MetaCase

Automating Safety Engineering with Model-Based Techniques

16 March, 2016

Juha-Pekka Tolvanen
jpt@metacase.com

Agenda

- Motivation
- A model-based approach
- Examples
- Demonstration
- Q&A

Motivation

- Safety engineering is quite expensive and tedious
  - Requires considerable about of manual work
  - Scales badly to larger systems

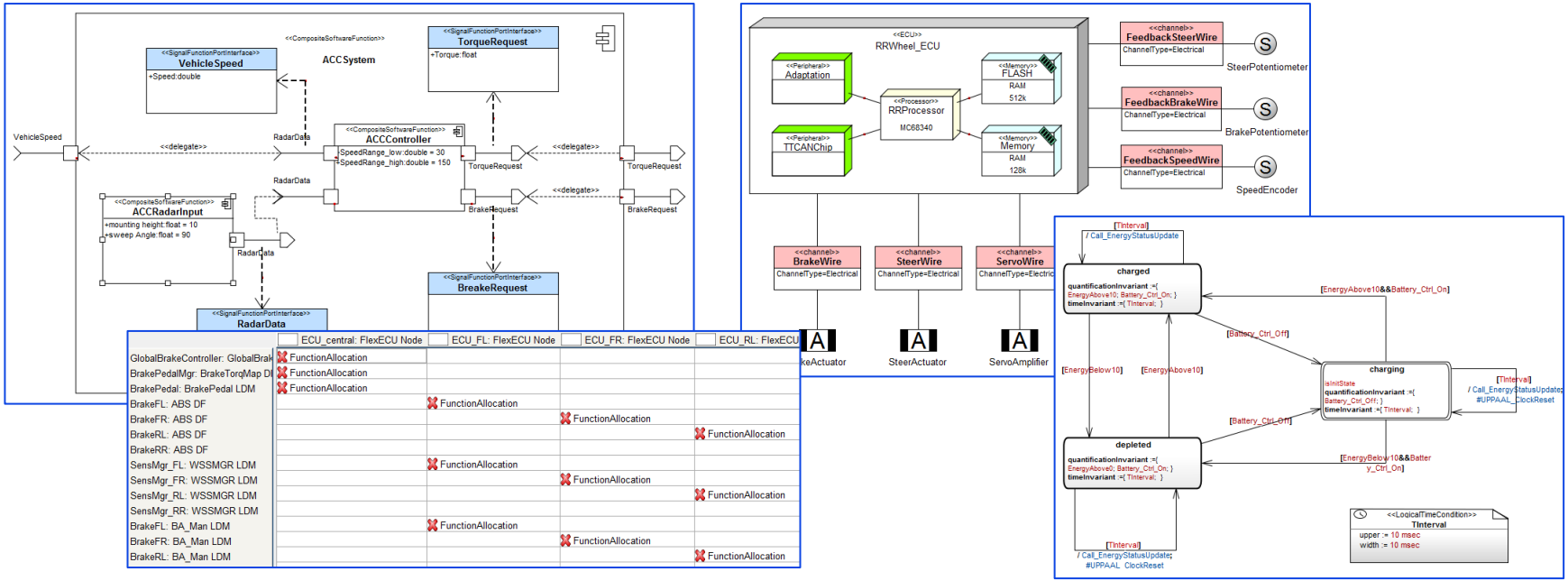
- Feedback to system and software design could be improved
  - Safety engineering flows do not always acknowledge typical iterative/incremental development approach

Model-based approach:

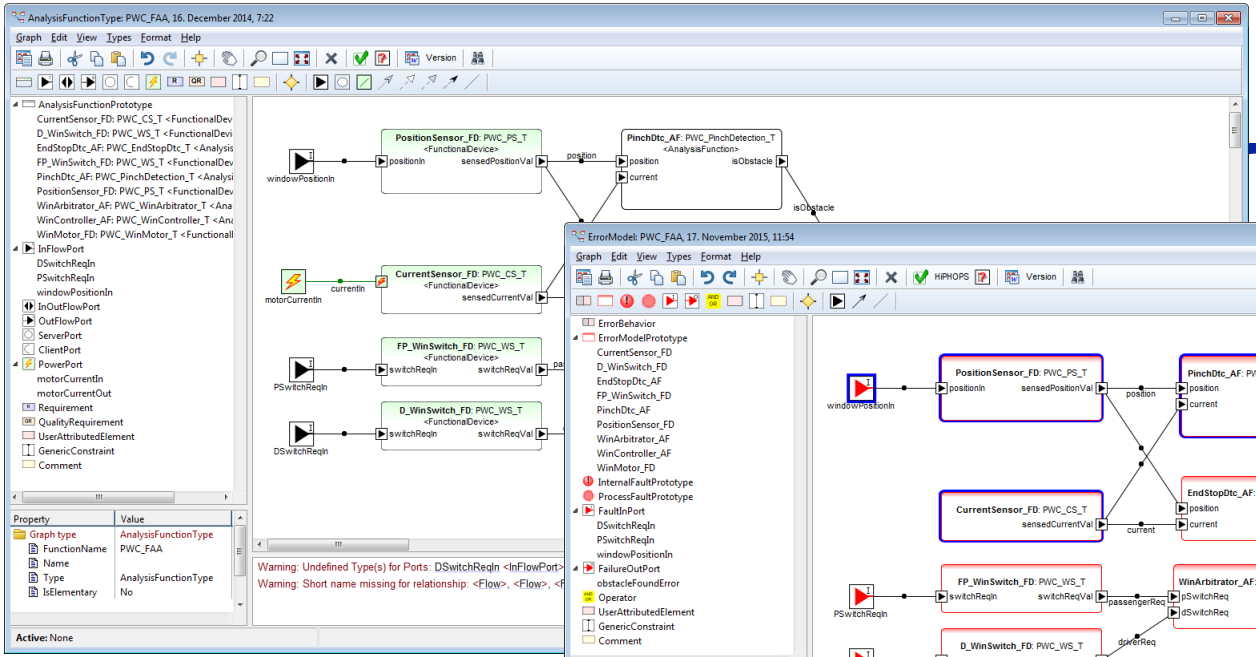
1. Utilize existing specifications with model transformations
   - Analyzes must be related to what is developed (or planned to be developed – early stages)
   - Usually such nominal specifications already exists
2. Apply directly safety concepts in models
   - Safety standards suggest already now own terminology
3. Link safety related models to analytical tools
   - Use models created (automatically) with various analysis tools
   - Different tools for different purposes

Existing system design as a basis

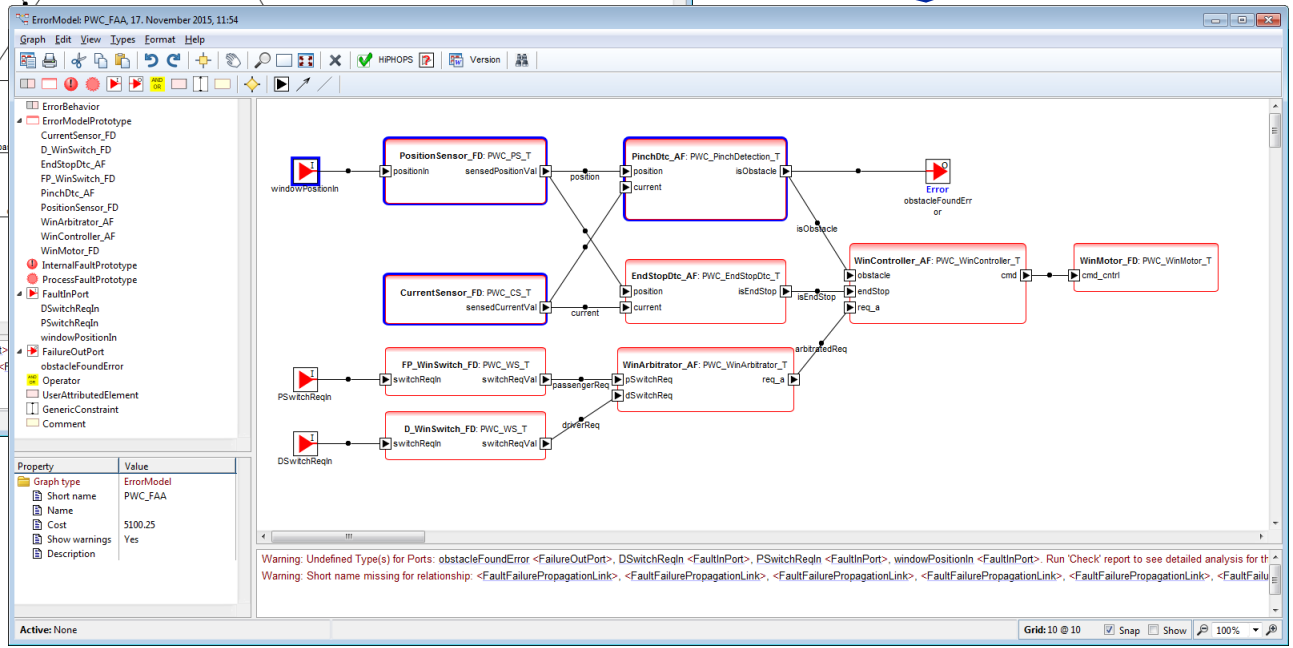
- Usually some designs or specifications already exist, e.g. logical functions, hardware specification, behavior etc.



1) Utilizing existing specifications

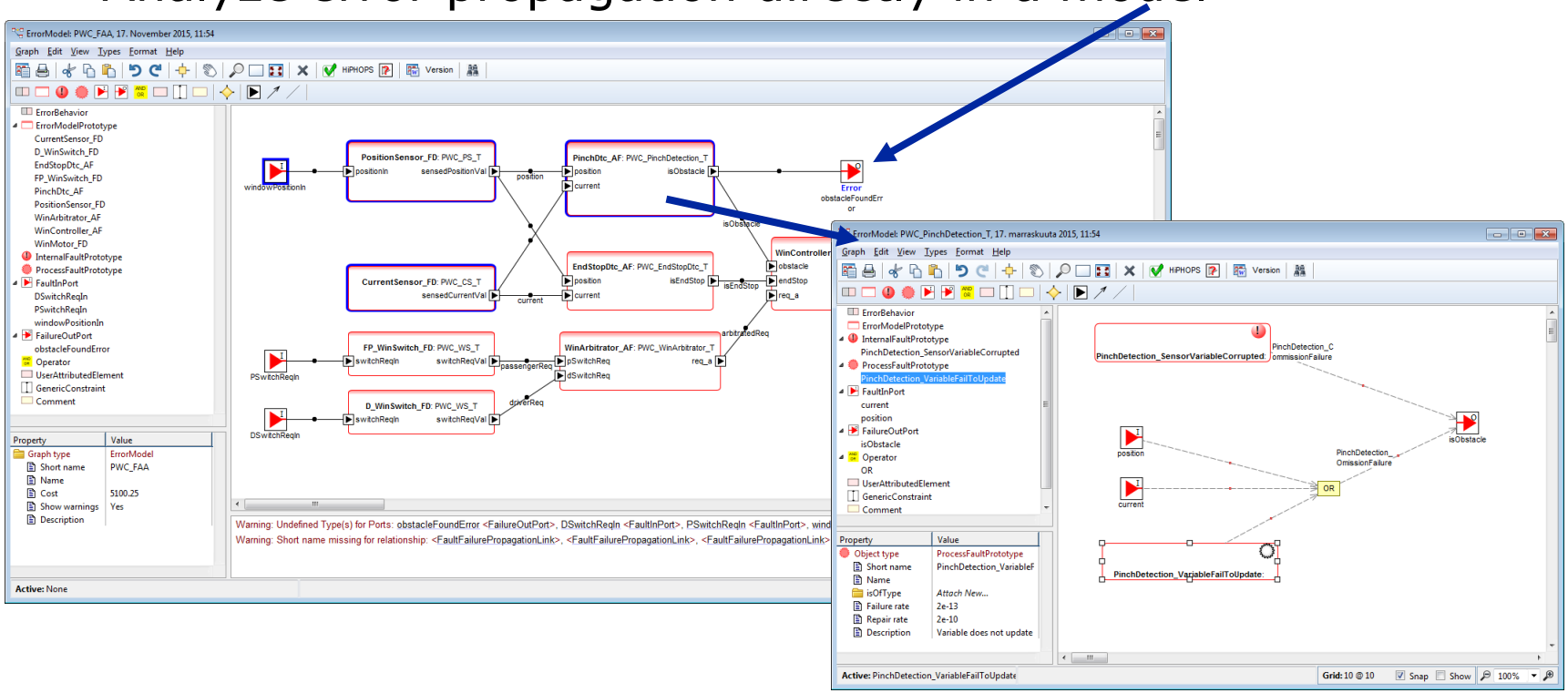


Model transformation in MetaEdit+ tool



Error logic – partly generated

- Analyze error propagation directly in a model



ISO 26262 from 10.000 feet

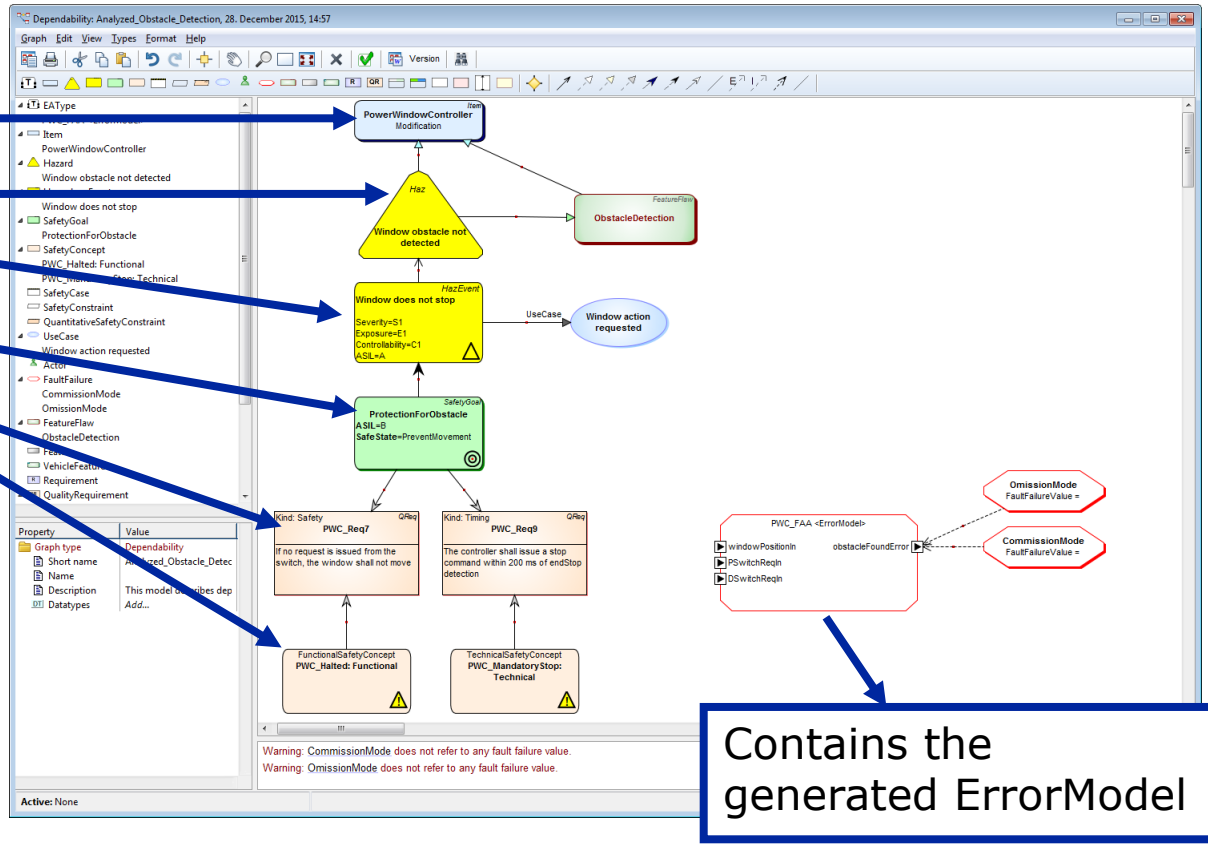
- Define the **item** (functions) and preliminary architecture
- Determine how the item can **fail** (HAZOP or FMEA)
- Determine the driving scenarios that make the failures **hazardous**
- Determine the **exposure** (E) to the **hazard** based on the driving scenario
- Evaluate the **severity** (S) of the hazard
- Evaluate the **controllability** (C) by the operator
- Calculate the ASIL
- Verify your E and C assumptions

ISO 13849-1 from 10.000 feet

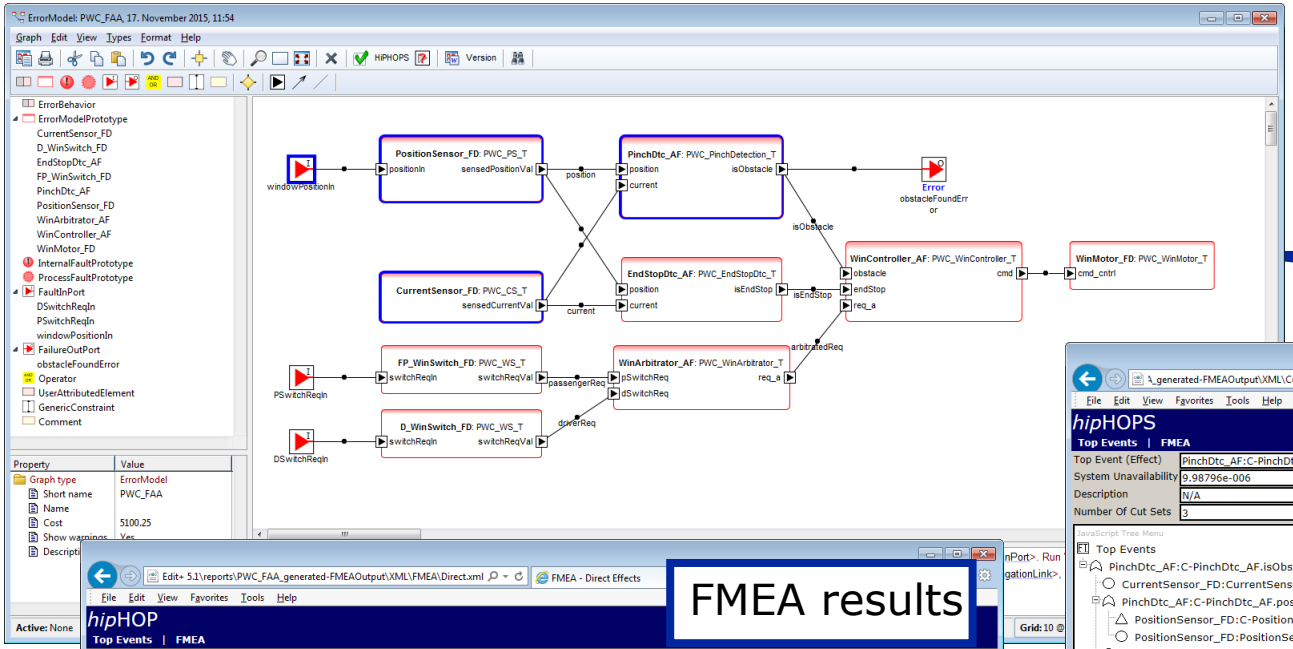
- Define the **scope** (usage, environment etc)
- Identify risk sources
- Estimate the risk
- Evaluate the risk
- Identify **safety functions**
- Calculate **risks**
- Use the results to reduce risks

2) Apply safety concepts directly while modeling

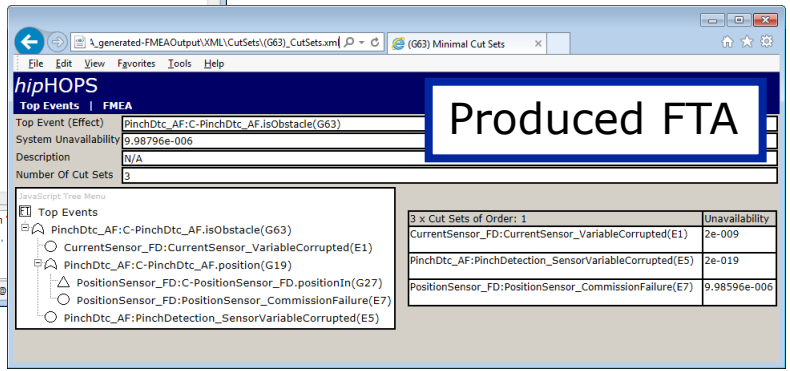
- ISO26262
  - Item
  - Hazard
  - HazardEvent
  - SafetyGoal
  - Requirement
  - SafetyConcept
  - …

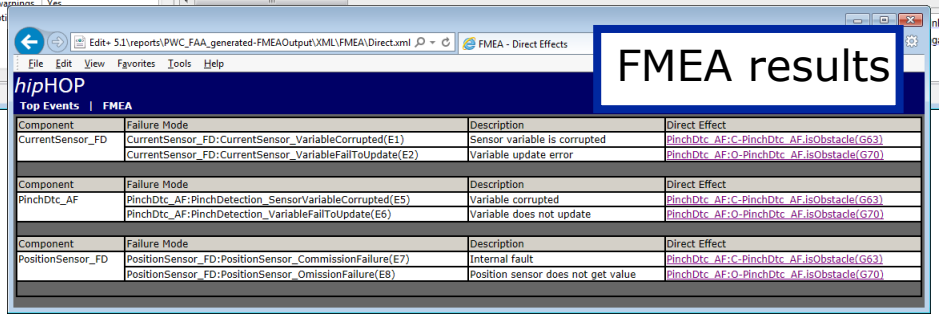


3) Link with analytical tools



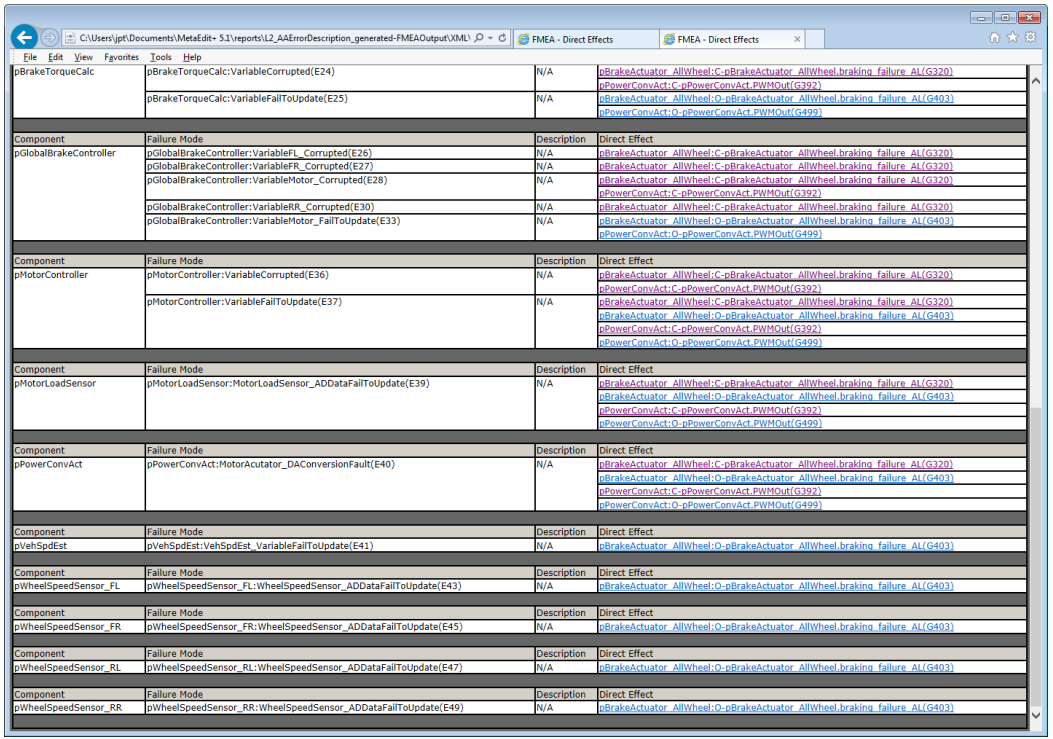
Exports the error model to HipHOPS tool



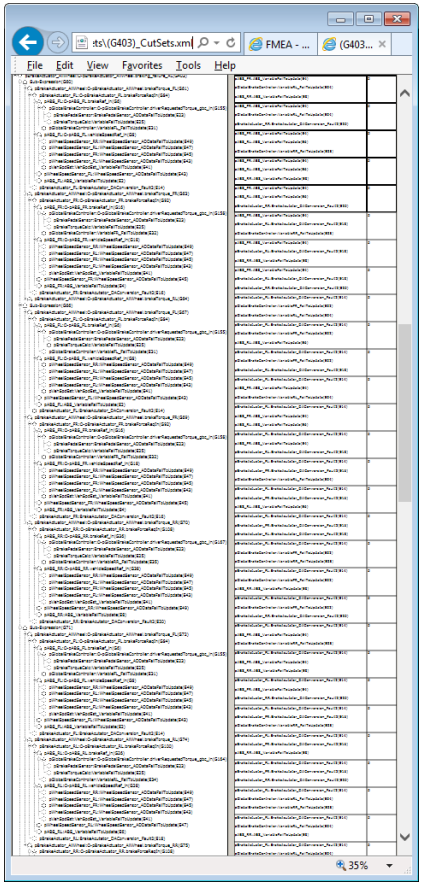


Scaled for larger systems

- FTA/FMEA with cut sets, unavailability, costs

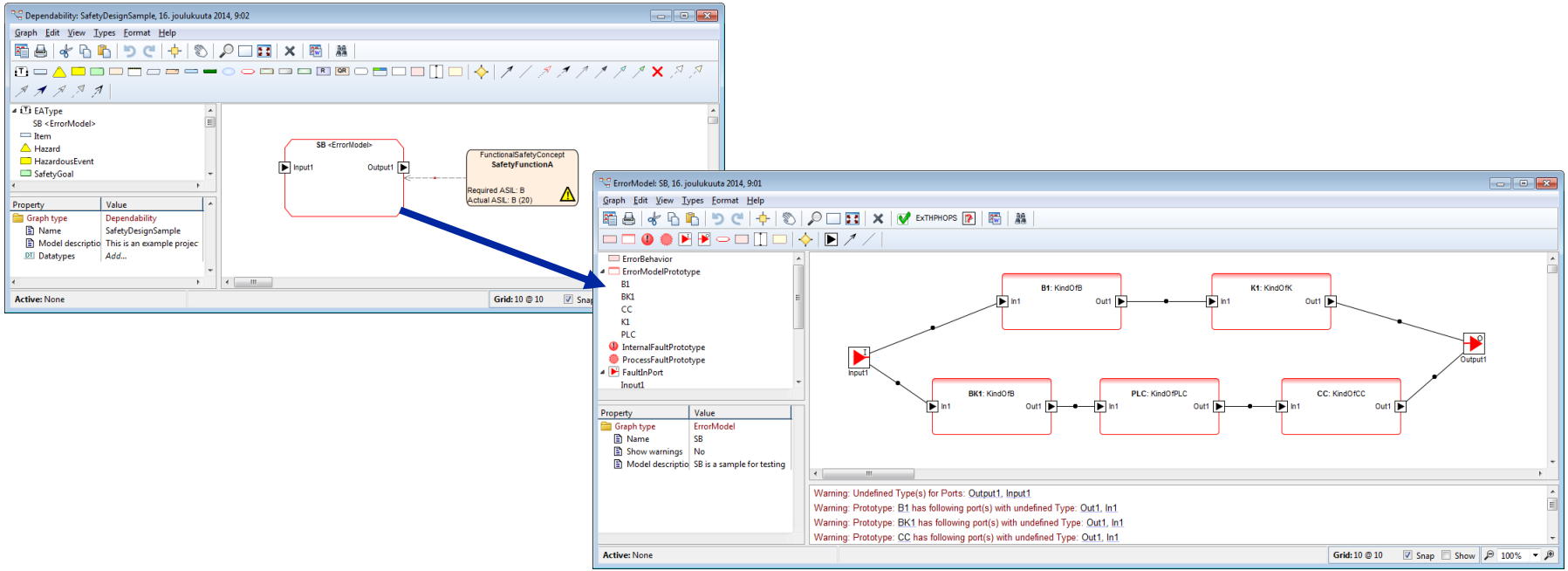


pBrakeTorqueCalc	pBrakeTorqueCalc:VariableCorrupted(E24)	N/A	pBrakeActuator_AllWheel:C-pBrakeActuator_AllWheel.braking_failure_AL(G320) pPowerConvAct:C-pPowerConvAct.PWMOut(G392)
	pBrakeTorqueCalc:VariableFailToUpdate(E25)	N/A	pBrakeActuator_AllWheel:O-pBrakeActuator_AllWheel.braking_failure_AL(G403) pPowerConvAct:O-pPowerConvAct.PWMOut(G499)
Component	Failure Mode	Description	Direct Effect
pGlobalBrakeController	pGlobalBrakeController:VariableFL_Corrupted(E26)	N/A	pBrakeActuator_AllWheel:C-pBrakeActuator_AllWheel.braking_failure_AL(G320)
	pGlobalBrakeController:VariableFR_Corrupted(E27)	N/A	pBrakeActuator_AllWheel:C-pBrakeActuator_AllWheel.braking_failure_AL(G320)
	pGlobalBrakeController:VariableMotor_Corrupted(E28)	N/A	pBrakeActuator_AllWheel:C-pBrakeActuator_AllWheel.braking_failure_AL(G320) pPowerConvAct:C-pPowerConvAct.PWMOut(G392)
	pGlobalBrakeController:VariableRR_Corrupted(E30)	N/A	pBrakeActuator_AllWheel:C-pBrakeActuator_AllWheel.braking_failure_AL(G320)
	pGlobalBrakeController:VariableMotor_FailToUpdate(E33)	N/A	pBrakeActuator_AllWheel:O-pBrakeActuator_AllWheel.braking_failure_AL(G403) pPowerConvAct:O-pPowerConvAct.PWMOut(G499)
Component	Failure Mode	Description	Direct Effect
pMotorController	pMotorController:VariableCorrupted(E36)	N/A	pBrakeActuator_AllWheel:C-pBrakeActuator_AllWheel.braking_failure_AL(G320) pPowerConvAct:C-pPowerConvAct.PWMOut(G392)
	pMotorController:VariableFailToUpdate(E37)	N/A	pBrakeActuator_AllWheel:C-pBrakeActuator_AllWheel.braking_failure_AL(G320) pBrakeActuator_AllWheel:O-pBrakeActuator_AllWheel.braking_failure_AL(G403) pPowerConvAct:C-pPowerConvAct.PWMOut(G392) pPowerConvAct:O-pPowerConvAct.PWMOut(G499)
Component	Failure Mode	Description	Direct Effect
pMotorLoadSensor	pMotorLoadSensor:MotorLoadSensor_ADDataFailToUpdate(E39)	N/A	pBrakeActuator_AllWheel:C-pBrakeActuator_AllWheel.braking_failure_AL(G320) pBrakeActuator_AllWheel:O-pBrakeActuator_AllWheel.braking_failure_AL(G403) pPowerConvAct:C-pPowerConvAct.PWMOut(G392) pPowerConvAct:O-pPowerConvAct.PWMOut(G499)
Component	Failure Mode	Description	Direct Effect
pPowerConvAct	pPowerConvAct:MotorAcutator_DAConversionFault(E40)	N/A	pBrakeActuator_AllWheel:C-pBrakeActuator_AllWheel.braking_failure_AL(G320) pBrakeActuator_AllWheel:O-pBrakeActuator_AllWheel.braking_failure_AL(G403) pPowerConvAct:C-pPowerConvAct.PWMOut(G392) pPowerConvAct:O-pPowerConvAct.PWMOut(G499)
Component	Failure Mode	Description	Direct Effect
pVehSpdEst	pVehSpdEst:VehSpdEst_VariableFailToUpdate(E41)	N/A	pBrakeActuator_AllWheel:O-pBrakeActuator_AllWheel.braking_failure_AL(G403)
Component	Failure Mode	Description	Direct Effect
pWheelSpeedSensor_FL	pWheelSpeedSensor_FL:WheelSpeedSensor_ADDataFailToUpdate(E43)	N/A	pBrakeActuator_AllWheel:O-pBrakeActuator_AllWheel.braking_failure_AL(G403)
Component	Failure Mode	Description	Direct Effect
pWheelSpeedSensor_FR	pWheelSpeedSensor_FR:WheelSpeedSensor_ADDataFailToUpdate(E45)	N/A	pBrakeActuator_AllWheel:O-pBrakeActuator_AllWheel.braking_failure_AL(G403)
Component	Failure Mode	Description	Direct Effect
pWheelSpeedSensor_RL	pWheelSpeedSensor_RL:WheelSpeedSensor_ADDataFailToUpdate(E47)	N/A	pBrakeActuator_AllWheel:O-pBrakeActuator_AllWheel.braking_failure_AL(G403)
Component	Failure Mode	Description	Direct Effect
pWheelSpeedSensor_RR	pWheelSpeedSensor_RR:WheelSpeedSensor_ADDataFailToUpdate(E49)	N/A	pBrakeActuator_AllWheel:O-pBrakeActuator_AllWheel.braking_failure_AL(G403)

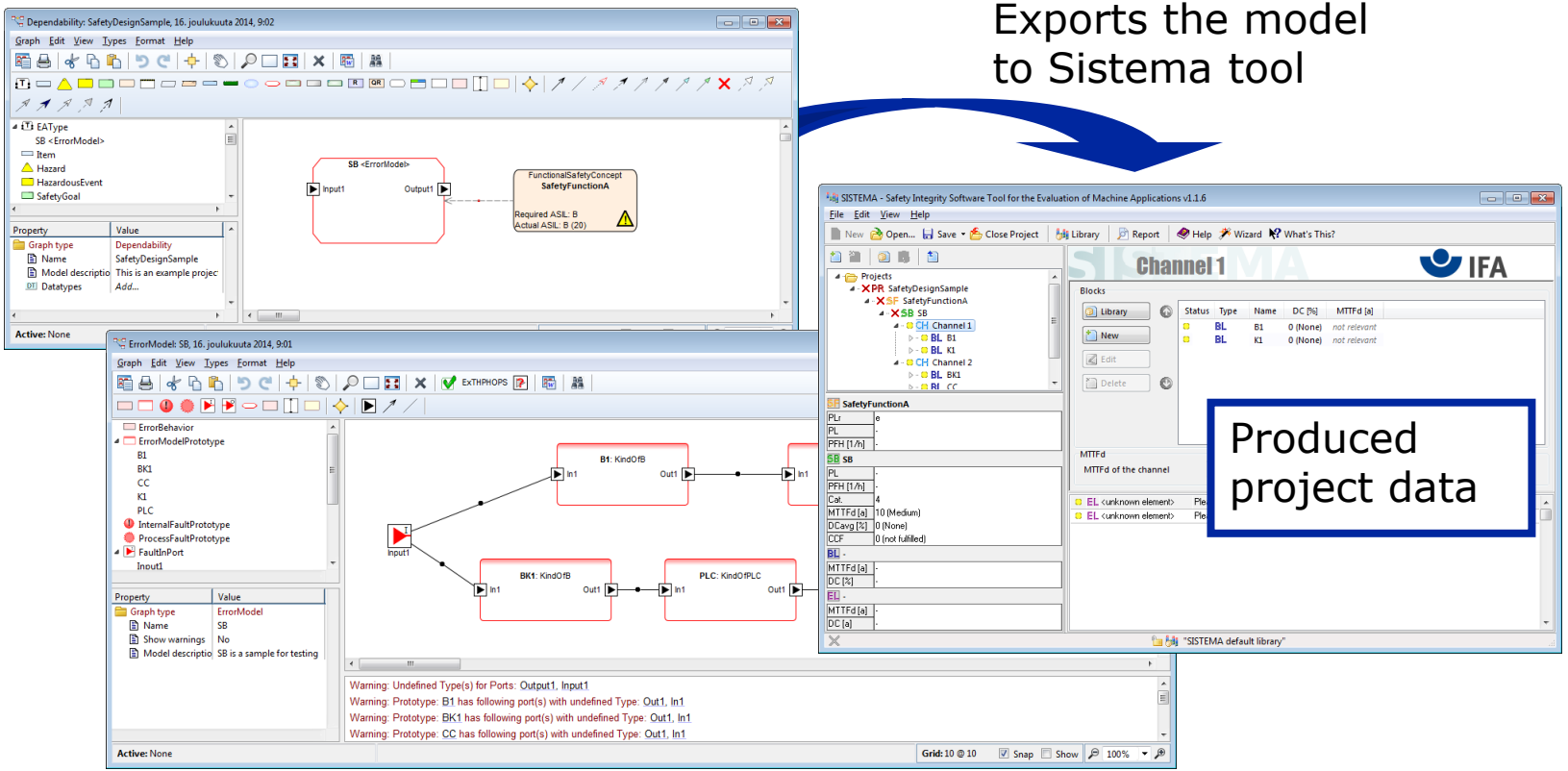


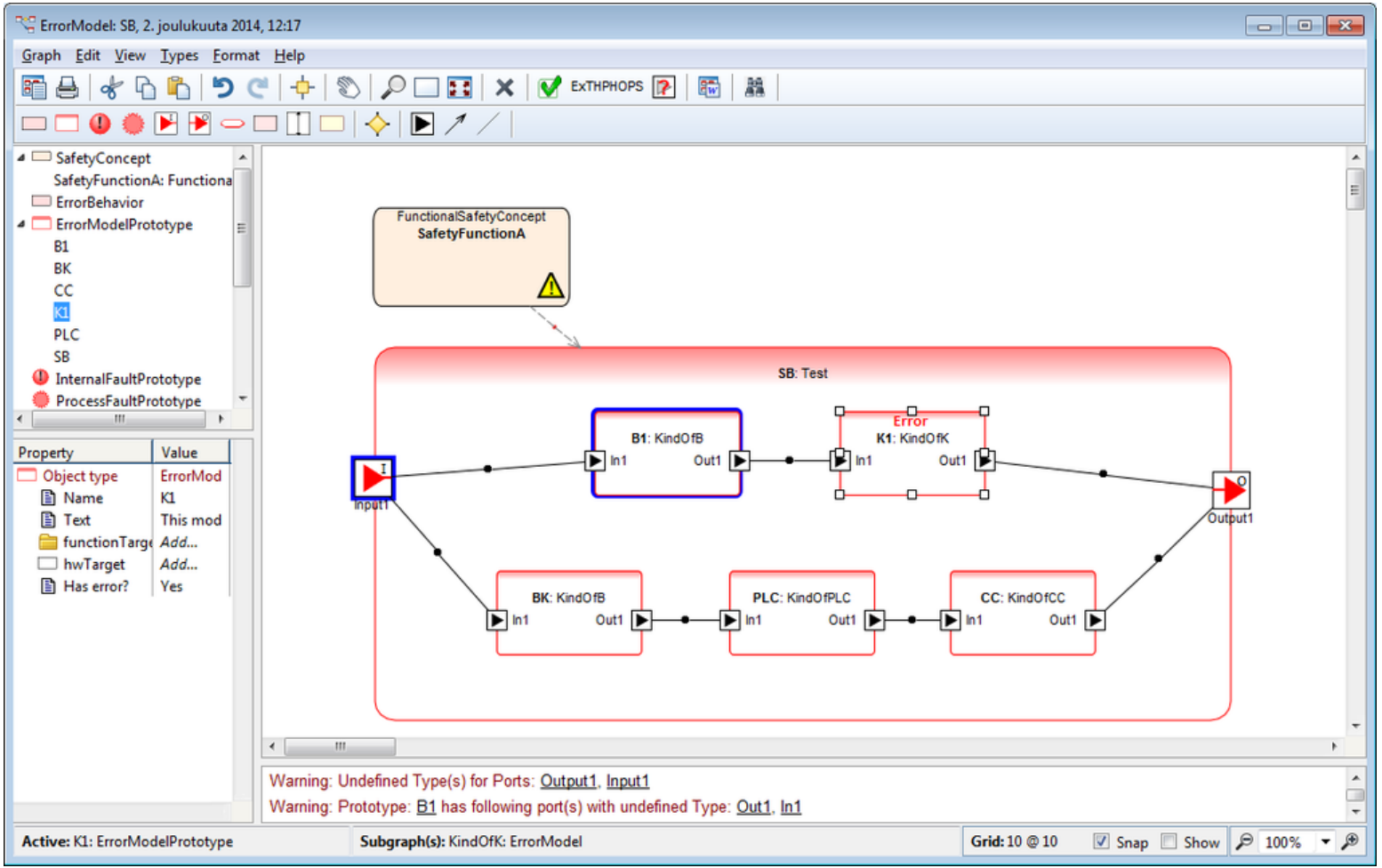
Different analytical tools

- Same model-based approach with another analysis tool
- Specification language adapted for specific needs



Link to analysis tool




ErrorModel: SB, 2. joulukuuta 2014, 12:17
Graph Edit View Types Format Help
ExTHPHOPS
SafetyConcept
SafetyFunctionA: Functiona
ErrorBehavior
ErrorModelPrototype
B1
BK
CC
K1
PLC
SB
InternalFaultPrototype
ProcessFaultPrototype
Property
Value
Object type
ErrorMod
Name
K1
Text
This mod
functionTarge
Add...
hwTarget
Add...
Has error?
Yes
FunctionalSafetyConcept
SafetyFunctionA
SB: Test
B1: KindOfB
In1
Out1
Error
K1: KindOfK
Input1
Output1
BK: KindOfB
PLC: KindOfPLC
CC: KindOfCC
Warning: Undefined Type(s) for Ports: Output1, Input1
Warning: Prototype: B1 has following port(s) with undefined Type: Out1, In1
Active: K1: ErrorModelPrototype
Subgraph(s): KindOfK: ErrorModel
Grid: 10 @ 10
Snap
Show
100%

Summary

- Use of model-based approach provides several benefits:
  - Ensures that safety analysis is done for the intended/designed architecture
  - Makes safety analysis faster as it is partly automated
  - Reduces error-prone routine work
  - Makes safety analysis easier to use and accessible
- The presented approach is not tied to any particular tool
- Specification languages and related transformations need to be flexible
- Extend the approach by providing feedback loop back from analysis to original source models

Thank you!
Questions, please?

For references on examples and cases contact:
Juha-Pekka Tolvanen, jpt@metacase.com
www.metacase.com