SD02

Model-Based Systems Engineering with Rhapsody

Graham Bleakley

Principal Consultant, Telelogic

an IBM Company

graham.bleakley@uk.ibm.com

IBM Rational Software Development Conference 2008

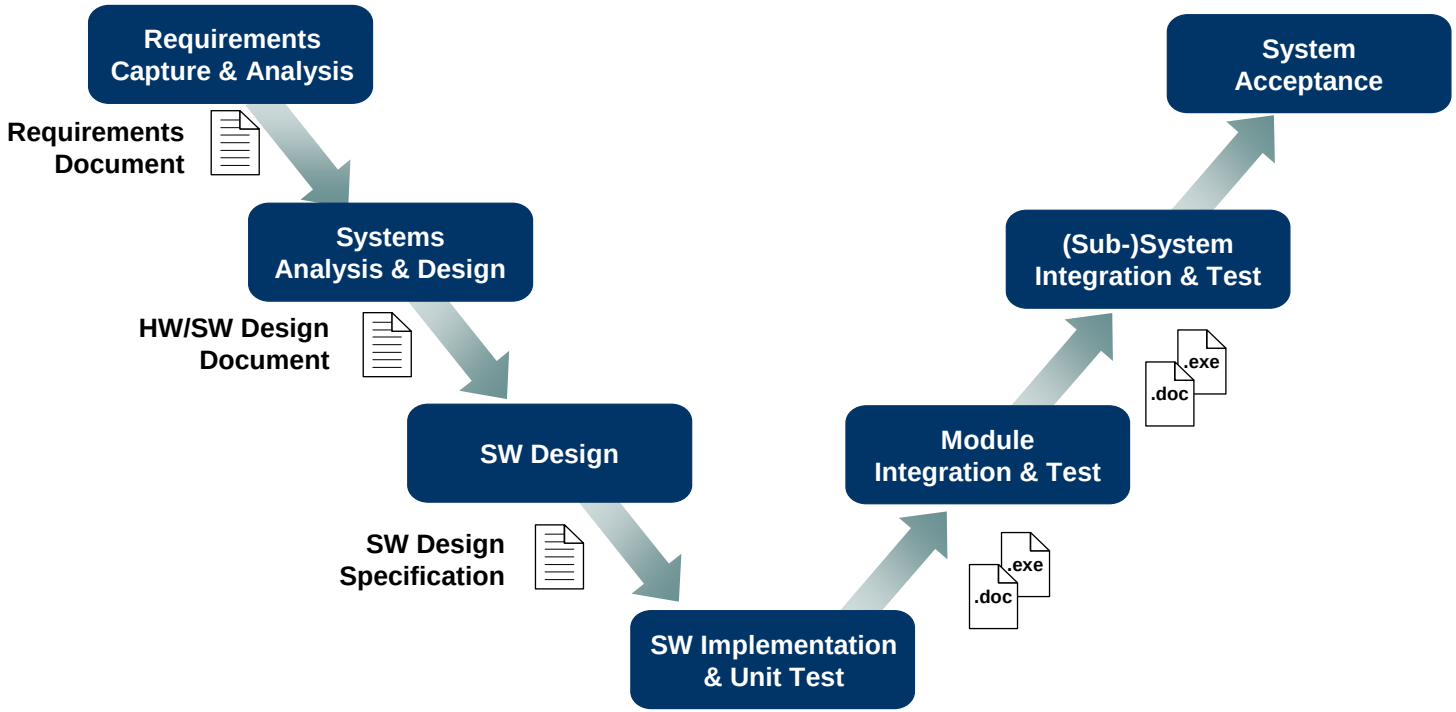
WHERE TEAMS ARE R-HEROES

IBM

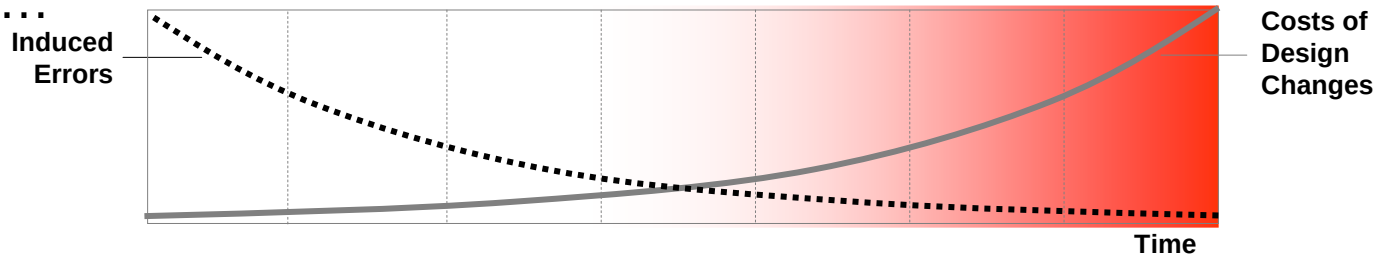
Rational. software

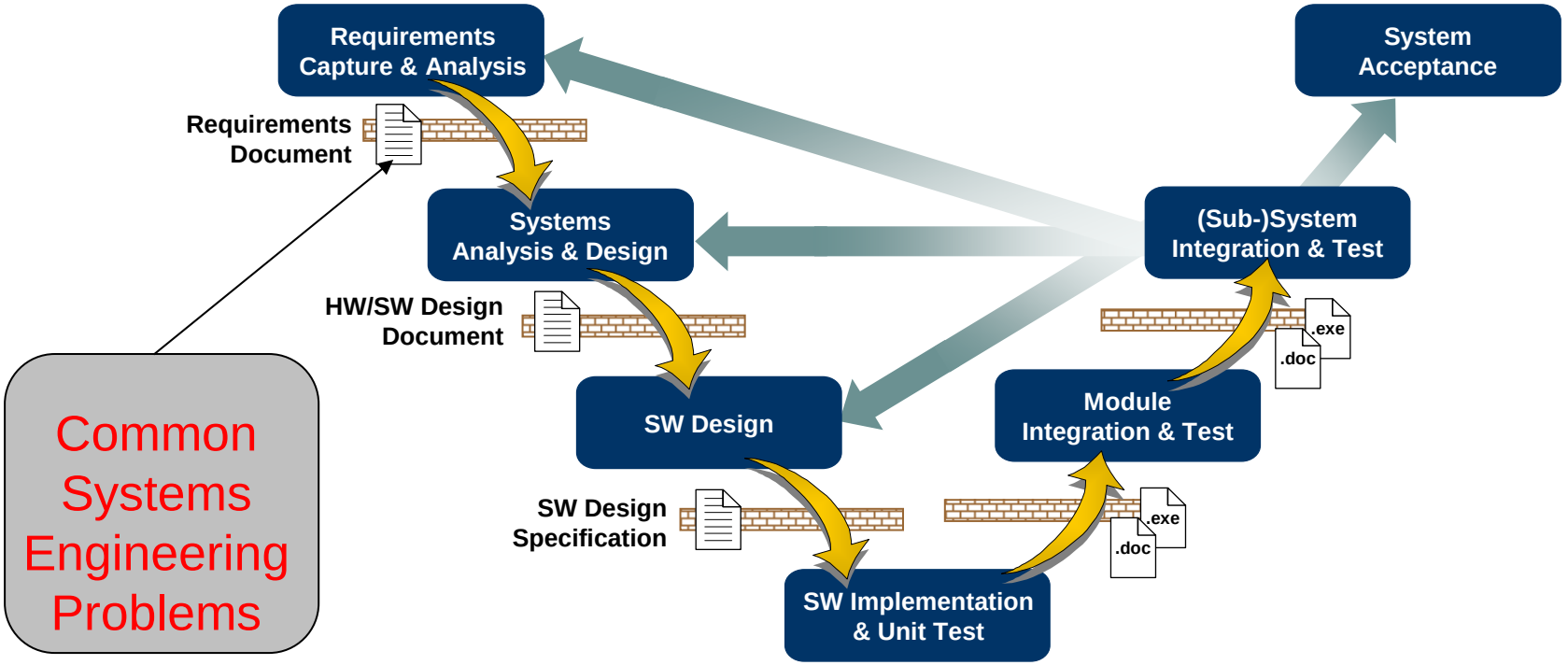
"The Rush to Code"
Why we need Systems Engineering

Document Driven Development of Embedded Systems: How It Should Work …



Document Driven Development of Embedded Systems: How It Actually Works …





The Benefits of Model Based Systems Engineering

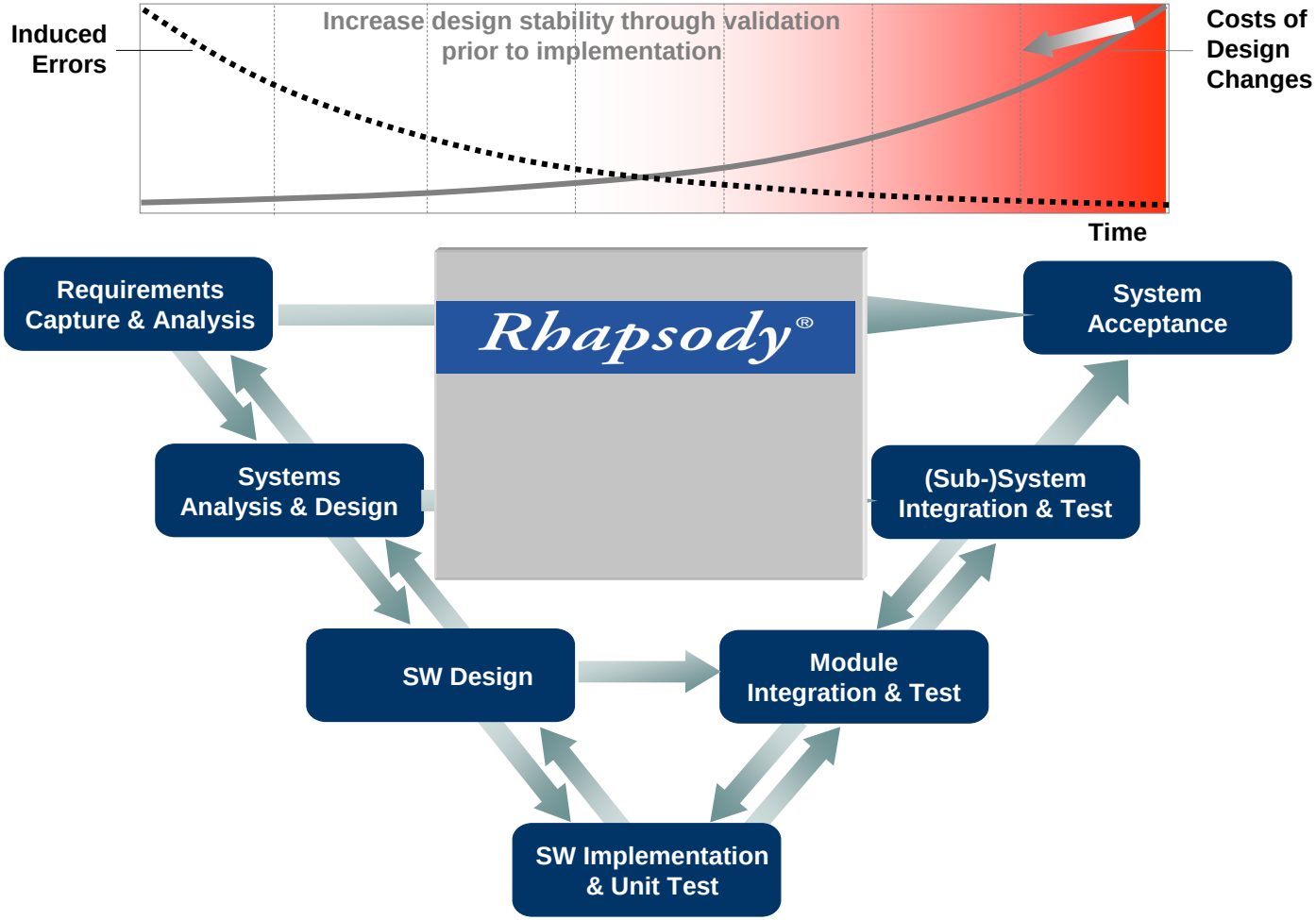
Simulation and execution of models allow Errors to be Eliminated Early in the Process, improving quality

Reduces overall costs by putting effort in early in the development lifecycle

Enables reuse of development scenarios to test the implementation

Enables a seamless Systems to Software handoff as you are using the same tool and effectively the same language

Synchonisation of model views enables a concurrent, approach to Systems Development

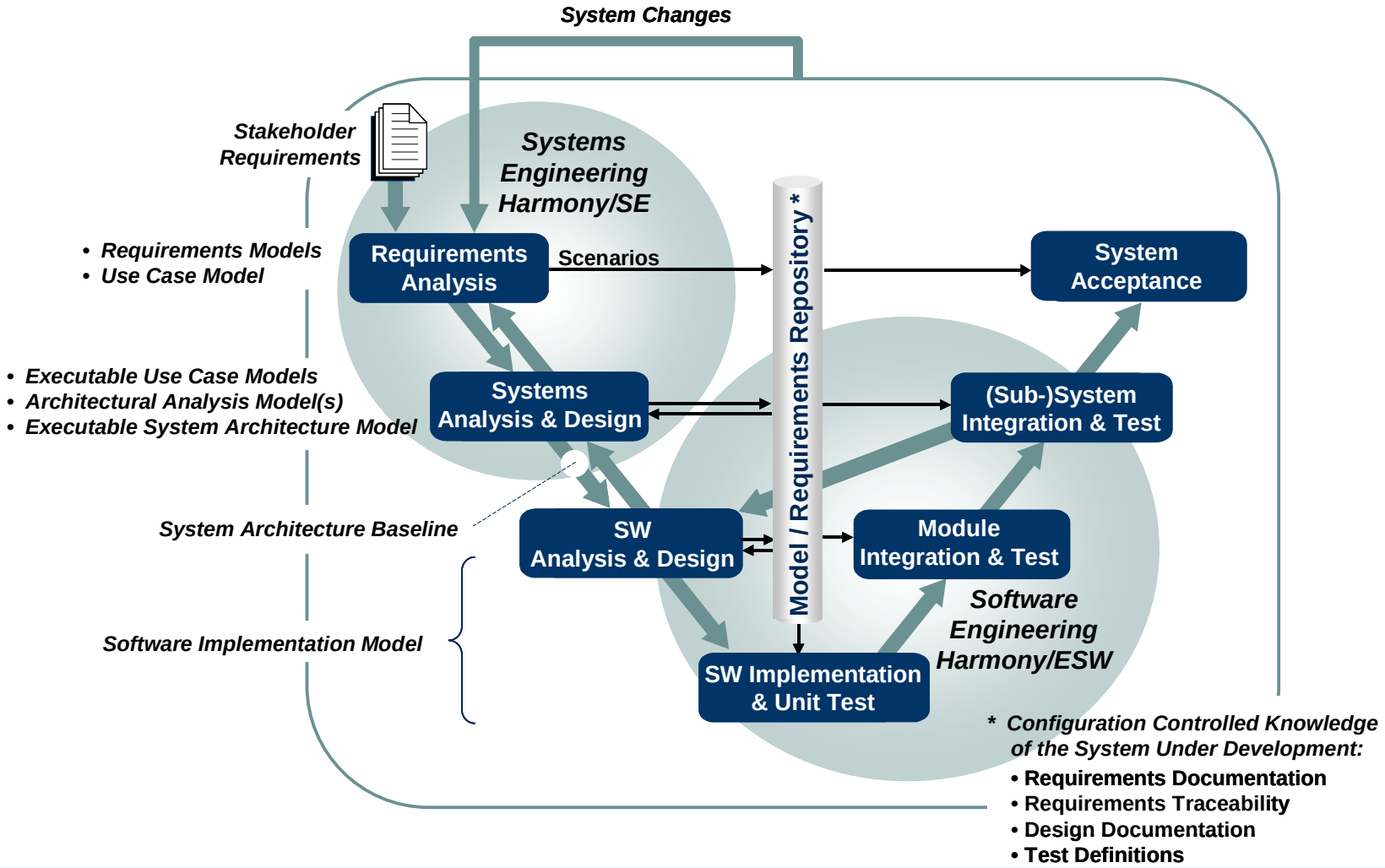


Model-Based Systems Engineering with *Rhapsody*

- **Integrated System / Embedded Software Development Process**
- **Fundamentals of Model-Based Systems Engineering (*Harmony/SE*)**
  - **Essential SysML Artifacts**
  - **Service Request-Driven Modeling Approach**
- **Task Flow and Work Products in *Harmony/SE***
- **Handoff to Hardware / Software Development**

Integrated System / Software Development Process

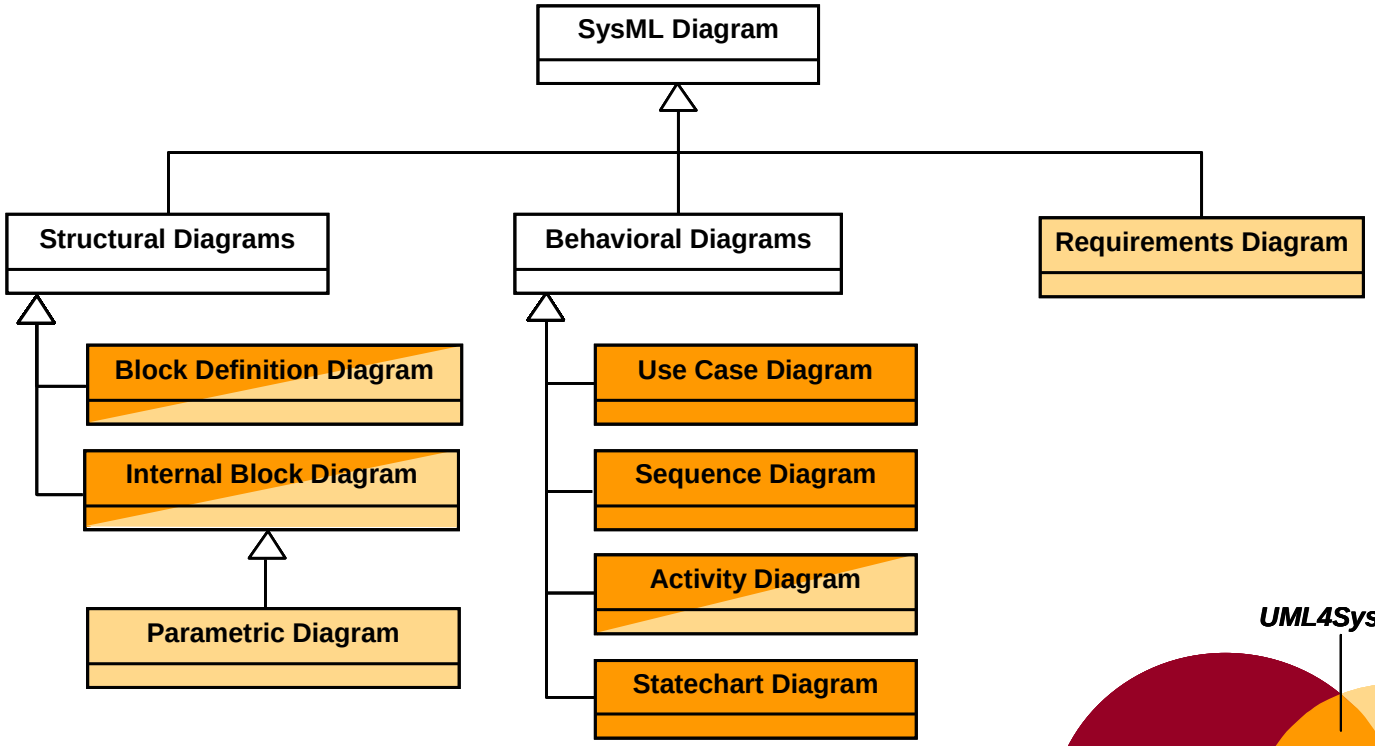
Model-Driven Development of Embedded Systems

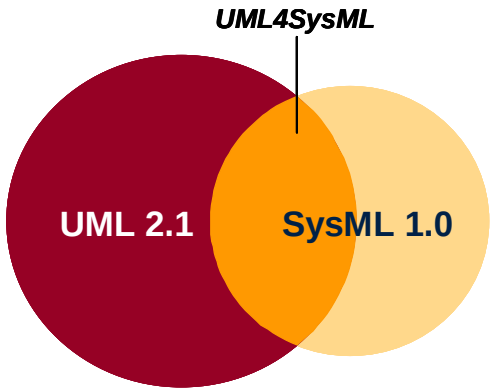


Model-Based Systems Engineering with *Rhapsody*

- Integrated System / Embedded Software Development Process
- **Fundamentals of Model-Based Systems Engineering (*Harmony/SE*)**
  - **Essential SysML Artifacts**
  - Service Request-Driven Modeling Approach
- Task Flow and Work Products in *Harmony/SE*
- Handoff to Hardware / Software Development

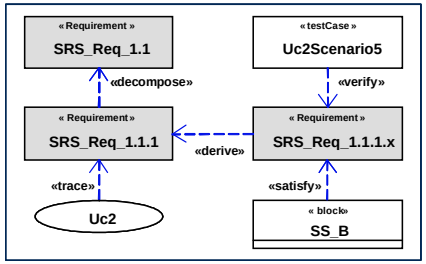
Essential SysML Artifacts for Model-Based Systems Engineering





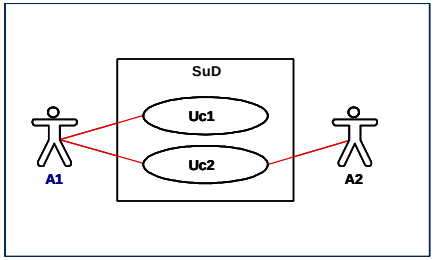
SysML Artifacts in Harmony/SE

Requirements Diagram



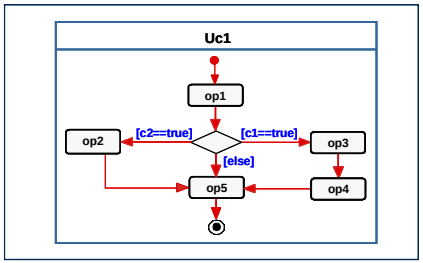
- *Taxonomy of Requirements*
- *Relationship between Model Elements and Requirements*

Use Case Diagram



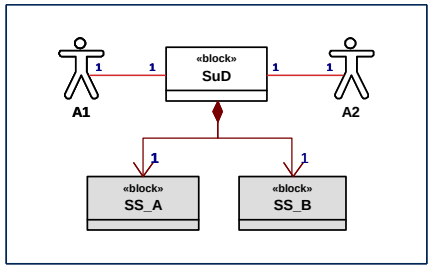
- *Definition of System Scope*
- *Grouping of Requirements into Use Cases*

Activity Diagram

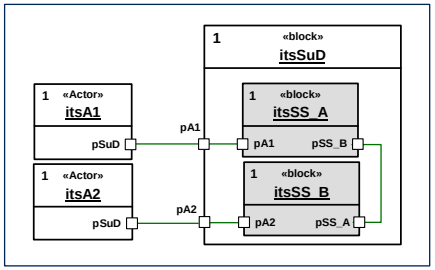


Functional Flow in Use Case / Block(s)

Structure Diagrams

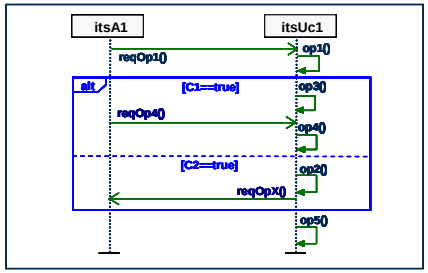


Block Definition Diagram: Structural Elements (Blocks) and their Relationship



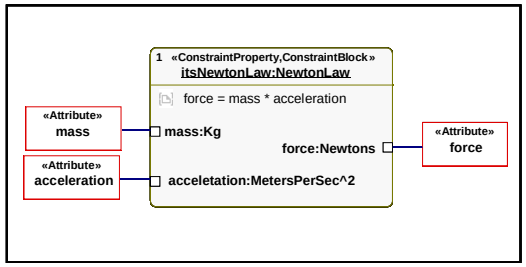
Internal Block Diagram: Realization of System Structure

Sequence Diagram



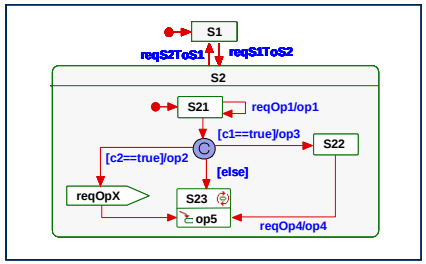
Message Interactions between Nodes

Parametric Diagram



Parametric Relationship between System Properties.

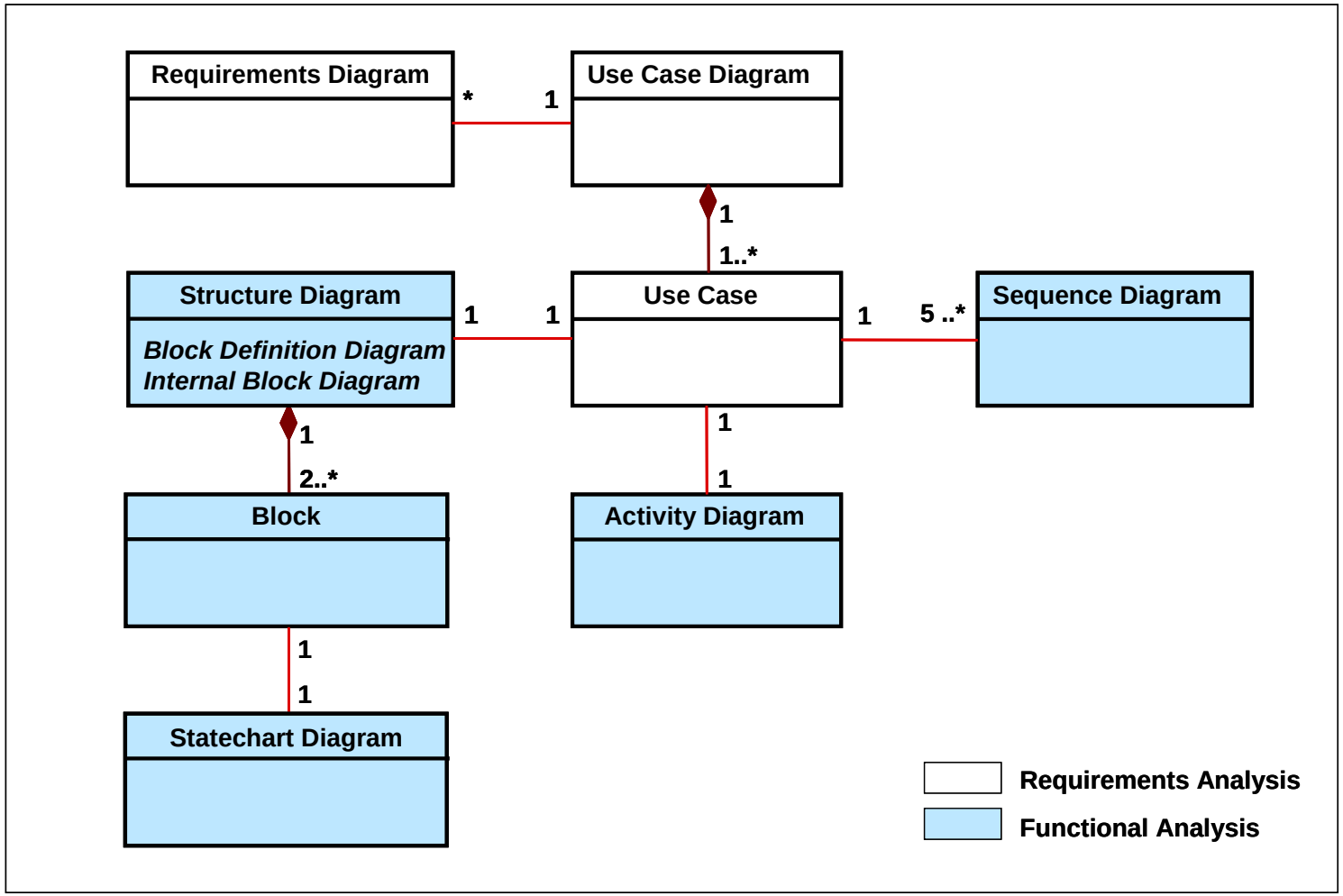
Statechart Diagram



State-based Behavior of Block

Model-Based Systems Engineering (*Harmony/SE*)

Artifact Relationships at the Requirements Analysis / Functional Analysis Level



Model-Based Systems Engineering with *Rhapsody*

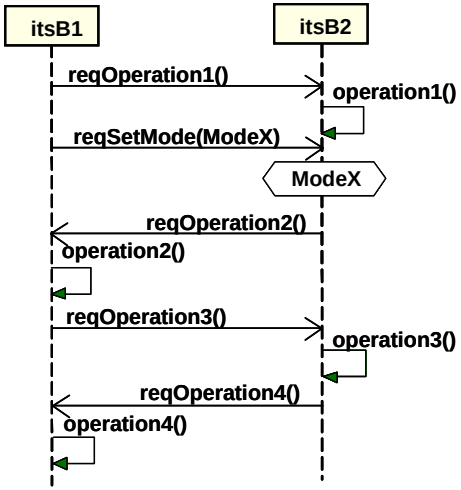
- Integrated System / Embedded Software Development Process
- **Fundamentals of Model-Based Systems Engineering (*Harmony/SE*)**
  - Essential SysML Artifacts
  - **Service Request-Driven Modeling Approach**
- Task Flow and Work Products in *Harmony/SE*
- Handoff to Hardware / Software Development

Service Request-Driven Modeling Approach

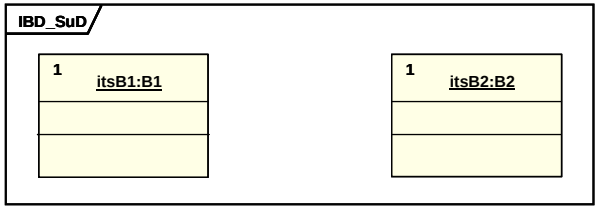
- Communication nodes described by means of *SysML* structure diagrams *(BDD, IBD)* using *Blocks* as basic structural elements.

- Communication described through *Operational Contracts* (*OpCon*), i.e.
  - asynchronous *service requests* via *SysML Service Ports* followed by
  - provided services at the receiving part (state/mode changes or operations)

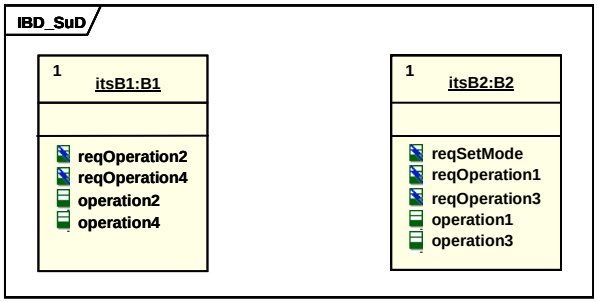
Step 2: Description of Internodal Communication



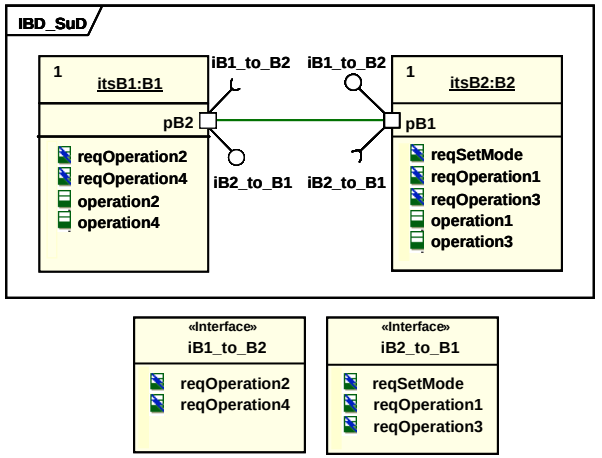
Step 1: Definition of Communication Nodes



Step 3: Allocation of Service Requests and Operations



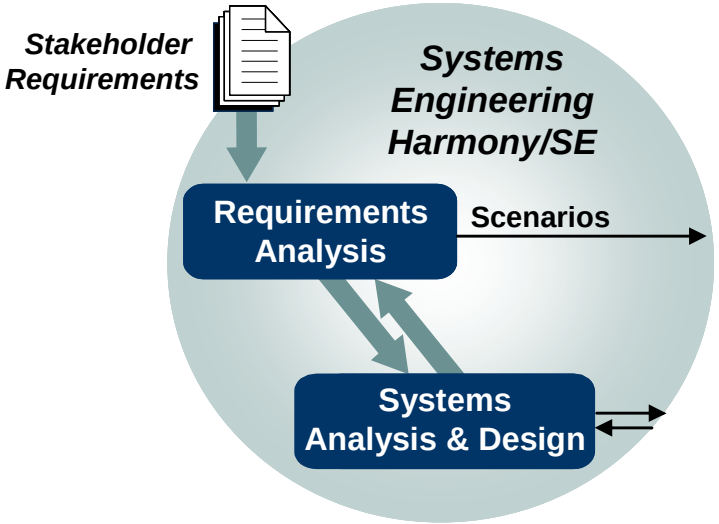
Step 4: Definition of Ports and Interfaces



Model-Based Systems Engineering with *Rhapsody*

- Integrated System / Embedded Software Development Process
- Fundamentals of Model-Based Systems Engineering (*Harmony/SE*)
  - Essential SysML Artifacts
  - Service Request-Driven Modeling Approach
- **Task Flow and Work Products in *Harmony/SE***
- Handoff to Hardware / Software Development

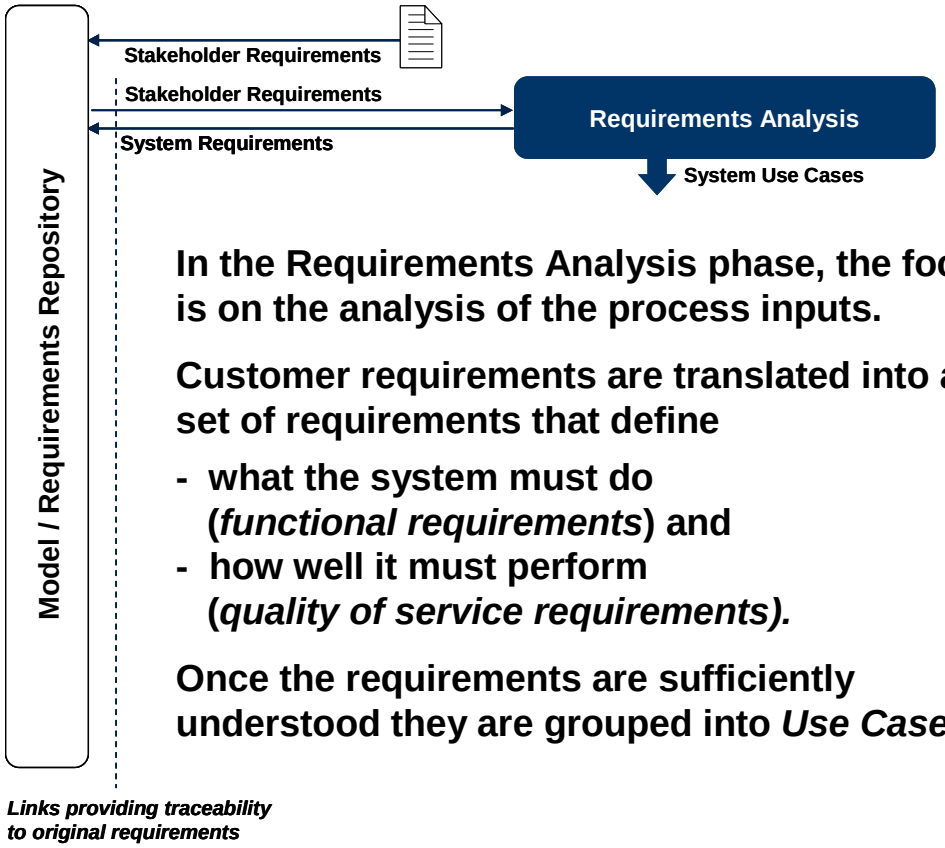
Key Objectives of the Model-Based Systems Engineering Process *Harmony/SE*



- **Identify / derive required system functionality**
- **Identify associated system modes and states**
- **Allocate system functionality / modes to a physical architecture**

Model-Based Systems Engineering (*Harmony/SE*)

Requirements Analysis

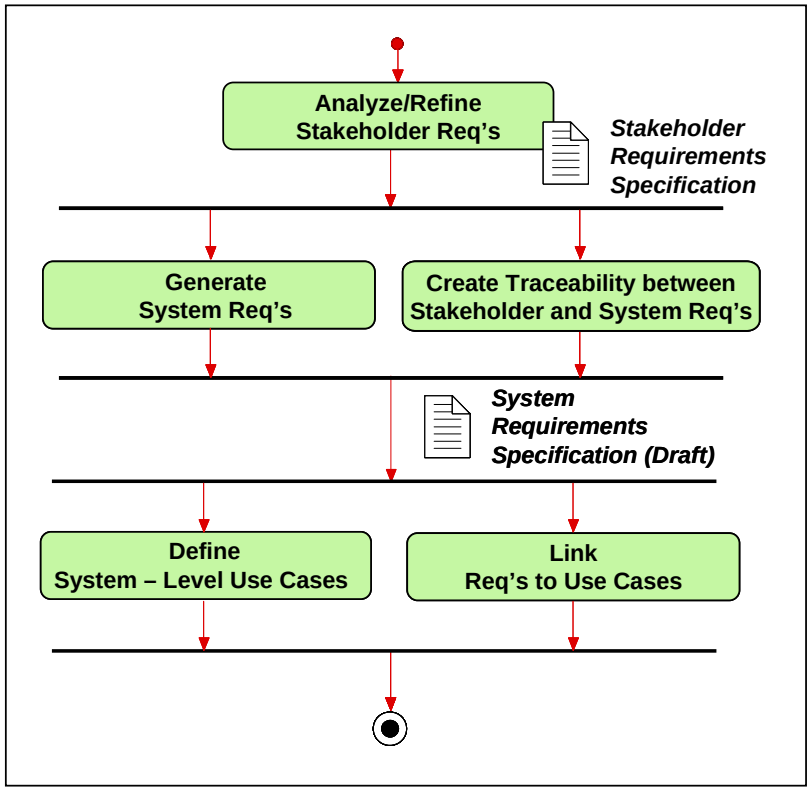


In the Requirements Analysis phase, the focus is on the analysis of the process inputs.

Customer requirements are translated into a set of requirements that define

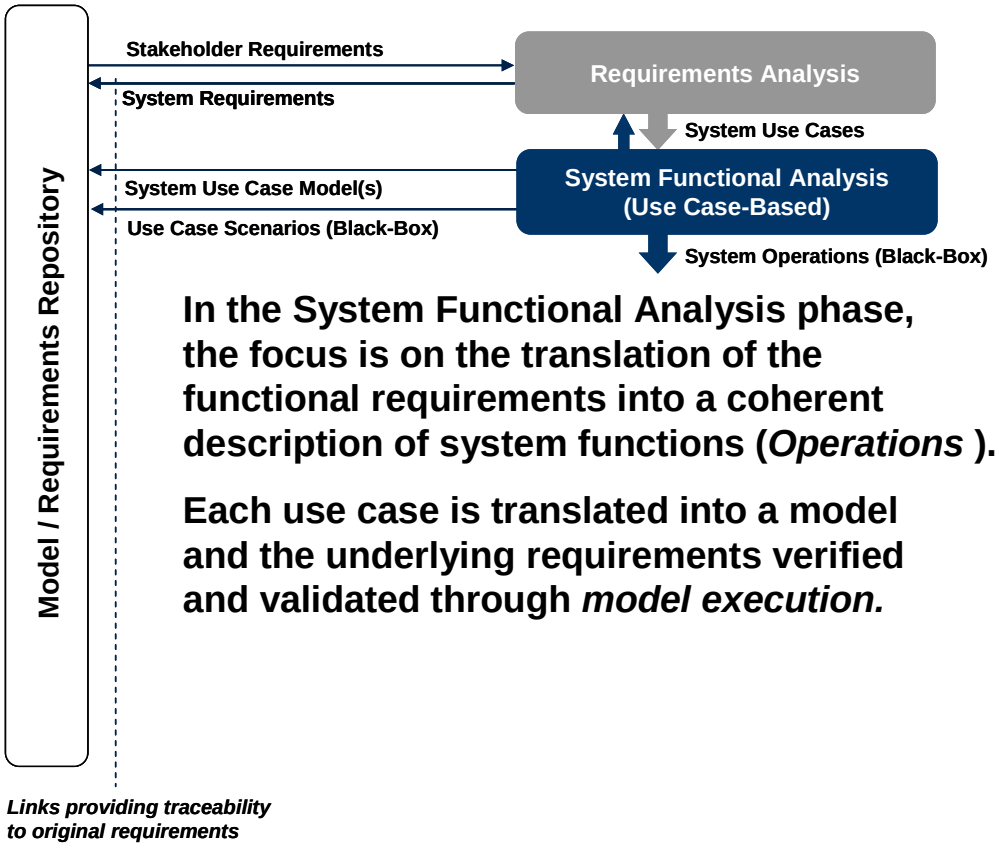
- **what the system must do (*functional requirements*) and**
- **how well it must perform (*quality of service requirements).***

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



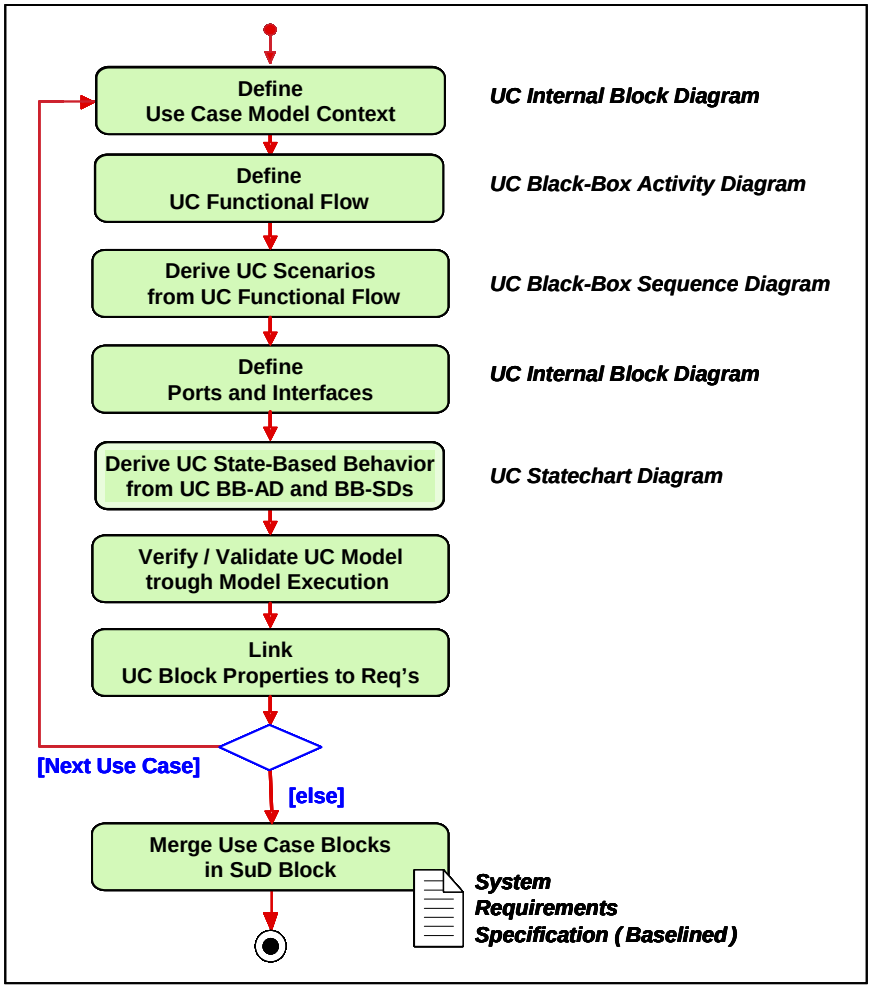
Model-Based Systems Engineering (*Harmony/SE*)

System Functional Analysis



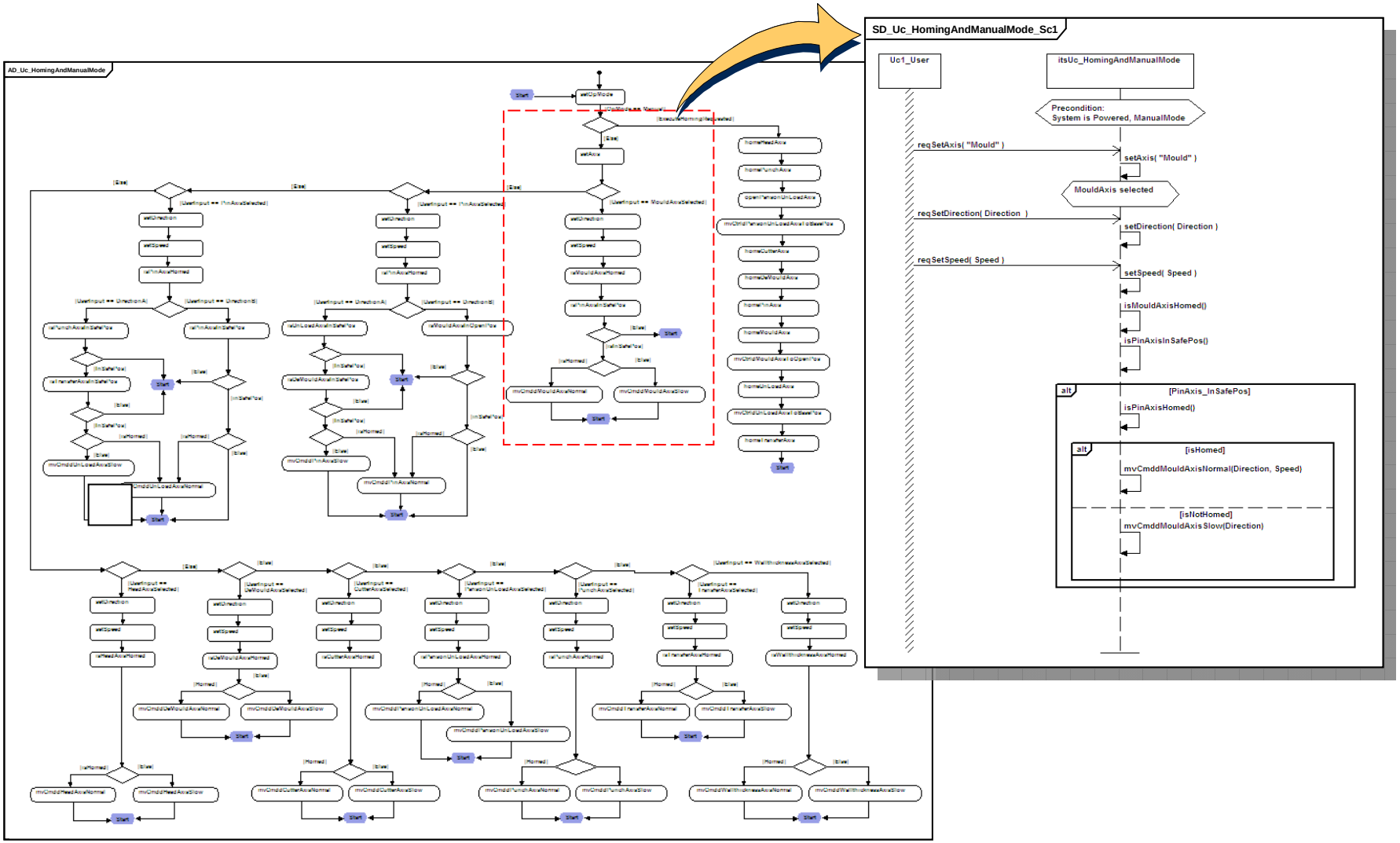
In the System Functional Analysis phase, the focus is on the translation of the functional requirements into a coherent description of system functions (*Operations*).

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



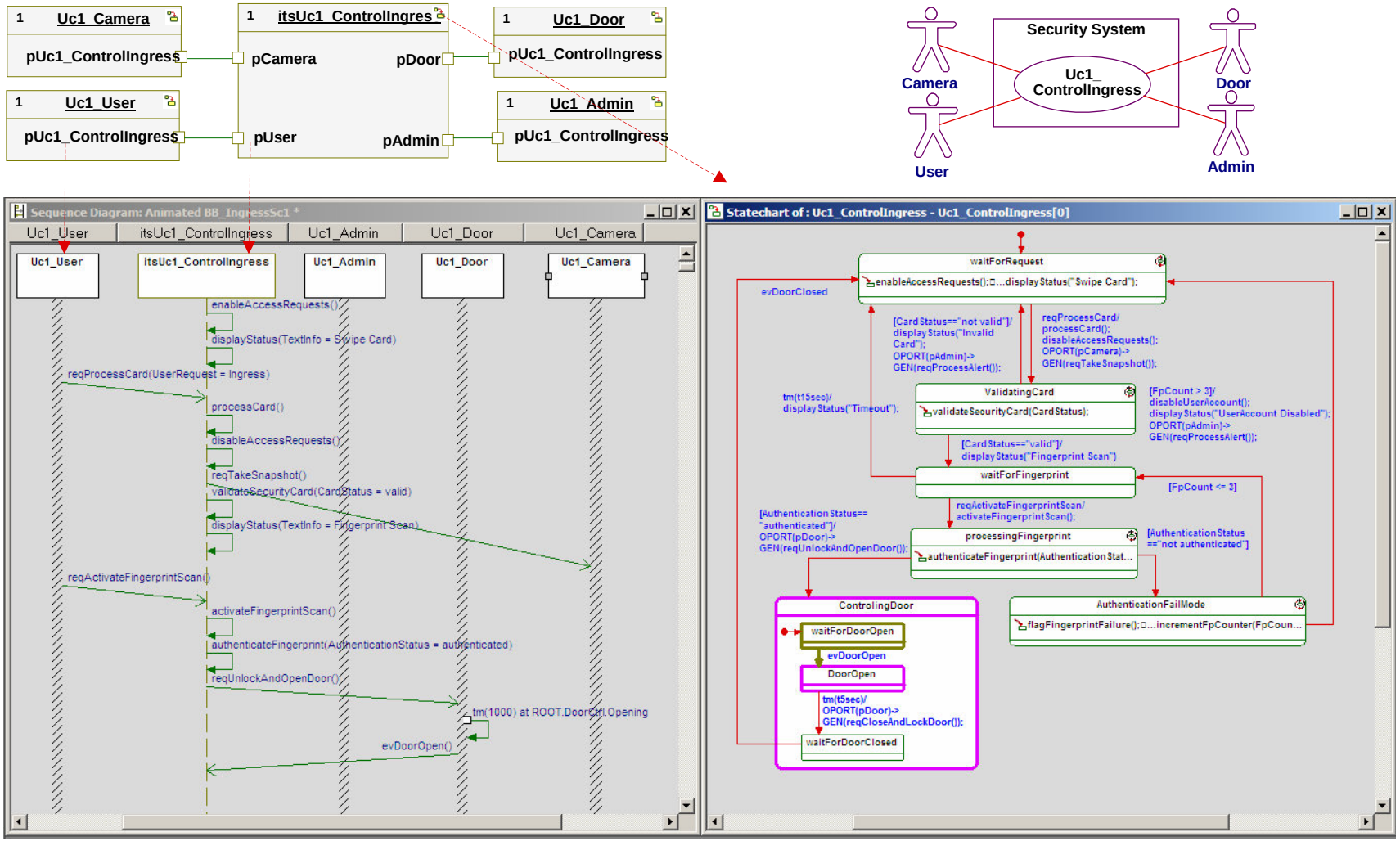
System Functional Analysis

Derivation of Use Case Scenarios from a UC Black-Box Activity Diagram



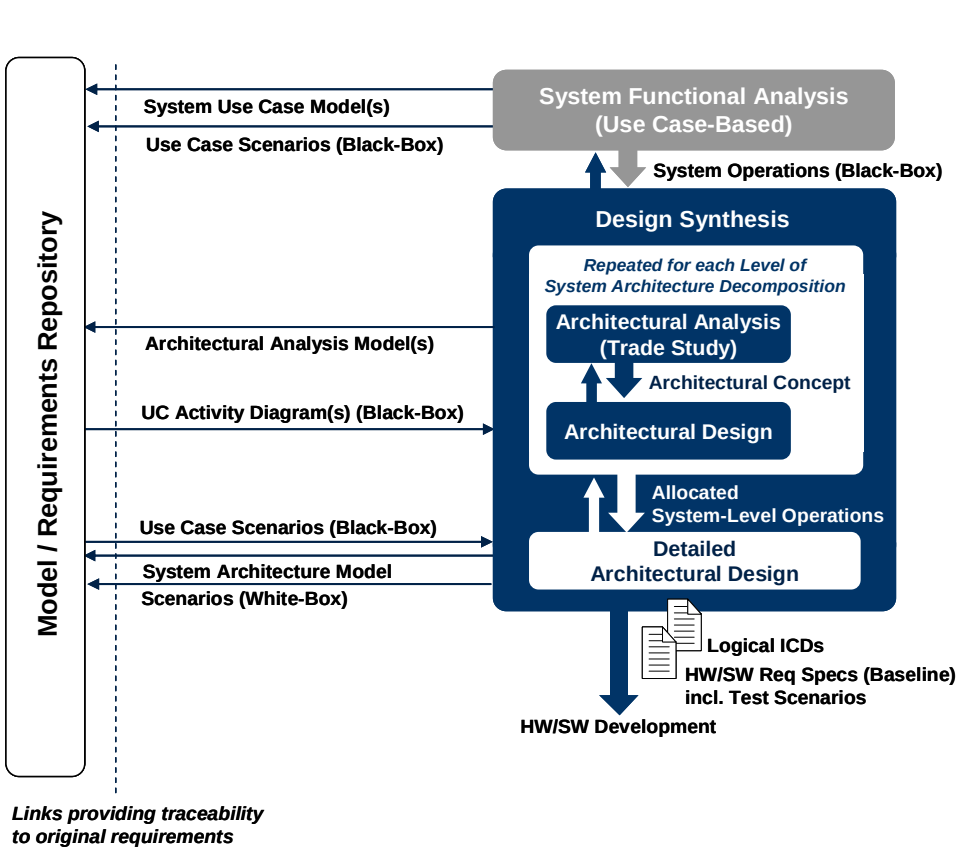
Use Case Model Verification and Validation Through Model Execution

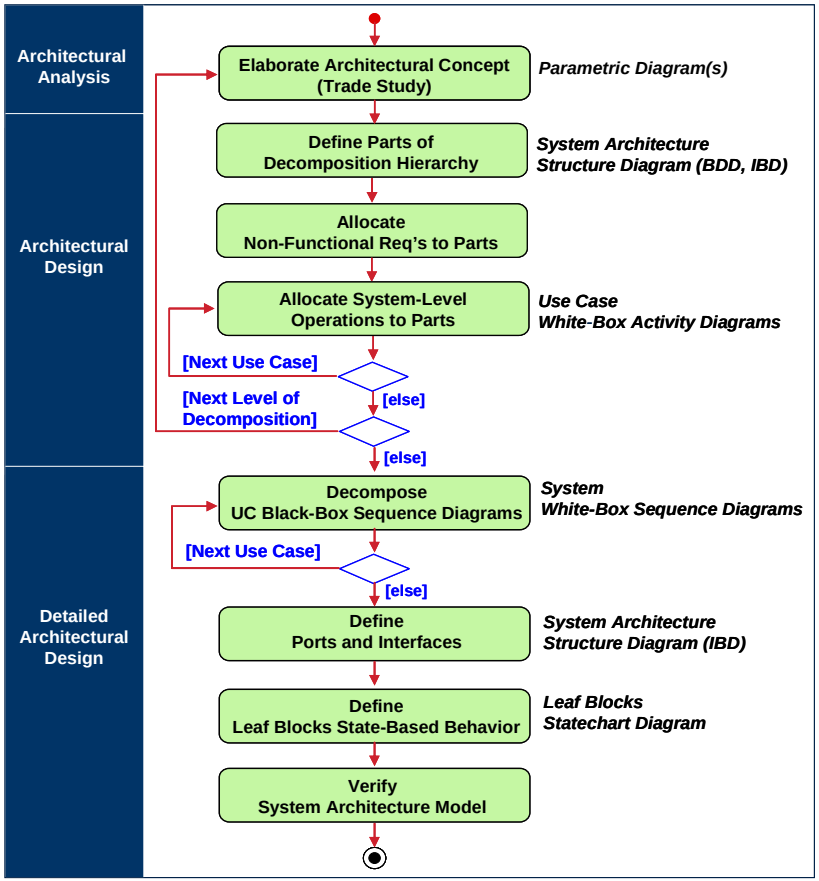
Example: Security System Use Case *ControlIngress* (ref. *Rhapsody® Deskbook*)



Model-Based Systems Engineering (*Harmony/SE*)

Design Synthesis

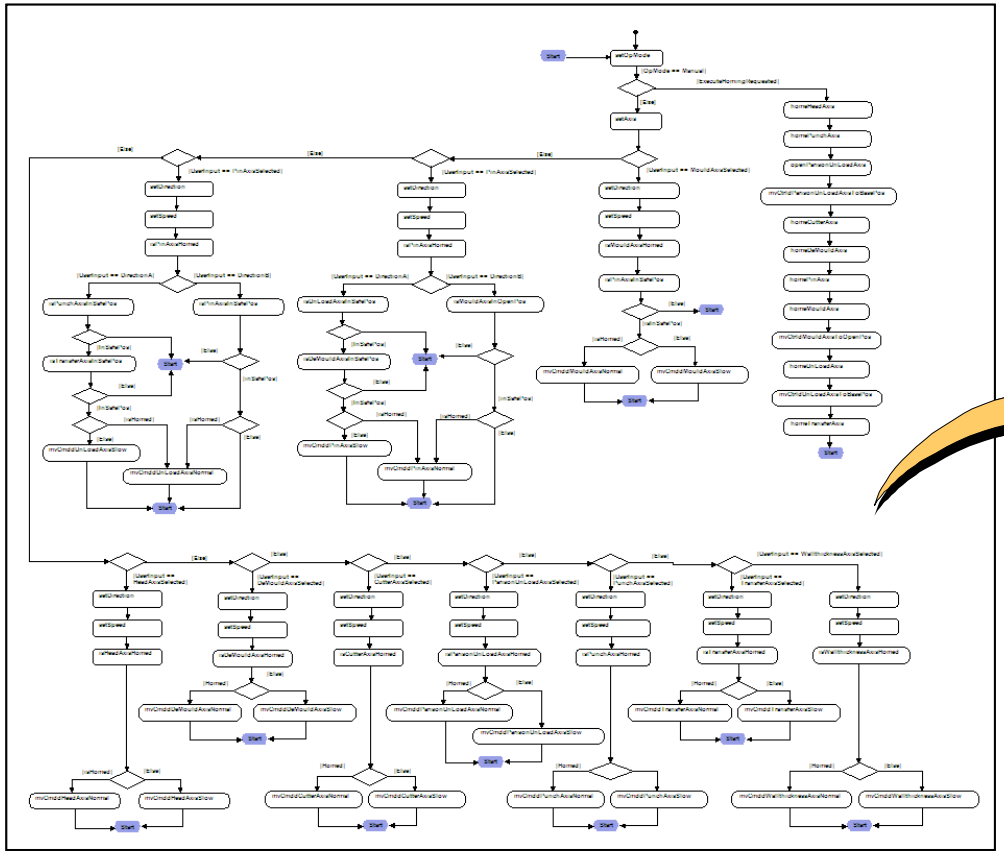




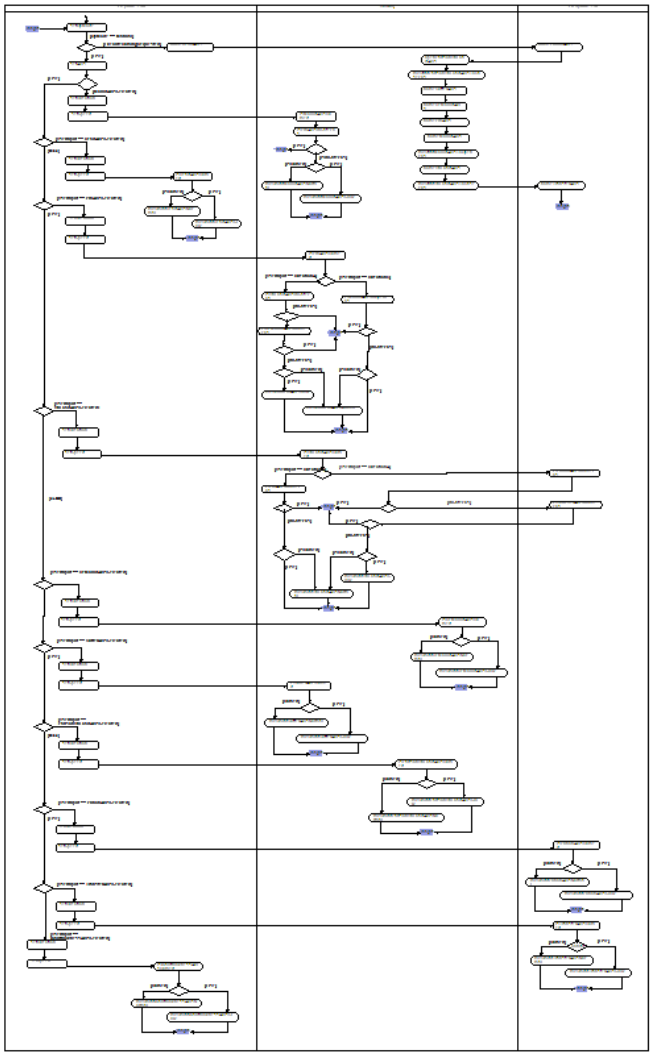
In the Design Synthesis phase, the focus is on the allocation of system-level operations to a system architecture - optionally elaborated through trade studies - and on the definition of ports / interfaces and state-based behavior at the lowest level of the structural decomposition.

Architectural Design

Allocation of Operations to Subsystems (Decomposition Level 1)



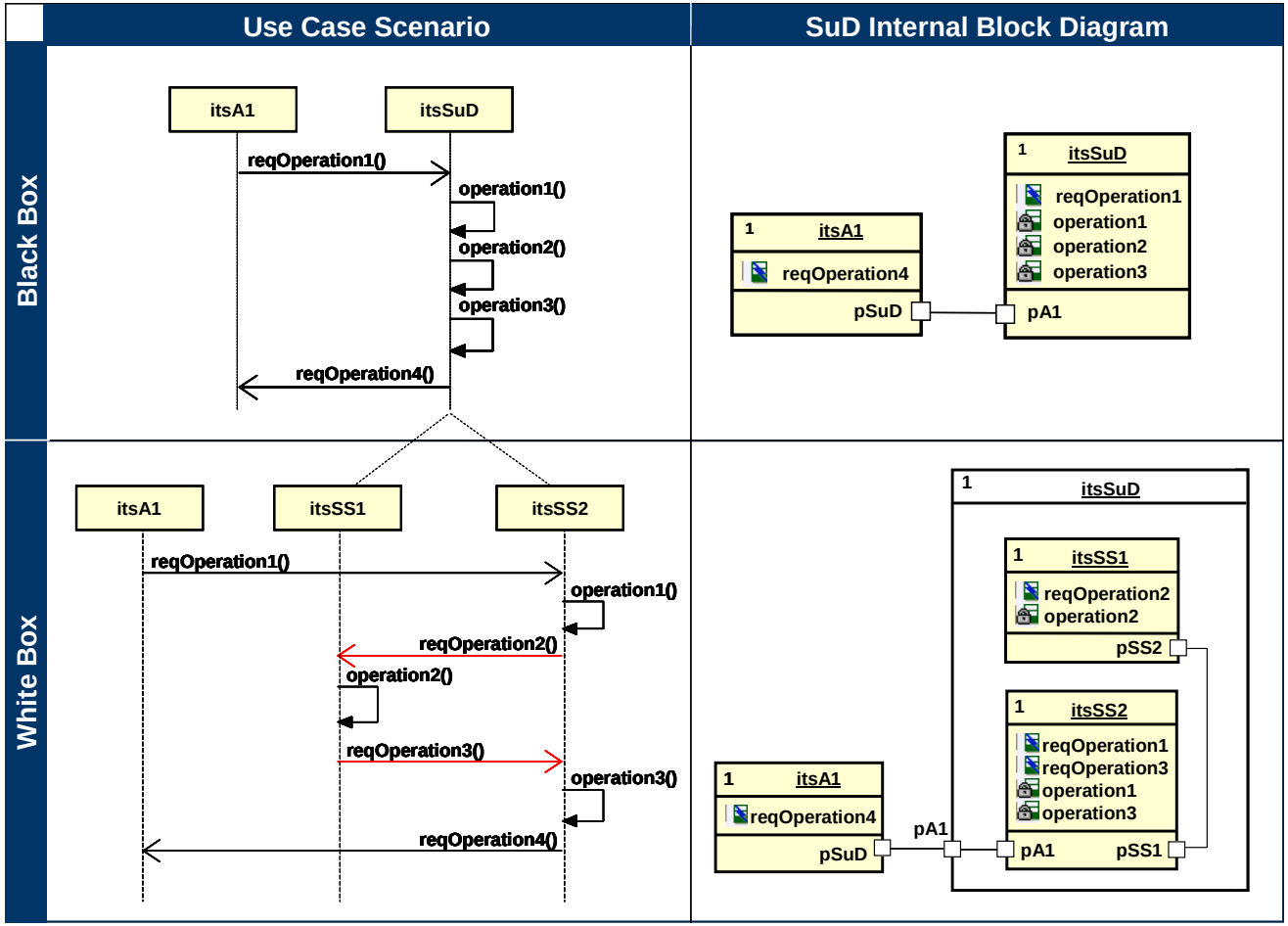
Use Case Black-Box Activity Diagram



Use Case White-Box Activity Diagram

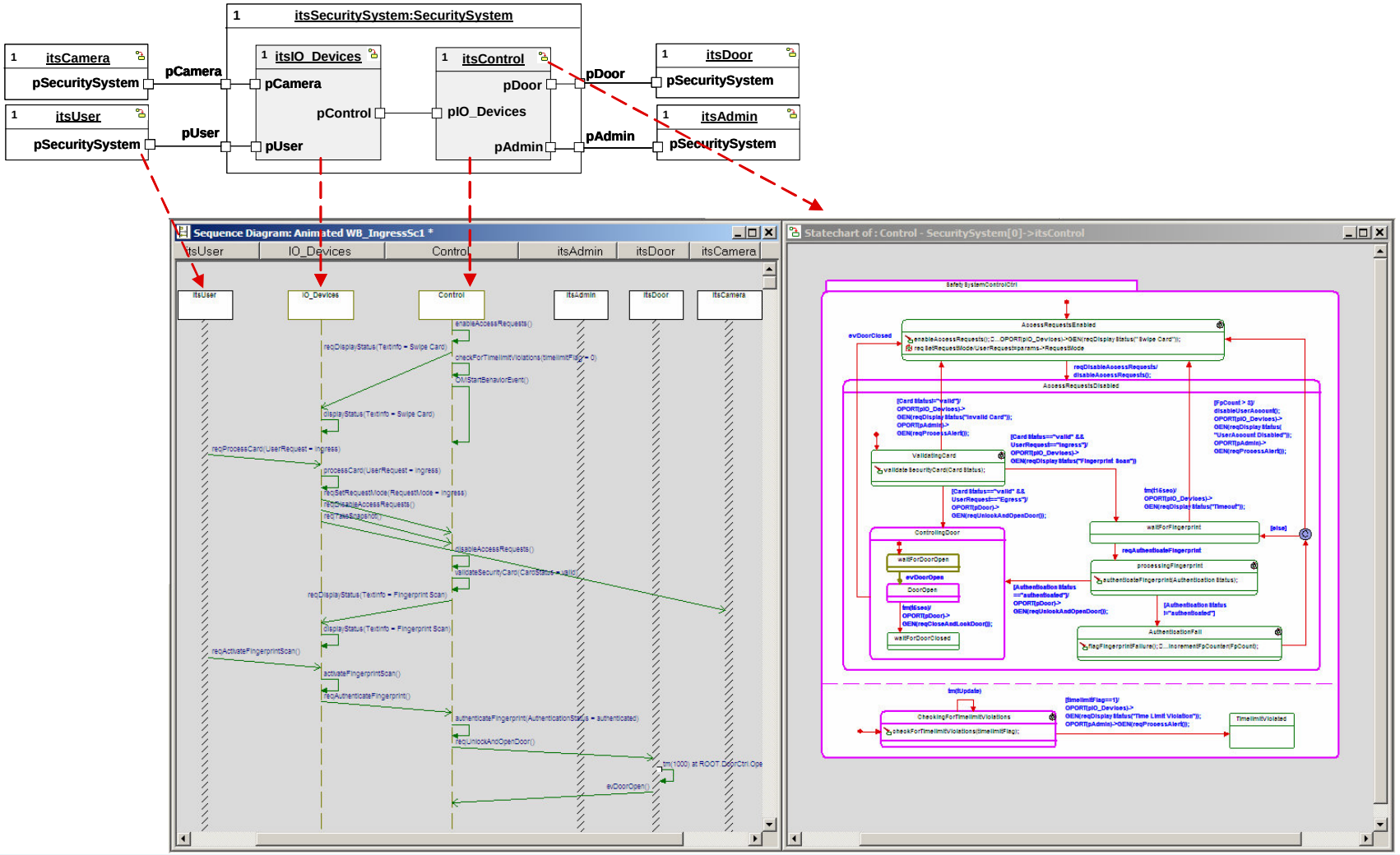
Detailed Architectural Design

Decomposition of UC Black-Box Scenarios and Definition of Ports / Interfaces

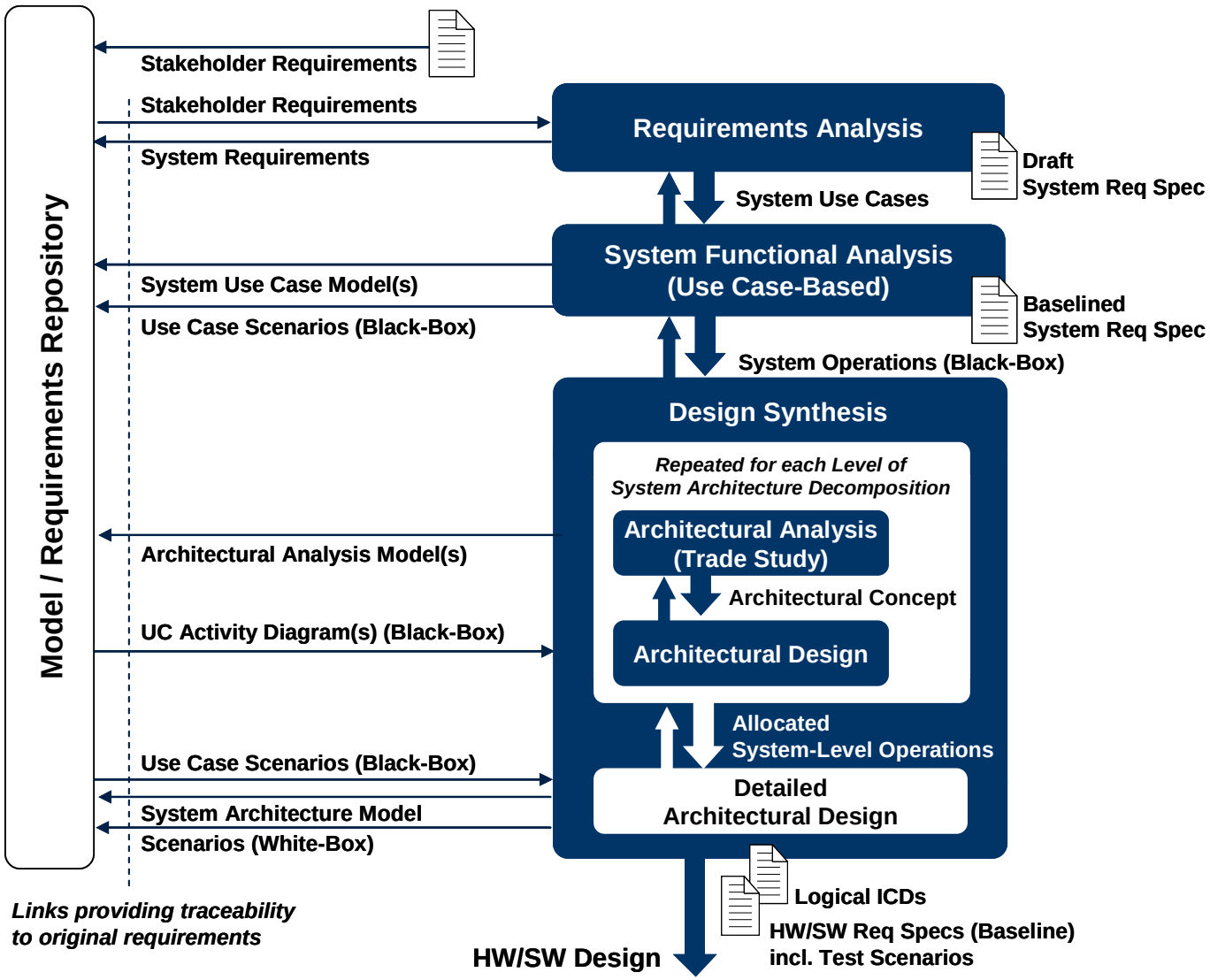


Architecture Model Verification and Validation Through Model Execution

Example: Security System (ref. *Rhapsody® Deskbook*)



Model-Based Systems Engineering (*Harmony/SE*)



Model-Based Systems Engineering with *Rhapsody*

- Integrated System / Embedded Software Development Process
- Fundamentals of Model-Based Systems Engineering (*Harmony/SE*)
  - Essential SysML Artifacts
  - Service Request-Driven Modeling Approach
- Task Flow and Work Products in *Harmony/SE*
- **Handoff to Hardware / Software Development**

Systems Engineering Handoff to Hardware and Software 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 Hardware and Software Development (1)

If the SuD is all software, systems engineering may stop at the functional analysis level. In this case, the hand-off are use case models.

- Baselined executable use case model
- Definition of use case operations including links to the associated system functional requirements,
- Definition of ports and logical interfaces of the use case block,
- Use case black-box activity diagram,
- Sunny day/rainy day use case scenarios,
- Statechart diagram of the use case block, and
- Non-functional requirements that are allocated to the use case.

Systems Engineering Handoff to Hardware and Software Development (2)

If - from the organizational point of view - there is a separation between systems engineering and subsystems engineering, the hand-off may stop at the first level of system architecture decomposition.

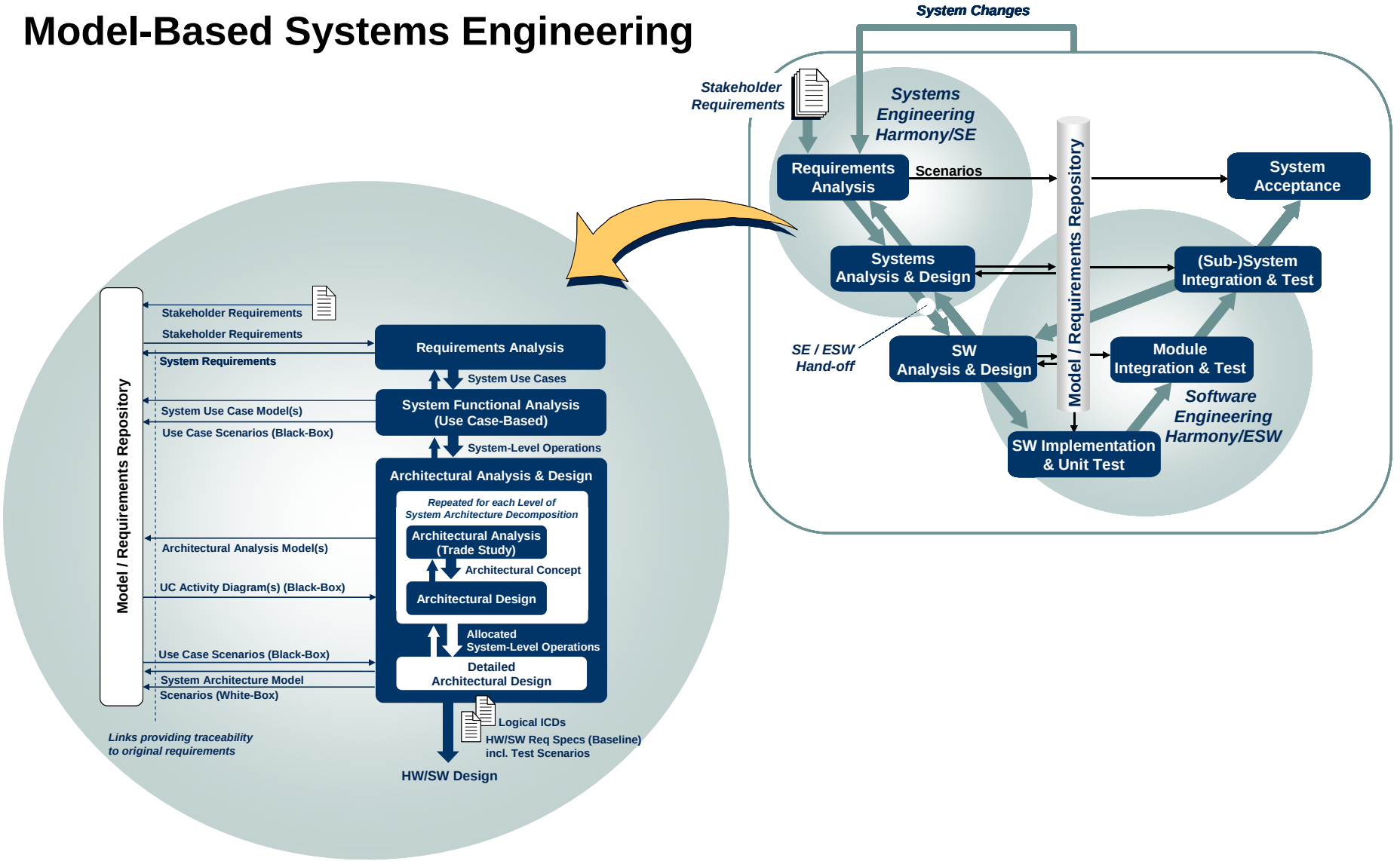
- Baselined executable subsystem model,
- Definition of allocated operations including links to the associated system functional requirements,
- Definition of subsystem ports and logical interfaces (Subsystem Logical Interface Document),
- Subsystem statechart diagram,
- Subsystem test scenarios, derived from the system white-box sequence diagrams, and
- Non-functional requirements that are allocated to the subsystem.

Systems Engineering Handoff to Hardware and Software Development (3)

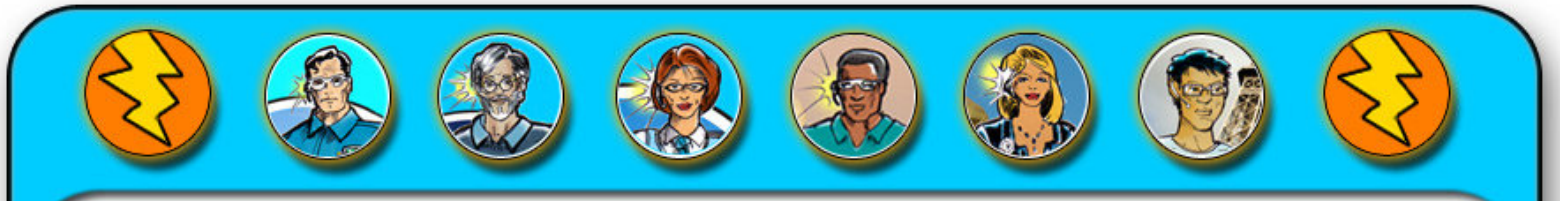
If systems engineers hand-off their specifications directly to HW/SW development, each HW/SW component specification will include the

- Baselined executable HW/SW component model,
- Definition of HW/SW component allocated operations including links to the associated system functional requirements,
- Definition of HW/SW component ports and logical interfaces (Component Logical Interface Document)
- HW/SW component statechart diagram,
- HW/SW component test scenarios, derived from the system white-box sequence diagrams, and
- Non-functional requirements that are allocated to the HW/SW component.

Model-Based Systems Engineering




QUESTIONS



THANK YOU

Learn more at:

IBM Rational software

IBM Rational Software Delivery Platform

Process and portfolio management

Change and release management

Quality management

Architecture management

Rational trial downloads

Leading Innovation Web site

developerWorks Rational

IBM Rational TV

IBM Rational Business Partners