International Council on Systems Engineering
A better world through a systems approach

Systematic Risk Analysis: FMEA and FTA Approaches for Multi-Level Systems Architectures

Presented by: Brian Pepper (3DS)

Co-Authors: Habibi Husain Arifin (3DS), Kyle Post (Ford), Saulius Pavalkis (3DS)

7/29/2025

INCOSE International Symposium 2025 | Ottawa, Canada

Team Bio

Brian Pepper is an Industry Business Senior Consultant for CATIA Magic (Dassault Systemes). He has 20+ years' experience in Systems Engineering with 10 years' Experience applying Model Based Systems Engineering Techniques to complex systems. Specializing in MBSE and Safety and Reliability Integration.

Saulius Pavalkis Global MBSE Ecosystem Director and MBSE R&D Cyber Portfolio Manager – NAM. He has 20 years of MBSE adoption experience in MBSE ecosystem, digital engineering, system architecture and simulation as Cameo co-creator and consultant.

Habibi Husain Arifin is a Research & Development professional, has over 14 years of experience as a Java and C software engineer, MBSE consultant, and solution architect, contributing to a variety of corporate and government projects.

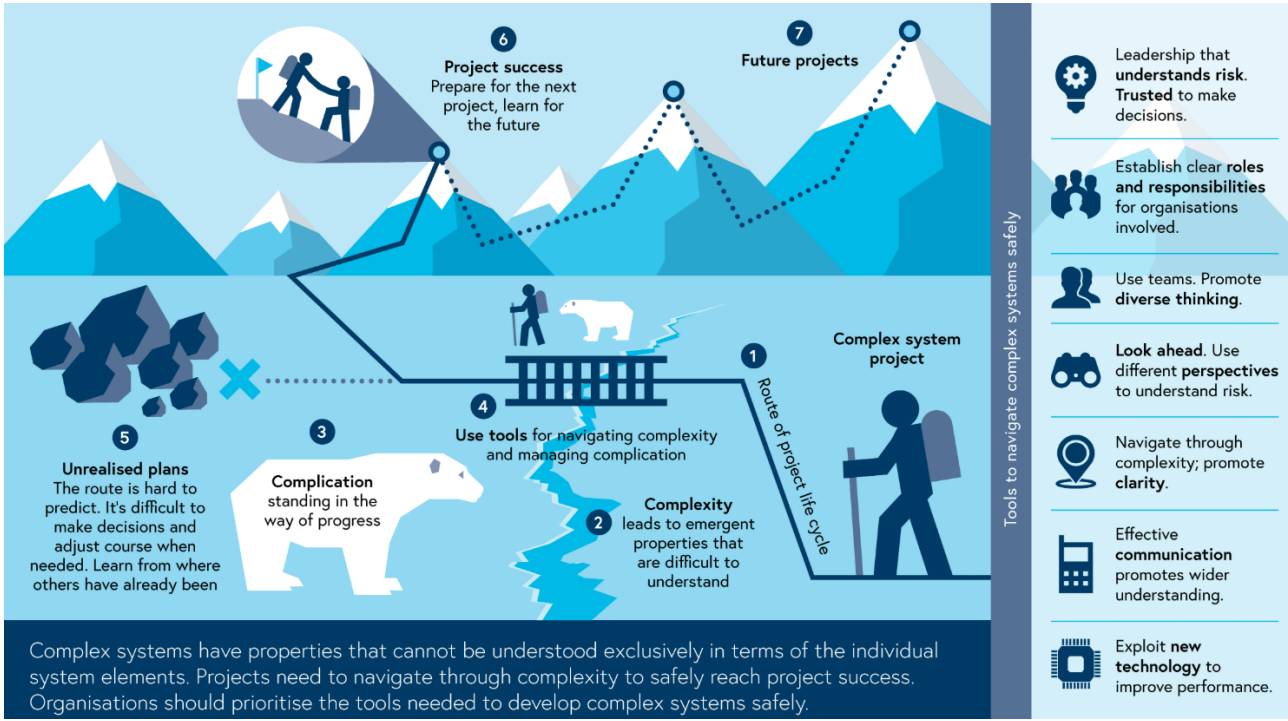
Kyle Post is the Systems Safety Technical Leader at Ford Motor Company; he has 24 years of controls and embedded software experience. Mr. Post is a co-chair of the Risk Analysis and Assessment Modeling Language (RAAML).

Agenda

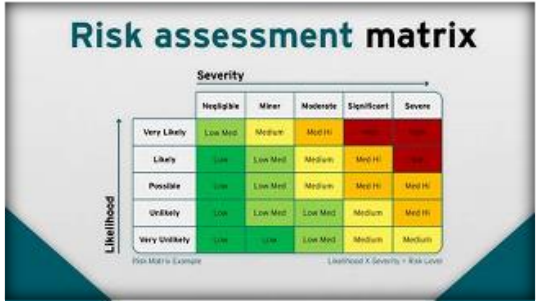
- Introduction: Problem Statement and Objectives
- Demonstration of Workflow
  - Steps of the Workflow process
- Application of Methodology
  - Architecture Levels
  - Process of Creating an Integrated Failure mode Effects Analysis (FMEA)
  - Process of Creating an Integrated Fault Tree Analysis (FTA)
  - Integrating the FMEA and FTA to the Model Architecture
- Conclusions / Future Work

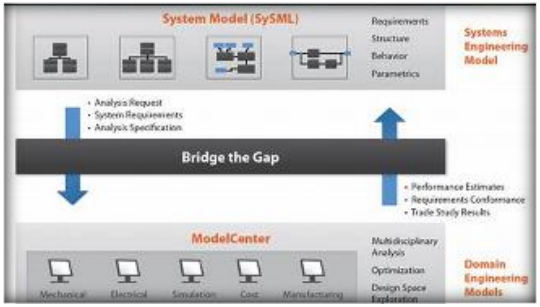
Challenges of Risk Analysis on the Increase of Complex Systems

- **Increasing Complexity in Systems**
  - Modern systems feature integrated functionalities and interfaces, complicating risk identification and traceability across architectures and reliability analyses.
- **Difficulty in Identifying Risks**
  - Traditional methods catalog component failures but lack holistic analysis of systemic interactions, external events, and partial failures.
- **Knowledge Gaps Between Teams**
  - Siloed teams using different reliability approaches and inconsistent terminologies hinder communication, traceability, and harmonization of risk analysis.



Managing Organization-Specific Needs and Customization Challenges





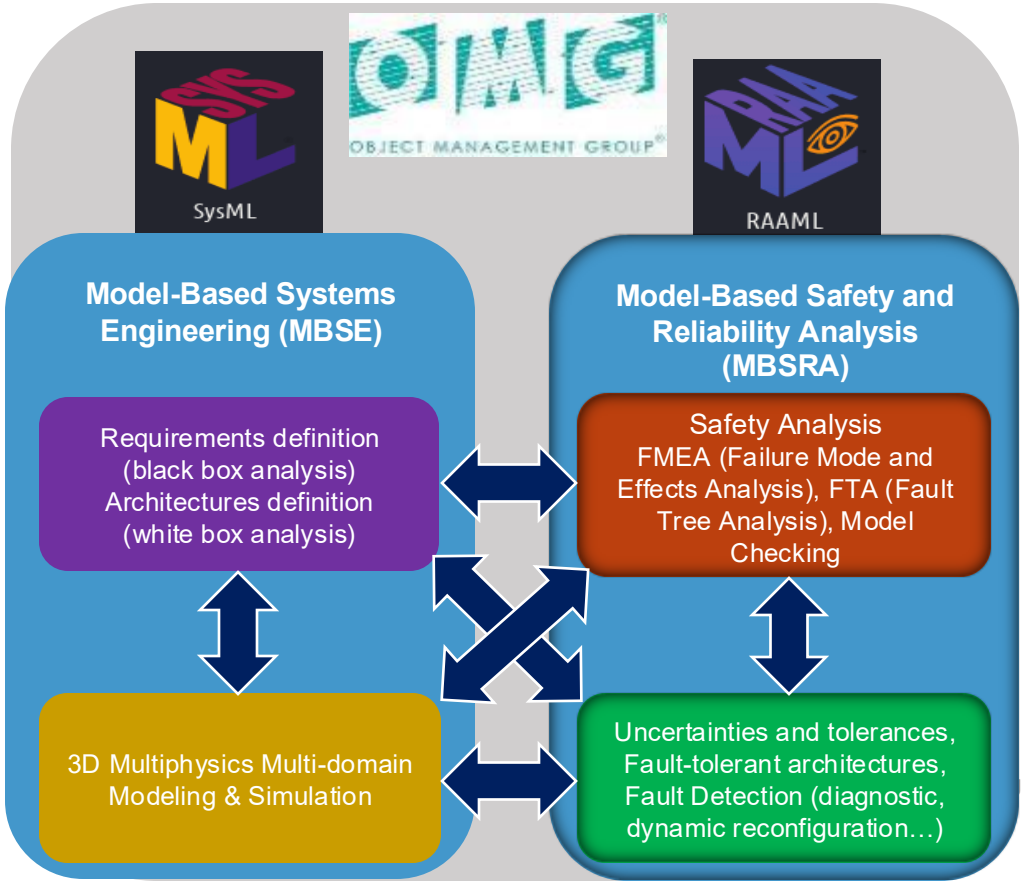
Variability in Terminologies Standards and Tools
Organizations use unique terminologies and risk analysis tools/ templates tailored to their processes and regulations, complicating the adoption of universal reliability tools.

Limitations of Existing Risk Analysis Tools
Current tools often lack compatibility with system architecture models, hindering seamless integration and traceability in complex multi-level systems.

Training and Adoption Challenges
Diverse tools and languages create a steep learning curve, increasing training costs and slowing risk management processes across teams.

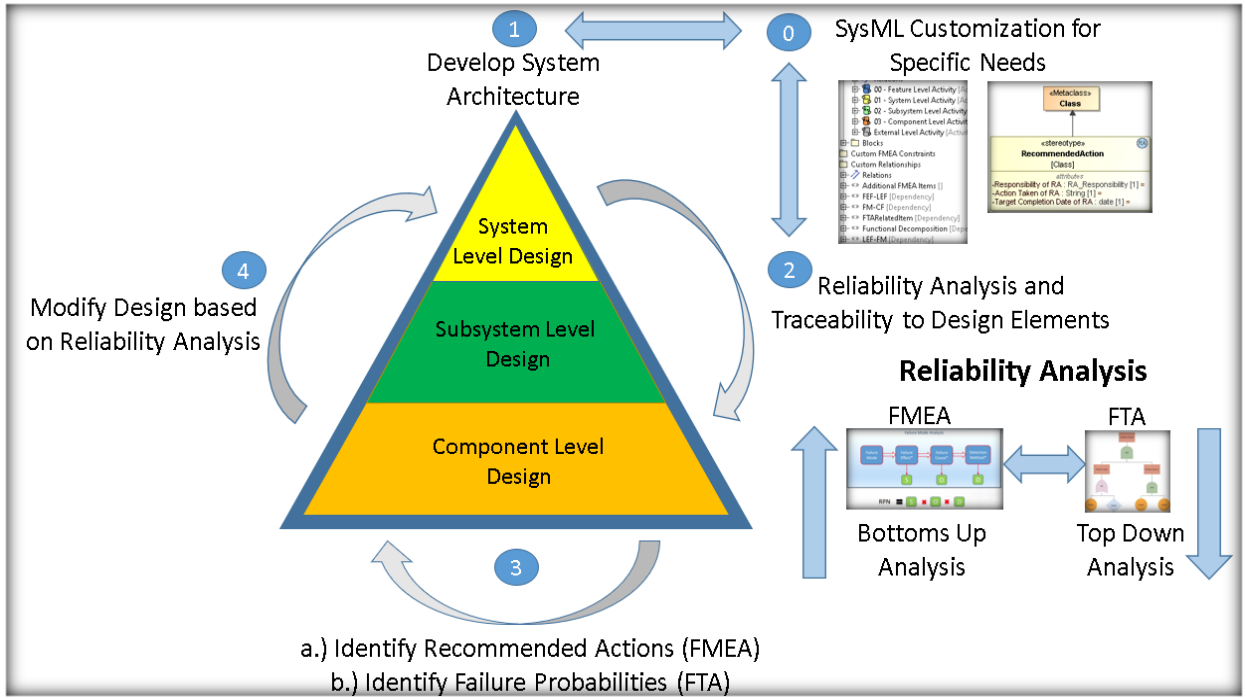
Model-Based Systems Engineering (MBSE) and Model-Based Safety & Reliability Analysis (MBSRA) Integration

- **MBSE for Managing Complexity**
  - MBSE uses graphical models to represent complex systems, improving design quality, communication, and enabling early issue detection throughout development.
- **MBSRA Bridging Architecture and Reliability**
  - MBSRA integrates with MBSE to link system architecture with reliability analysis, enhancing risk assessment, traceability, and validation processes.
- **Information Reusability Across Teams**
  - MBSE and MBSRA promote reuse of system models across teams, improving consistency, collaboration, and ensuring updates propagate effectively.
- **Use of Semi-Formalized Specifications**
  - Leveraging MBSE with SysML and RAAML (Risk Analysis and Assessment Modeling Language) addresses fragmentation by enforcing consistent, structured specifications that integrate system and reliability models for digital continuity.
- **Standardizing Architecture Models**
  - Standardizing system architecture with SysML and RAAML unifies system functions and reliability concerns, enabling coherent analyses and synchronization across diverse engineering teams. **RAAML extends SysML to unify safety and reliability modeling, enabling traceability and integration of safety analyses with system architecture.**



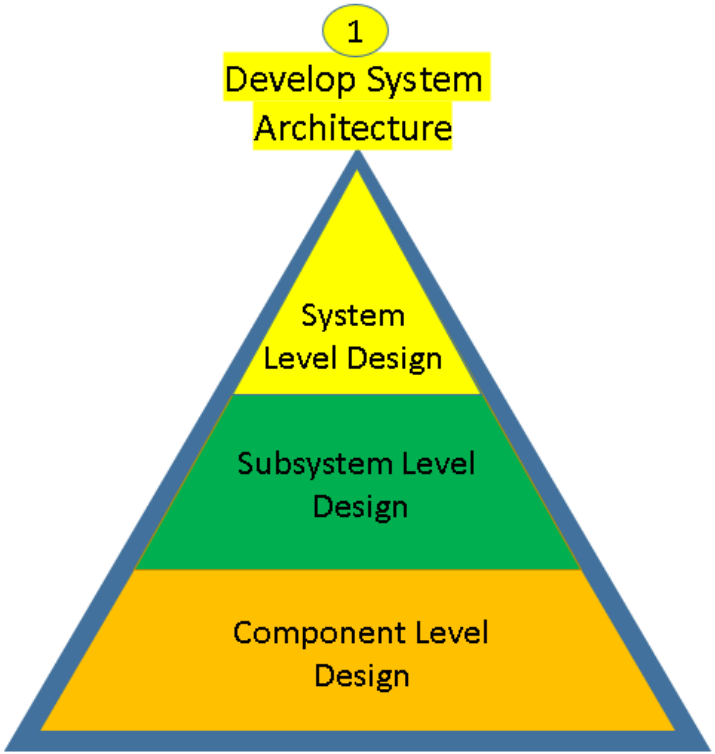
Demonstration/ Methodology of Workflow

- Four-step process embeds reliability into system design
- Encourages completion of each step before advancing, but allows flexibility
- Iterative process supports revisiting steps throughout development
- Promotes continuous refinement and optimization of reliability
- Ensures early and ongoing focus on system robustness and dependability



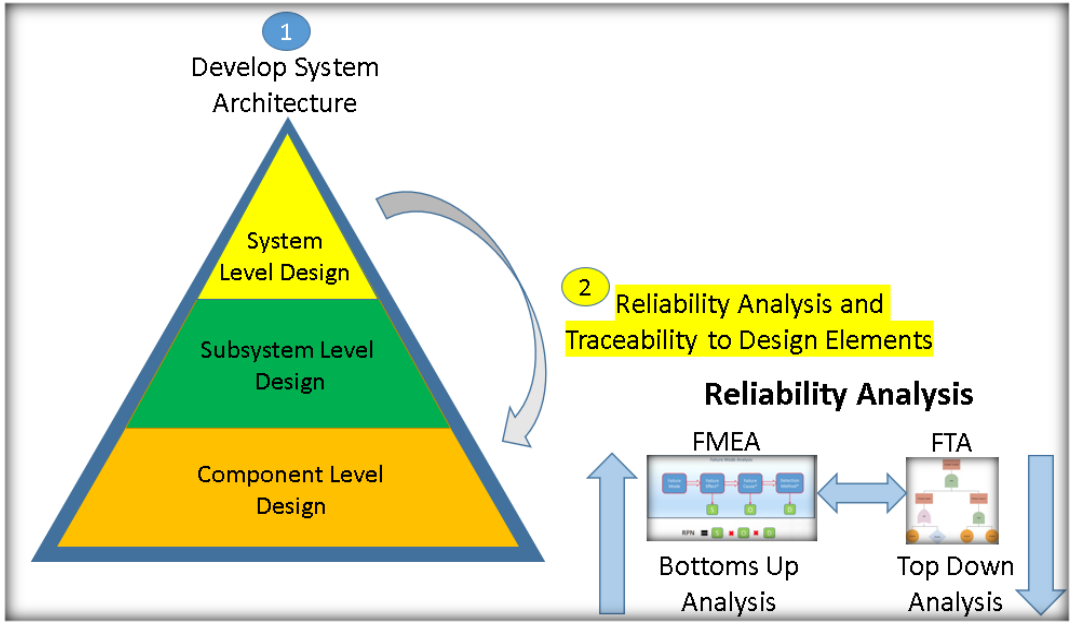
Step 1: Develop System Architecture

- Start with clear understanding of system requirements
- Create a structured framework with multiple architecture levels
- Number of levels should align with organizational needs
- This example uses three levels:
  - System, Subsystem, and Component
- Architecture serves as a communication tool and development blueprint
- Use iterative feedback for continuous refinement and adaptation



Step 2: Reliability Analysis and Traceability to Design Elements

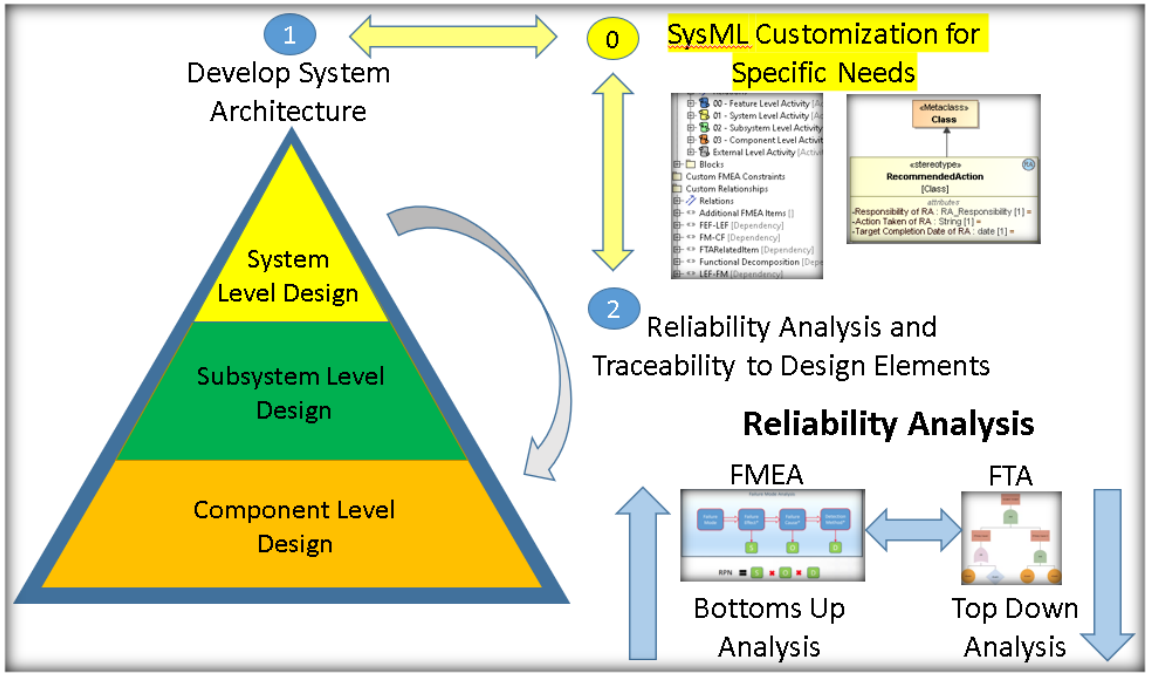
- Ensures alignment between design decisions and reliability/performance expectations
- Maps reliability requirements to specific systems and components
- Supports analysis of how design changes impact reliability
- Enhances failure mode identification and informed decision-making
- Facilitates compliance with industry standards and improves product resilience



- **FMEA (Failure Mode and Effects Analysis)**
  - Starts at component/subsystem level and works upward
  - Identifies potential failure modes and their effects on the system
  - Prioritizes risks based on severity, occurrence, and detectability
  - Helps uncover granular issues often missed in top-down approaches
- **FTA Fault Tree Analysis**
  - Begins with identifying a top-level failure event (“top event”)
  - Analyzes contributing causes using logical gates (AND, OR)
  - Visualizes complex failure pathways for better understanding
  - Supports risk mitigation by highlighting critical failure paths
  - Commonly used in safety-critical industries (aerospace, automotive, nuclear)

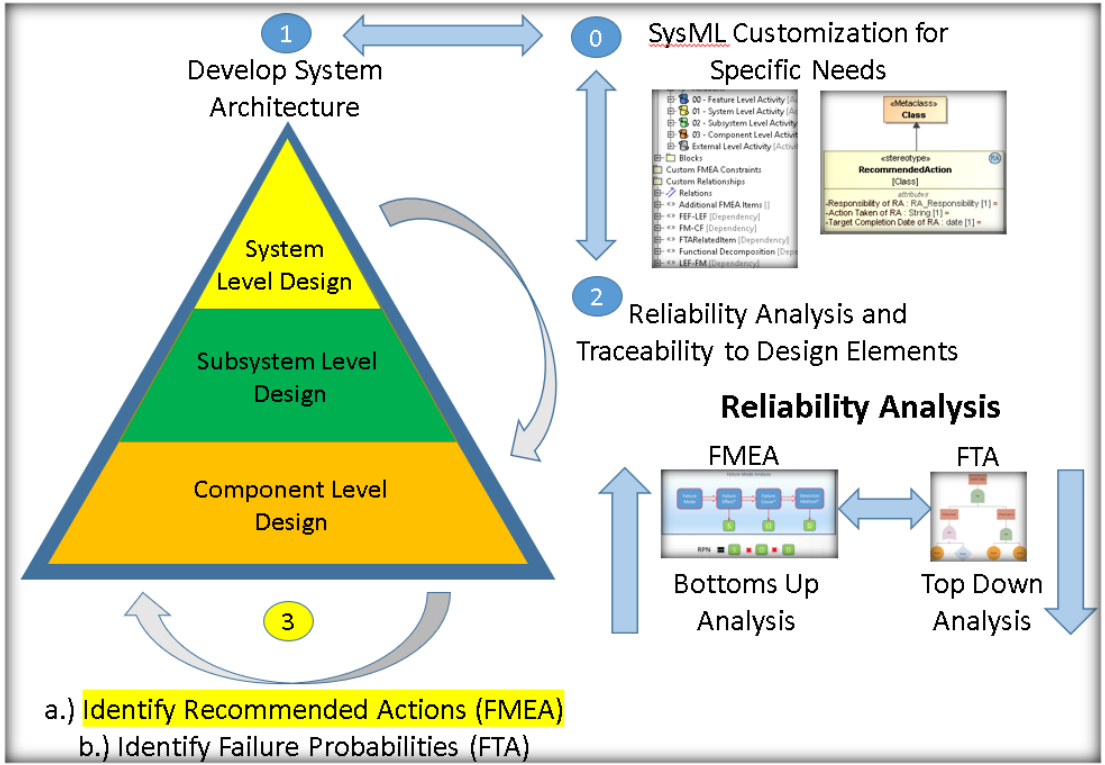
Step 0: SysML Customization for Specific Needs

- Tailor SysML to fit specific project domains (e.g., aerospace, automotive, software)
- Introduce specialized concepts, notations, and semantics
- Improve stakeholder communication with a shared, relevant vocabulary
- Enhance clarity and understanding through domain-specific representations
- Enable model reuse and streamline development
- Support industry standard compliance
- Customize both elements and relationships for system architecture and reliability analysis during design phase



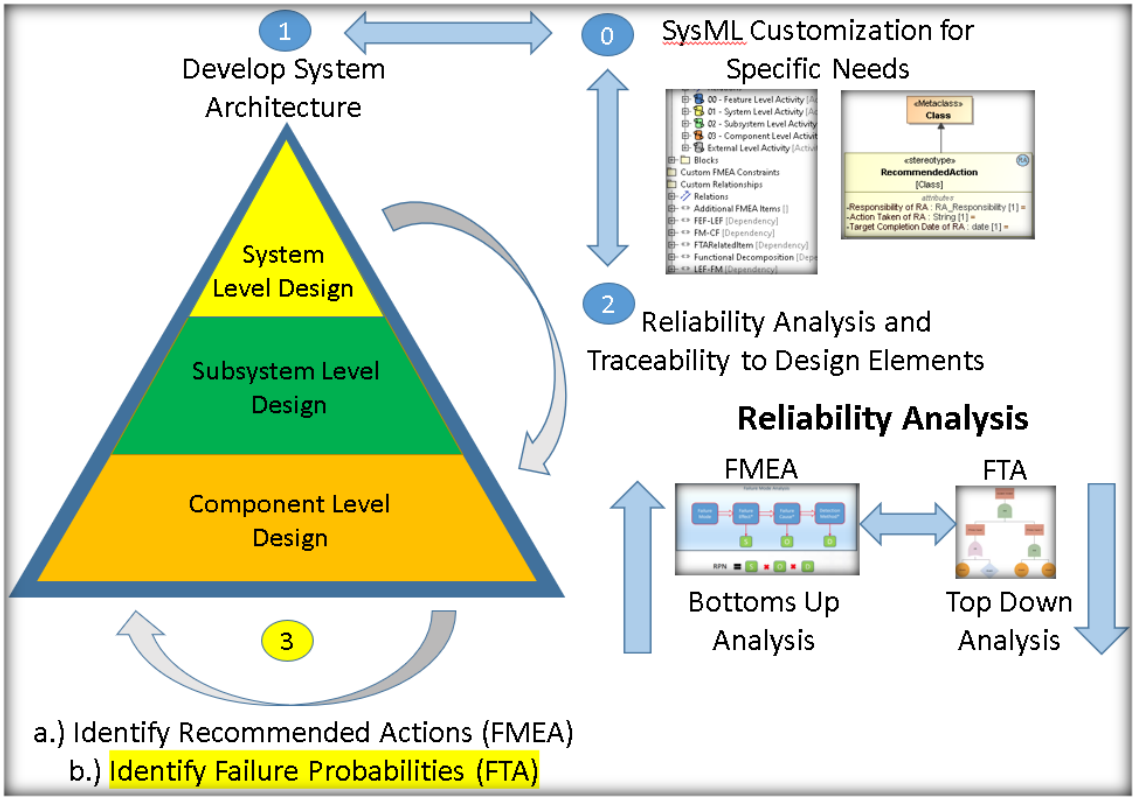
Step 3a: Identify Recommended Actions from FMEA

- Address failure modes to prevent issues before they occur
- Enhance safety and reduce likelihood of costly failures
- Promote clear communication among stakeholders about risk strategies
- Help prioritize resources toward high-impact risks
- Support continuous improvement and higher product quality
- Demonstrate commitment to excellence and customer satisfaction
- Translate FMEA insights into tangible system improvements
- Recommended actions are a critical part of FMEA, aimed at mitigating identified risks and improving system reliability



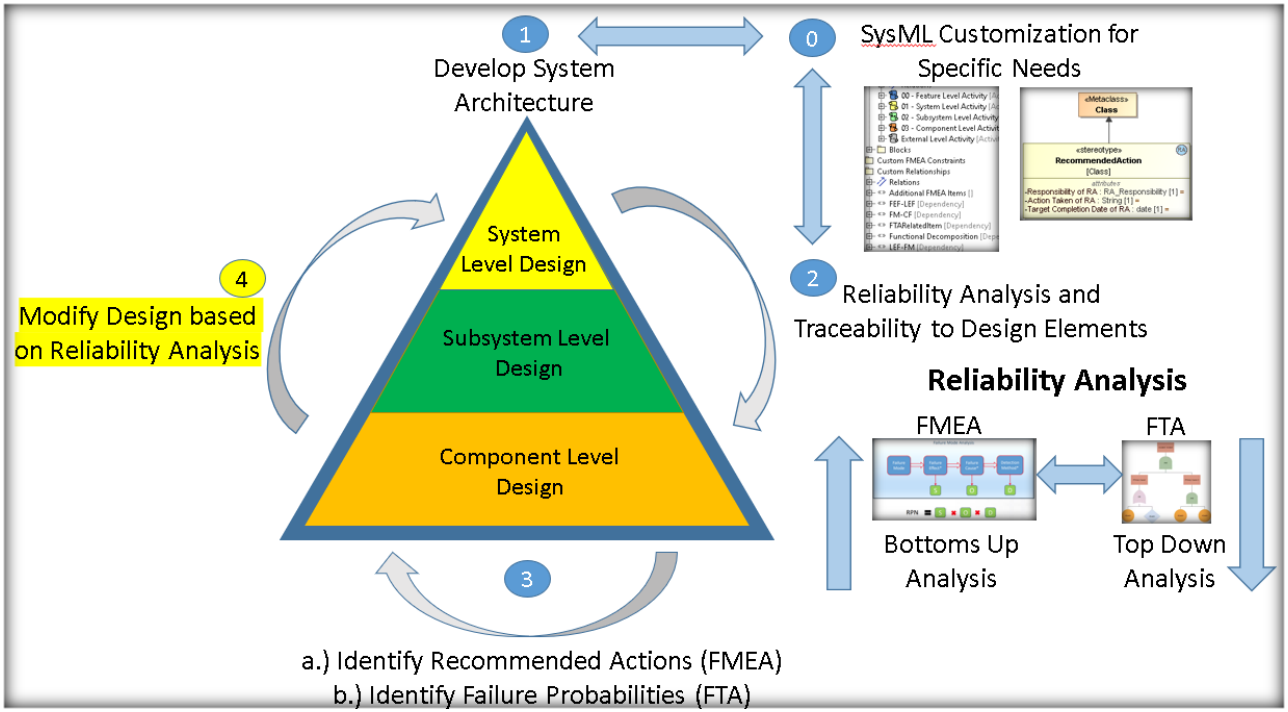
Step 3b: Identify Failure Probabilities from FTA

- Quantify likelihood of failure modes to prioritize critical risks
- Identify system vulnerabilities and necessary safeguards
- Integrate redundancies (e.g., data paths, processing units) to enhance reliability
- Design for resilience and operational continuity under failure conditions
- Improve decision-making through understanding of failure interdependencies
- Support development of robust, efficient, and user-aligned systems



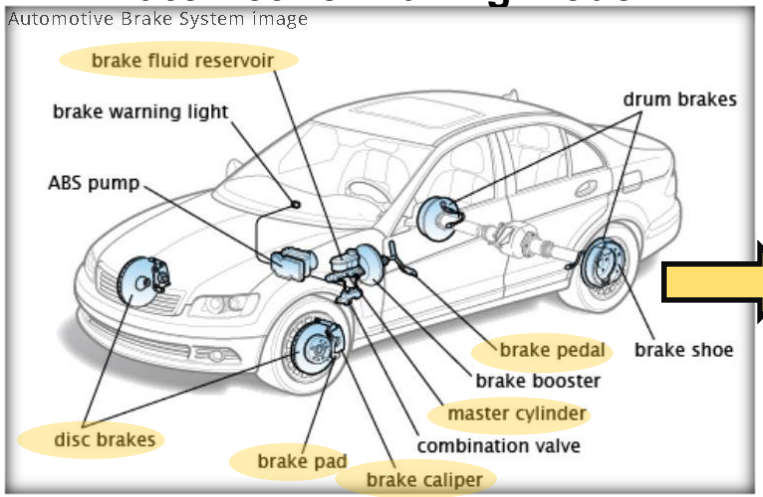
Step 4: Modify Design Based on Reliability Analysis

- Reliability data helps identify weak points and failure-prone areas for targeted improvements.
- Enhancements may include incorporating redundancy to the model architecture to improve reliability.
- Applying insights from reliability analysis ensures compliance with safety and performance standards.
- The result is a more resilient product aligned with market expectations and long-term operational goals.



Example of Applied Methodology

Automotive Braking Model

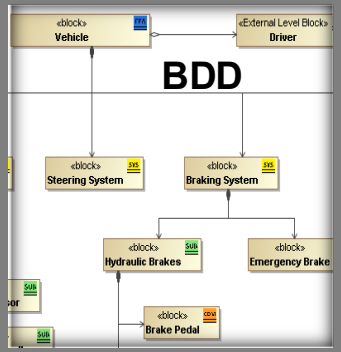


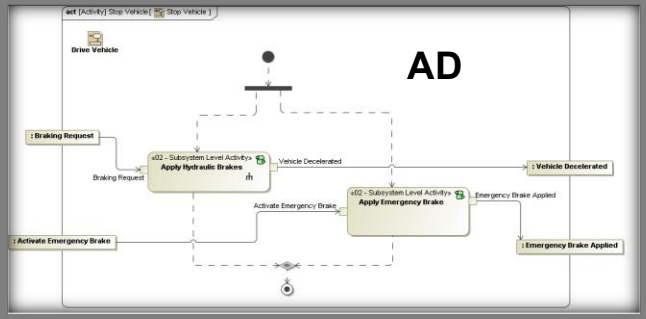
- CATIA Magic used as the SysML tool to develop system architecture
- Demonstrates integration of reliability analysis into architecture
- Actions from FMEA and FTA inform design modifications
- Enhances overall reliability of the braking system model

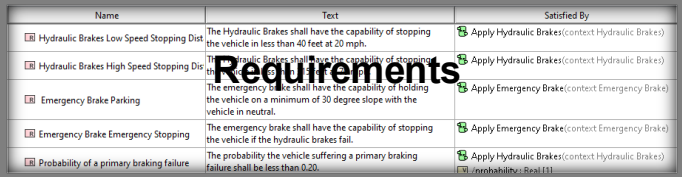
CATIA Magic

OMG SYSTEMS MODELING LANGUAGE SysML

Model Architecture





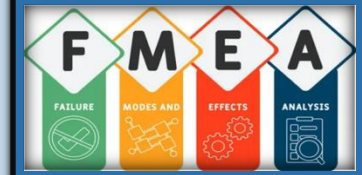


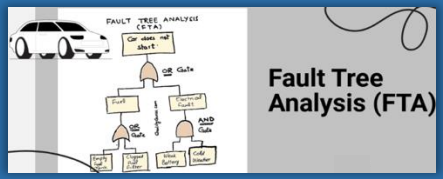
Requirements

Design Modifications

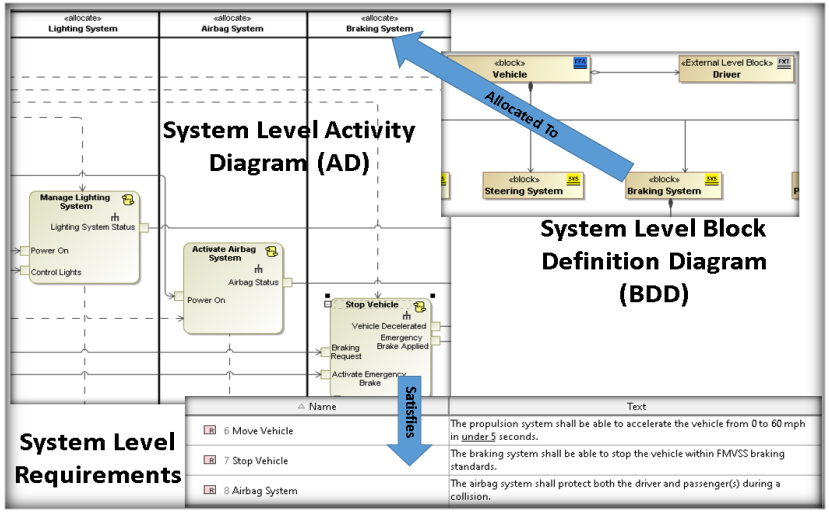
Perform Analysis

Reliability Analysis



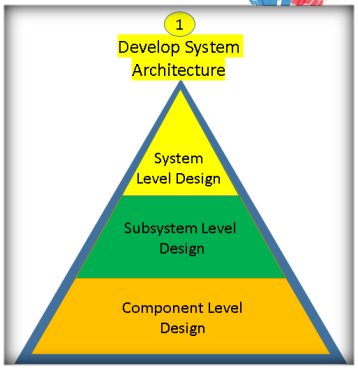


Step 1: Develop System Architecture



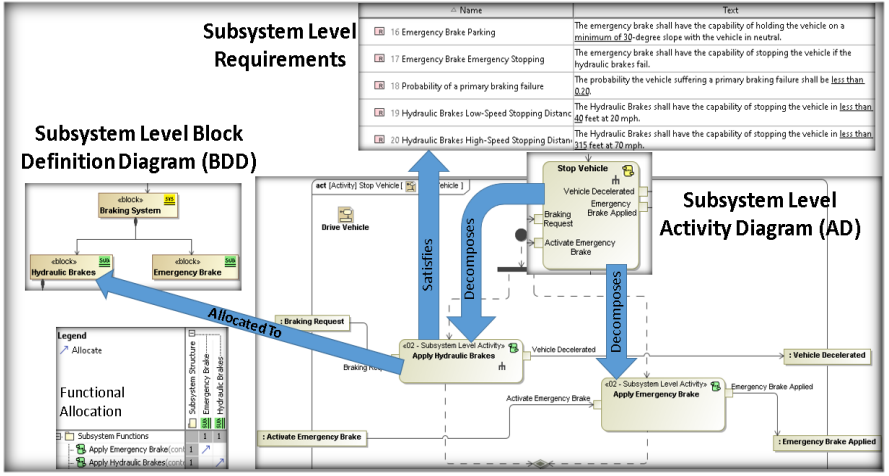
System-Level Design in SysML

- Expands architecture by detailing components, interactions, and behaviors
- Translates conceptual architecture into practical implementation
- Swim lanes represent allocations using blocks from the Block Definition Diagram (BDD)
- Activities fulfill system-level functional requirements
- Demonstrates integration of diagrams within the overall system architecture

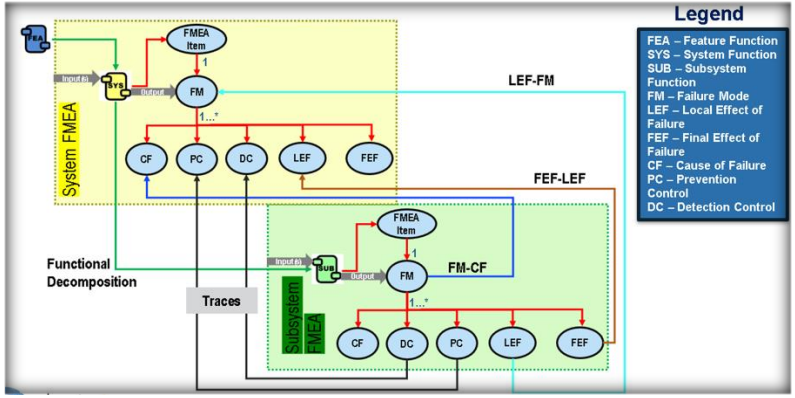


Subsystem-Level Design

- Decomposes system architecture into smaller, functional subsystems
- Each subsystem addresses specific responsibilities within the overall system
- Example (Figure 4): "Stop Vehicle" function split into “Apply Hydraulic Brakes” and “Apply Emergency Brakes”
- Subsystem requirements are derived from system-level requirements
- Block Definition Diagram (BDD) shows subsystem structure and functional allocation



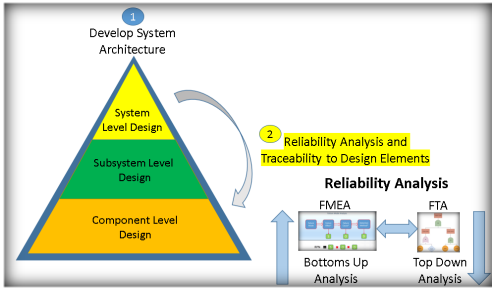
Step 2: Reliability Analysis and Traceability to Design Elements (FMEA)



- Prevention Controls (PC) and Detection Controls (DC) at the subsystem level are loosely linked to their system-level counterparts through a <<Trace>> relationship

3 Key Relationships

- Failure Mode → Cause of Failure (FM-CF): Subsystem failure mode links to system-level cause of failure.
- Local Effect of Failure → Failure Mode (LEF-FM): Subsystem local effect links to system failure mode.
- Final Effect of Failure → Local Effect of Failure (FEF-LEF): Subsystem final effect links to system local effect via decomposition.

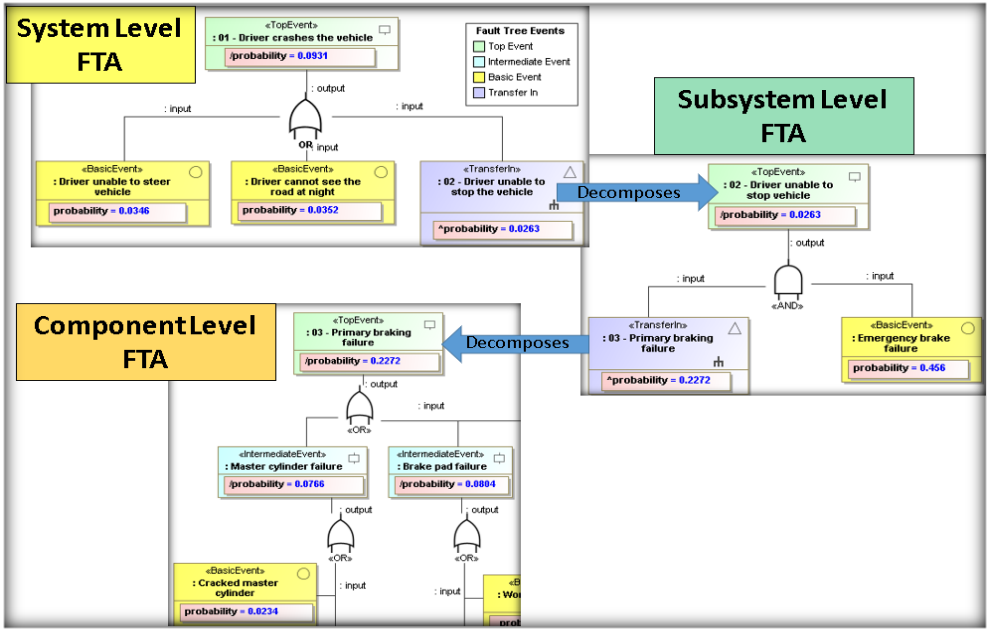


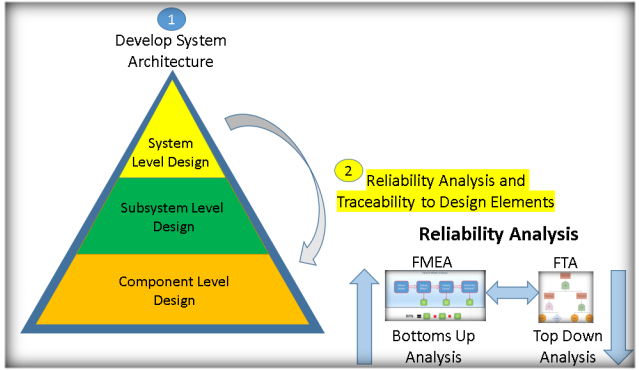
1. Subsystem Local Effect of Failure → System Failure Mode
2. Subsystem Final Effect of Failure → System Local Effect of Failure
3. Subsystem Failure Mode → System Cause of Failure
4. Subsystem Prevention Control → System Prevention Control
5. Subsystem Detection Control → System Detection Control

6. Component Local Effect of Failure → Subsystem Failure Mode
7. Component Final Effect of Failure → Subsystem Local Effect of Failure
8. Component Failure Mode → Subsystem Cause of Failure
9. Component Prevention Control → Subsystem Prevention Control
10. Component Detection Control → Subsystem Detection Control

Item	Failure Mode	Local Effect Of Failure	Final Effect Of Failure	Cause Of Failure	Prevention Control	Detection Control
Stop Vehicle (context Braking System)	Primary braking failure	Driver unable to stop the vehicle	Driver crashes the vehicle	Master cylinder failure Pneumatic brake line failure Caliper failure Brake pad failure Brake pedal failure Rotor failure	Design braking system per automotive safety standard 135	Driver applies the brakes and the system does not respond as expected
Apply Hydraulic Brakes (context Hydraulic Brakes)	Master cylinder failure	Primary braking failure	Driver unable to stop the vehicle	Cracked master cylinder Master cylinder seal leak	Abide by the commercial vehicle safety alliance hydraulic brake system inspection procedures	Brake fluid leak on ground under vehicle Brake pedal goes to the floor when the driver applies pressure
Generate Hydraulic Pressure	Cracked master cylinder	Master cylinder failure	Primary braking failure	High pressure in the system	Design and test master cylinder pressures per automotive standard	Perform pressure test on master cylinder during brake inspection

Step 2: Reliability Analysis and Traceability to Design Elements (FTA)





FTA is a systematic, top-down method for identifying the causes of system failures

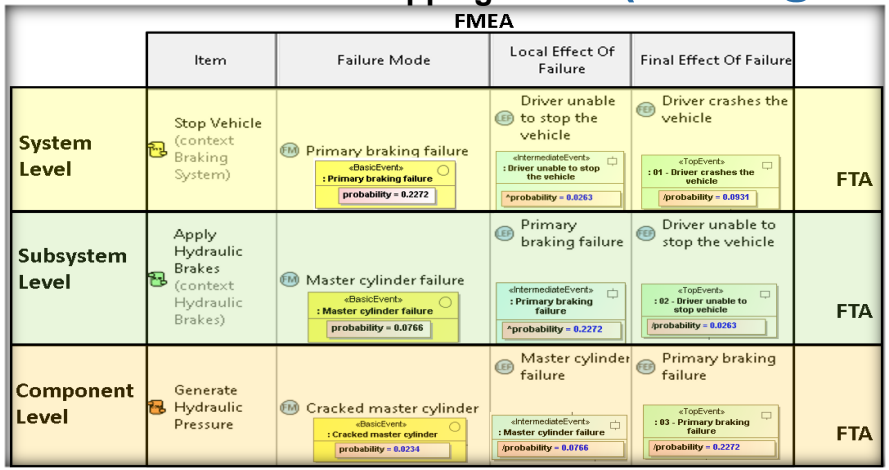
- The process begins by defining the undesired event (top event) to analyze.
- Logic gates (e.g., “AND”, “OR”) are used to represent relationships between causes.
- System Level Top Event: “Driver crashes the vehicle”.
- Subsystem Level Top Event: “Driver unable to stop the vehicle”.
- Component Level Top Event: “Primary braking failure”.
- Decomposed down to the basic event: “Cracked master cylinder failure”.

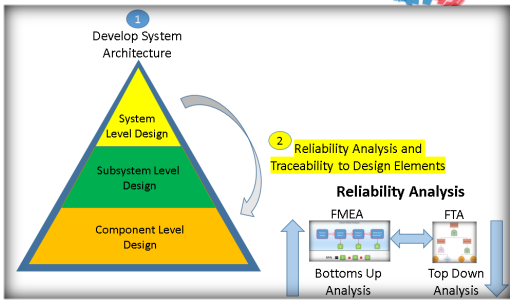
Simulation and Analysis

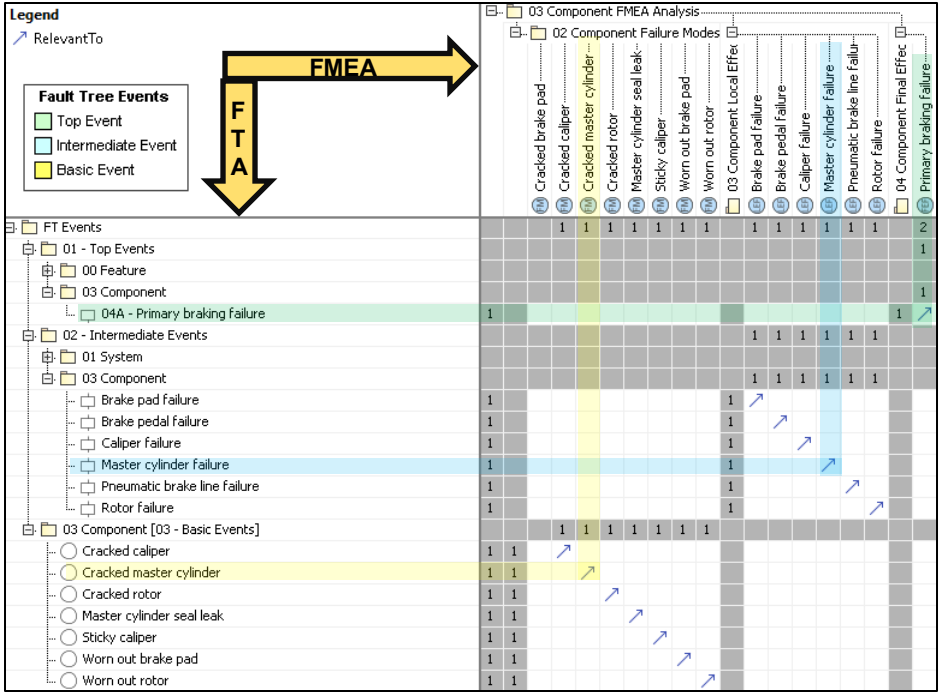
- Intermediate and top event probabilities are calculated using built-in simulation equations within the tool.
- Calculations are based on user-entered default values for basic events.

Step 2: Reliability Analysis and Traceability to Design Elements (Linking FMEA and FTA)

FTA-FMEA Element Mappings



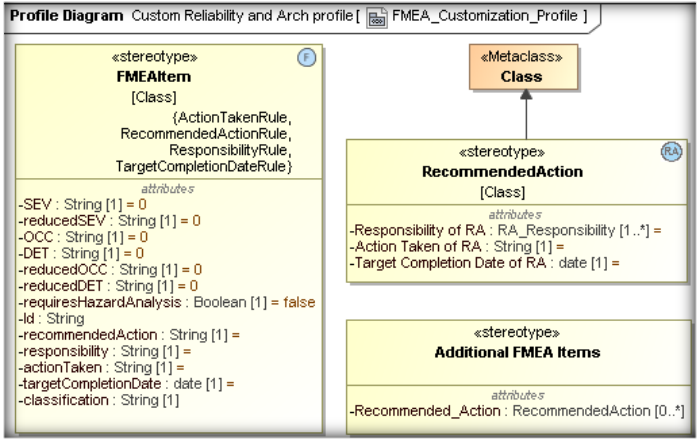




Custom stereotypes are necessary due to limitations in standard SysML for full traceability.

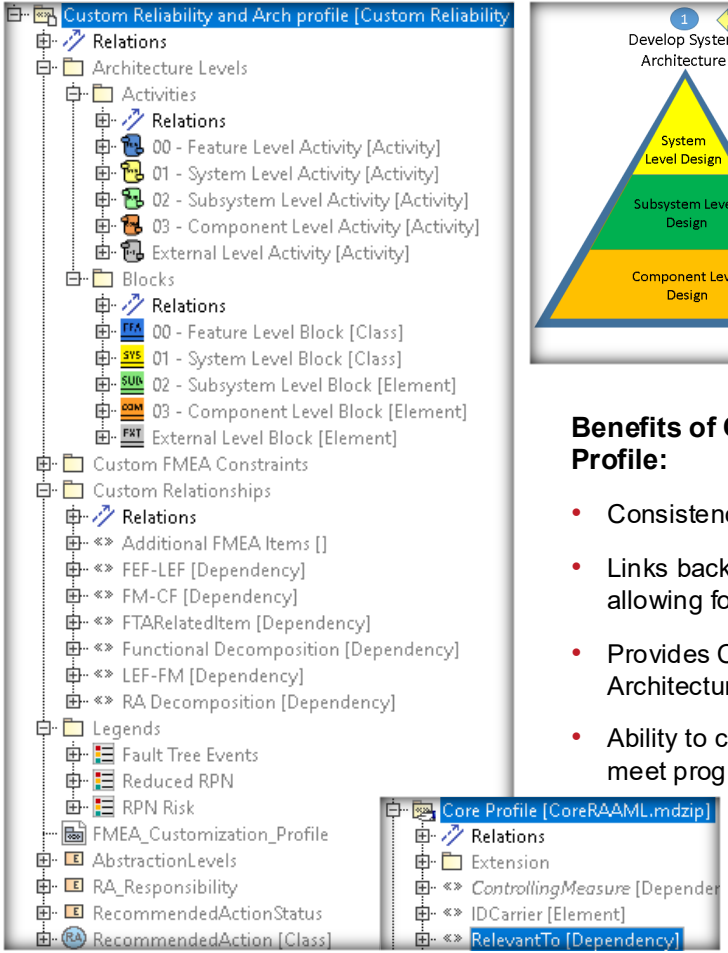
- "RelevantTo" is a custom SysML stereotype that links FMEA and FTA elements, and is defined in the RAAML Specification
- FTA Top Event ↔ FMEA Final Effect of Failure
- FTA Intermediate Event ↔ FMEA Local Effect of Failure
- FTA Basic Event ↔ FMEA Failure Mode
- Primary braking failure appears at all levels (system, subsystem, component)
- Maintains consistent probability values across levels for accurate analysis and traceability

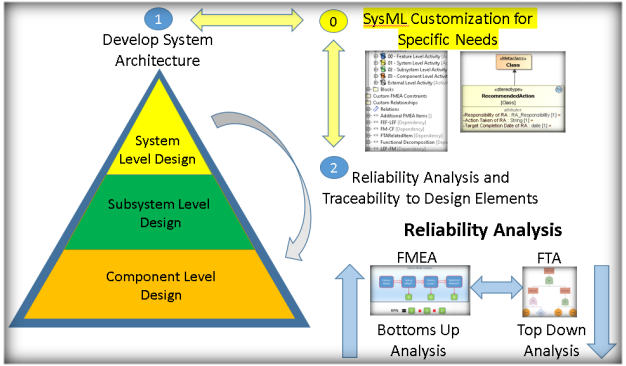
Step 0: SysML Customization for Specific Needs



Customized Profile used in this automotive braking model example

- Architecture Levels for Activities and Blocks
- Custom Relationships: FEF-LEF, FM-CF, LEF-FM, FTARelatedItem, etc.
- Legends: Fault Tree Events, RPN, and Reduced RPN
- Custom FMEA Constraints/ Rules
- Custom Elements: Recommended Action Element

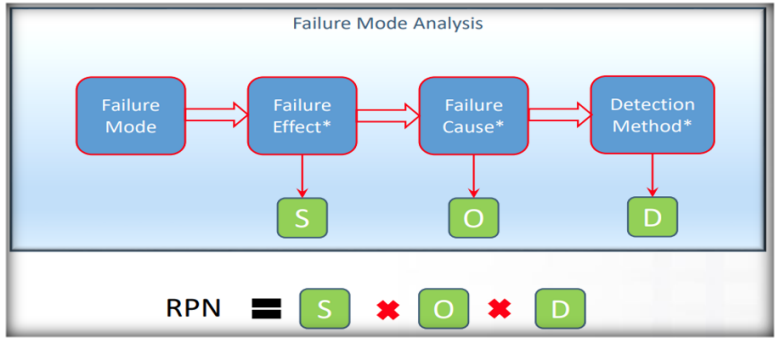




Benefits of Creating Customized Profile:

- Consistency and Reuse across projects
- Links back into the Model Architecture allowing for gap analysis
- Provides Clarity at all Levels of Architecture
- Ability to create custom elements to meet program/ organization needs

Step 3a: Identify Recommended Actions (FMEA)



- **FMEA Process for Calculating Risk Priority Number (RPN)**

1.) Start by identifying failure modes:

2.) Assess final effects of each failure mode, consider impacts on performance, safety, and functionality.

3.) Assign a severity rating (scale 1–4), where 4 = most severe.

4.) Determine root causes of each failure mode, evaluate all possible causes.

5.) Assign an occurrence rating (scale 1–5), where 5 = most likely to occur.

6.) Evaluate detection controls, include measures like testing and inspections.

7.) Assign a detection rating (scale 1–5), where 1 = easiest to detect.

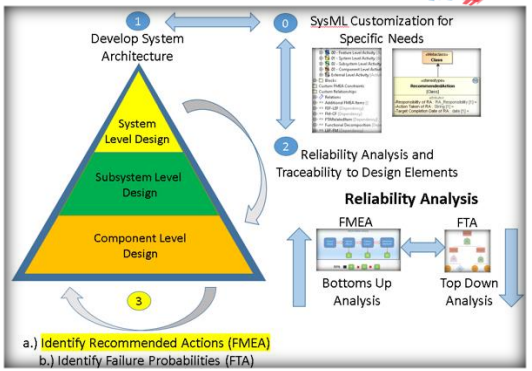
8.) Calculate RPN using the formula: RPN = Severity × Occurrence × Detection

9.) Use RPN values to prioritize risks, Higher RPN = greater risk

10.) Focus mitigation efforts on high-RPN failure modes.

Recommended Actions for "Stop Vehicle" System Function

- The RPN value of 80 indicates a high-risk failure mode, prompting the need for recommended actions.
- Recommended Action (RA4) Calls for additional redundancy in the braking system to improve safety.
- Action Taken: Modify the system architecture to include the downshift capability of the transmission to help slow and stop the vehicle.
- Based on insights from FMEA, a System Architecture Design Change is required

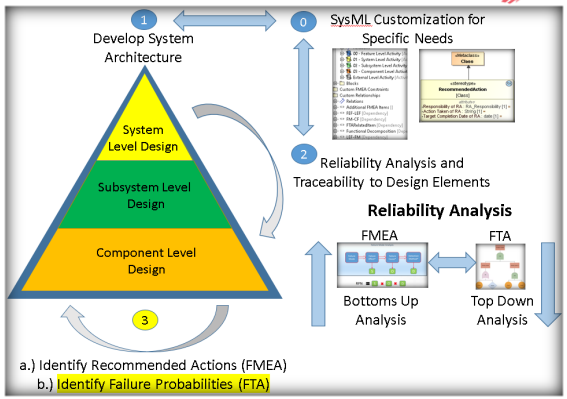


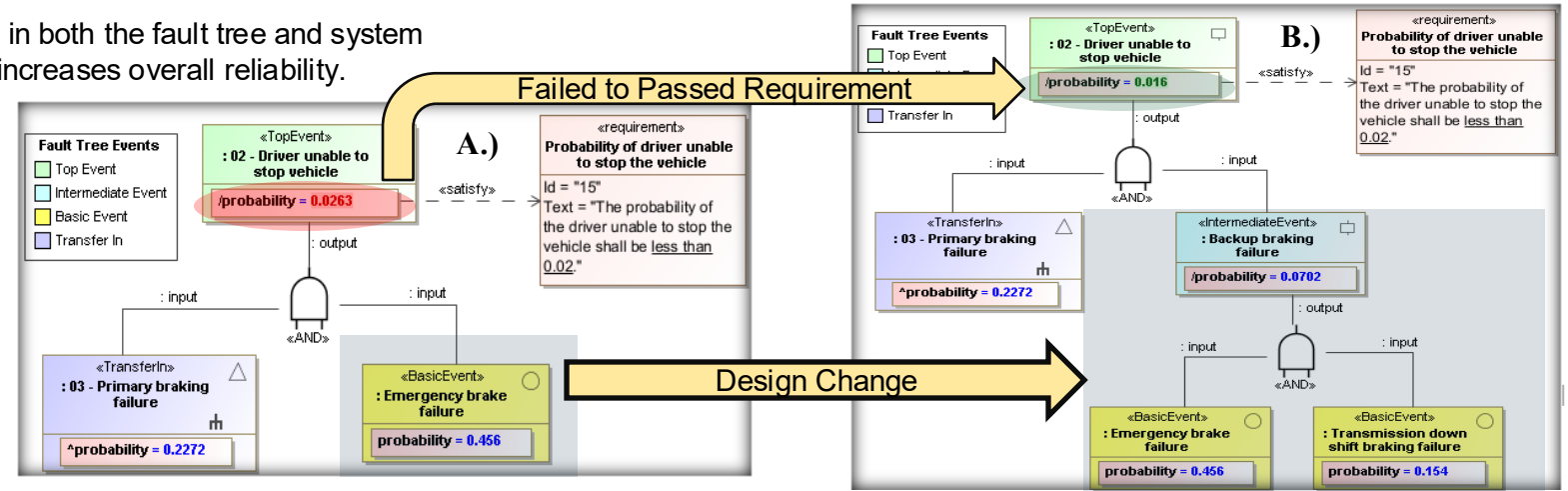
Item	Failure Mode	RPN	Recommended_Action	Action Taken of RA
Stop Vehicle (context Braking System)	Primary braking failure	80.0	RA1 Install sensor on caliper and display notification to driver on display when unusual wear is occurring between the brake pads and rotors	Installed sensor on calipers to sense undesired friction between pads and rotors. Also included an indicator light on the cluster to warn driver of a possible issue.
			RA3 Display notification for driver to bring in vehicle for maintenance check which includes inspecting and replacing brakes as necessary	Programmed the vehicle to display a notification on the display for when to bring the vehicle in for maintenance.
			RA4 Add an emergency braking capability that does not include the emergency brake or primary braking capabilities	Use the down shift ability of the transmission to stop the vehicle if the primary and emergency brake fail
			RA2 Install OEM brake pads and rotors	Install OEM brake pads and rotors

Step 3b: Identify Failure Probabilities (FTA)

- FTA allows the simulation of failure probabilities for intermediate and top-level events.
- Requirement values can be linked to derived probabilities in the fault tree
- Enables verification of whether system requirements are met.
- Helps identify how to optimize the allocation of failure probabilities.
- A key strategy is to add redundancy to the system to improve probability outcome
- Redundancy in both the fault tree and system architecture increases overall reliability.

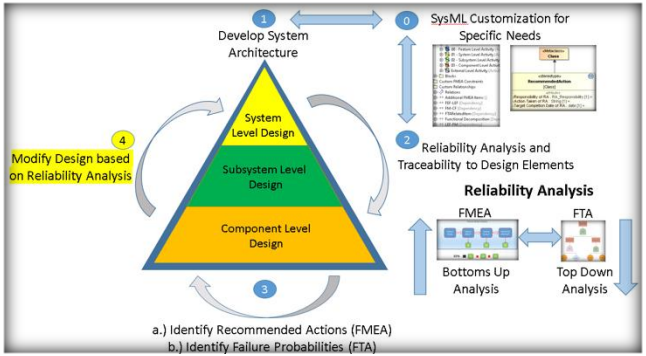
- Design Evolution (Figures A & B)
- Figure A.) Original design includes emergency brake failure and primary braking.
- Figure B.) Updated design adds transmission downshifting as redundancy.
- This improves the fault tree outcome, now meeting the driver stopping requirement within a safe probability range.
- These architectural changes need to be propagated throughout the system model, as outlined in Step 4 of the process.

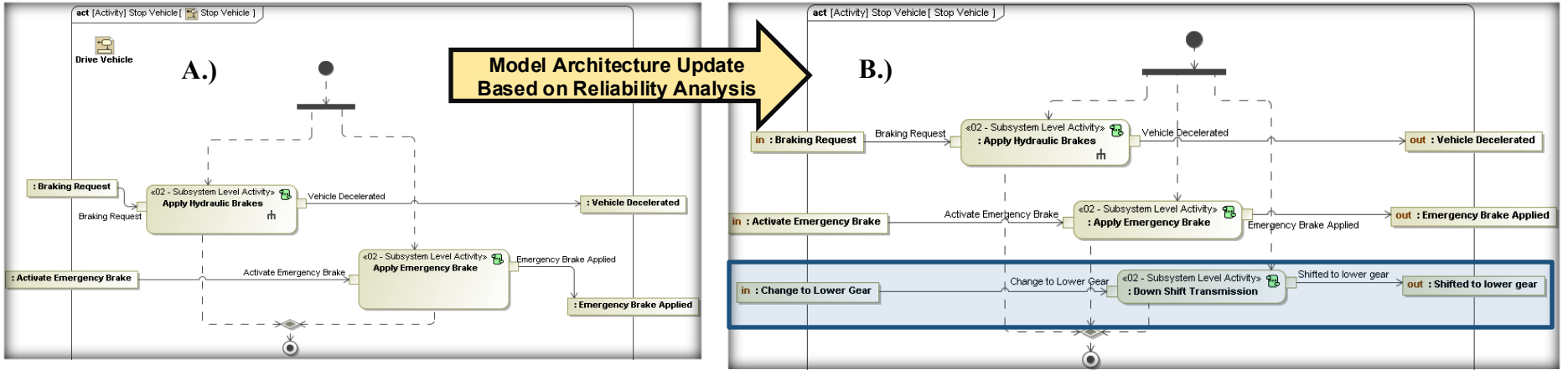




Step 4: Modify Design based on Reliability Analysis

- Enhancing System Architecture with FMEA & FTA Insights
- Use FMEA recommended actions and FTA failure probabilities to refine system architecture
- Integrated analysis offers a complete view of system reliability
- Enables easy traceability of impacted elements when requirements change
- Improves design robustness and adaptability
- Recommended Action: Add transmission downshifting as backup braking method
- Made Recommended Action update to FTA, and the probability now passes the Requirement
- Original subsystem activity diagram (Figure A): Only included hydraulic and emergency brakes
- Updated diagram (Figure B): Includes transmission downshifting, enhancing overall system reliability





Conclusions/ Future Work

- Model-Based Safety & Reliability Analysis (MBSRA) bridges Architecture and Reliability enhancing risk assessment, traceability, and validation processes.
- Promotes continuous refinement and optimization of reliability
- This integration improves understanding of risk interdependencies, enhances traceability, continuity, standardization, and strengthens communication across multidisciplinary teams.
- Future Work:
  - More advanced metrics like Mean Time Between Repairs (MTBR) and Mean Time to Repair (MTTR) can be derived through FTA, specialized tools (e.g., ReliaSoft BlockSim) may be required for these calculations. Integrate back into model architecture.
  - SysMLv2 and RAAML Integration

Meet Dassault Systemes Team at IS2025!

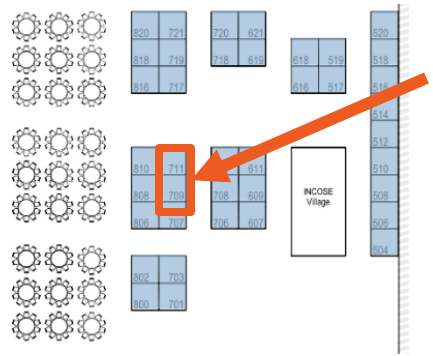


Exhibit Hall

Booth 709-711

QR code with Session Details

Track	Day	Start	End	Room	Type	Dassault Systemes's Sessions at IS25
1.1.1	Mon	10:00	10:40	Hall 3	Presentation	Case Studies for Querying the Model - **SysML V2**
1.7.1	Mon	11:00	11:20	201	Presentation	Exploring the Next Frontier: **SysML V2**
1.1.3	Mon	11:30	12:10	Hall 3	Paper 185	Exploring the Use of **SysMLv2** for Solution Architecture Development with the MagicGrid Framework
1.2.3	Mon	11:30	12:10	214	Paper 320	Towards a **Digital Engineering Ontology** to Support Information Exchange
2.4.1	Mon	13:30	14:10	215	Paper 340	**Systems Engineering with Attitude**
2.4.2	Mon	14:15	14:55	215	Presentation	Taming the Beast: Best Practices of Extending **SysML V2**
4.7.1	Tue	10:00	10:20	201	Presentation	**Digital Engineering and MBSE with Virtual Twins**: Streamlining Robotic Arm Design and Deployment
5.3.1	Tue	13:30	13:55	213	Paper 26	**Systematic Risk Analysis**: FMEA and FTA Approaches for Multi-Level System Architectures
5.3.2	Tue	14:00	14:25	213	Paper 270	SysML4Sec – Methodology for **Security modeling** in the context of large-scale product development with multiple design levels
5.3.3	Tue	14:30	14:55	213	Paper 147	A System-of-Systems Modeling, Simulation and Data Analytics Framework for Resilient **Sustainment and Support Readiness** Strategies
6.5.3	Tue	16:30	16:55	208	Paper 128	Model-Based Systems Engineering **for Industrial Systems**
7.2.1	Wed	10:00	10:40	214	Paper 361	A Transformative Process for **Model-Based Design Reviews**
8.1	Wed	13:30	14:55	Hall 3	Panel	Bridging the Divide: **Linking Architectural Specification and Verification** by System Simulation
9.1	Wed	15:30	16:55	Hall 3	Panel	Cost Impacts of **Generative AI** in Systems Engineering Processes
9.5.2	Wed	16:00	16:25	208	Paper 30	Navigating Innovation: **MBSE Adoption** at Turkish Aerospace Industries
9.5.3	Wed	16:30	16:55	208	Presentation	**Configuration Management** Challenges in Multi-Team Collaboration Using Linked Models
10.2.1	Thu	10:30	11:10	214	Paper 164	**Enterprise Transformation** Planning with UAF
11.5.3	Thu	14:00	14:25	208	Paper 108	Integration of MBSE and **Agile** Development by Seamlessly Creating **Test Plans from Model Simulations in SDV Development**

35th Annual INCOSE international symposium

hybrid event

Ottawa, Canada
July 26 - 31, 2025