"Harmony for Systems Engineering
Best Practices of Model-Based Systems Engineering using Rhapsody"

Workflow-oriented Tool/SysML Training

Course Outline

italic: Hands-on Practice, regular: Lecture

1. **Introduction**
   - Model-based Systems Engineering in the Context of MDD
   - Fundamentals of model-based systems engineering
     - Essential SysML artifacts
     - Service request-driven modeling approach
   - Task flow and work products in Rational *Harmony™ for Systems Engineering*
   - Systems engineering handoff to hardware and software development

2. **Getting started**
   - *Create Harmony Project (SE-Toolkit Feature: #2)*
   - Introduction to Harmony Project Structure

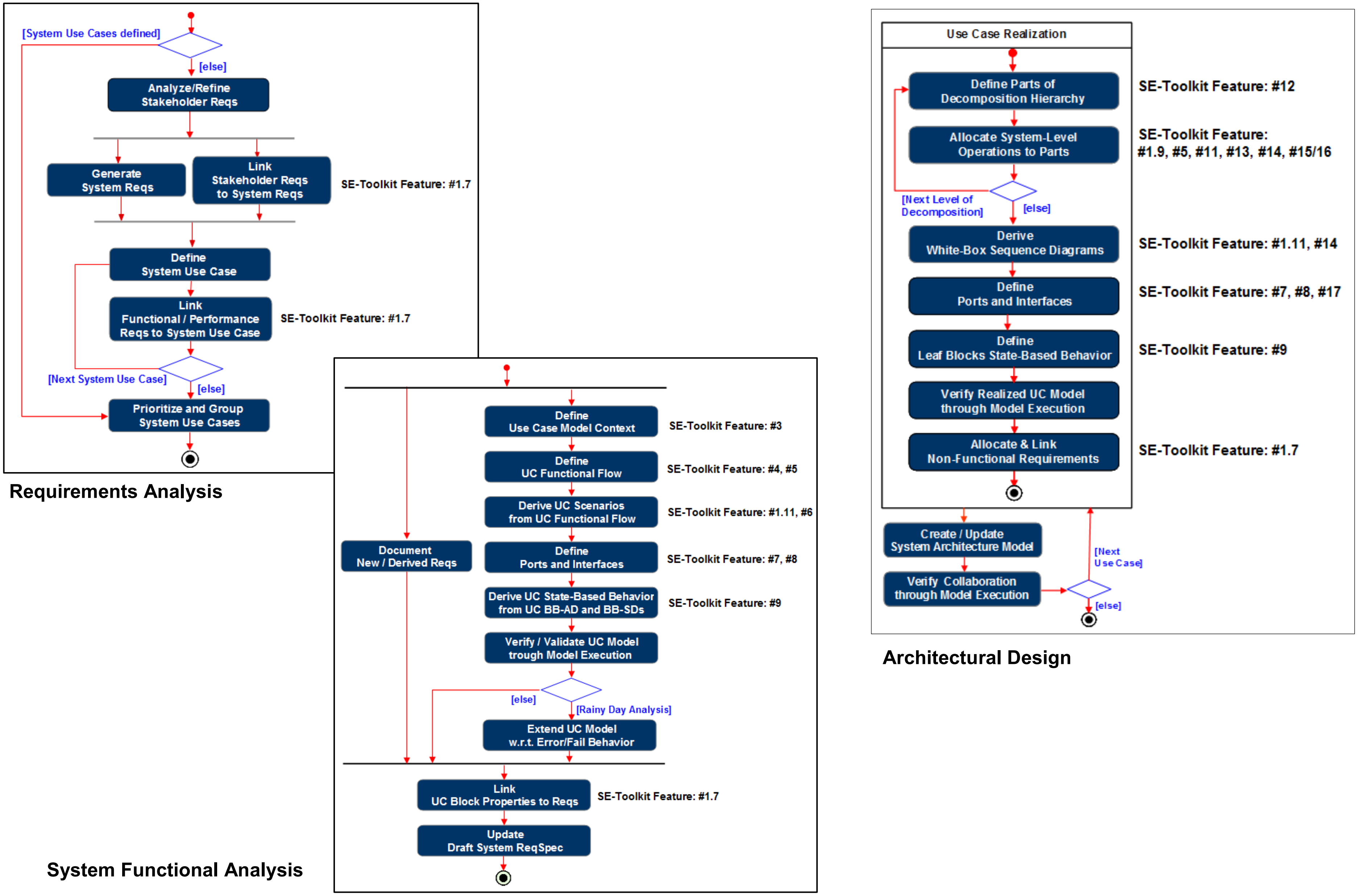
3. **Requirements Analysis**
   - Overview: Requirement Analysis Workflow
   - Role of Requirements Diagrams in Harmony/SE and essential elements
   - Rhapsody Gateway essentials
   - *Import system requirements (training example) into Rhapsody through Gateway*
   - Intro: Role of use cases in Harmony/SE and essential elements of a Use Case Diagram
   - *Define use cases of training example*
   - *Link use cases to system requirements (SE-Toolkit Feature: #1.7)*
   - *Check requirements coverage (Gateway).*

4. **Functional Analysis**
   - Overview: Functional Analysis Workflow (Part 1)
   - *Create a use case model project structure (SE-Toolkit Feature: #3)*
   - Introduction to use case model project structure (Functional Analysis Package)
     - Role of Block Definition Diagram (BDD) and Intern Block Diagram (IBD) in Harmony/SE and essential elements of BDD and IBD – Part 1
     - Role of Activity Diagram (AD) in Harmony/SE and essential elements of an AD.
   - *Capture the use case functional flow in the black-box activity diagram (SE-Toolkit Feature: #4.5).*
   - Role of Sequence Diagrams (SD) in Harmony/SE and essential elements of an SD.
   - *Derive use case scenarios from the black-box activity diagram (SE-Toolkit Feature: #1.10, #6)*
   - Essential elements of BDD and IBD - Part 2: Ports and Interfaces
   - *Create use case model ports and interfaces (SE-Toolkit Feature: #7, #8).*
   - Role of a Statechart Diagram (SC) in Harmony/SE and essential elements of a SC.
     - Guideline: How to derive a Statechart from the information captured in an AD and SDs.
   - *Describe the state-based behavior of the use case block and the actors (SE-Toolkit Feature: #9).*
   - Role of model execution in Harmony/SE and introduction to the different ways of running model execution in Rhapsody.
   - *Verify/validate the use case model through model execution (use SD Compare feature).*
   - Functional Analysis Workflow Part 2: Use case "Rainy Day Analysis"
   - *Perform use case "rainy day" analysis*

- *Link use case block properties to system requirements (SE-Toolkit Feature: #1.7).*

5. **Design Synthesis - Architectural Design**
   - Overview: Architectural Design Workflow (*Use Case Realization*)
   - *Create an Architectural Design package in Rhapsody and merge properties of use case block in SuD block (SE-Toolkit Feature: #10).*
   - *Create system architecture BDD and IBD (SE-Toolkit Feature: #12).*
   - *Decompose black-box use case activity diagram(s) into use case white-box activity diagram(s) and graphically allocate operations (SE-Toolkit Feature: #5, #11, #15/16).*
   - *Formalize allocation of operations (SE-Toolkit Feature: #1.9, #13, #14).*
   - *Allocate non-functional requirements and define traceability links. (SE-Toolkit Feature: #1.7).*
   - *Derive white-box sequence diagrams (SE-Toolkit Feature: #1.10, #14)*
   - Essential elements of IBD - Part 3: Delegation Ports
   - *Define and document system architecture ports and interfaces (SE-Toolkit Feature: #7, #8, #17)*
   - *Define state-based behavior of subsystem blocks and extend the state-based behavior of the actors (SE-Toolkit Feature: #9)*
   - *Verify / validate the system architecture model through model execution (use SD Compare feature)*

6. **Systems Engineering Hand-off**
   - Overview: Hand-off artifacts to SW
   - *Exercise: Hand-off a subsystem and verify the hand-off artifacts through model execution.*


[System Use Cases defined]
[else]
Analyze/Refine Stakeholder Reqs
Generate System Reqs
Link Stakeholder Reqs to System Reqs
SE-Toolkit Feature: #1.7
Define System Use Case
Link Functional / Performance Reqs to System Use Case
SE-Toolkit Feature: #1.7
[Next System Use Case]
[else]
Prioritize and Group System Use Cases
Define Use Case Model Context
SE-Toolkit Feature: #3
Define UC Functional Flow
SE-Toolkit Feature: #4, #5
Derive UC Scenarios from UC Functional Flow
SE-Toolkit Feature: #1.11, #6
Document New / Derived Reqs
Define Ports and Interfaces
SE-Toolkit Feature: #7, #8
Derive UC State-Based Behavior from UC BB-AD and BB-SDs
SE-Toolkit Feature: #9
Verify / Validate UC Model trough Model Execution
[else]
[Rainy Day Analysis]
Extend UC Model w.r.t. Error/Fail Behavior
Link UC Block Properties to Reqs
SE-Toolkit Feature: #1.7
Update Draft System ReqSpec
Use Case Realization
Define Parts of Decomposition Hierarchy
SE-Toolkit Feature: #12
Allocate System-Level Operations to Parts
SE-Toolkit Feature:
#1.9, #5, #11, #13, #14, #15/16
[Next Level of Decomposition]
[else]
Derive White-Box Sequence Diagrams
SE-Toolkit Feature: #1.11, #14
Define Ports and Interfaces
SE-Toolkit Feature: #7, #8, #17
Define Leaf Blocks State-Based Behavior
SE-Toolkit Feature: #9
Verify Realized UC Model through Model Execution
Allocate & Link Non-Functional Requirements
SE-Toolkit Feature: #1.7
Create / Update System Architecture Model
Verify Collaboration through Model Execution
[Next Use Case]
[else]

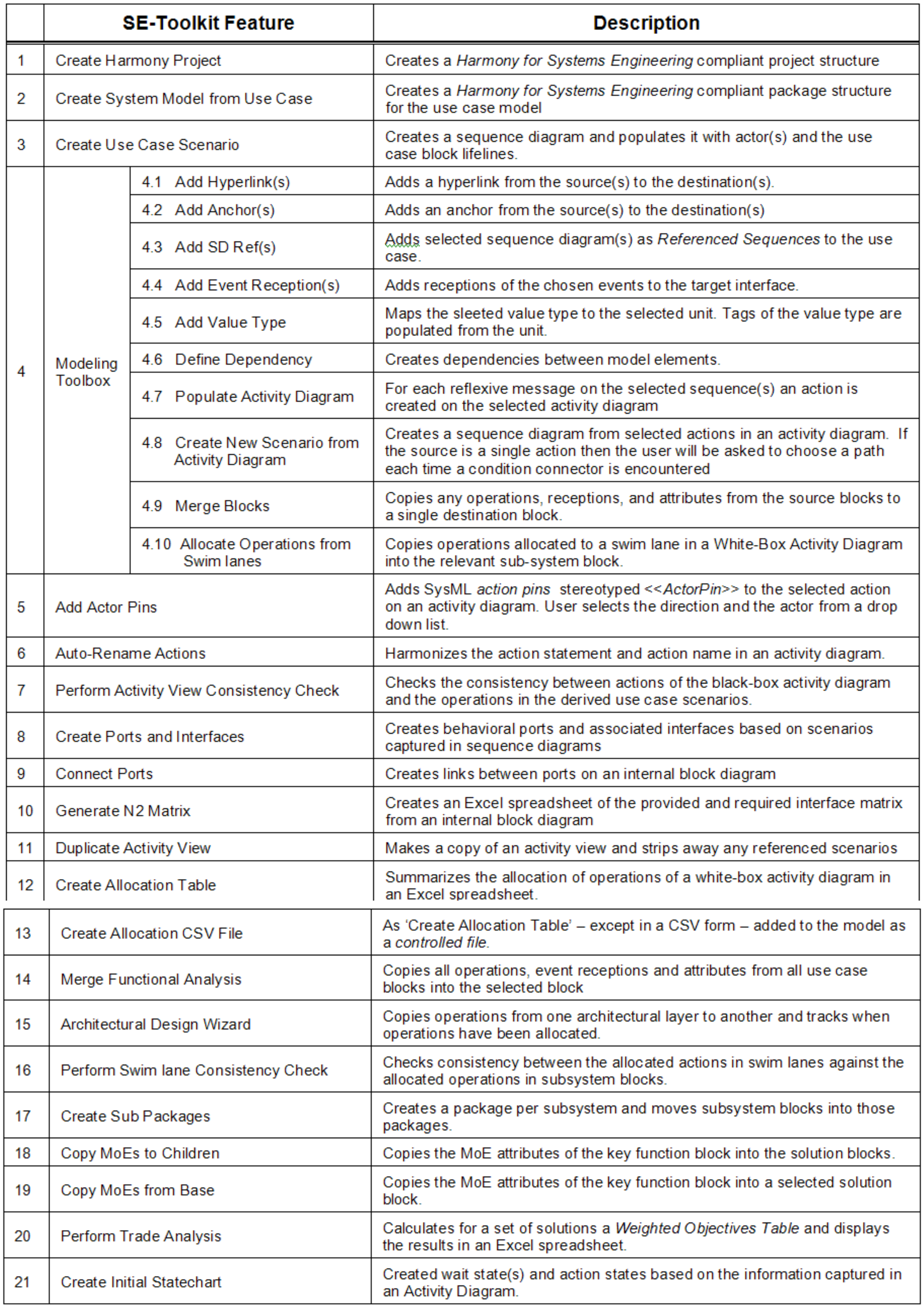
Requirements Analysis

System Functional Analysis

Architectural Design

Harmony/SE Workflow and its Support through the Rhapsody Toolkit

Rhapsody SE-Toolkit



	SE-Toolkit Feature		Description
1	Create Harmony Project		Creates a *Harmony for Systems Engineering* compliant project structure
2	Create System Model from Use Case		Creates a *Harmony for Systems Engineering* compliant package structure for the use case model
3	Create Use Case Scenario		Creates a sequence diagram and populates it with actor(s) and the use case block lifelines.
4	Modeling Toolbox	4.1 Add Hyperlink(s)	Adds a hyperlink from the source(s) to the destination(s).
		4.2 Add Anchor(s)	Adds an anchor from the source(s) to the destination(s)
		4.3 Add SD Ref(s)	Adds selected sequence diagram(s) as *Referenced Sequences* to the use case.
		4.4 Add Event Reception(s)	Adds receptions of the chosen events to the target interface.
		4.5 Add Value Type	Maps the sleeted value type to the selected unit. Tags of the value type are populated from the unit.
		4.6 Define Dependency	Creates dependencies between model elements.
		4.7 Populate Activity Diagram	For each reflexive message on the selected sequence(s) an action is created on the selected activity diagram
		4.8 Create New Scenario from Activity Diagram	Creates a sequence diagram from selected actions in an activity diagram. If the source is a single action then the user will be asked to choose a path each time a condition connector is encountered
		4.9 Merge Blocks	Copies any operations, receptions, and attributes from the source blocks to a single destination block.
		4.10 Allocate Operations from Swim lanes	Copies operations allocated to a swim lane in a White-Box Activity Diagram into the relevant sub-system block.
5	Add Actor Pins		Adds SysML *action pins* stereotyped <<*ActorPin*>> to the selected action on an activity diagram. User selects the direction and the actor from a drop down list.
6	Auto-Rename Actions		Harmonizes the action statement and action name in an activity diagram.
7	Perform Activity View Consistency Check		Checks the consistency between actions of the black-box activity diagram and the operations in the derived use case scenarios.
8	Create Ports and Interfaces		Creates behavioral ports and associated interfaces based on scenarios captured in sequence diagrams
9	Connect Ports		Creates links between ports on an internal block diagram
10	Generate N2 Matrix		Creates an Excel spreadsheet of the provided and required interface matrix from an internal block diagram
11	Duplicate Activity View		Makes a copy of an activity view and strips away any referenced scenarios
12	Create Allocation Table		Summarizes the allocation of operations of a white-box activity diagram in an Excel spreadsheet.
13	Create Allocation CSV File		As 'Create Allocation Table' – except in a CSV form – added to the model as a *controlled file*.
14	Merge Functional Analysis		Copies all operations, event receptions and attributes from all use case blocks into the selected block
15	Architectural Design Wizard		Copies operations from one architectural layer to another and tracks when operations have been allocated.
16	Perform Swim lane Consistency Check		Checks consistency between the allocated actions in swim lanes against the allocated operations in subsystem blocks.
17	Create Sub Packages		Creates a package per subsystem and moves subsystem blocks into those packages.
18	Copy MoEs to Children		Copies the MoE attributes of the key function block into the solution blocks.
19	Copy MoEs from Base		Copies the MoE attributes of the key function block into a selected solution block.
20	Perform Trade Analysis		Calculates for a set of solutions a *Weighted Objectives Table* and displays the results in an Excel spreadsheet.
21	Create Initial Statechart		Created wait state(s) and action states based on the information captured in an Activity Diagram.