

Digital Engineering: MBSE Quick and Easy with AI

think

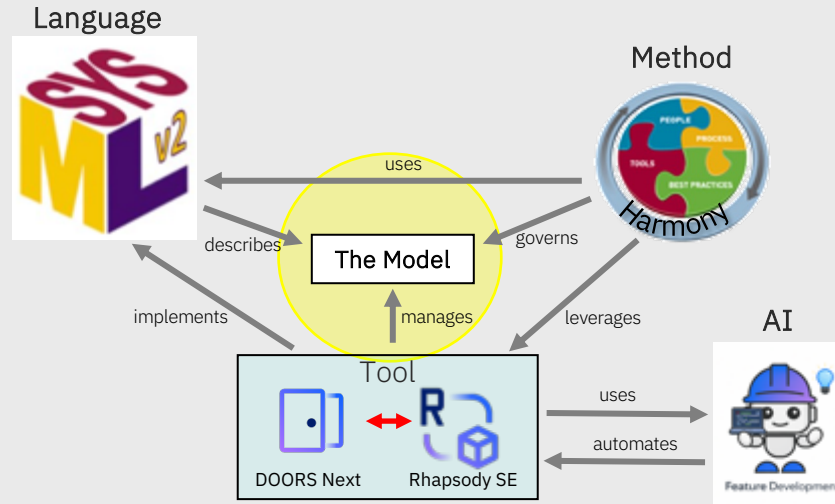


Peter Schedl
Program Manager
IBM Engineering Lifecycle Management
peter.schedl@de.ibm.com

Executive Summary - Be Smart use AI wisely orchestrated in Engineering

Engineering requires models defined in Model-based Systems Engineering (MBSE).

MBSE consists of Language + Method + Tools **enhanced by AI**

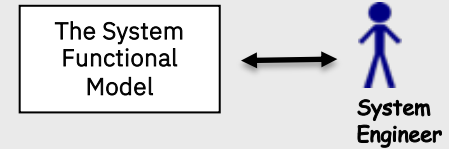


What is MBSE

Model-Based Systems Engineering (MBSE) is a methodology that uses **formal models** as the primary means of information exchange and system development, replacing traditional document-centric approaches

...Or...

A visual approach to understanding **requirements** and realizing them into a robust **system design**



Vision 2035: The Future of Systems Engineering is **Predominantly Model-Based**

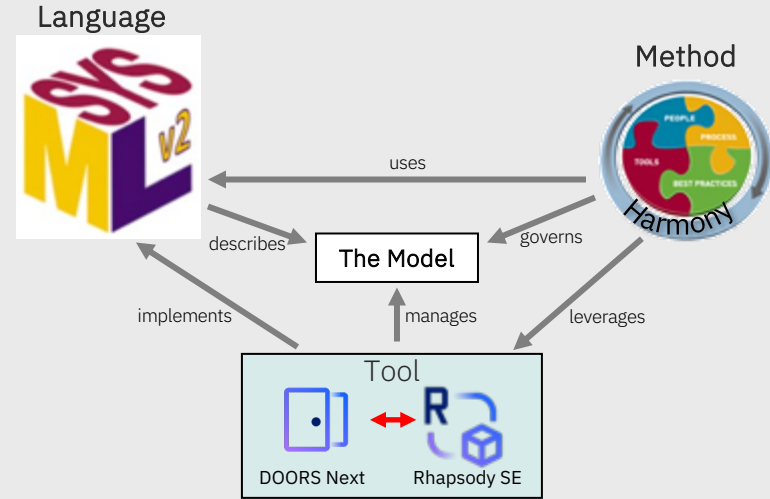
MBSE Modeling Solution

MBSE is a Systems Engineering practice that incorporates:

- Modeling language
- Modeling method
- Modeling + Requirements Mgt tool

MBSE transforms system development by

- Providing a **single source of truth** for all stakeholders
- Enabling early **validation** and error detection via standard, **formal notation** (SysML)
- Improving **communication** through visual models
- **Automating** repetitive tasks and documentation
- Ensuring **traceability** throughout the lifecycle
- **Managing complexity** in large-scale systems



MBSE is not just a tool –
It's a fundamental shift in how we think about
and develop complex systems

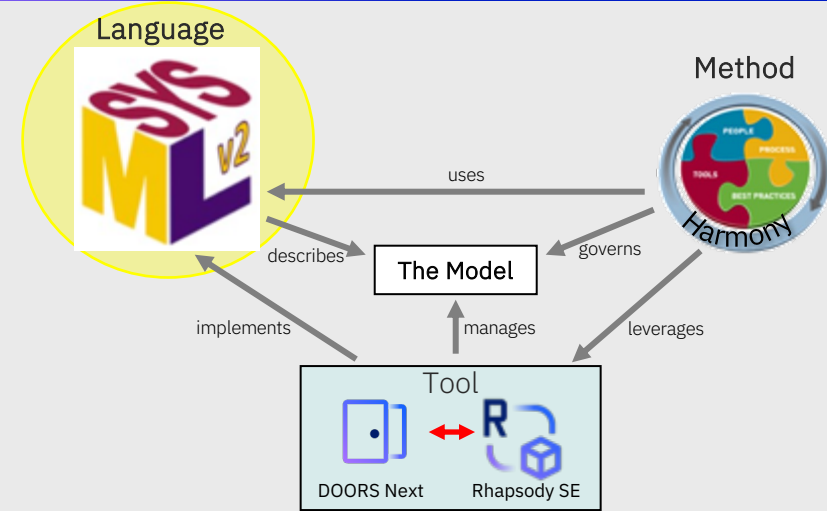
There has been an industry-wide call for a more comprehensive MBSE standard

Requires **NEW**:

- Modeling language – **SysML v2**
- Modeling method
- Modeling tool

SysML v2 Key Enhancements

- For System Engineers (not based on UML/SysML v1)
- Well defined Graphical & Textual **language semantics**
- **Integration API**
- **Extensibility** to Domain Specific Languages
- Greater coverage of systems engineering activities (e.g. geometry)



AI in Product Engineering?

Engineering teams face increasing complexity

- More software-defined products with cross-domain dependencies (mechanic, electric, ...)
- Faster release cycles with stricter compliance
- Large volumes of requirements, models, interfaces, variants and test artifacts

AI advantages

- Faster analysis of large engineering datasets
- Better traceability across lifecycle artifacts
- Higher automation in repetitive knowledge work
- Improved consistency in models, code and documentation
- Decision support for architects and systems engineers

AI for Engineering Challenge

Engineering as high trust domain **vs** GenAI as probabilistic technology

Approach for GenAI supported teams

- Agents augment tasks
- Orchestration and stepwise automation across workflows
- Human-centered decisions

The Goal: From Requirements to Models to Code (and back)

- Requirements analyzed by AI
- Candidate SysML v2 MBSE models generated stepwise
- Models reviewed and approved by engineers stepwise
- Code and interfaces generated from models
- Tests and documentation generated from the same source
- Feedback from implementation refines model and requirements

AI enables continuity across engineering artifacts

SysML v2 Textual vs API: Two complementary ways to work

Textual

- Human-readable and version-control friendly
- Easy to diff and merge
- Good for AI-assisted generation

– Textual is good for AI authoring

API

- Programmatic access to model elements and relationships
- Enables integration with engineering toolchains
- Supports automation, synchronization, validation and analytics
- Better for scalable system-to-system workflows

– API is ideal for automation and enterprise integration

ChatBot, MCP and Agents

ChatBot

- Conversational interface
- Good for Q&A, summarization and simple assistance
- Usually reactive and **user-prompt driven**

MCP

- Model Context Protocol for **standardized tool access**
- Connects models to tools, files, APIs and external systems
- Gives LLM-based solutions reliable context and action capabilities

Agents

- Goal-oriented software entities
- Can plan, reason, call tools, coordinate tasks and **act autonomously**
- Useful for multi-step engineering workflows

Examples

- “Explain this requirement”
- “Retrieve related model”
- “Analyze requirement, update model and generate code stub”
[stepwise orchestrated](#)

Target Operating Model for AI in Engineering – Human-in-the-loop

- **AI proposes** requirements interpretations, incremental models and code
- **Engineers approve** safety-critical and architecture-relevant decisions
- Toolchain **integration via MCP** and APIs
- **MBSE backbone** ensures consistency and structure

Operating principles

- Separate proposal from doing / approval
- Log all AI-generated changes
- Validate against engineering rules and standards

AI for Engineering Examples



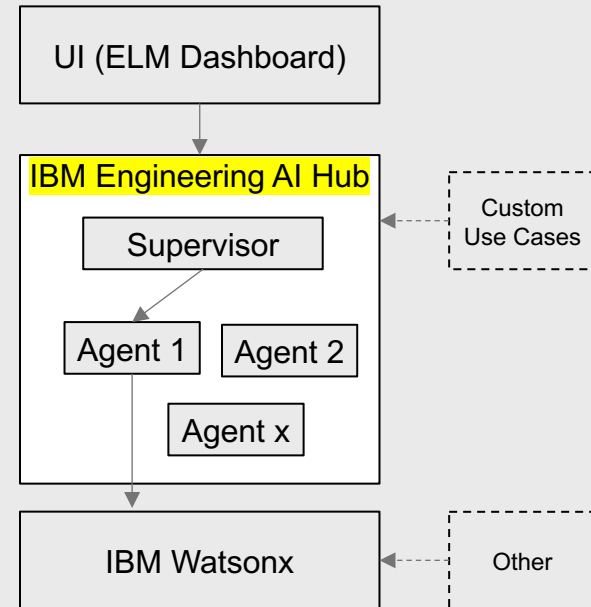
AI for Engineering Example – IBM Engineering AI Hub utilizing Agents

AI agents to improve clarity and speed in complex, mission-critical systems and software

- Automate repetitive, error-prone tasks to focus on high-value work
- Outputs grounded in your engineering data
- AI agents propose; Engineers review and confirm
- Designed to grow with your needs

Actual Agents / Use Cases

- Engineering Assistant
- Requirements Quality Analysis & Recommendation
- Work Item Synopsis & Work Item Compose
- **Actors and Use Case Discovery**



AI for Engineering Examples – Engineering AI Hub w DOORS Next

Engineering Assistant

The screenshot shows the IBM Engineering Assistant interface. On the left, a chat window displays a user query: "Do the requirements mention the lowest speed the cruise control would work at?". The assistant's response is: "Yes, the requirements do specify the lowest speed for cruise control activation. According to Requirement 14688, the cruise control is expected to work at 30 mph or higher." Below this, a follow-up question is asked: "Would you like to see if any requirements mention the highest speed for cruise control?". A list of items is shown, including Requirement 14688. On the right, a table of requirements is visible, with the first row highlighted: "14651 • 1 Basic Cruise Control Subsystem S Requirements".

Requirements Analysis

The screenshot shows the IBM Requirements Analysis interface. On the left, a sidebar displays "Requirement details" for requirement 14655: "When entering active software control, the vehicle speed shall be controlled by commanding engine output, in or...". Below this, a "Score" section shows "Low quality" with a progress bar indicating "2 out of 10". An "Issues" section lists "Imprecise verb", "Missing unit", and "Passive voice". A "Recommendation" section provides advice: "Upon entering Active mode, the system shall regulate vehicle speed to maintain the set speed within +/- 2 mph, using available control mechanisms." and shows a "Resulting score: 9". On the right, a table of requirements is visible, with the first row highlighted: "14655 When entering active software control, the vehicle speed shall be controlled by commanding engine output, in or...".

AI for Engineering Examples – “Identify use cases and their system actors” w Rhapsody SE

from textual requirements

generate Use Cases and Actors

The screenshot displays the IBM Rhapsody SE interface, which is powered by IBM Watson. The main window is titled "requirements table" and shows a list of requirements. A context menu is open over the table, with the "AI Extension" option selected. This extension has generated AI suggestions for use cases and actors.

AI Suggestions Panel:

- Use cases and actors discovered by AI**
Showing 5 use cases and 2 actors (7 of 7 total items).
Review, edit, and add the ones you want to your model.
- Search by name or description...**
- Table of suggestions:**

Name	Type
Vehicle Sensor System	System Actor
User	System Actor
Door Status Monitoring	Use Case
Side Door Event Detect	Use Case
Trunk Event Detection	Use Case
Courtesy Light Control	Use Case
Temperature Control	Use Case

UML Diagram:

The diagram illustrates the relationships between the generated elements. It shows a central "subject" system (represented by a white box) connected to several "system actors" (represented by orange boxes) and "use cases" (represented by green boxes). The actors include "Vehicle Sensor System" and "User". The use cases include "Door Status Monitoring", "Side Door Event Detect", "Trunk Event Detection", "Courtesy Light Control", and "Temperature Control". Arrows indicate the associations between the actors and the use cases.

Requirements Table:

Declared name	Requirement doc
doorDetection	The system shall detect the open
sideDoorOpenDetection	The system shall detect the open
doorStatusTracking	The system shall keep track of l
sideDoorCloseDetection	The system shall detect the clor
trunkOpenDetection	The system shall detect the open
trunkCloseDetection	The system shall detect the clor
courtesyLightOn	The system shall switch on the i
courtesyLightOff	The system shall switch off the i
Dim	The system shall dim the courte
control	The system shall maintain a set

AI for Engineering Examples – Engineering AI Hub Extended (A2A) Examples

User creates extended uses cases combining existing agents and MCP functionalities

- Quality Analyze recently changed requirements (i.e. time or changeset)
- Impact analysis on recently changed requirements

Requirement Change Agent 2:03 PM Show Reasoning

Impact Summary

Metric	Count
Total Root Requirements Changed	2
Total Unique Affected Requirements*	6

*Unique affected requirements do not include the root requirements (299, 298).

Detailed Impact per Root Requirement

The change to 299 will potentially affect the following downstream requirements: 229, 228, 506, 230, 507

Id	Primary Text
229	The software shall cancel BCC Operation when the 'OFF' button is pressed
228	The software shall turn off all control functions to cancel the system when ignition key is turned to off position.

You 2:00 PM

Give me the requirements that have changed in tab last 5 minutes.

Quality Assessment Agent 2:00 PM Show Reasoning

Recently Modified Requirements (last5minutes)

ID	Summary	Primary Text (text inside <p> tags)
299	The Basic Cruise Control operation shall be canceled by the operator or automatically via the BCC system. Any of the following conditions shall deactivate BCC: Ignition key is turned to the OFF position for 100-ms 'OFF' button is pressed for 100-ms Vehicle Speed > 0 mph and no car is detected from rear-camera.A Relevant Fault is detected The Steering Wheel is turned with a torque greater than 5.4Nm.	The Basic Cruise Control operation shall be canceled by the operator or automatically via the BCC system. Any of the following conditions shall deactivate BCC: Ignition key is turned to the OFF position for 100-ms 'OFF' button is pressed for 100-ms Vehicle Speed > 0 mph and no car is detected from rear-camera.A Relevant Fault is detected The Steering Wheel is turned with a torque greater than 5.4Nm.
298	The Driver Control Subsystem shall provide power for 700ms following a command to initialize.	The Driver Control Subsystem shall provide power for 700ms following a command to initialize.

I can now score the wording of each requirement and offer feedback on how to improve them. May I proceed?

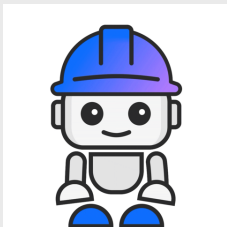
Test Case Summary		
Root Requirement ID	Test Case ID	Owner
299	34	elmuser02
299	35	elmuser03

AI for Engineering - The complete Software Development Lifecycle in the Age of AI with IBM Bob

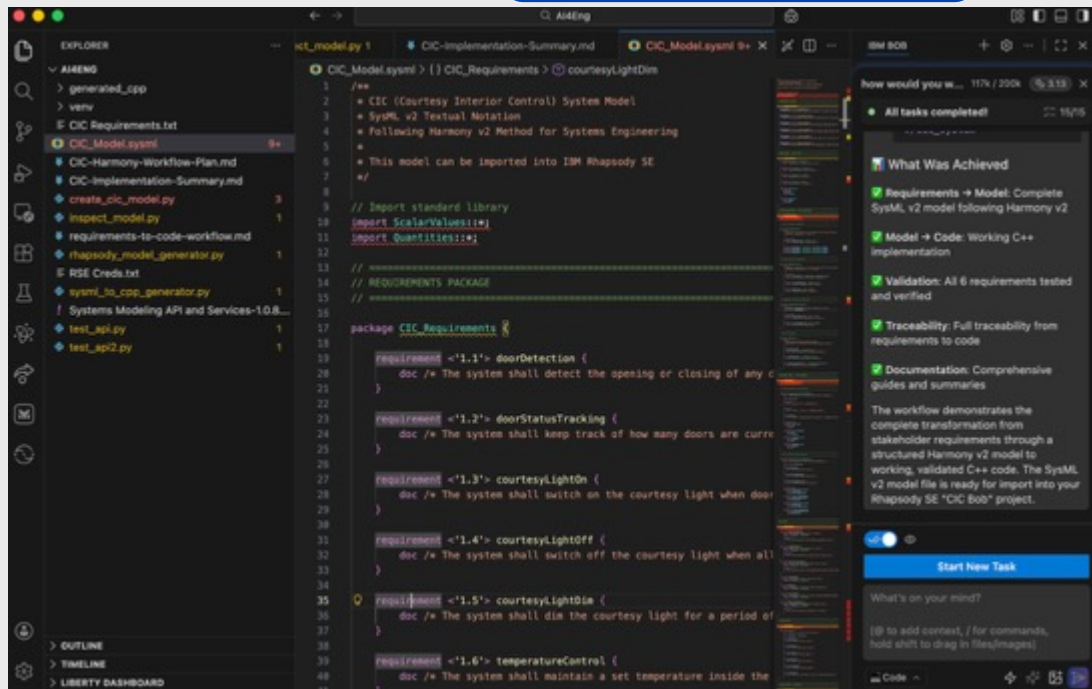
Hi, I'm Bob!

I'm more than a code assistant

- Modes & Skills
- MCP Library
- Diagrams
- Security Findings & Resolution
- Documentation
- DevOps & Automation



I can help you to



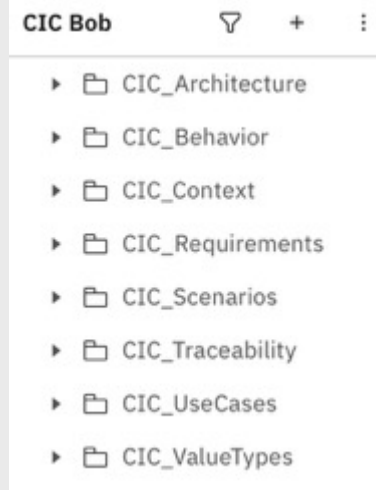
AI for Engineering Examples – IBM Bob used to build SysML v2 Model from Requirements

Input: 6 CIC Requirements

Generates SysML v2 textual notation

Imported in Rhapsody SE

	Declared name	Requirement doc
1	courtesyLightDim	The system shall dim the courtesy light for a period of 30 seconds before...
2	courtesyLightOff	The system shall switch off the courtesy light when all doors are closed.
3	courtesyLightOn	The system shall switch on the courtesy light when door is opened.
4	doorDetection	The system shall detect the opening or closing of any door on the vehicle.
5	doorStatusTracking	The system shall keep track of how many doors are currently open.
6	temperatureControl	The system shall maintain a set temperature inside the vehicle.



Views Populated



AI for Engineering Examples – IBM Bob to build Rhapsody SE Extensions

Task

Generate an Rhapsody SE extension which reads the type Use Case lines including the used columns from the file UCD.xlsx. For example the Primary Text shall be the name. Generates these as use cases in Rhapsody SE with the appropriate attributes.

Todo List 11/12

- Analyze UCD.xlsx structure and identif...
- Set up new extension project structure ...
- Install required dependencies (xlsx libr...
- Create Excel file reader utility to parse ...
- Implement data extraction logic for Use...
- Create mapping between Excel column...
- Implement Use Case creation logic usin...
- Create user action to trigger import fro...
- Add error handling and validation
- Add user notifications for import progr...
- Test the extension with sample data
- Document the extension usage and dep...

UCD Excel Importer Extension - Implementation Complete ✓

I have successfully created a complete Rhapsody SE extension that reads Use Case data from UCD.xlsx files and generates Use Cases in Rhapsody SE with appropriate attributes.

What Was Created

Core Implementation Files

1. **src/types.ts** (82 lines) - TypeScript interfaces for Use Case data, Excel rows, and results
2. **src/excel-reader.ts** (186 lines) - Excel file parsing, validation, and data extraction
3. **src/use-case-creator.ts** (189 lines) - Use Case creation in Rhapsody SE model with RSE API
4. **src/extension-actions.ts** (128 lines) - User action registration and import orchestration

✨ Key Features

Excel File Processing

- Parses .xlsx files using the xlsx library
- Extracts rows where "Artifact Type" = "Use Case"
- Maps 10+ Excel columns to Use Case attributes
- Validates file structure and data integrity

Use Case Creation

- Creates UseCase elements in Rhapsody SE model

Status: ✓ Implementation Complete
Build: ✓ Successful (321 KB bundle)
Documentation: ✓ Comprehensive (6 guides, 2,562 lines)
Code: ✓ Production-ready (618 lines TypeScript)
Ready for: Deployment and Testing

Generated Use Case Importer:

Copy link

Import Use Cases from Excel

Import Abstract Effect Chain Parts fr...

Move

Rhapsody SE UI

Key Takeaways

AI is a practical accelerator for engineering products

- AI reduces effort in requirements analysis, modeling and coding
- **AI + MBSE** creates a strong foundation for structured automation
- **SysML v2** textual and API approaches are complementary
- MCP enables reliable access to tools and engineering context
- Agents extend beyond chat into **step-by-step engineering workflows**
- Highest value comes from human-governed, traceable, end-to-end integration

The future is not just AI-assisted chat -

It is AI-enabled engineering workflows across requirements, models, code and validation

Thank You

Further information:

[IBM Engineering Lifecycle Management Automotive Compliance](#)

[IBM Engineering Lifecycle Management Overview](#)

[IBM Rhapsody Systems Engineering](#)

[IBM DevOps Loop](#)

think

© Copyright IBM Corporation 2026. 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. Any statement of direction represents IBM's current intent, is subject to change or withdrawal, and represent only goals and objectives. IBM, the IBM logo, and ibm.com are trademarks of IBM Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available at [Copyright and trademark information](#).

