Model-based Specification of safety-critical Rail Systems

Randolf Berglehner, DB Netze AG
Andreas Korff, Atego Systems
Agenda

- Project Introduction: CCS Strategy Neupro
- History: Document-based Approach
- Next Step: Requirements Management
- Model-based Approach to formalize
- SysML
- Validation Rules / Artisan Studio Reviewer
- State-based Simulation with Automatic Code Synchronization
- SysML-based Simulation (Artisan Studio SySim)
- Future: From Interface models to whole system models
Modular target architecture for electronic interlockings: standard interfaces

Status of the works:
- Open specification of interfaces (=without supplier IPR) developed
- Unified communication protocol (RaSTA) for all interfaces defined
- Reference implementation of SCI-RBC, SCI-LX and SCI-ILS under contract
- Supplier under contract will deliver interface and test specifications, the other suppliers will validate
Experts specify in documents, how a new system or version of a system should

- Comply with standards
- Behave
- Interface to other systems
- Structured internally

Side effects:
- Huge number of document references
- Acceptance against these documents
Challenges

- Amount of information and references
- Levels of abstraction often mixed
- Inconsistencies possible
- Implicit knowledge in the head of the experts
- Aging of documents
- Propagation of changes
- No formal interface definition leads to missing interchangeability
Specifications in an RM Tool

- Atomic requirements
- Traceable references
- Possibly annotated with diagrams

=> Still not formalized, but textual information
Introduction of Model-based views

Incremental use of SysML:

- First textual information is annotated with diagrams
- Then diagrams are leading
  - In case of discrepancies, the model information „wins“
- More and more the model leads: Visual Modeling to clarify the requirements
What is OMG SysML™?

A graphical modeling language in response to the UML for Systems Engineering RFP developed by the OMG, INCOSE, and AP233

- a UML Profile that represents a subset of UML 2 with extensions

Supports the specification, analysis, design, verification and validation of systems that include hardware, software, data, personnel, procedures, and facilities

Provides model and data interchange via XMI and the AP233 standard
Nine SysML Diagram Types

- Structural Diagrams
  - Package Diagram
  - Block Definition Diagram
  - Internal Block Diagram
  - Parametric Diagram

- Requirement Diagram

- Behavioral Diagrams
  - Use Case Diagram
  - State Machine Diagram
  - Activity Diagram
  - Sequence Diagram
SysML explained by its four Pillars (INCOSE)
3 of 4 Pillars of SysML, according to INCOSE:

- System Structure (BDD, IBD)
- System Behavior (UC, SEQ, SM, ACT)
- System Requirements (REQ)

Parametric View currently not needed
- As Focus is logical behavior and interface structure
Static Model Analysis

Artisan Studio Reviewer
- Web-site Style Output

VBS-based Reviews
- Checks against SysML Language Rules
- Checks against best Practice
- Custom Checks against NeuPro rules
Model Validation Step 1

Hierarchy of State Machines

Simulation Executable generated with C++ as Action Language on Windows

- Multi-threaded to simulate communication partners

Sequence Diagrams define Test Scenarios
Execution of Scenarios against Simulation with State Machine Animation
Model Validation Step 2

Simulation Setup using standardized I/O and defined in Simulation IBDs

Connector-based communication

Execution Generation via VB.NET
- Atego Structured Action Language (ASAL) and VB as Action Language
- Windows Executable with Domain-specific Front-End

Domain Experts can validate without analyzing complicated State Machines

- Simulation Logging into MS Excel
Next Steps

From Interface modelling to modelling the complete Interlocking System