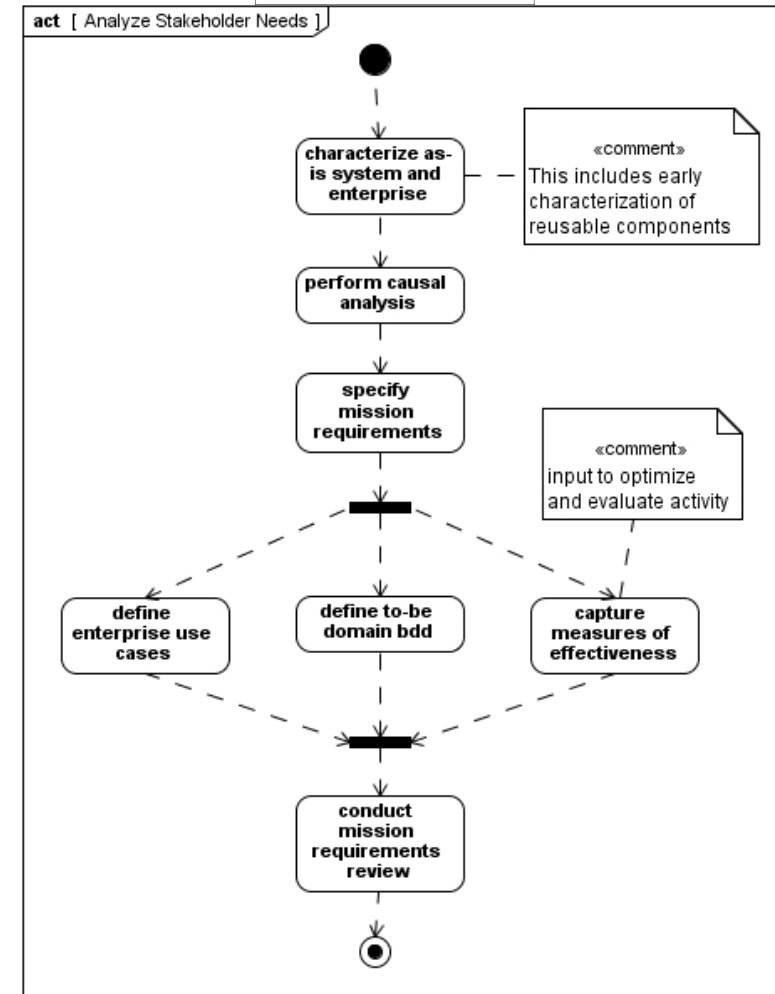
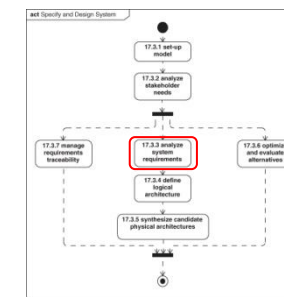


Figure 17.6



Analyze System Requirements

- Perform scenario analysis for each enterprise use case
- Specify the system requirements in terms of its input and output responses and other black box features
 - Interfaces
 - Functions
 - States
 - Performance, physical and quality characteristics
 - Stores

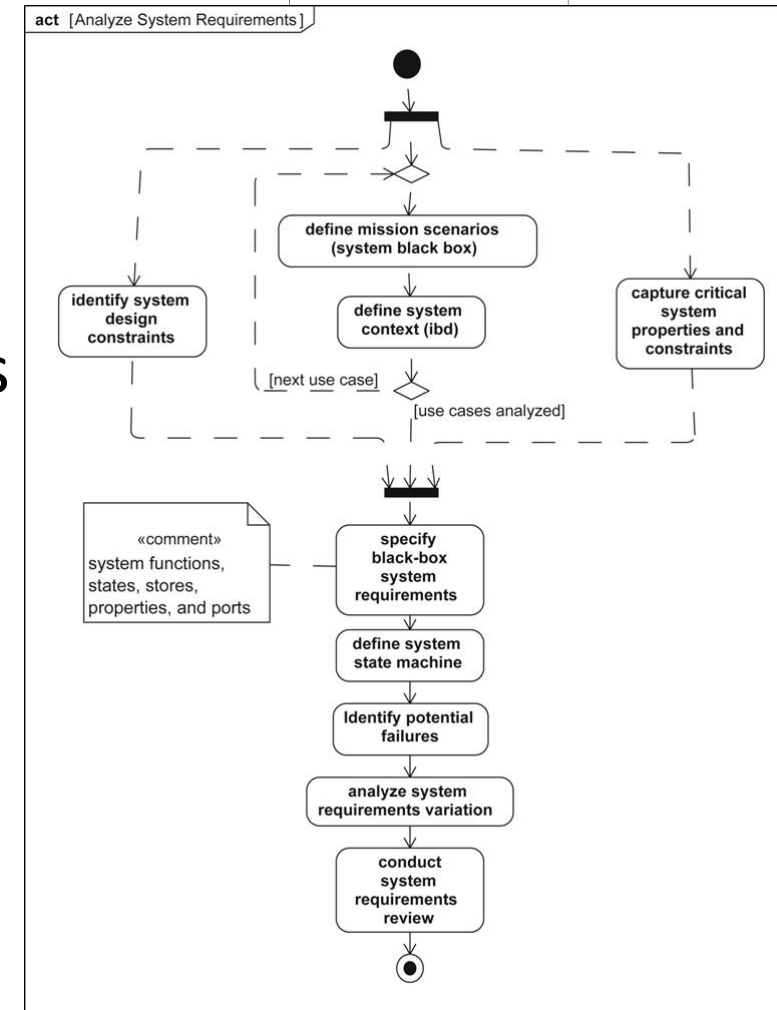


Figure 17.14

Define Logical Architecture

- Decompose the system into logical components
- Define how the logical components interact to realize the system black box behavior
- Define the interconnection between the logical components

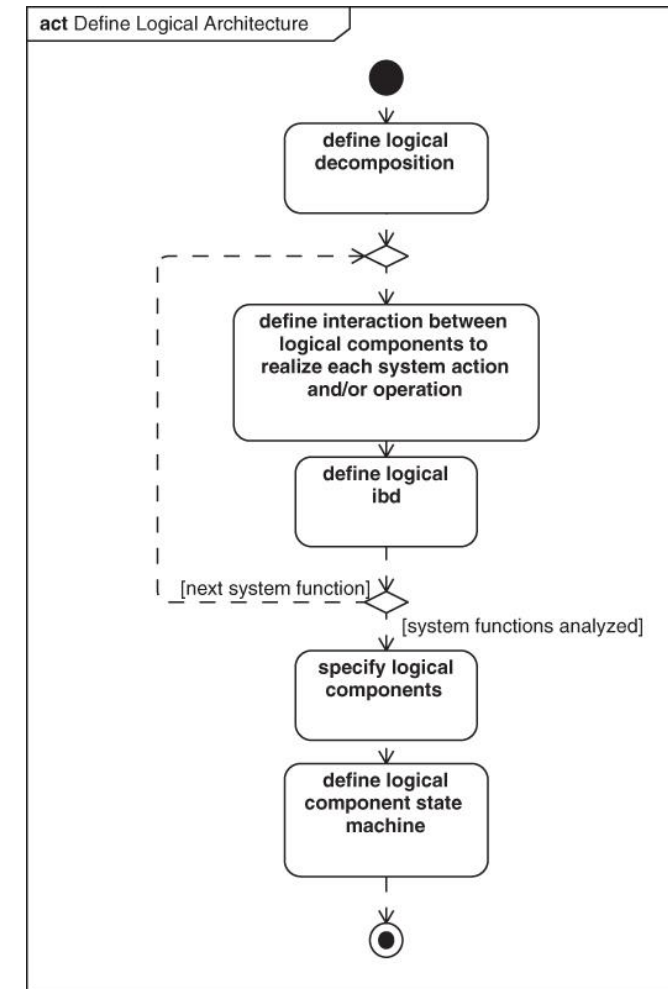


Figure 17.22

Synthesize Candidate Physical Architectures

- Define how logical components are distributed across nodes
- Allocate the logical components to physical components that are implemented in hardware, software, data, and procedures
- Define hardware, software, data, and other cross-cutting views of the system architecture

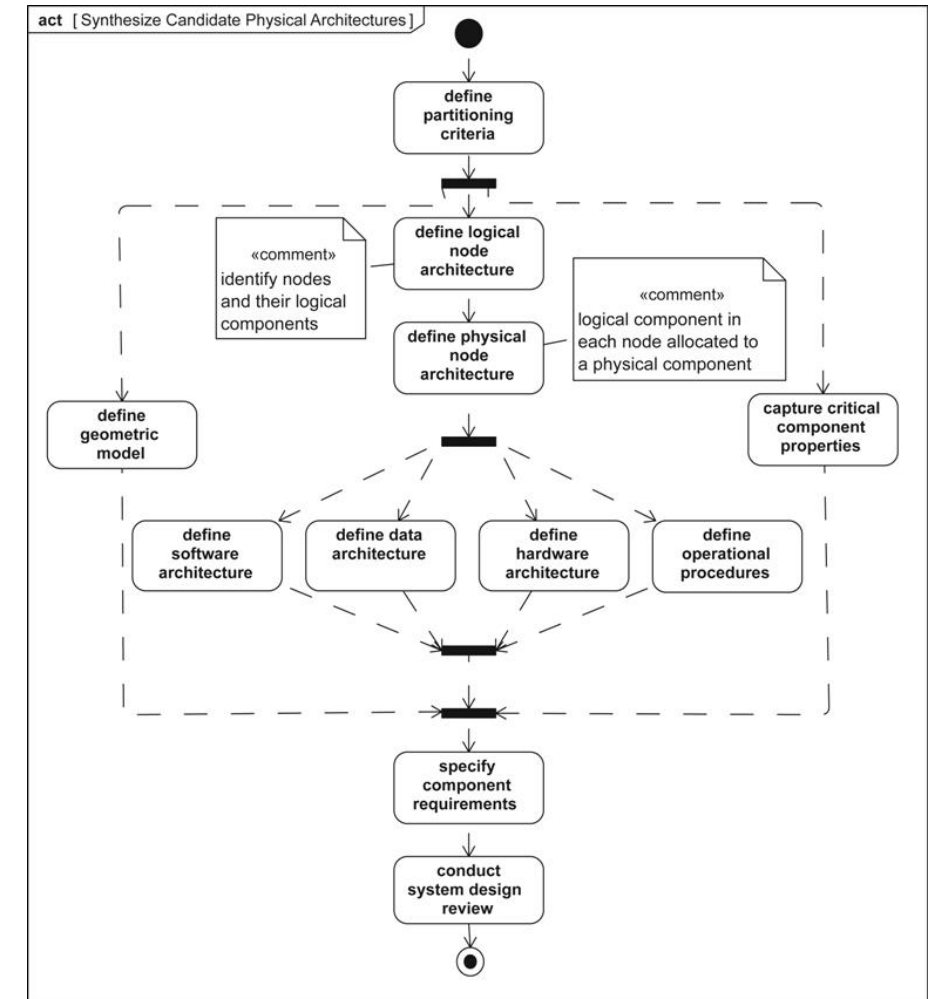


Figure 17.26

Optimize and Evaluate Alternatives

- Invoked throughout the process
 - Perform engineering analysis that supports system design trade studies and design optimization
- Parametric diagrams used to specify each critical analysis of the system

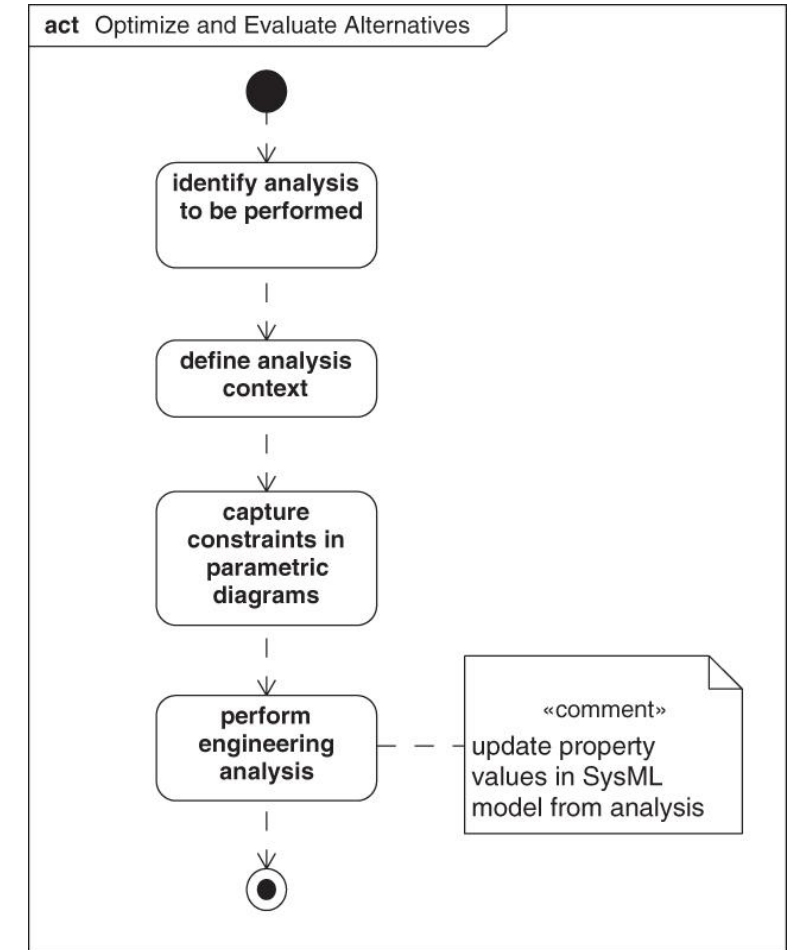
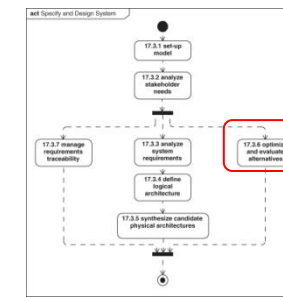


Figure 17.48

Manage Requirements Traceability

- Invoked throughout the process
- Manage traceability from the mission-level requirements to the component requirements
 - Trace requirements to design, analysis, and verification elements

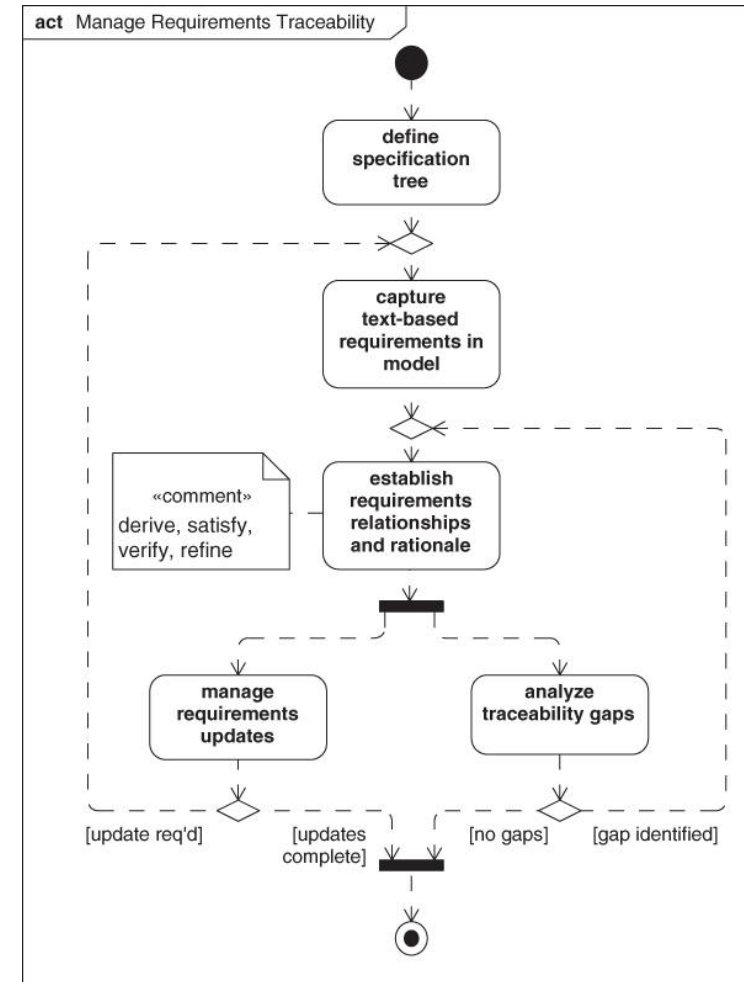
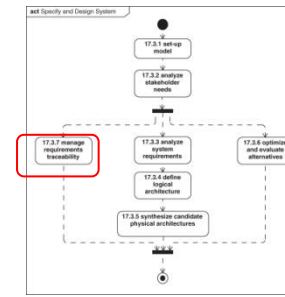
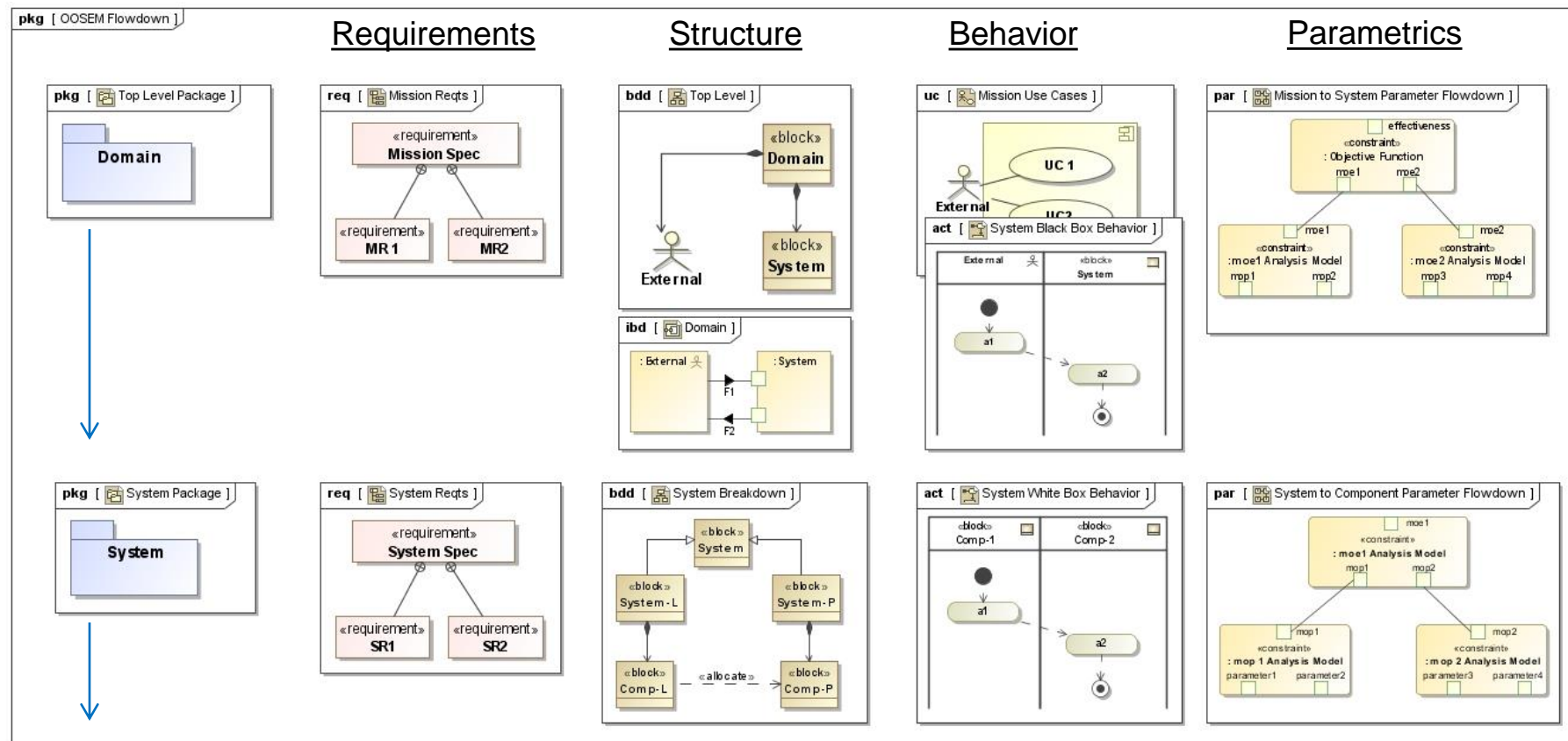


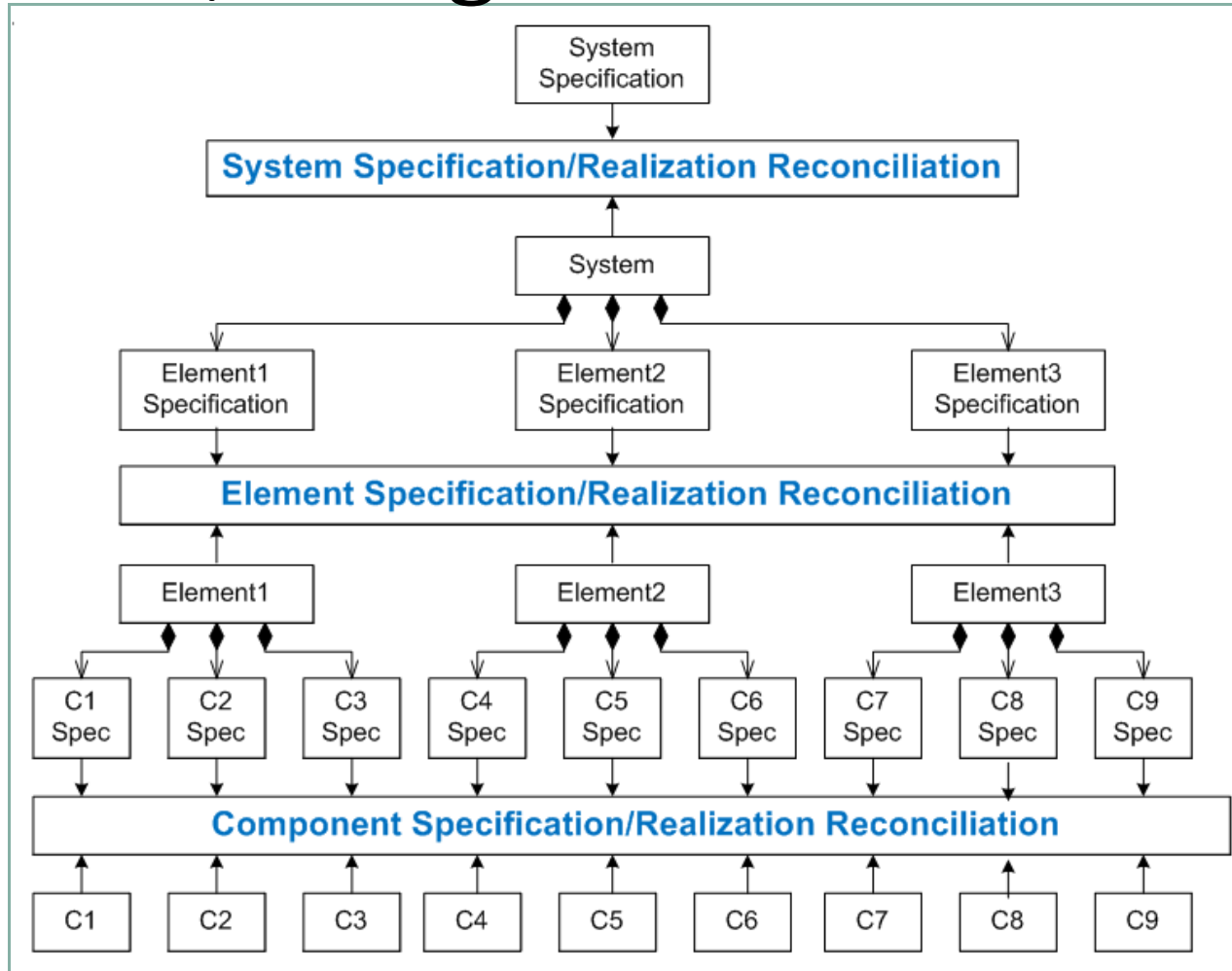
Figure 17.53

OOSEM Flowdown

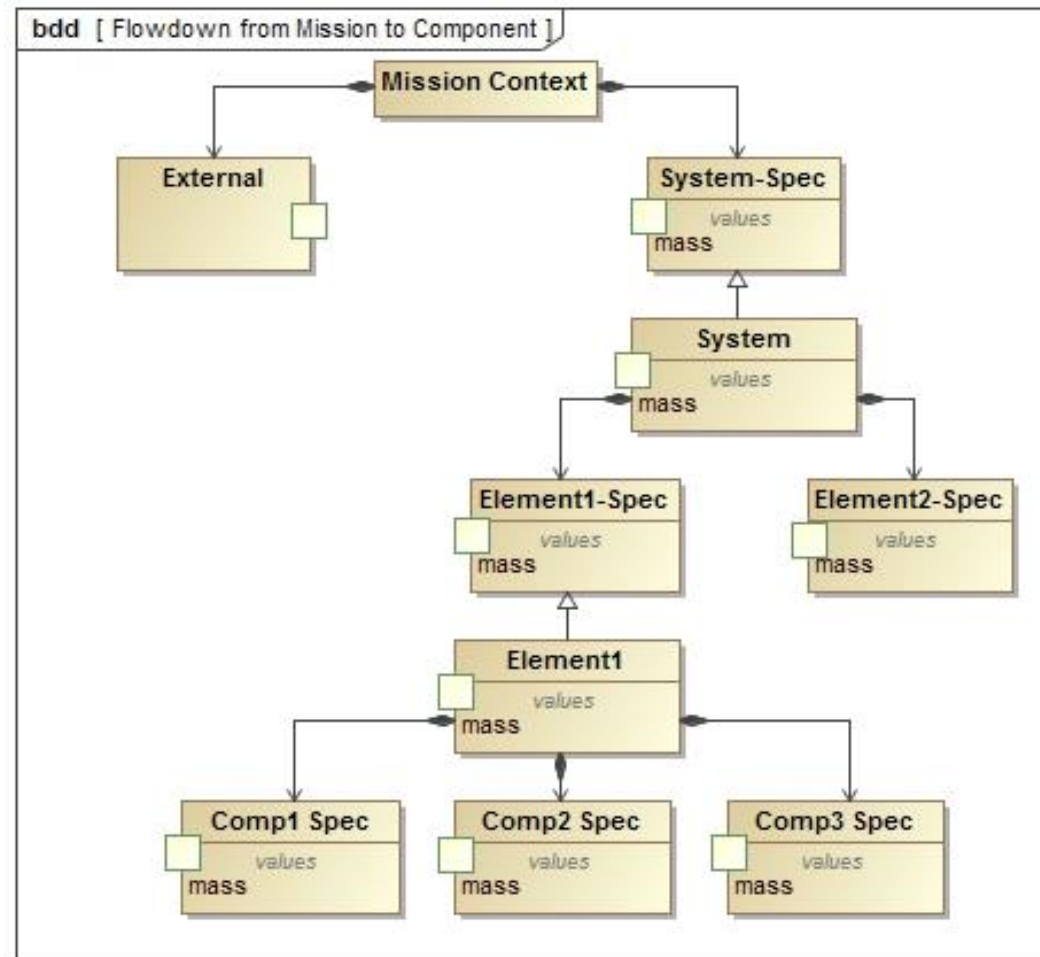
- Flowdown of Requirements, Structure, Behavior, Parametrics
- Ensure model integration horizontally and vertically



Top Down-Bottoms Up Specification/Design Reconciliation



Mission to Component Flowdown



Applying OOSEM to Enabling Systems

- OOSEM can be applied to develop systems that support the operational system throughout its life cycle
 - Verification system
 - Manufacturing system
 - Installation system
 - Support system
 - ...

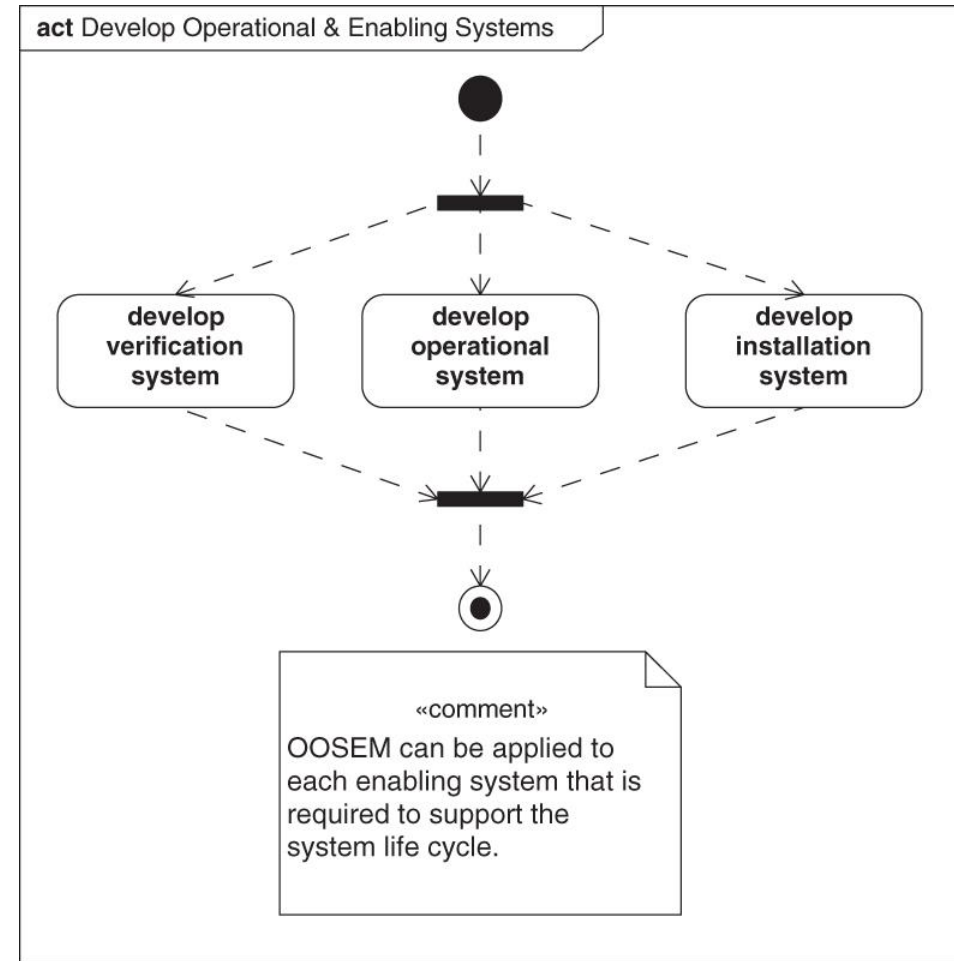


Figure 17.58

Tailoring OOSEM

- OOSEM should be tailored to meet project objectives within schedule and resource constraints
- Tailoring involves selecting the appropriate level of rigor to apply to each of the OOSEM activities
 - Selecting the activities and their ordering
 - Selecting the artifacts to generate
 - Selecting the level of detail
- Emphasis shifts with the phase of development
 - Top level mission requirements, architecture design and physical sizing trade-offs in conceptual design phase
 - Detailed architecture trade-offs and component requirements in later design phases

Security System Example Background

- Security Inc. has been in business for several years providing residential security systems
- Current system provides break-in detection, alarms, and reporting to a central system
- Company has been losing market share to competitors
- Security Inc. is initiating development of an Enhanced Security System (ESS) to increase its business

Systems Engineering Team Technical Management Role



- Technical Planning
- Risk Management
- Managing the Technical Baseline
- Conducting Technical Reviews

Systems Engineering Team Early Planning Activities

- Established the Project Plan and Infrastructure
 - Define model objectives
 - Scope the model to meet objectives
 - Select & tailor the method and modeling guidelines
 - Select the tools and infrastructure
 - Define the schedule of modeling activities and artifacts
 - Staff and organize the modeling effort
 - Provide training in the language, method, and tools

Model Development Milestones

- Modeling artifact maturity defined for each project milestone
- Integrate model artifact delivery into project schedule
- Track against plan like any other project deliverable

		Maturity Level				
Milestone		SRR	SDR	PDR	CDR	TRR
Level	Artifact					
System	Black box specification	H				
	Logical architecture	M	H			
	Physical architecture	L	M	H		
	Analysis					
	Test cases	L	M		H	
	Verification results					L
Element	Black box specification	M	H			
	Logical architecture		M	H		
	Physical architecture		L	M	H	
	Analysis					
	Test cases		L	M	H	
	Verification results					M
Component	Black box specification		M	H		
	Design		L	M	H	
	Analysis					
	Test cases		L	M	H	
	Implementation					H
	Verification results					H

Summary

- OOSEM is an MBSE method that provides a top-down, scenario-driven process to analyze, specify, design, and verify a system
 - System model is a primary artifact
 - Controlled as part of the technical baseline
- Provides a way to deal with system design and development complexity using separation of concerns as a fundamental tenet
 - Black box, white box
 - Logical, physical
 - Distribution
 - Operational system, Enabling systems
- Should be tailored to satisfy project objectives and constraints