Digital Engineering

Requirements to Systems Engineering with HarmonyMBE

Peter Schedl
Program Manager
IBM Engineering Lifecycle Management
peter.schedl@de.ibm.com

Questions to be answered

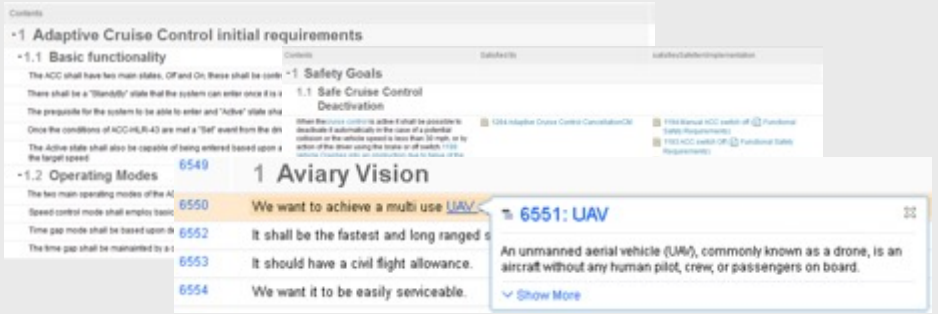
- What is Requirements Engineering ?
- Does Modeling help ?
- How much Modeling do I need?
- What is MBSE and how does it relate to Requirements Engineering ?
- What is HarmonyMBE ?

- How to valdidate Requirements - or - what is this ?

Requirements Definition and Management

Requirements Engineering involves:

- Requirements elicitation – gathering requirements from stakeholders/customers
- Requirements analysis and negotiation – check clarity, completeness, resolve issues
- Requirements specification – document with text, maybe with use cases and scenarios
- Functional analysis
- Requirements validation
- Requirements management – continuous activity, traceability, change handling



Requirements Challenges

Misunderstood requirements by stakeholders and engineers

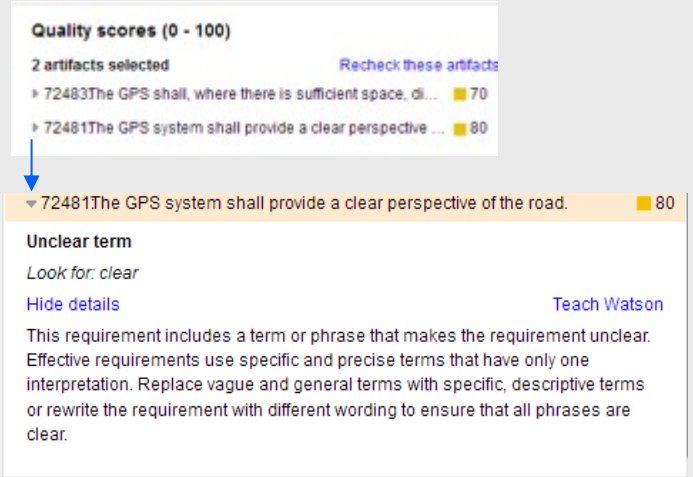
Poorly expressed requirements → RQA: AI based quality checker

Misunderstanding or omission by development

Missed test coverage

Requirement change impact misunderstandings

Little reuse



Requirements Challenges

Misunderstood requirements by stakeholders and engineers

Poorly expressed requirements → ChatGPT example

Misunderstanding or omission by development

Missed test coverage

Requirement change impact misunderstandings

Little reuse

"Ok. That's what we need to do:
Thread A will pass event X to thread B and that will change B's state to Running from what it was before which was Idle. When B changes to Running it will send back an event Y to A and then wait for 2 second and then go back to Idle. Thread A will have started in Idle also and will go to Run after B sends back event Z which happens after the 2 seconds before going to Idle. All this should happen in less then 5 seconds."

Functional Requirements:
1.Thread A shall pass event X to Thread B.
2.Upon receiving event X, Thread B shall change its state from Idle to Running.
3.After changing to Running state, Thread B shall send event Y to Thread A.
4.Thread B shall wait for 2 seconds after sending event Y.
5.After waiting for 2 seconds, Thread B shall change its state from Running to Idle.
6.Thread A shall start in Idle state.
7.After receiving event Y from Thread B, Thread A shall change its state from Idle to Running.
8.Thread A shall wait for event Z before changing its state to Idle.
9.Event Z shall be sent by Thread B after waiting for 2 seconds in Running state.
10.The entire process of passing event X from Thread A to Thread B, changing states, waiting
Non-Functional Requirements:
1.The system shall ensure that the communication between Thread A and Thread B is reliable
2.The system shall ensure that the time delay between sending events, changing states, and
3.The system shall be designed to handle multiple instances of the process simultaneously w

Requirements Challenges

Misunderstood requirements by stakeholders and engineers

Poorly expressed requirements

Misunderstanding or omission by development

Missed test coverage

Requirement change impact misunderstandings

Little reuse

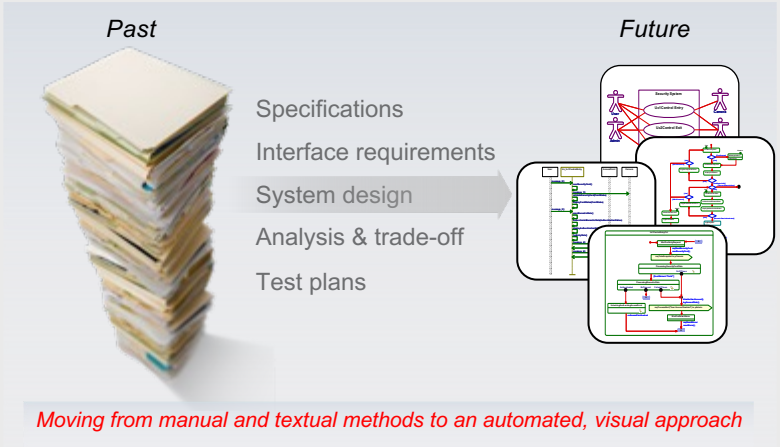
"Ok. That's what we need to do:
Thread A will pass event X to thread B and that will change B's state to Running from what it was before which was Idle. When B changes to Running it will send back an event Y to A and then wait for 2 second and then go back to Idle. Thread A will have started in Idle also and will go to Run after B sends back event Z which happens after the 2 seconds before going to Idle. All this should happen in less then 5 seconds."

Complexity is the biggest challenge facing organizations today!

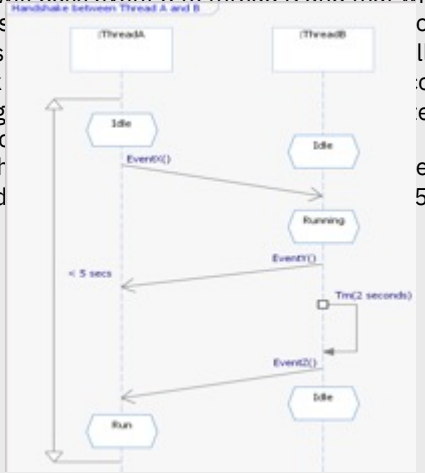
Modeling in Requirements Engineering

Requirements Engineering involves:

- Requirements elicitation
- Requirements analysis and negotiation
- **Requirements specification**
- **Functional analysis**
- **Requirements validation**
- Requirements management



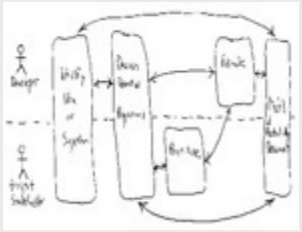
"Ok. That's what we need to do:
Thread A will pass event X to thread B and that will
change B's ore
which was ll
send back ond
and then g ed
in Idle alsc
event Z wh e
going to Id 5
seconds."



Why Modeling?

Manage complexity

– Complicated applications need a visual plan



Simplify and abstract ! essential aspects of a system

– Increase understanding of requirements

Enhance communication

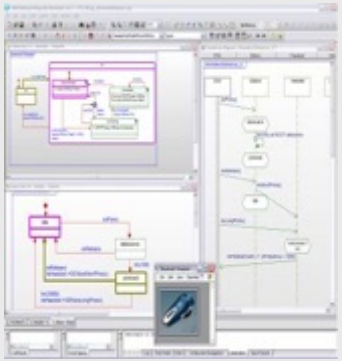
– Common language promotes common understanding across disciplines

Reduce risk

– Model execution increases knowledge and reduces uncertainty and risk

Provide traceability

– Models document what you have done



Modeling in Requirements Engineering

Requirements Engineering involves:

- Requirements elicitation
- Requirements analysis and negotiation
- **Requirements specification – use cases and scenarios for describing user interactions**
- **Functional analysis – functional flows, interface definition, documented rationale**
- **Requirements validation – testing of requirements through model execution**
- Requirements management

2 Functional Requirements

2.1 Power car

2.1.1 Move car

2.1.1.1 Move forwards

The car shall be able to move forwards at all speeds from 0 to 200 kilometers per hour on standard flat roads with winds of 0 kilometers per hour, with 180 BHP.

2.1.1.2 Move backwards

The car shall be able to move backwards to a maximum speed of 20 Kilometers per hour on standard flat roads with winds of 0 kilometers per hour, with 180 BHP.

2.1.2 Accelerate car

The car shall be able to accelerate from 0 to 100 Kilometers per hour in 10 seconds on standard flat roads with winds of 0 kilometers per hour.

The car shall be able to accelerate from 100 to 150 kilometers per hour at a rate of 5 kilometers per second on standard flat roads with winds of 0 kilometers per hour.

The car shall be able to accelerate from 150 to 200 kilometers per hour at a rate of 3 kilometers per second on standard flat roads with winds of 0 kilometers per hour.

2.2 Control car

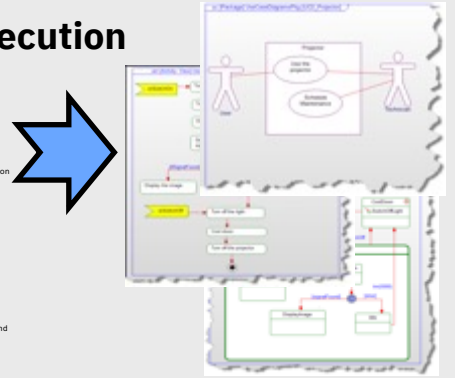
2.2.1 Switch on car

The car shall be able to discriminate which authorized people shall be able to switch on and operate the car.

2.2.2 Control speed

The car shall have a foot mechanism to control the speed of the car.

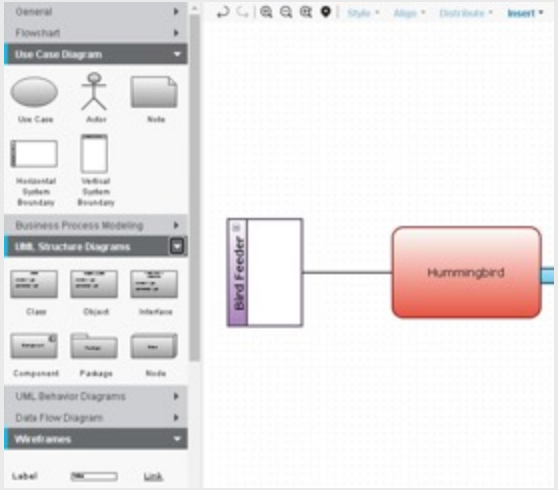
The speed control shall be infinitely variable from zero to maximum speed.



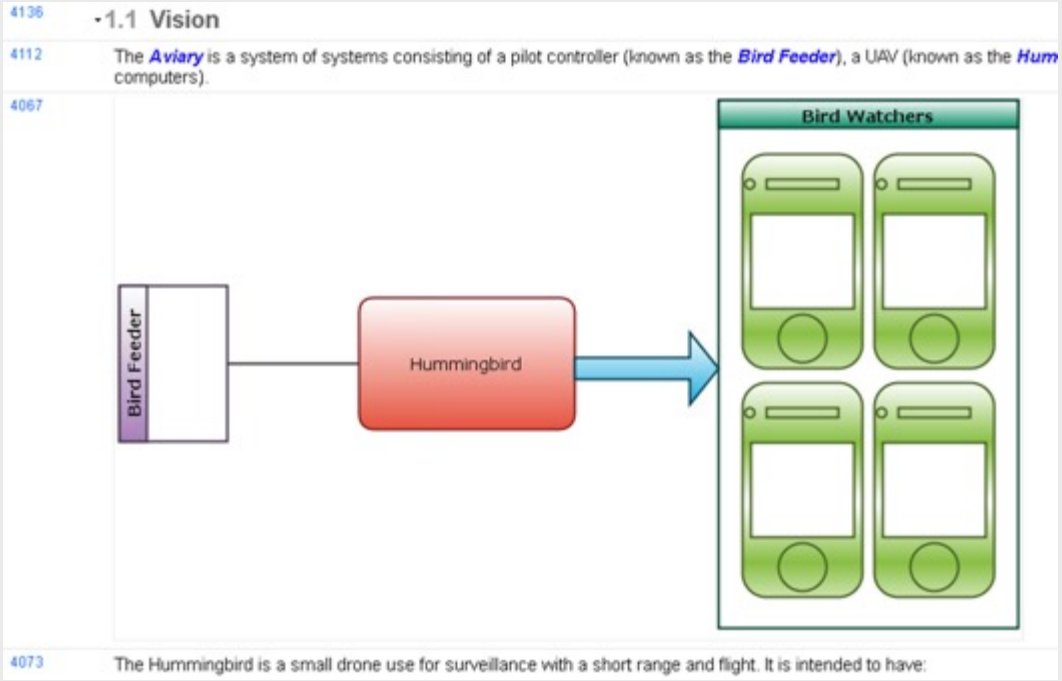
Improve Requirements specification by just adding “Diagrams”

Fully integrated (like textual requirement)

Offers rich notations



Not a formal model !



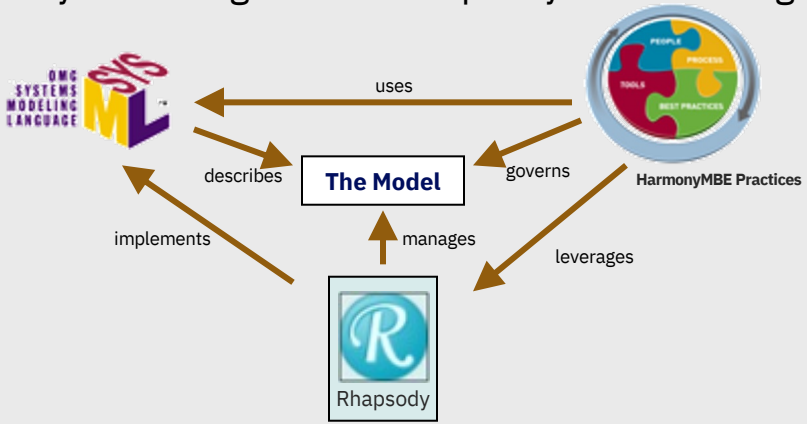
IBM’s Model Based Systems Engineering (MBSE) Solution

MBSE is a standards based Systems Engineering practice that incorporates:

– Modeling language – SysML

– Modeling method – Harmony Systems Engineering Practices

– Modeling tool – Rhapsody for Systems Engineers & Rhapsody Model Manager

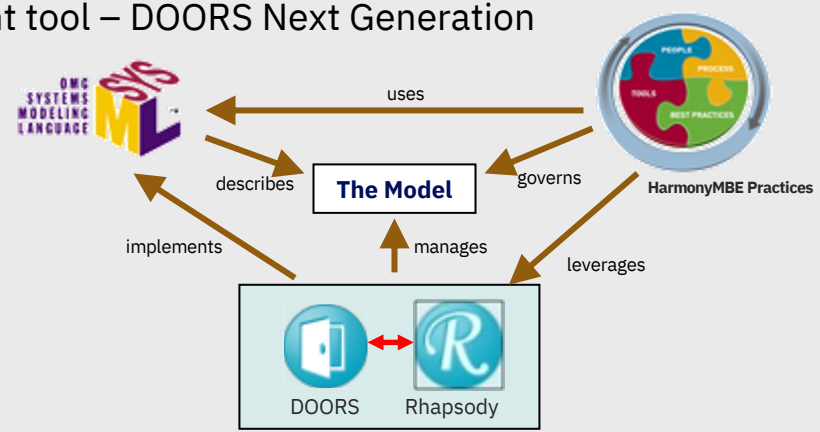


Requirements ?

IBM’s Model Based Systems Engineering (MBSE) Solution

MBSE is a standards based Systems Engineering practice that incorporates:

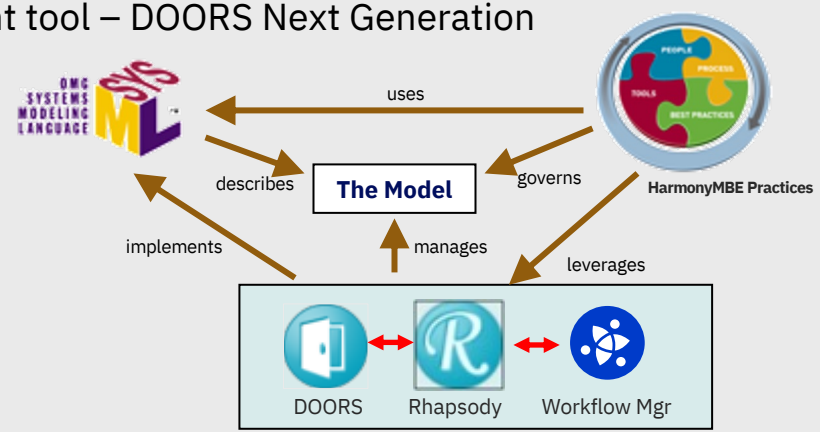
- Modeling language – SysML
- Modeling method – Harmony Systems Engineering Practices
- Modeling tool – Rhapsody for Systems Engineers & Rhapsody Model Manager
- Requirements management tool – DOORS Next Generation



IBM's Model Based Systems Engineering (MBSE) Solution

MBSE is a standards based Systems Engineering practice that incorporates:

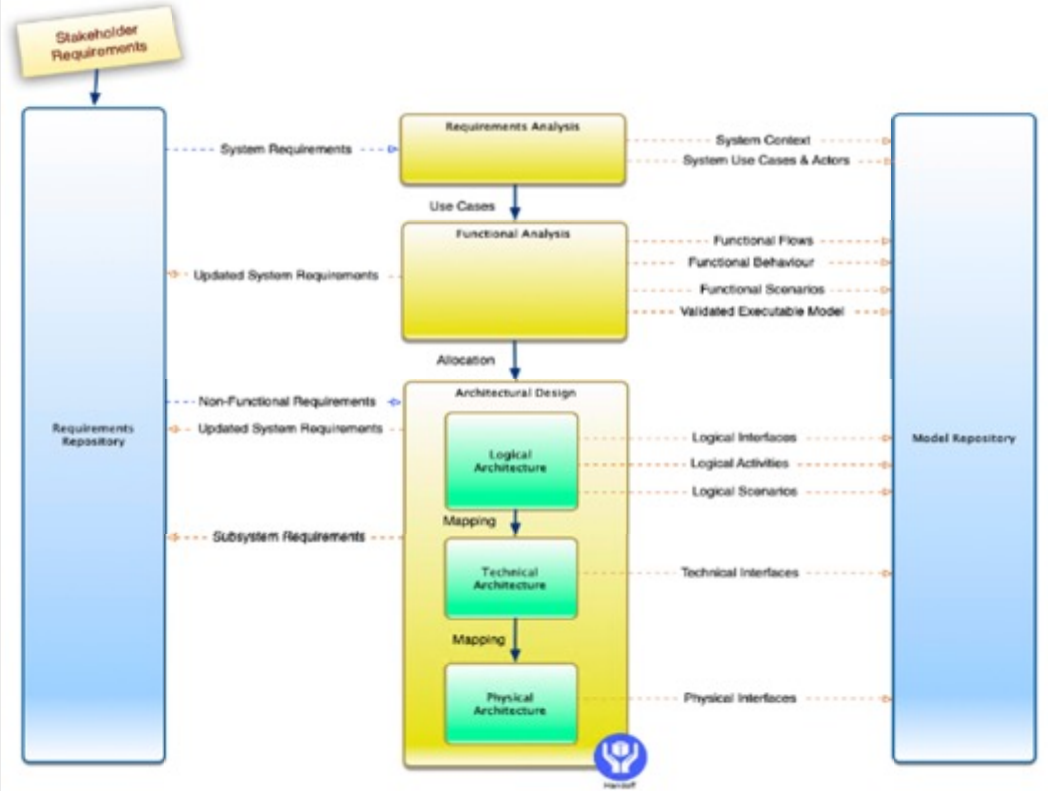
- Modeling language – SysML
- Modeling method – Harmony Systems Engineering Practices incl. Ticket System for Guidance & Com.
- Modeling tool – Rhapsody for Systems Engineers & Rhapsody Model Manager
- Requirements management tool – DOORS Next Generation



HarmonyMBE Practice provides Guidance & Automation

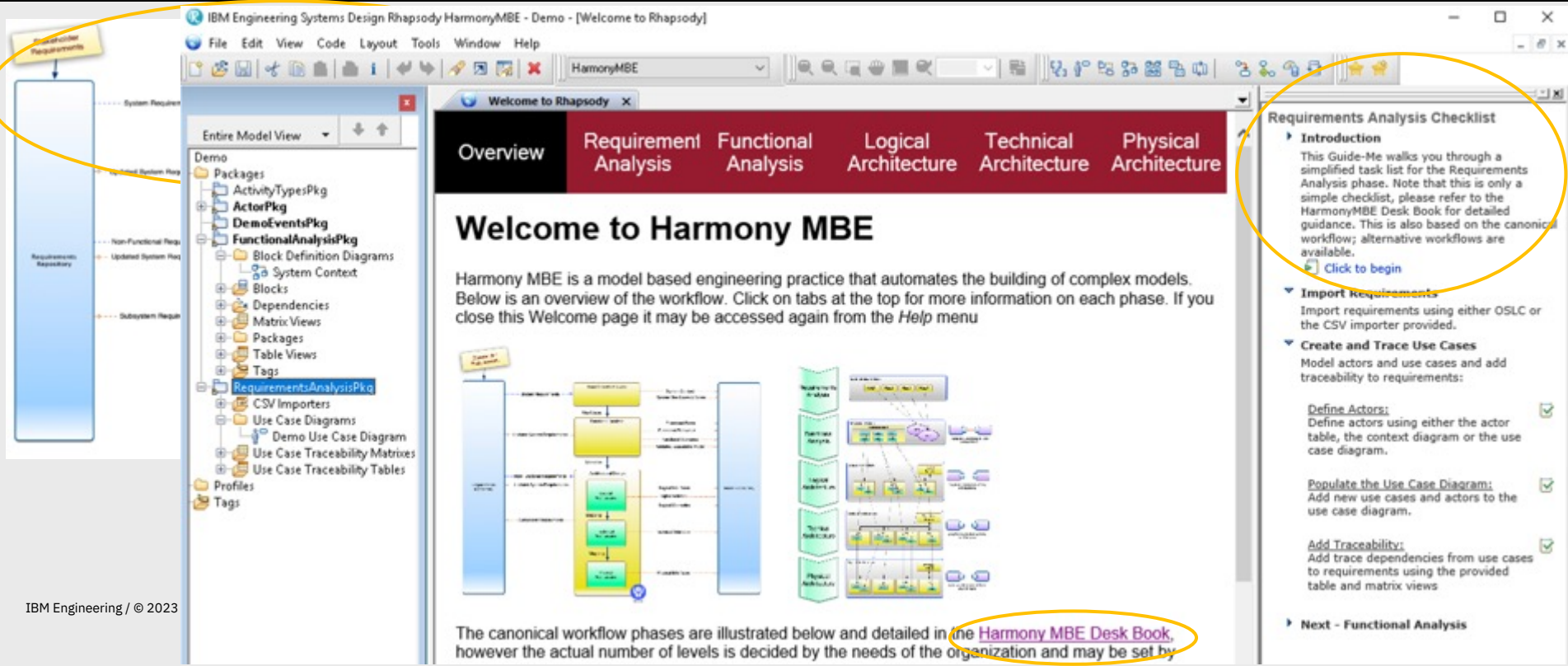
Model Based Systems Engineering complements traditional requirements analysis techniques

- during Requirements Analysis, we organize requirements into functional groups (use cases)
- during Functional Analysis, we identify system functions and explore the system's dynamic behavior using activity diagrams and model execution
- during Architectural Design, system operations are allocated to decomposed Logical Architecture
- Finally either direct Hand-Off to the engineering teams or first map to a Technical Architecture



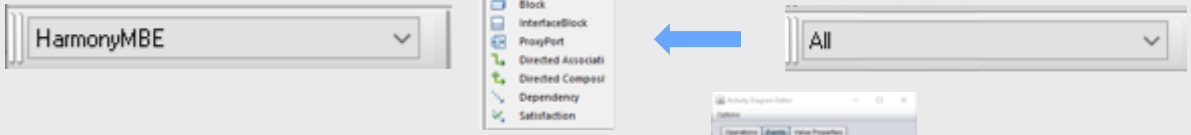
HarmonyMBE Workflow – Integrated in Tool Rhapsody

HarmonyMBE Workflow - Can be used with any Tool

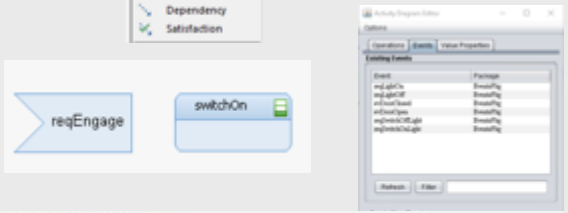


HarmonyMBE simplifies MBSE

Reduced Menu's & Toolbars



Model creation Support (e.g. Activity Diagram Editor)



Auto generation of ports, interfaces, delegation ports, connectors and diagrams

New architecture layers are generated on demand

– Allocation & Re-use of blocks, ports and interfaces

Harmony MBE Benefits

Proven Method with simplified Tool UI and automation

Only create core information: Use Cases, Activities and Subsystems (BDD) incl. State Charts



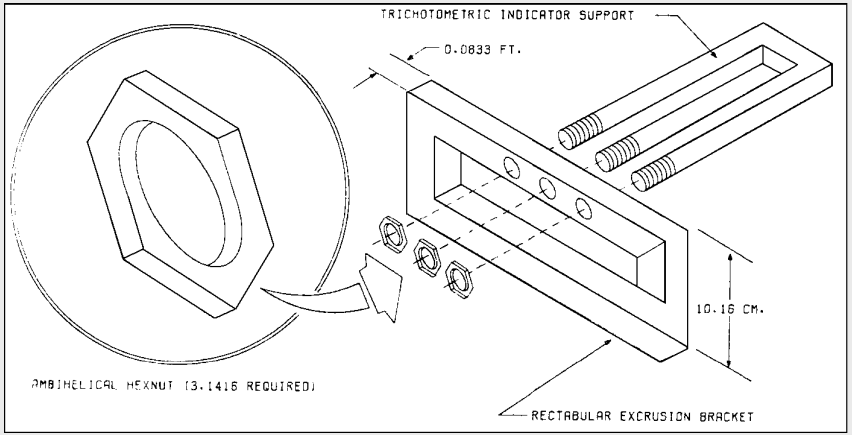

Get verified System Architecture incl. Interfaces (ICD) & Test Scenarios



Linked to Requirements & Tasks

Is my Model correct ?

„How do you validate that you have a ‚useful' model?



Continuously validate and verify thru Model Execution

Once the system behavior has been captured it can be verified through execution

– The Sequence Diagrams should be used as the basis for stimulating the model and to record the test run

Advantages of execution

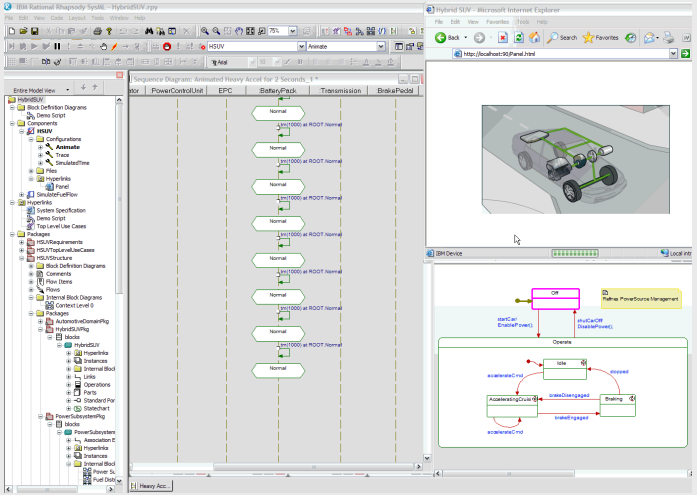
– Helps eliminate bugs and design flaws

– Helps build a robust system as unhandled conditions are exposed

– Helps test interfaces…Reducing integration issues further down

– Helps test for regressions

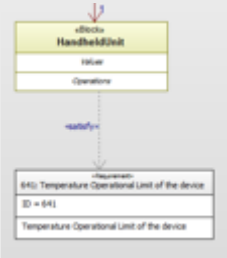
– Verified Test Cases can be handed down the life-cycle



Requirements Revisited – Modellers Point of View

Functional Requirements have been traced to Model Elements like Use Cases, Blocks, Operations, Interface Data

Non-functional requirements are traced to Subsystem Blocks



Visualize the Traceability in a Matrix View

From: block, PrimitiveOperation **Scope:** ParkingCtrlPkg

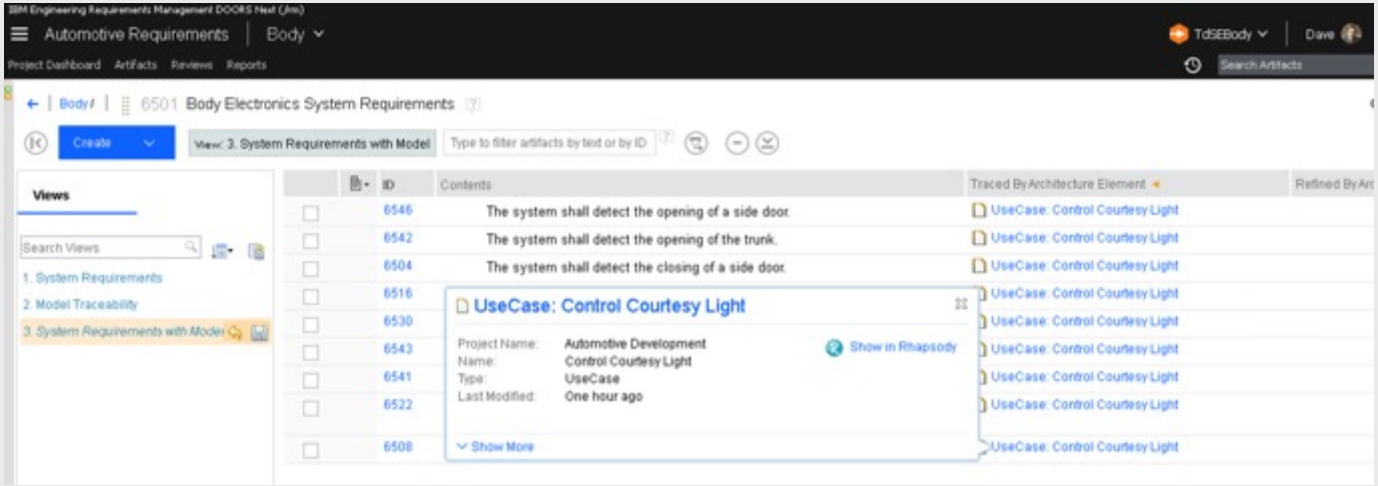
	ParkingCtrl	confirmGuest
SR001 - Entry		SR001 _ Entry
SR007 - Allocation of Parking Spaces		SR007 _ Allocation of Parking Spaces
SR009 - Price Information	SR009 _ Price Information	
SR014 - Pay via Checkout		SR014 _ Pay via Checkout

When a Requirement changes – the impact of that change may be analyzed

Requirements traced to Model Artefacts – RE Point of View

Requirements tight integration with Models

Join also my other talk about Importance of Traceability



When a Requirement changes – the impact of that change may be analyzed

Summary

Model Based Systems Engineering (MBSE) complements traditional requirements definition and management techniques

- Work at appropriate level of abstraction graphically
- Can be scaled from light “drawing” to full MBSE incl. model simulation
- Continuously validate and verify
- Provide detailed traceability

IBM provides with HarmonyMBE a SysML-based Systems Engineering practice library with application guidance

Functional Analysis …

- is a Requirements Engineering activity
- emphasizes the transformation of functional system requirements into a coherent description of system functions
- improves upon the quality of system requirements

Thank You

Peter Schedl
peter.schedl@de.ibm.com

Further information:

[IBM Engineering Lifecycle Management Automotive Compliance]
[IBM Engineering Management Overview]