

# Innovate2011

The Premier Software and Product Delivery Event



## Why MbSE? Streamlining the Development of Complex Systems

**Hans-Peter Hoffmann, Ph.D.**

*Chief Systems Methodologist*

*IBM Rational Software*

*hoffmape@us.ibm.com*

# What is Model-based Systems Engineering?

- **Model-based Systems Engineering** (MbSE) is a structured approach for the development of complex systems across the mechanical, electrical, electronic and software disciplines
  - Helps ensure that all requirements are fulfilled; that functional behaviors are realized while non-functional constraints are met
  - Employs **models** as the primary artifacts throughout the systems development lifecycle
  - Facilitates improved communication among stakeholders on a systems development team
  - Provides a disciplined way to manage complexity through abstraction



# Complex systems are everywhere



## Aerospace and Defense

e.g.  
Flight control  
Weapons  
Navigation  
Guidance  
Communications  
Autonomous systems

...



## Automotive

e.g.  
Power train  
Safety  
Entertainment  
Comfort and convenience  
Instrumentation  
Communications

...



## Electronics ...

e.g.  
Medical  
Industrial  
Consumer  
Transport  
Telecommunications

...

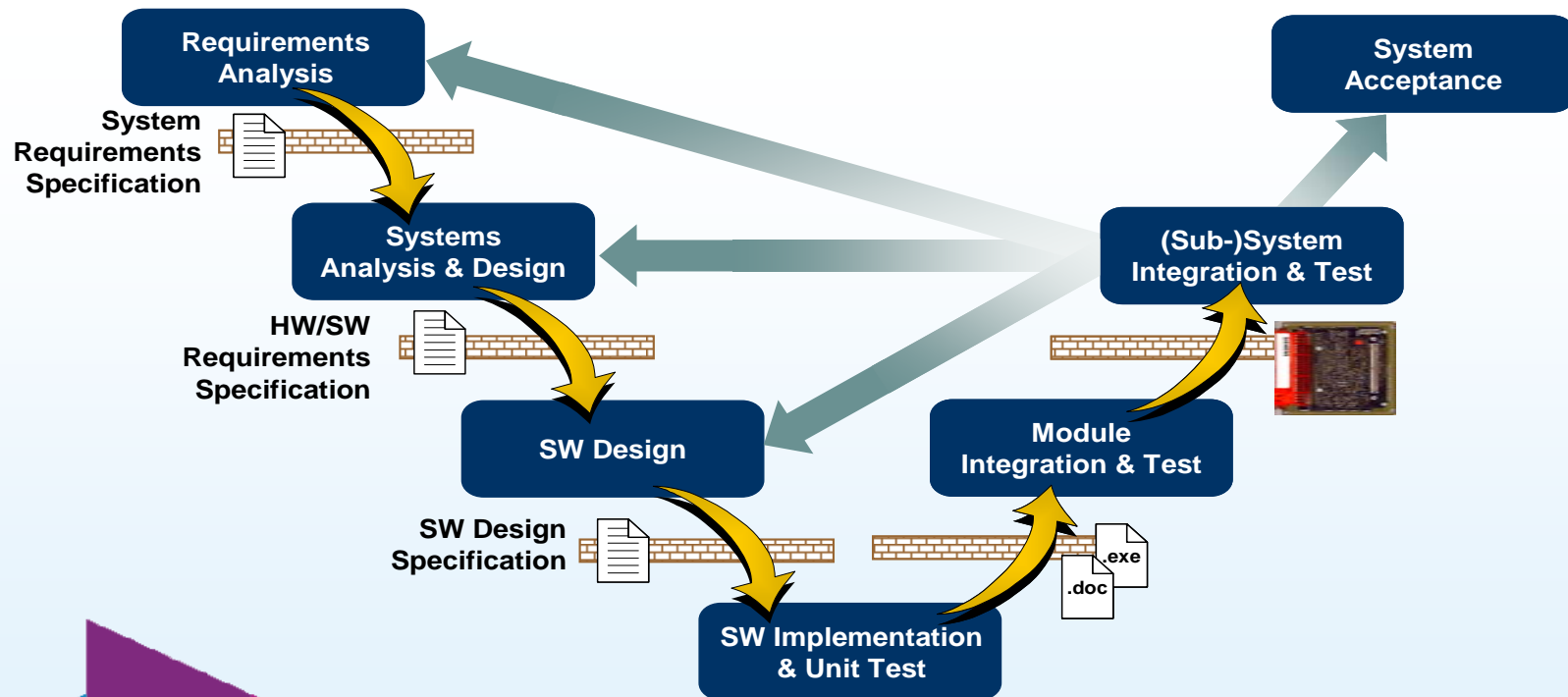
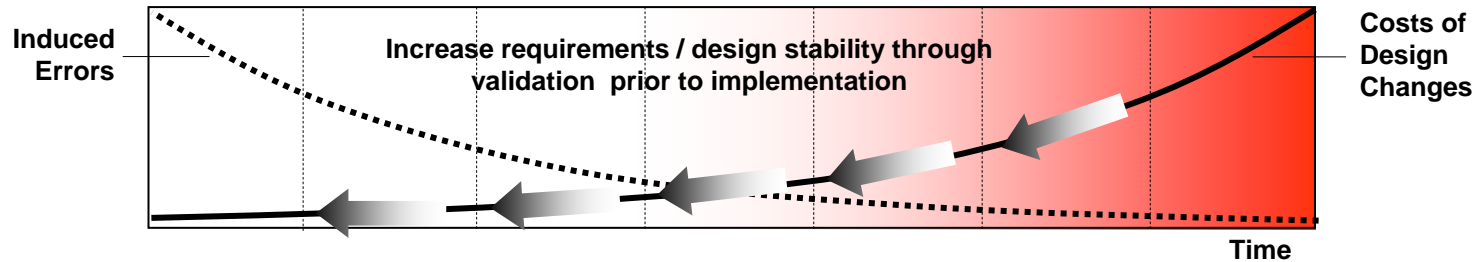
# Agenda



- **Model-based systems engineering in a model-driven development lifecycle**
- Essential SysML artifacts of the Rational MbSE approach
- Task flow in Rational *Harmony™ for Systems Engineering*
- Deploying MbSE with Rational® *Rhapsody®*
- Documentation of Rational *Harmony™ for Systems Engineering*

# Document driven development of embedded systems

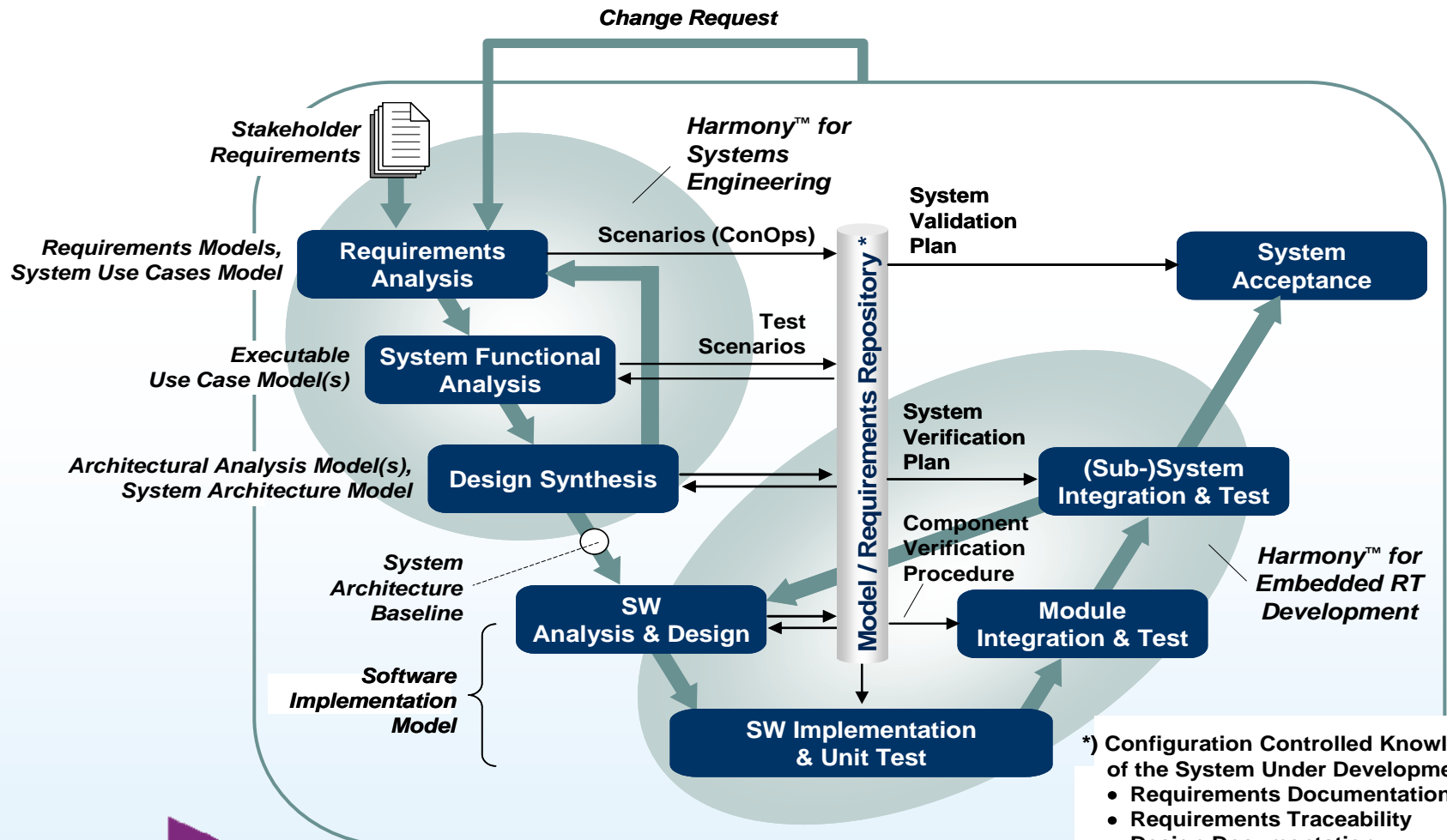
## The “Throw-it-over-the-Fence” approach





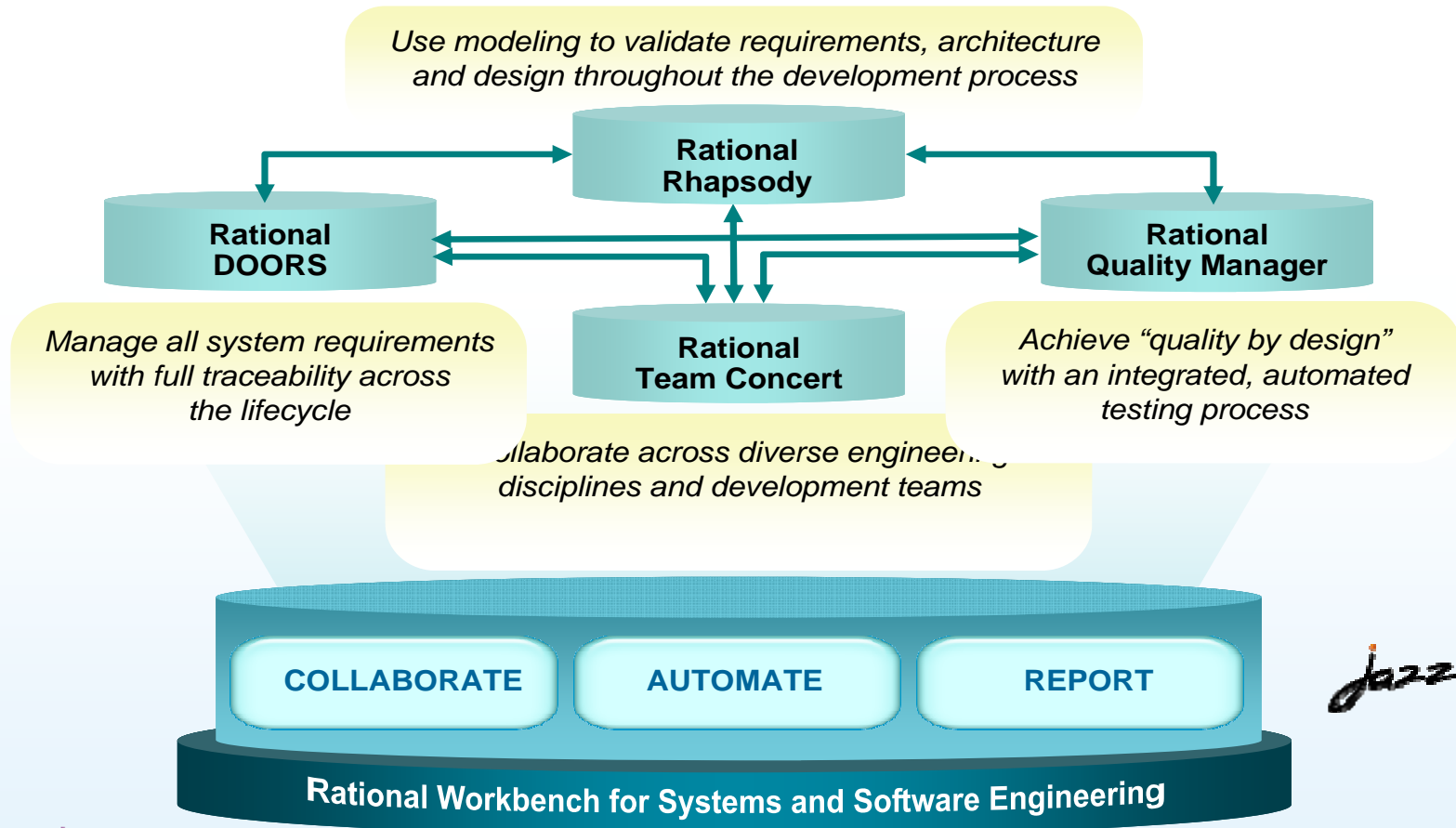
# Integrated system / software development process

## Model-driven development of embedded systems



# Rational Workbench for Systems and Software Engineering

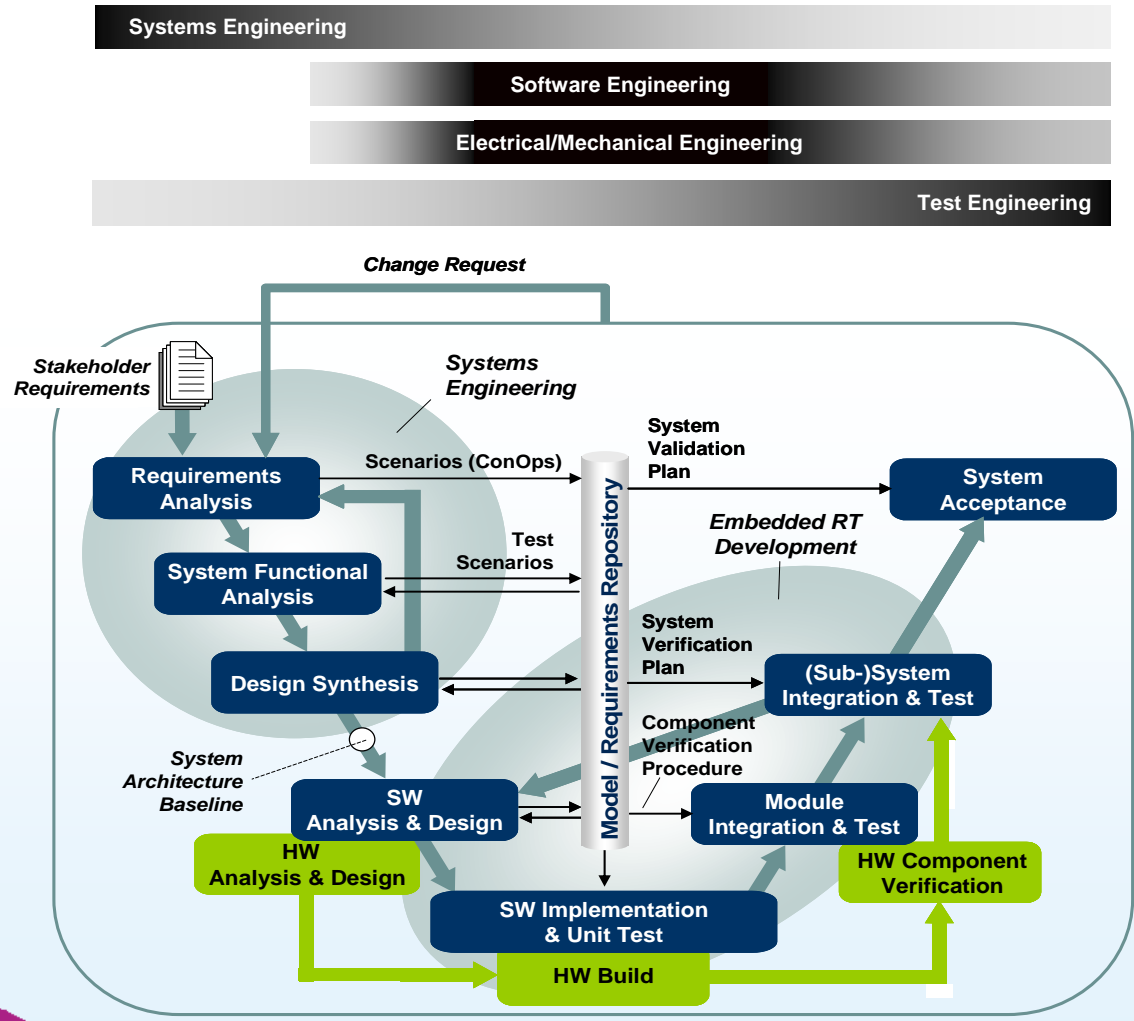
Built on a core solution set





# Integrated system / software development process

## Domains involved in the different phases of the model-driven development



# Agenda



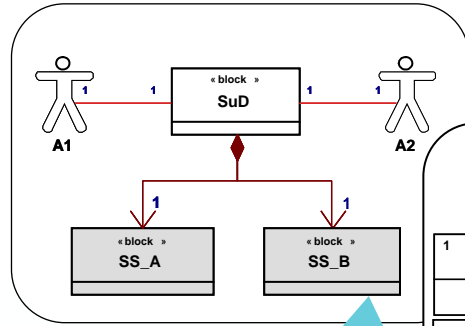
- Model-based systems engineering in a model-driven development lifecycle
- **Essential SysML artifacts of the Rational MbSE approach**
- Task flow in Rational *Harmony™ for Systems Engineering*
- Deploying MbSE with Rational® *Rhapsody®*
- Documentation of Rational *Harmony™ for Systems Engineering*

# SysML artifacts in Rational *Harmony*<sup>TM</sup> for Systems Engineering

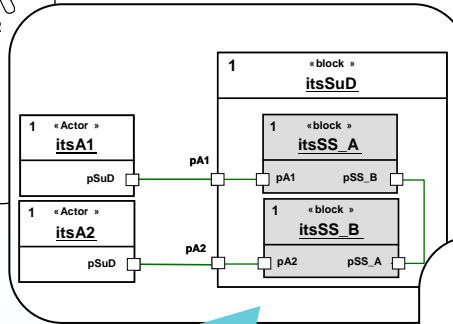
## Capturing the static view



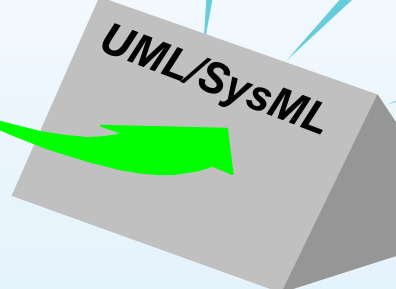
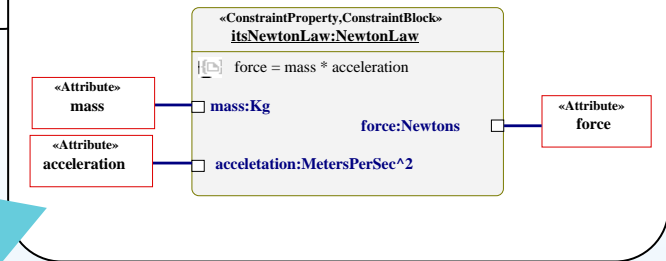
Defines structural elements (Blocks) and their relationship



Defines the realization of system structure



Defines the parametric relationship between system properties



Block Definition Diagram

Internal Block Diagram

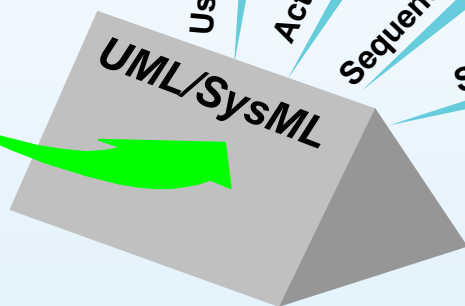
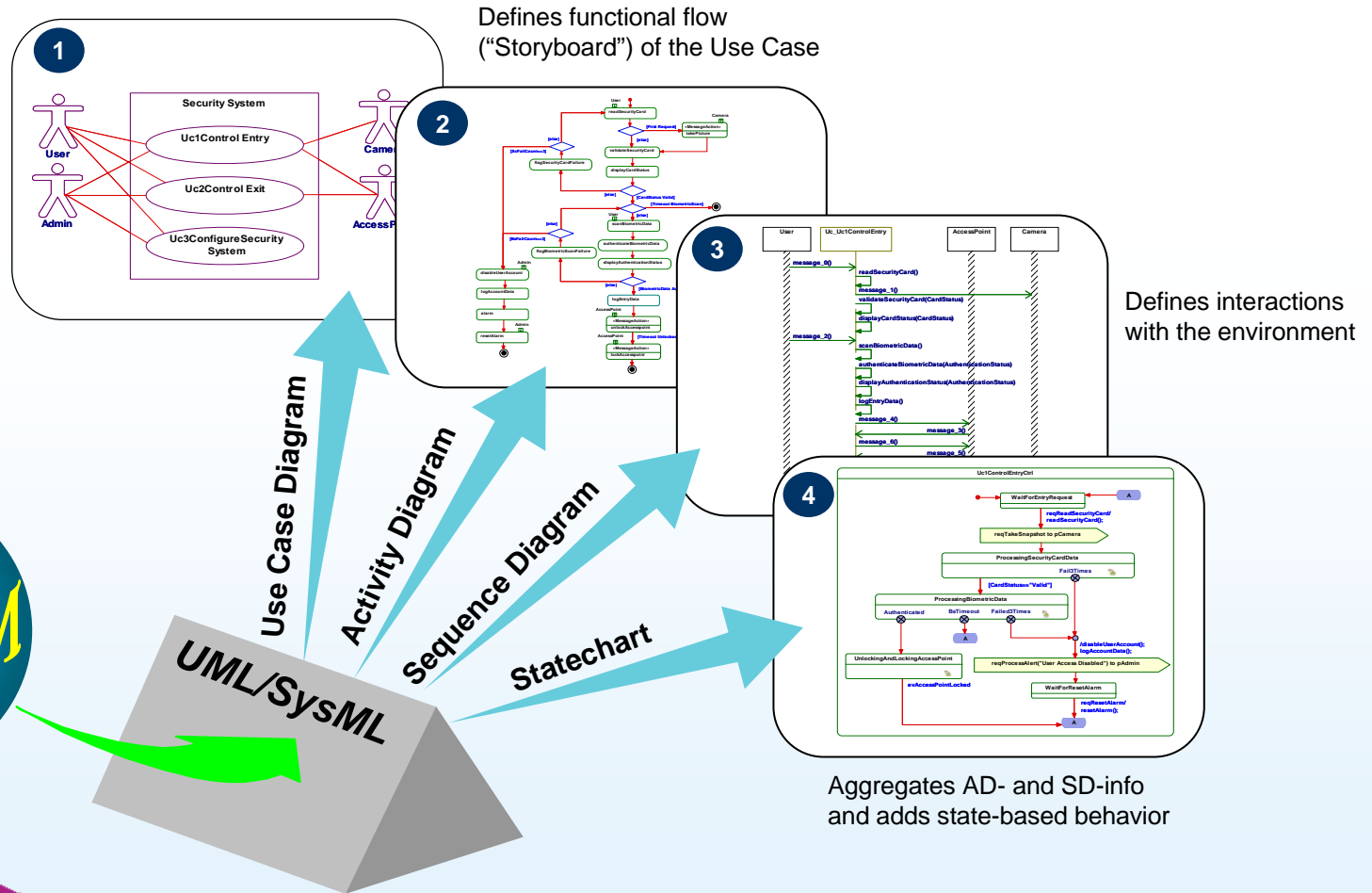
Parametric Diagram

# Capturing system behavior in a model-based approach

## Example: Creating an executable use case model



- Defines system scope
- Groups requirements into Use Cases ("Table of Contents")

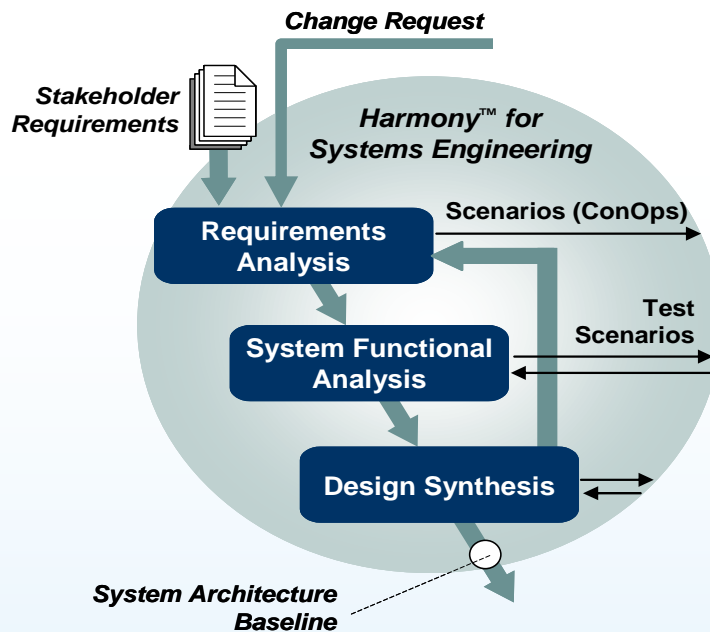


# Agenda



- Model-based systems engineering in a model-driven development lifecycle
- Essential SysML artifacts of the Rational MbSE approach
- **Task flow in Rational *Harmony™ for Systems Engineering***
- Deploying MbSE with Rational® *Rhapsody®*
- Documentation of Rational *Harmony™ for Systems Engineering*

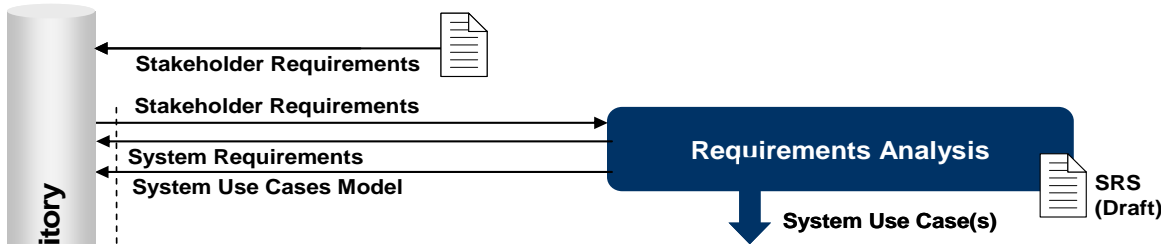
# Key objectives of the Rational *Harmony*<sup>TM</sup> for Systems Engineering workflow



- Identify / derive required system functionality
- Identify associated system states and operational modes
- Allocate required system functionality to a system architecture taking into account non-functional aspects of the requirements

# Rational Harmony™ for Systems Engineering

## Requirements analysis



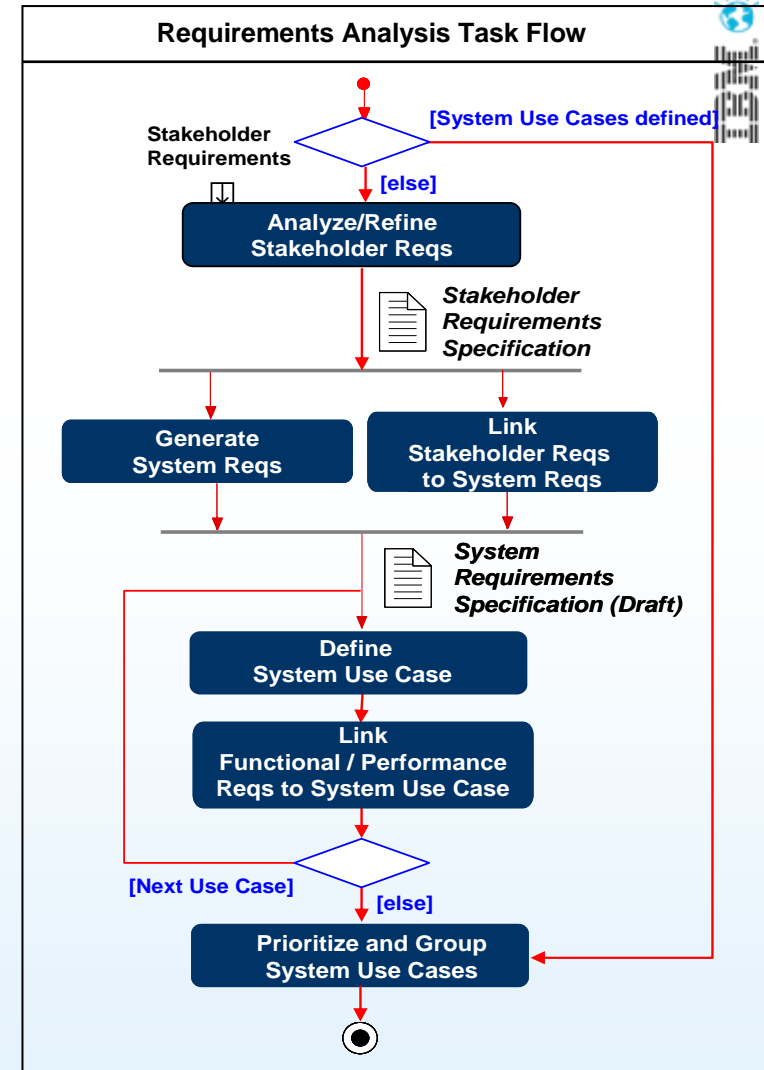
In the **requirements analysis** phase, the focus is on the analysis of the process inputs.

Stakeholder requirements are translated into system requirements that define

- what the system must do (*functional requirements*) and
- how well it must perform (*quality of service requirements*).

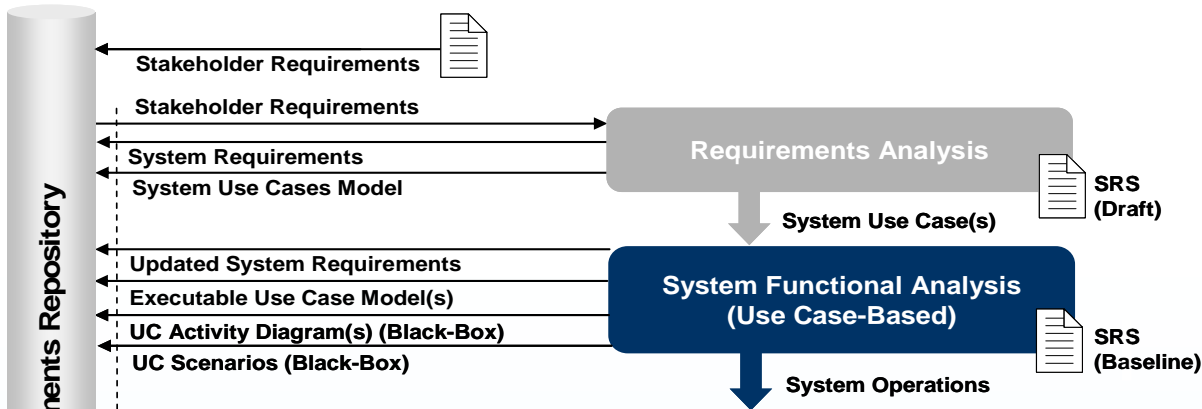
Once the requirements are sufficiently understood they are grouped into *Use Cases*.

Links providing traceability to original requirements



# Rational Harmony™ for Systems Engineering

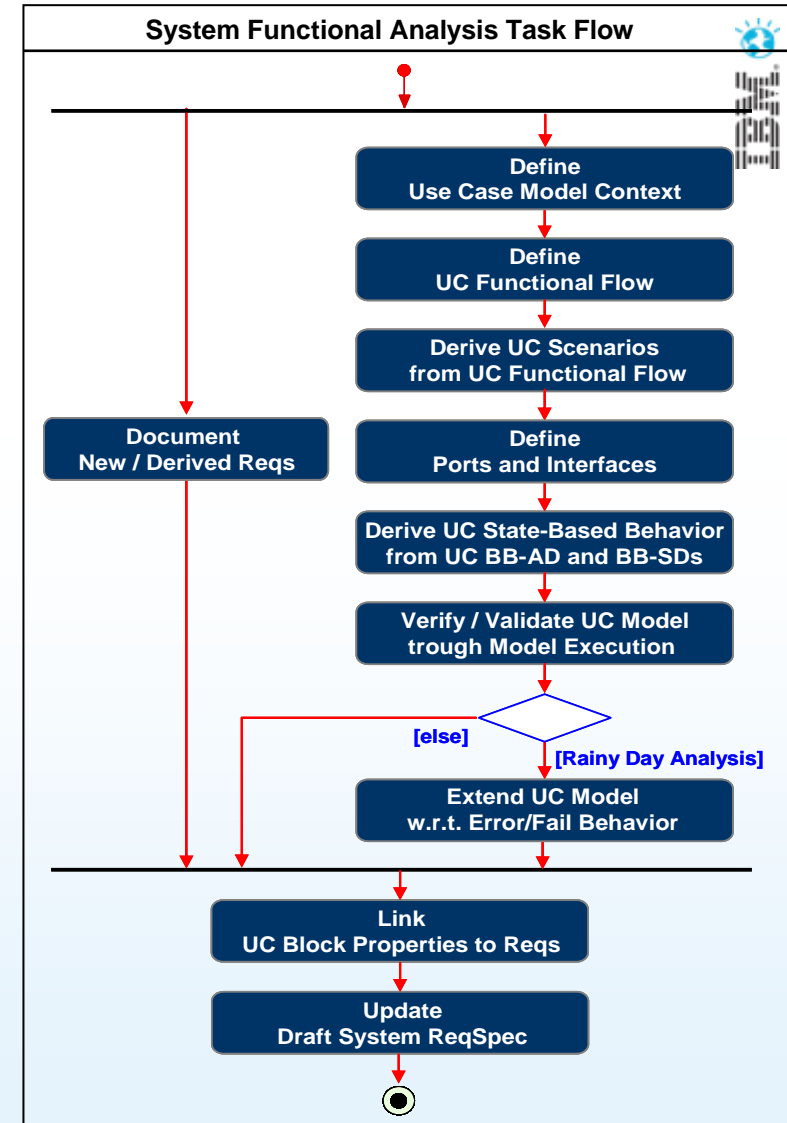
## System functional analysis



In the **system functional analysis** phase, the focus is on the translation of the functional requirements into a coherent description of system *operations*.

Each use case of an iteration is translated into a model and the underlying requirements verified and validated through *model execution*.

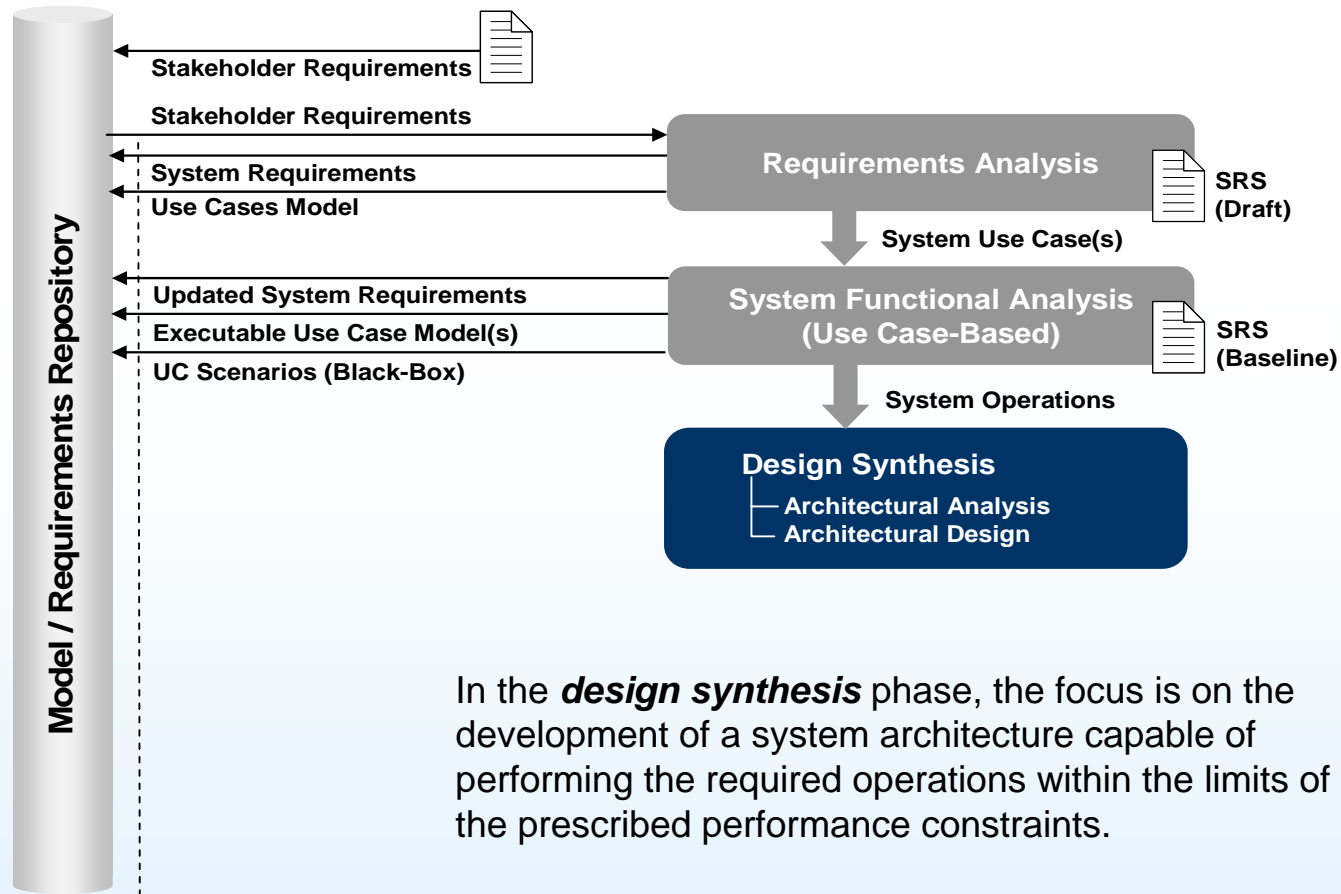
Links providing traceability to original requirements





# Rational *Harmony*<sup>TM</sup> for Systems Engineering

## Design synthesis

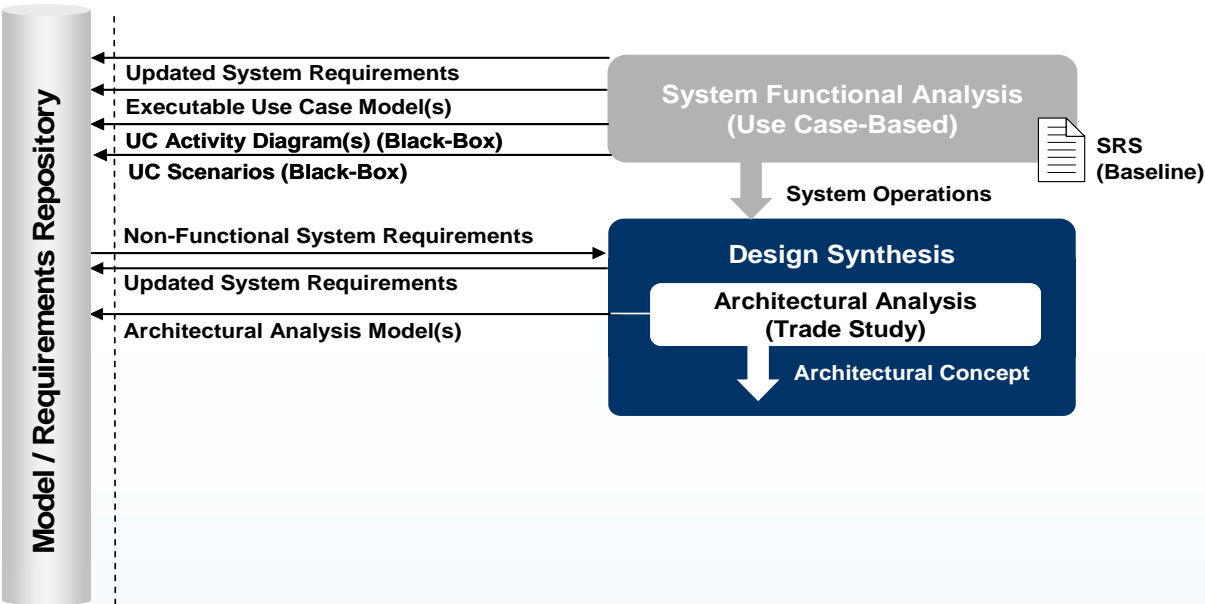


In the **design synthesis** phase, the focus is on the development of a system architecture capable of performing the required operations within the limits of the prescribed performance constraints.

Links providing traceability  
to original requirements

# Design synthesis

## Architectural analysis



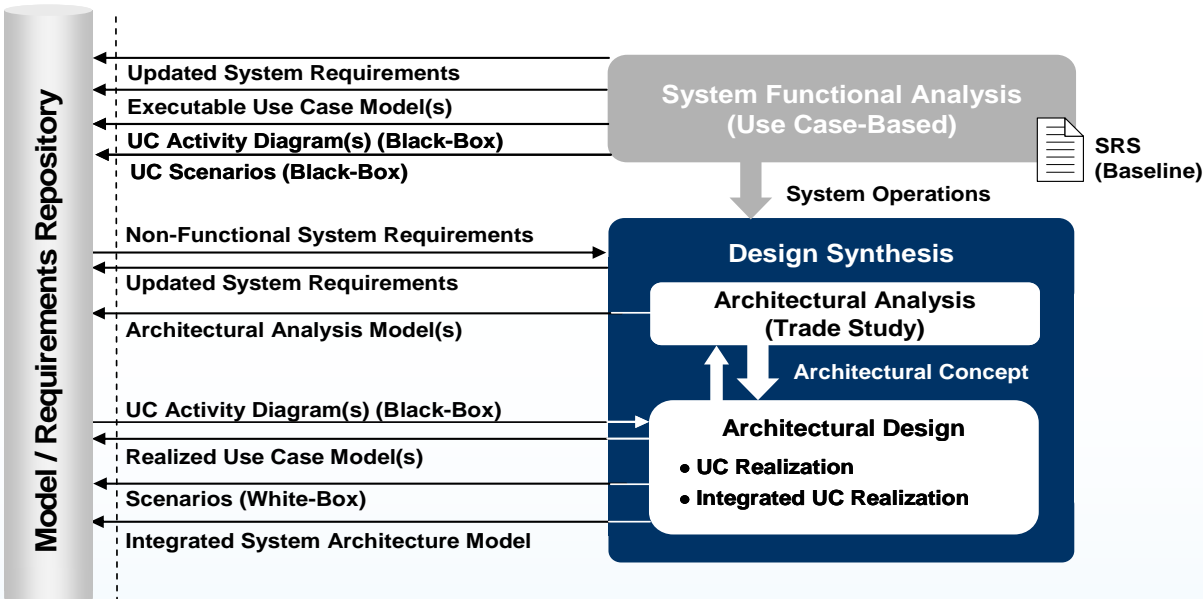
Links providing traceability to original requirements

The objective the **architectural analysis** phase is to elaborate the optimum design concept based upon a set of criteria (e.g. *Measures of Effectiveness*, MoEs) that are weighted according to their relative importance.



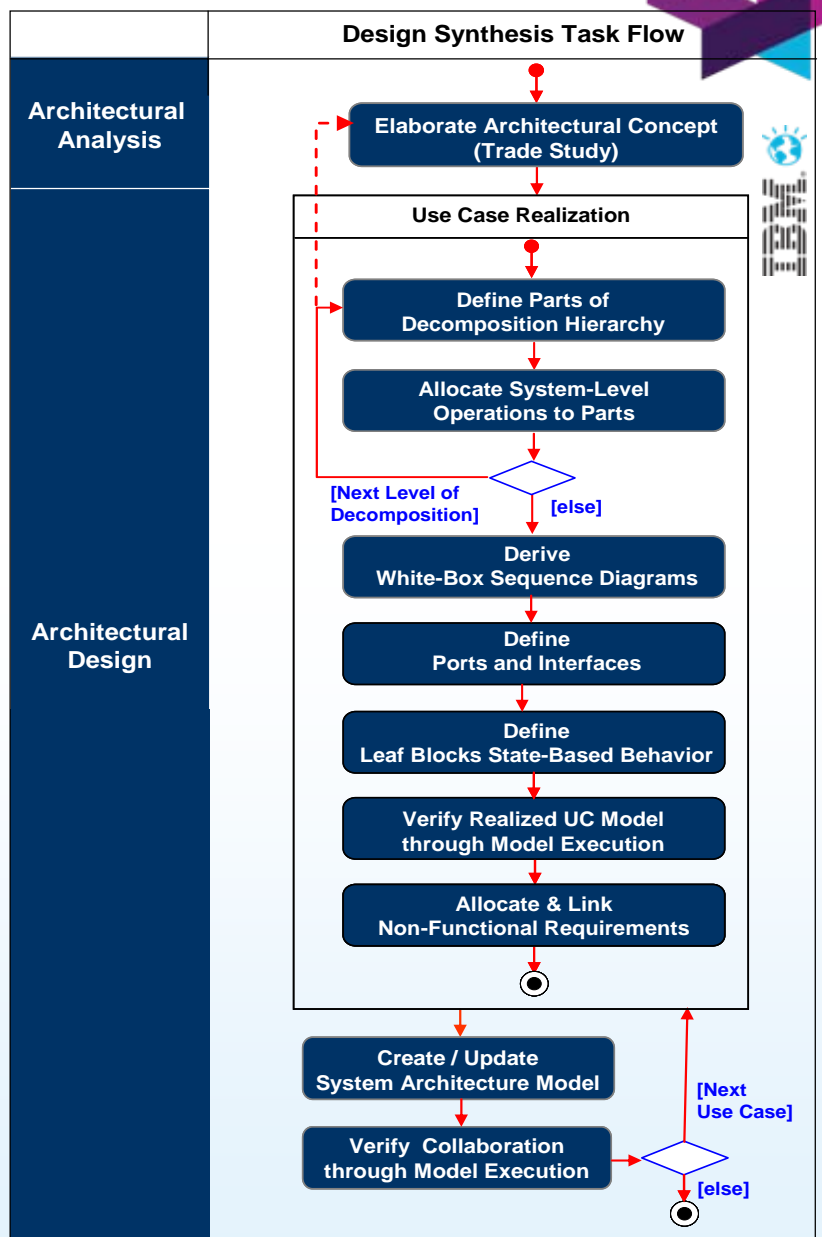
# Design synthesis

## Architectural analysis

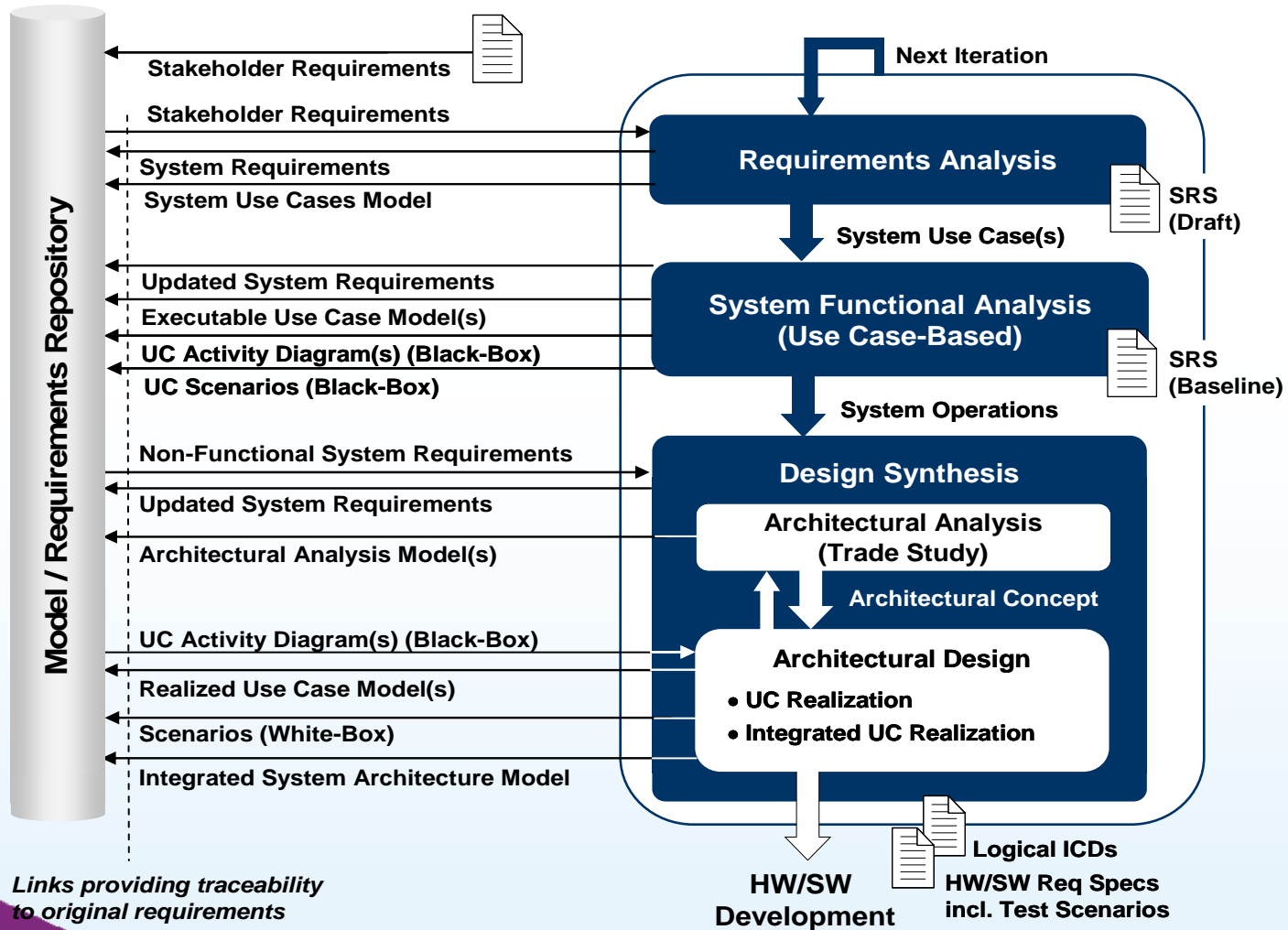


Links providing traceability to original requirements

The focus of the **architectural design** phase is on the allocation of the functional requirements (= system operations) and non-functional requirements to an architectural structure. The architectural design is performed *incrementally* for each use case of an iteration by transitioning from the black-box view to the white-box view – also referred to as **use case realization**.



# Rational Harmony™ for Systems Engineering



# Systems engineering handoff to the subsequent system development



In a model-driven development the key artifact of the handoff from systems engineering to the subsequent system development is the *baselined* executable model.

This model is the repository from which specification documents (e.g. HW/SW requirements specifications, ICDs, ...) are generated.

Scope and content of the hand-off is dependent on the characteristics of the project and the organizational structure systems engineering is embedded.

# Systems engineering handoff to the subsequent system development



The hand-off packages typically are composed of baselined executable CI model(s) which contain

- The definition of allocated operations including their links to the associated system functional and performance requirements
- The definition of the associated state-based behavior, captured in a statechart diagram
- The allocated and linked non-functional requirements
- The definition of ports and *logical* interfaces
- Test scenarios, derived from system-level use case scenarios

# Agenda



- Model-based systems engineering in a model-driven development lifecycle
- Essential SysML artifacts of the Rational MbSE approach
- Task flow in Rational *Harmony™ for Systems Engineering*
- **Deploying MbSE with Rational® *Rhapsody®***
- Documentation of Rational *Harmony™ for Systems Engineering*

# Deploying model-based systems engineering

## Project specific model-based Systems Engineering Handbook



*Table of Contents*

- 4.2.4.1 Architectural Analysis Package.....
- 4.2.4.2 Architectural Design Package.....
- 4.2.5 System-Level Definitions.....
- 4.3 Configuration Management.....
- 4.3.1 Strategy.....
- 4.3.1.1 Requirements Model.....
- 4.3.1.2 Black-Box Use Case Model.....
- 4.3.1.3 White-Box Use Case Model.....
- 4.3.1.4 Integrated System Architecture Model.....
- 4.3.2 Model Integration.....
- 4.3.2.1 Sharing Model Elements between Models.....
- 5 Requirements Management.....
- 5.1 Strategy.....
- 5.2 Workflow.....
- 5.2.1 Identifying new or derived Requirements.....
- 5.2.2 Setting up the Gateway.....
- 5.2.3 Loading the DOORS Module into Gateway.....
- 5.2.4 Importing Requirements into the Rhapsody Model.....
- 5.2.5 Setting up Traceability.....
- 5.2.6 Exporting SRD Information.....
- 5.2.7 Repository Files used by Gateway.....
- 6 Documentation.....
- 6.1 Strategy.....
- Appendix.....
- A1 Rhapsody Action Language.....
- A2 Usage of Activity Chart Information in the SE Workflow.....
- A3 Guideline: Logical Decomposition of a Use Case Black-Box Activity Diagram.....
- A4 Guideline: Deriving a Statechart Diagram.....
- A5 Rhapsody SE-Toolkit (Overview).....
- Application Notes.....
- List of Acronyms.....
- References.....

*Model-Based Systems Engineering Handbook*

*Table of Contents*

- 1 Introduction.....
- 1.1 Scope.....
- 1.2 Document Overview.....
- 2 Model-based Systems Engineering.....
- 2.1 Model-based Systems Engineering in a Model-driven System Development <Project> Model-based Systems Engineering Workflow.....
- 2.2 MbSE Workflow Overview.....
- 2.2.1 MbSE Workflow Overview.....
- 2.2.2 Requirements Analysis.....
- 2.2.3 System Functional Analysis.....
- 2.2.4 Architectural Design.....
- 2.2.4.1 Use Case Realization.....
- 2.2.4.2 Integrated Use Case Realization.....
- 3 Modeling Guidelines.....
- 3.1 Essential SysML Artifacts.....
- 3.2 Service Request-Driven Modeling Approach.....
- 3.3 General Guidelines and Drawing Conventions.....
- 3.4 SysML Diagram Guidelines.....
- 3.4.1 Use Case Diagram.....
- 3.4.1.1 Elements and Artifacts.....
- 3.4.1.2 Guidelines and Drawing Conventions.....
- 3.4.1.3 Naming Conventions.....
- 3.4.2 Block Definition Diagram.....
- 3.4.2.1 Elements and Artifacts.....
- 3.4.2.2 Guidelines and Drawing Conventions.....
- 3.4.2.3 Naming Conventions.....
- 3.4.3 Internal Block Diagram.....
- 3.4.3.1 Elements and Artifacts.....
- 3.4.3.2 Guidelines and Drawing Conventions.....
- 3.4.3.3 Naming Conventions.....
- 3.4.4 Activity Diagram.....
- 3.4.4.1 Elements and Artifacts.....
- 3.4.4.2 Guidelines and Drawing Conventions.....
- 3.4.4.3 Naming Conventions.....
- 3.4.5 Sequence Diagram.....
- 3.4.5.1 Elements and Artifacts.....
- 3.4.5.2 Guidelines and Drawing Conventions.....
- 3.4.5.3 Naming Conventions.....
- 3.4.6 Statechart Diagram.....
- 3.4.6.1 Elements and Artifacts.....
- 3.4.6.2 Guidelines and Drawing Conventions.....
- 3.4.6.3 Naming Conventions.....
- 3.5 Profiles.....
- 4 Model Management.....
- 4.1 Organization.....
- 4.2 Rhapsody Project Structure.....
- 4.2.1 Project Structure Overview.....
- 4.2.2 Requirements Analysis Package.....
- 4.2.3 Functional Analysis Package.....
- 4.2.4 Design Synthesis.....

*Model-Based Systems Engineering Handbook*

IBM.

IBM Software Group | Rational software

< Project Name >

Model-Based Systems Engineering Handbook

Release/Revision	Release/Revision Date







## 2.2 Model-based Systems Engineering Workflow

The model-based systems engineering workflow applied in the xxxx project is a specialization of the Harmony/SE workflow outlined in the previous chapter. This chapter outlines the details of the MbSE workflow. It starts with a general overview. Then, for each phase the associated tasks and associated work products are described in detail.

### 2.2.1 MbSE Workflow Overview

Fig. 2.2-1 provides an overview of the MbSE workflow. It shows for each of the SE phases the generated models together with the associated Rhapsody projects. How the different projects are managed from the configuration point of view will be detailed in section 4.

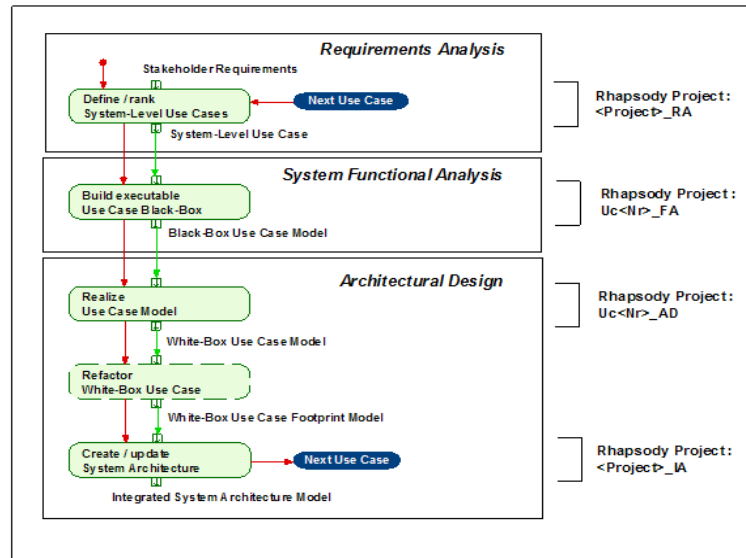


Fig. 2.2-1 Model-based Systems Engineering Workflow in the xxxx Project

The MbSE workflow is use case based. It starts with the definition of the system use cases. They are listed in the requirements package of the Rhapsody project <ProjectName>\_RA (ref. Section 4). The system use cases are ranked according to their importance for the development of

# MbSE Handbook

## Standardizing the MbSE workflow and Rhapsody tool usage

# The Alternative: MbSE Handbook in RMC



**Rational. Method Composer**

SE Handbook

- Introduction
- Model-Based Systems Engineering**
  - Modeling Guidelines
  - Model Management
  - Requirements Management
  - Documentation
  - Appendix

Model-Based Systems Engineering > Model-based Systems Engineering

## Capability Pattern: Model-based Systems Engineering

Provides an overview of the MbSE workflow. It shows for each of the SE phases the generated models together with the associated Rhapsody projects.

**Work Breakdown Structure**

**Description**

The MbSE workflow is use case based. It starts with the definition of the system use cases. They are listed in the requirements package of the Rhapsody project \_RA. The system use cases are ranked according to their importance for the development of the system architecture. Each of these system use cases defines the increment of the SE workflow iteration.

In the subsequent functional analysis phase the chosen system use case is transformed into an executable use case model. The modeling is performed in a separate Rhapsody project (Uc\_FA). Output of the functional analysis phase is the baselined verified/validated executable Black-Box Use Case Model.

The first step in the architectural design phase is the realization of the Black-Box Use Case Model. The realization is performed in a separate Rhapsody project (Uc\_AD). Based on design decisions – optionally elaborated in a trade study – and dependent on the hand-off to the subsequent development, the use case model is decomposed, and system-level operations then are allocated to the relevant subsystems/ subsystem components. The collaboration of the decomposed subsystems is verified through model execution. Output of the use case realization is the baselined White-Box Use Case Model.

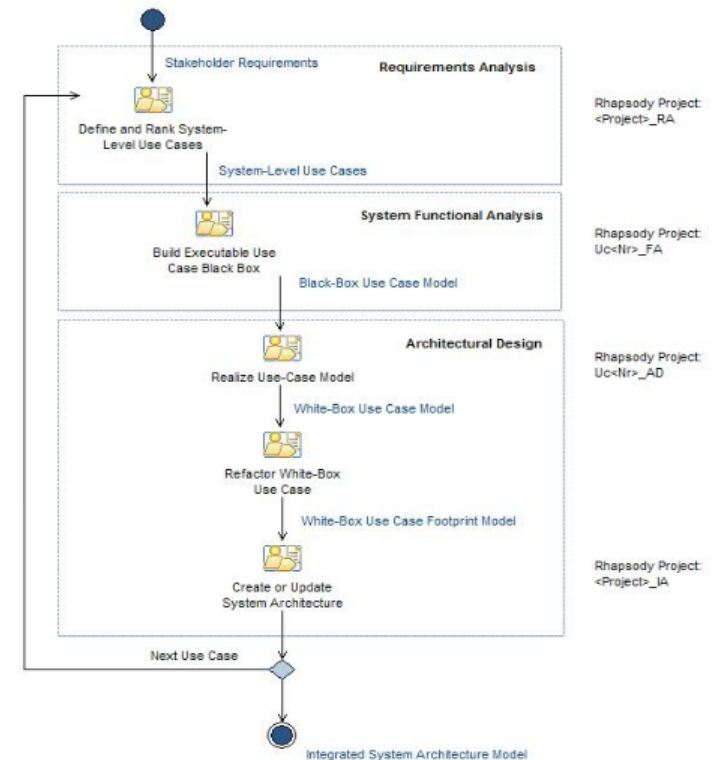
The final task in the architectural design phase is the creation/update of the Integrated System Architecture Model (Rhapsody project \_IA). This model is the aggregate of the baselined realized use case models. In order to comply with the chosen standardized architectural structure of the Integrated System Architecture, the White-Box Use Case Model needs to be refactored accordingly (ref. section 2.2.4.2) prior to its integration. The correctness and completeness of the updated Integrated System Architecture Model may be verified through model execution. The baselined Integrated System Architecture Model is the key artifact of the hand-off to the subsequent system development. It is the repository from which specification documents (HW/SW Requirements Specifications, ICDs ...) are generated.

The outlined workflow is repeated for all system use cases.

Back to top

Done

## Workflow



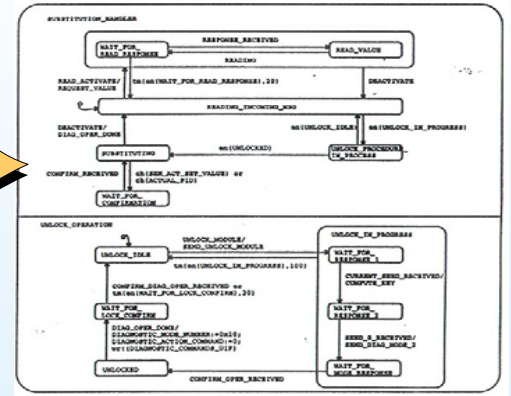
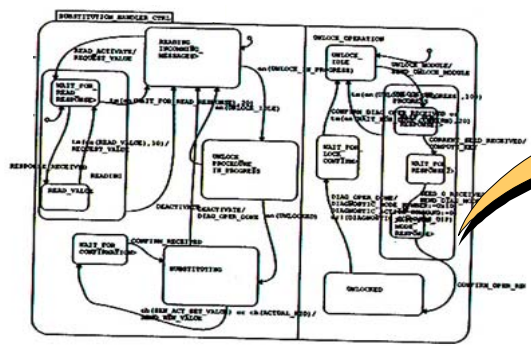
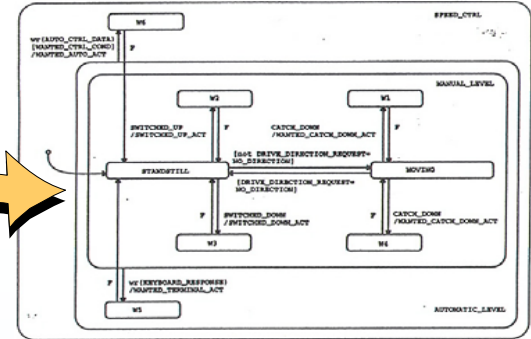
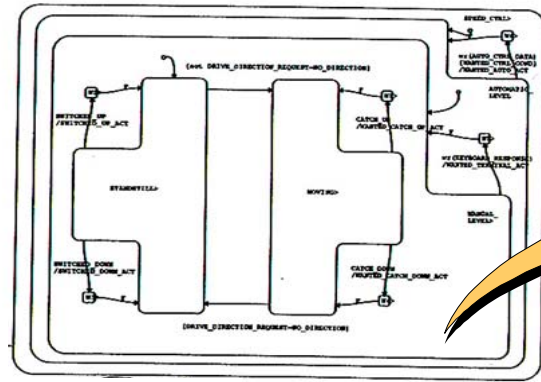
# MbSE Handbook

## Standardizing the usage of the modeling language



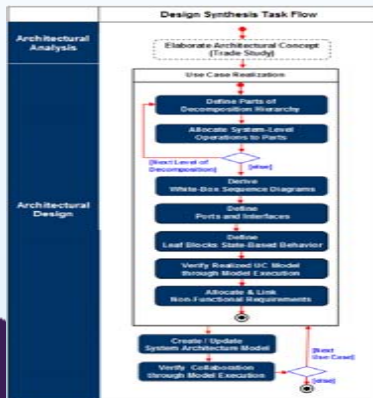
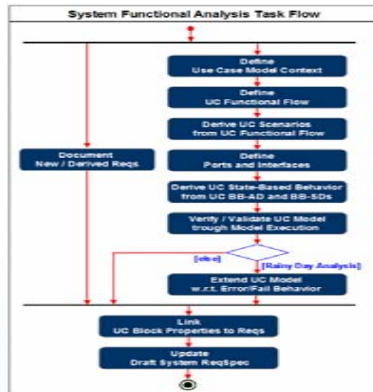
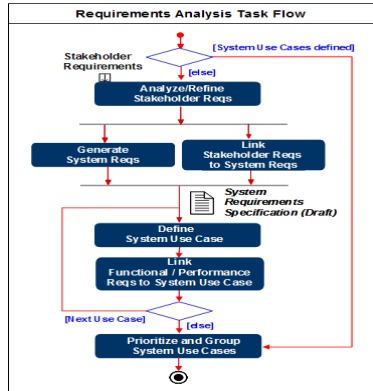
**3 Modeling Guidelines**

- 3.1 Essential SysML Artifacts
- 3.2 Service Request-Driven Modeling Approach
- 3.3 General Guidelines and Drawing Conventions
- 3.4 SysML Diagram Guidelines
  - 3.4.1 Use Case Diagram
    - 3.4.1.1 Elements and Artifacts
    - 3.4.1.2 Guidelines and Drawing Conventions
    - 3.4.1.3 Naming Conventions
  - 3.4.2 Block Definition Diagram
    - 3.4.2.1 Elements and Artifacts
    - 3.4.2.2 Guidelines and Drawing Conventions
    - 3.4.2.3 Naming Conventions
  - 3.4.3 Internal Block Diagram
    - 3.4.3.1 Elements and Artifacts
    - 3.4.3.2 Guidelines and Drawing Conventions
    - 3.4.3.3 Naming Conventions
  - 3.4.4 Activity Diagram
    - 3.4.4.1 Elements and Artifacts
    - 3.4.4.2 Guidelines and Drawing Conventions
    - 3.4.4.3 Naming Conventions
  - 3.4.5 Sequence Diagram
    - 3.4.5.1 Elements and Artifacts
    - 3.4.5.2 Guidelines and Drawing Conventions
    - 3.4.5.3 Naming Conventions
  - 3.4.6 Statechart Diagram
    - 3.4.6.1 Elements and Artifacts
    - 3.4.6.2 Guidelines and Drawing Conventions
    - 3.4.6.3 Naming Conventions
- 3.5 Profiles





# Deploying model-based systems engineering Managing the modeling activities: Project plan



Task Name	Duration	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10
1 <b>Requirements Analysis</b>	2 days	[Task bar spanning W1 to W2]									
2 Define Use Case	1 day	[Task bar at start of W2]									
3 Allocate Reqs to Use Case	1 day	[Task bar at start of W2]									
4 <b>System Functional Analysis</b>	21 days	[Task bar spanning W2 to W4]									
5 Create Black-Box Activity Diagram	10 days	[Task bar spanning W2 to W3]									
6 Review & Update Black-Box Activity Diagram	3 days	[Task bar at end of W3]									
7 Derive Use Case Black-Box Sequence Diagrams	2 days	[Task bar at end of W3]									
8 Define System-Level Ports & Interfaces	1 day	[Task bar at end of W3]									
9 Create Use Case Statechart	2 days	[Task bar at end of W3]									
10 V&V Use Case Model through Model Execution	2 days	[Task bar at end of W3]									
11 Link Model Properties to Requirements	1 day	[Task bar at end of W3]									
12 Black-Box Use Case Model Baseline	0 days	[Task bar at end of W3]									
13 <b>Architectural Design</b>	21 days	[Task bar spanning W5 to W7]									
14 Define System Architectural Structure (BDD, IBD)	1 day	[Task bar at start of W5]									
15 Create White-Box Activity Diagram	4 days	[Task bar at start of W5]									
16 Review White-Box Activity Diagram	2 days	[Task bar at start of W5]									
17 Derive Use Case White-Box Sequence Diagrams	3 days	[Task bar at start of W5]									
18 Define System Architecture Ports & Interfaces	1 day	[Task bar at start of W5]									
19 Create Subsystem Components Statechart	3 days	[Task bar at start of W5]									
20 Verify Realized Use Case Model through Model Execution	2 days	[Task bar at start of W5]									
21 Allocate & Link Non-functional Requirements	1 day	[Task bar at start of W5]									
22 Realized Use Case Model Baseline	0 days	[Task bar at start of W5]									
23 Generate Report	2 days	[Task bar at start of W5]									
24 Merge Realized UC Model in Integrated System Architecture Model	2 days	[Task bar at start of W5]									
25 Integrated System Architecture Model Baseline	0 days	[Task bar at start of W5]									

# Agenda



- Model-based systems engineering in a model-driven development lifecycle
- Essential SysML artifacts of the Rational MbSE approach
- Task flow in Rational *Harmony™ for Systems Engineering*
- Deploying MbSE with Rational® *Rhapsody®*
- **Documentation of Rational *Harmony™ for Systems Engineering***

# Rational Harmony™ for Systems Engineering Deskbook

## Rational Rhapsody® tool focused documentation



**Case Study: Hand-Off to Subsystem Development**

**Case Study: Design Synthesis**

**Case Study: System Functional Analysis**

**Case Study: Requirements Analysis**

**Rhapsody Project Structure**

**Fundamentals of Harmony for Systems Engineering**

IBM Software Group

February 2011

**Systems Engineering Best Practices with the Rational Solution for Systems and Software Engineering**

**Deskbook** Release 3.1.2

Model-Based Systems Engineering with *Rational Rhapsody* and *Rational Harmony for Systems Engineering*

Hans-Peter Hoffmann, Ph.D.  
Chief Systems Methodologist

[hoffmape@us.ibm.com](mailto:hoffmape@us.ibm.com)

© 2011 IBM Corporation

**Table of Contents:**

- 4.4 Design Synthesis.....
  - 4.4.1 Architectural Analysis (Trade-Off Analysis).....
    - 4.4.1.1 Definition of Key System Functions.....
    - 4.4.1.2 Definition of Candidate Solutions.....
    - 4.4.1.3 Definition of Assessment Criteria.....
    - 4.4.1.4 Assigning Weights to Assessment Criteria.....
    - 4.4.1.5 Definition of a Utility Curve for each Criterion.....
    - 4.4.1.6 Assigning Measures of Effectiveness (MoE) to each Solution.....
    - 4.4.1.7 Determination of Solution.....
  - 4.4.2 Architectural Design.....
    - 4.4.2.1 Security System Decomposition.....
    - 4.4.2.2 Graphical Allocation of Operations.....
    - 4.4.2.3 Formalizing the Allocation of Operations.....
    - 4.4.2.4 Allocation of Non-functional Requirements.....
  - 4.4.3 Detailed Architectural Design.....
    - 4.4.3.1 Decomposition of Black-Box Use Case Scenarios.....
    - 4.4.3.2 Definition of Ports and Interfaces.....
    - 4.4.3.3 Definition of State-based Behavior.....
    - 4.4.3.4 System Architecture Model Verification and Validation.....
- 2 FUNDAMENTALS OF HARMONY FOR SYSTEMS ENGINEERING.....
  - 2.1 Integrated Systems / Embedded Software Development Process Harmony.....
  - 2.2 Model-Driven Systems Engineering Process.....
    - 2.2.1 Requirements Analysis.....
    - 2.2.2 System Functional Analysis.....
    - 2.2.3 Design Synthesis.....
      - 2.2.3.1 Architectural Analysis (Trade Study).....
      - 2.2.3.2 Architectural Design.....
      - 2.2.3.3 Detailed Architectural Design.....
    - 2.2.4 Systems Engineering Hand-Off.....
  - 2.3 Essential SysML Artifacts of Model-Driven Systems Engineering.....
    - 2.3.1 Requirements Diagram.....
    - 2.3.2 Structure Diagrams.....
      - 2.3.2.1 Block Definition Diagram.....
      - 2.3.2.2 Internal Block Diagram.....
      - 2.3.2.3 Parametric Diagram.....
    - 2.3.3 Behavior Diagrams.....
      - 2.3.3.1 Use Case Diagram.....
      - 2.3.3.2 Activity Diagram.....
      - 2.3.3.3 Sequence Diagram.....
      - 2.3.3.4 Statechart Diagram.....
    - 2.3.4 Artifact Relationships at the Requirements Analysis / Functional Analysis Level.....
  - 2.4 Service Request-Driven Modeling Approach.....

# Rational Systems Engineering Practices captured in RMC



Rational. Method Composer

Systems Delivery

- SE Practices - Introduction
  - SE Practices
    - Elaborate Draft System Requirements Specification
    - Detailed Use-Case Requirements Analysis
    - Build and Validate Use Cases
    - Architectural Analysis - Key System Functions
    - Architectural Analysis - Operation Based
    - Trade Study - Weighted Objectives Method
    - Architectural Design - Operation Based
    - Architectural Design - Use-Case Based
    - Joint Realization
  - SE Lifecycle
    - SE Overview
    - SE - Use-case focused
    - SE - Operation focused
  - SE Workflows
  - SE Artifacts
    - Requirements
    - Architecture
    - Traceability
    - External
  - SE Roles
    - Requirements Engineer
    - System Architect
    - Stakeholder
  - Release Info

SE Practices - Introduction

A short description of the Systems Engineering practices.

### Main Description

#### Introduction

The new systems engineering practices reflect the vast experience accumulated by the IBM Rational consultants during various Systems Engineering engagements in a number of different industries. This process guidance covers an area starting from stakeholder needs and finishing with a hand-off to the systems development groups of the System Architecture baseline. These practices could be combined in different ways and could play as part of different types of lifecycles. One example is the traditional V lifecycle shown below where the SE practices are part of the front-end domain (see red arrow).

The diagram illustrates the V-model lifecycle. The left side of the 'V' represents development phases: Requirements Analysis, System Functional Analysis, Design Synthesis, and SW Analysis & Design. The right side represents testing phases: SW Implementation & Unit Test, Module Integration & Test, (Sub-)System Integration & Test, and System Acceptance. A central vertical bar represents the Model/Requirements Repository. A red arrow labeled 'Systems Engineering Practices' points to the Requirements Analysis phase. A dashed arrow labeled 'System Architecture Baseline' points from SW Analysis & Design to the Model/Requirements Repository. A feedback loop labeled 'System Changes' connects the top of the V back to the start. Various plans and procedures are associated with the phases, such as System Validation Plan, System Verification Plan, Component Verification Procedure, Test Scenarios, and Scenarios (ConOps).

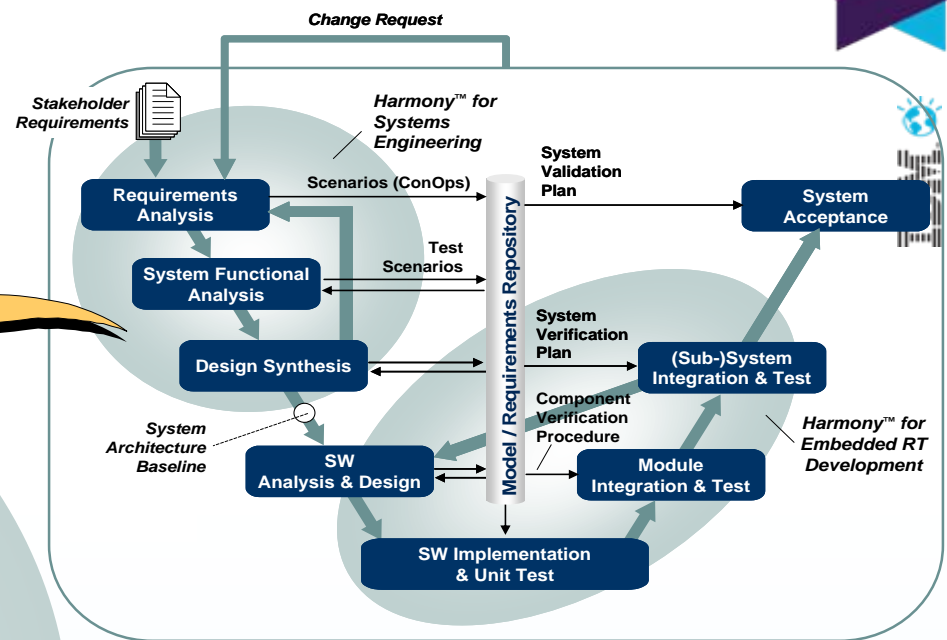
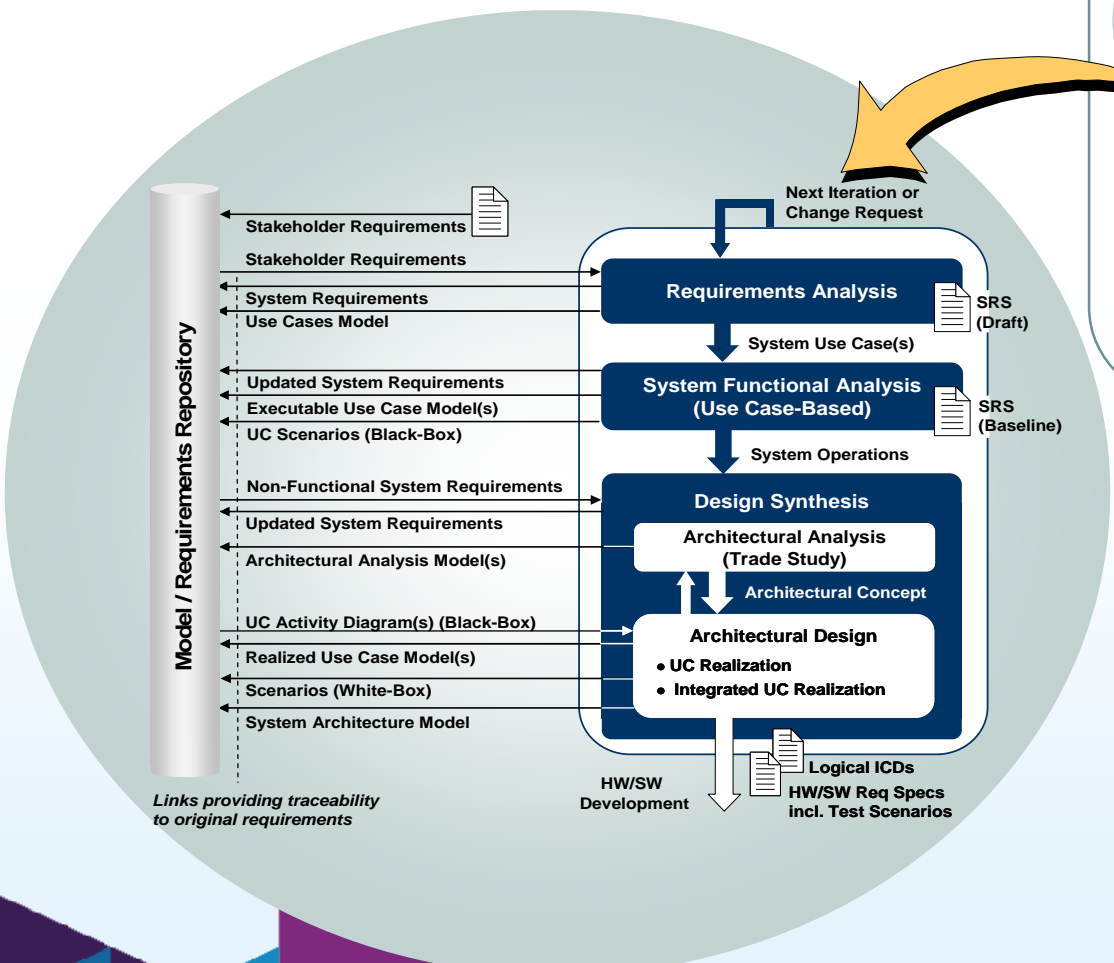
# Why MbSE?

## Experiences summary

- 60% Reduction in time to develop a specification
  - Due to improved clear up rate of issues
  - Improved communication
  - Models act as means of negotiating understanding between customer and supplier
- Improved understanding of interfaces between contractors
  - Interfaces particularly complex as the boundary between the interfacing systems was not on the perceived physical boundary of the systems
- Improvements in HW/SW of 60% less errors
  - Due to better understanding of requirements
  - Tying the model and code together
- Use of a General Systems Architecture (library of reference models) has resulted in a reduction of 75% of the time taken to do initial bid work
  - 1 year to 3 months
  - Leads to reduced bid costs
  - Rapid response to bids







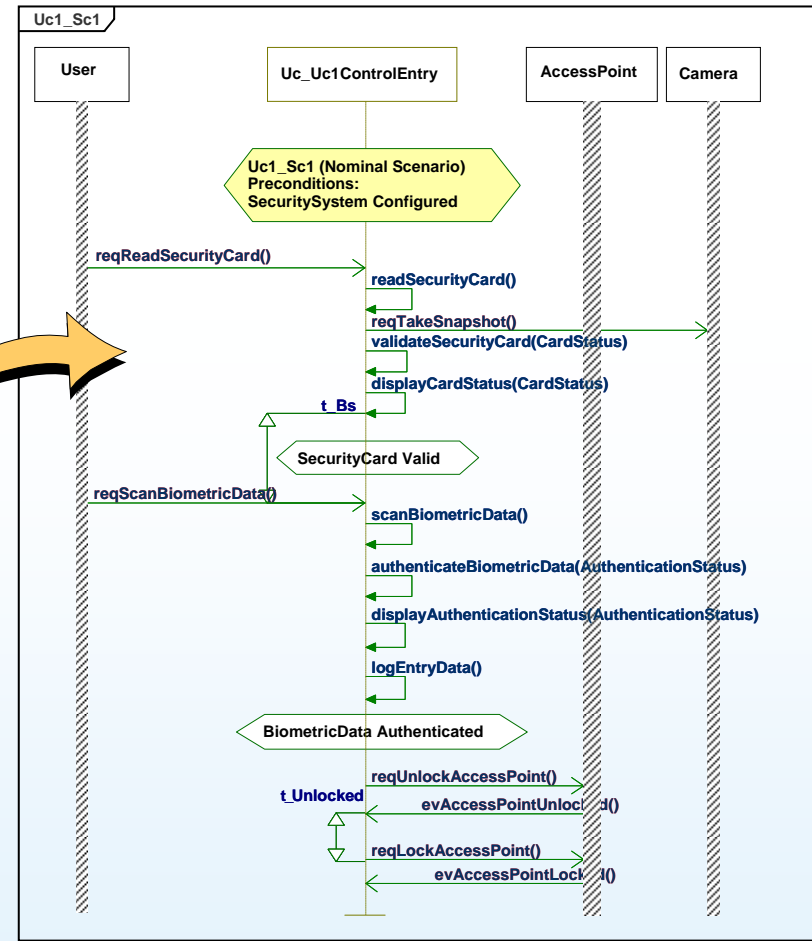
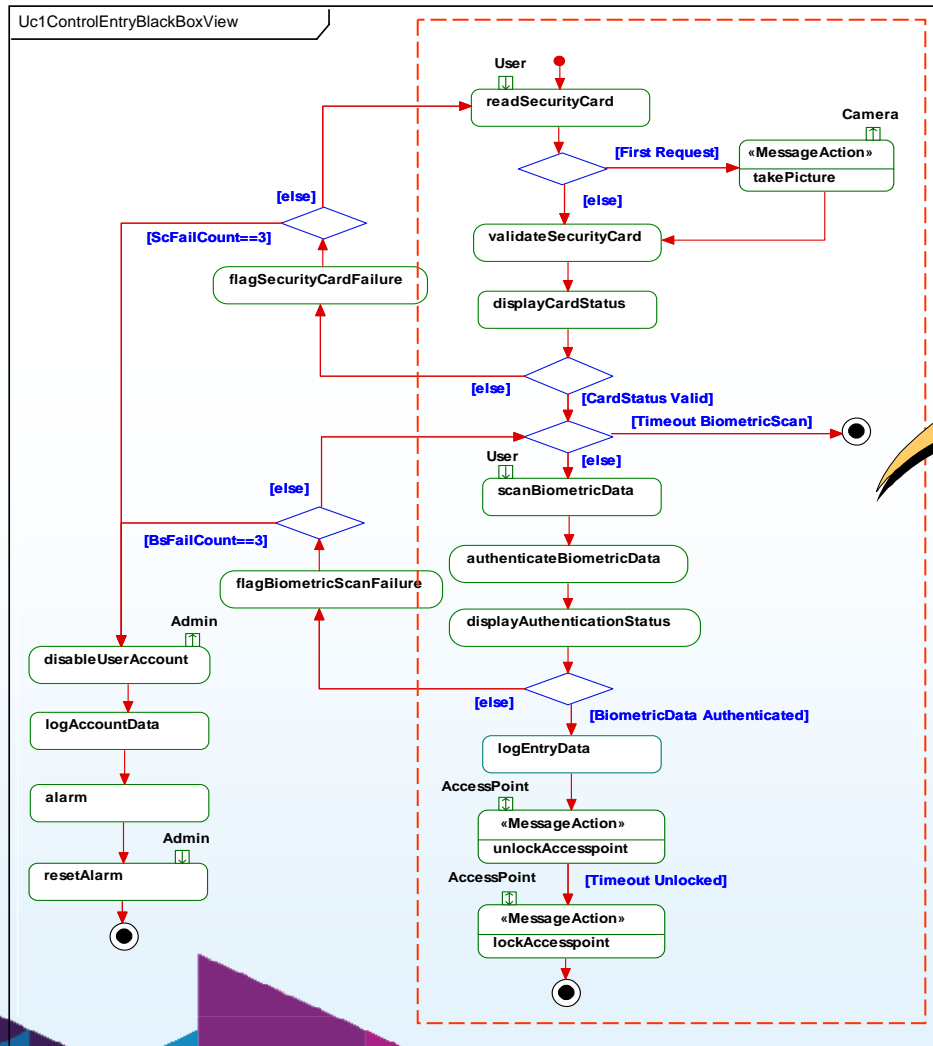
# Rational Harmony™ for Systems Engineering



# Backup

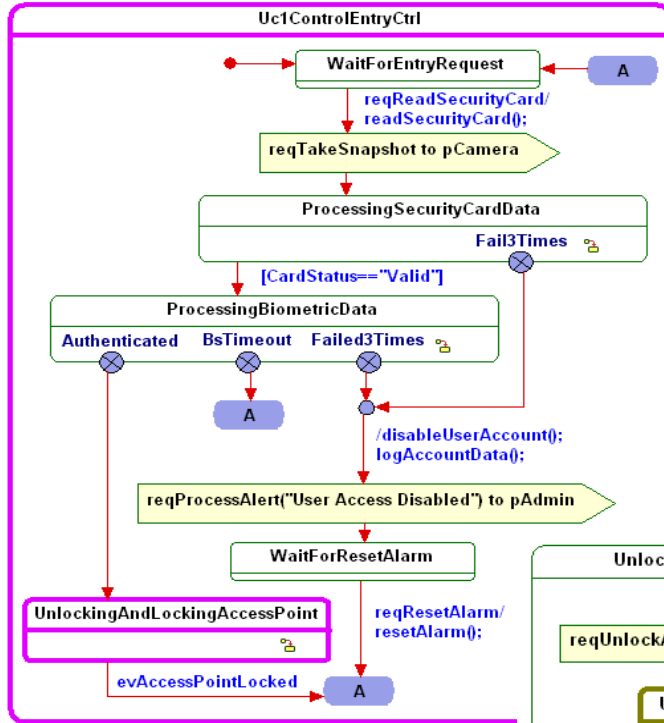
# Case study

## Derivation of a use case scenario from a use case black-box activity diagram

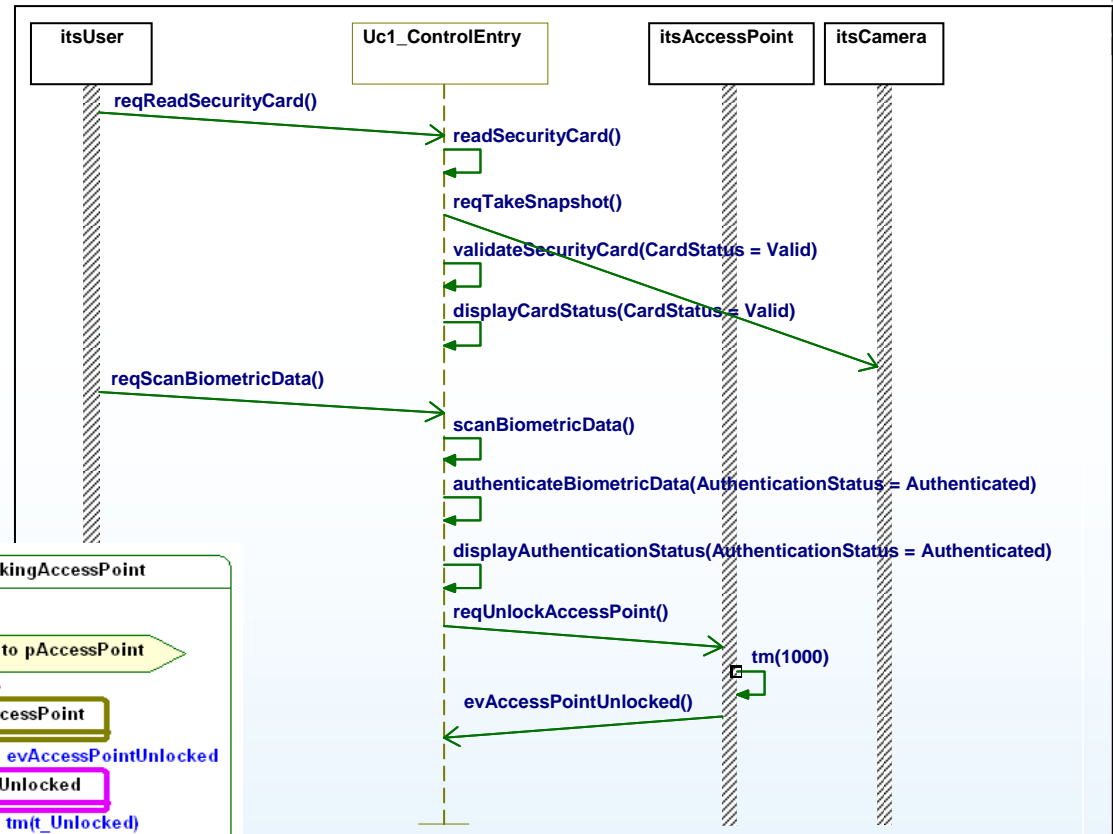
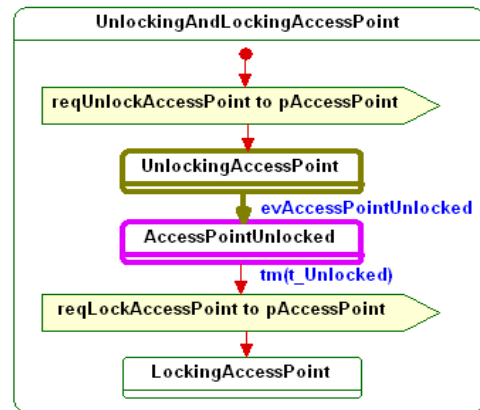


# Case study

## Model verification and validation through model execution (Rational Rhapsody tool)



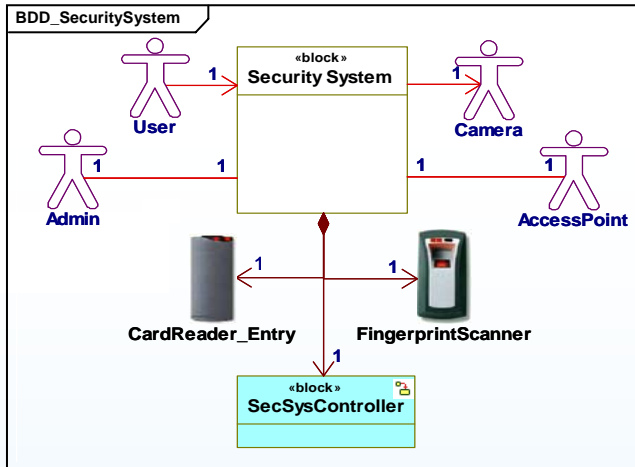
Animated Statechart Diagram  
(Uc1ControlEntry)



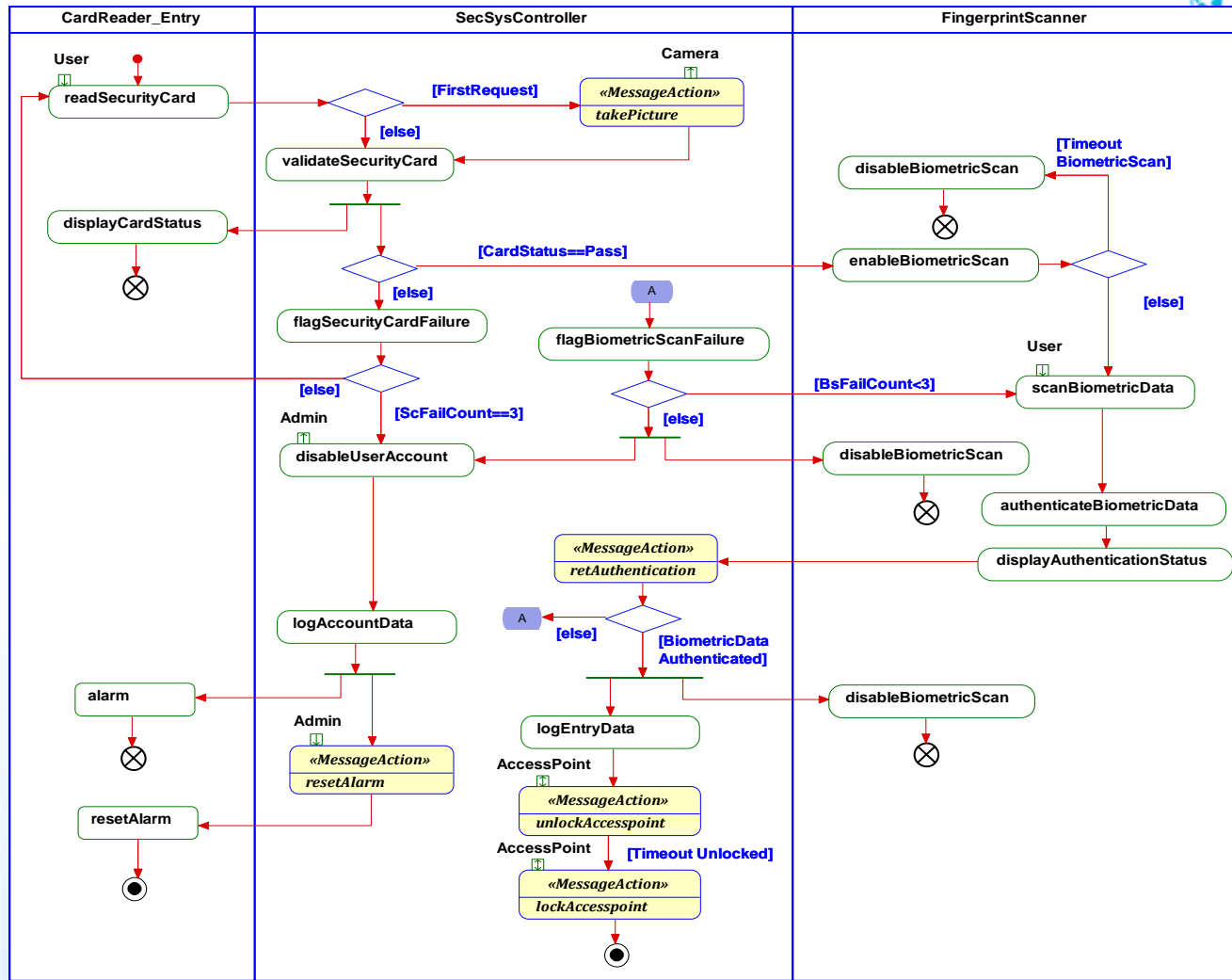
Animated Sequence Diagram (Uc1Sc1)

# Case study – Uc1 (Control Entry) realization

## Allocation of system-level operations to parts (subsystems)



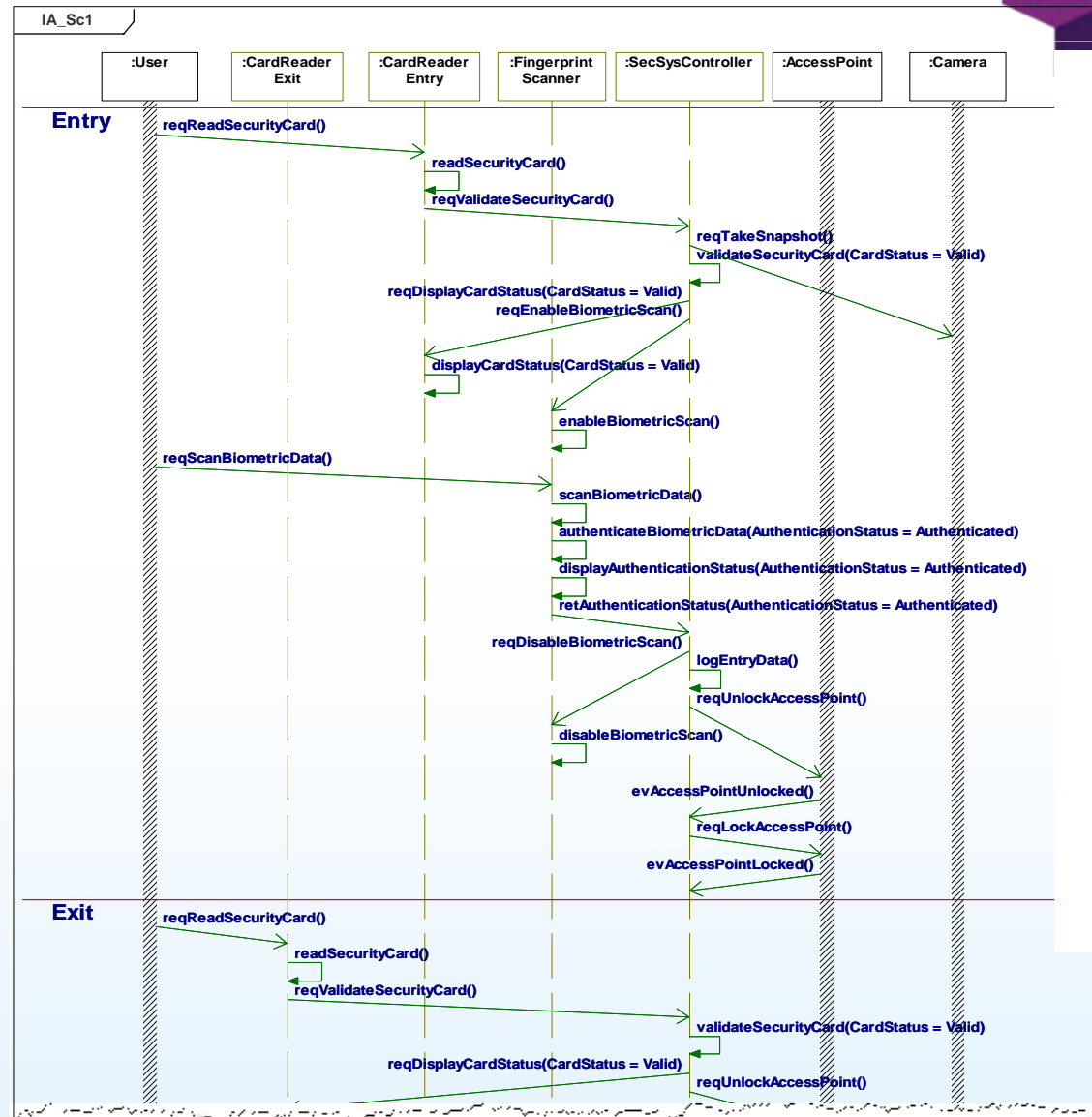
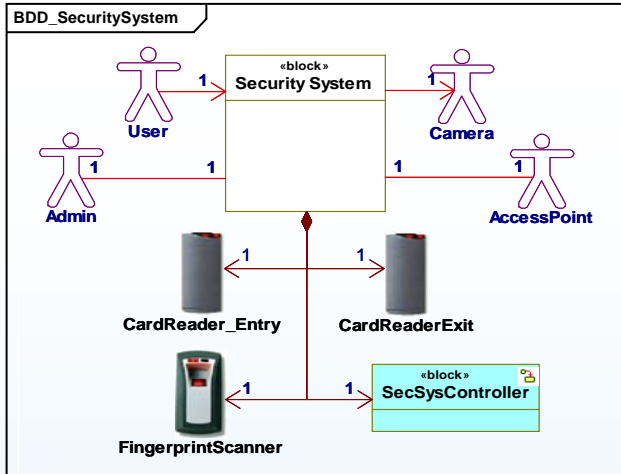
Chosen system architecture captured in a Block Definition Diagram



Uc1ControlEntry White-Box Activity Diagram

# Case study

## Integrated System Architecture verification through model execution



Animated Sequence Diagram (Uc1, Uc2 Nominal Flows)



[www.ibm/software/rational](http://www.ibm/software/rational)

© Copyright IBM Corporation 2011. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. IBM, the IBM logo, Rational, the Rational logo, Telelogic, the Telelogic logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.


Innovate2011

 Software. Everywhere.



# Software. Everywhere.

Innovate2011

 Software. Everywhere.