MetaCase

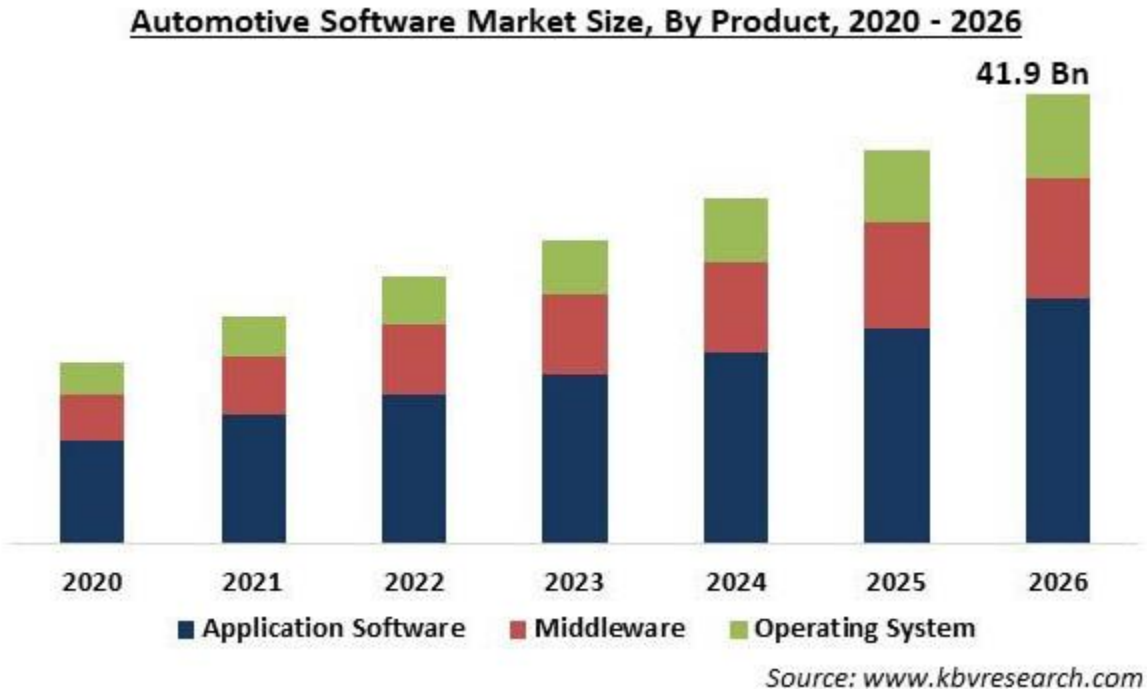
安全性とセキュリティでMBSEを強化

MetaEdit+ for Automotive Systems Engineering

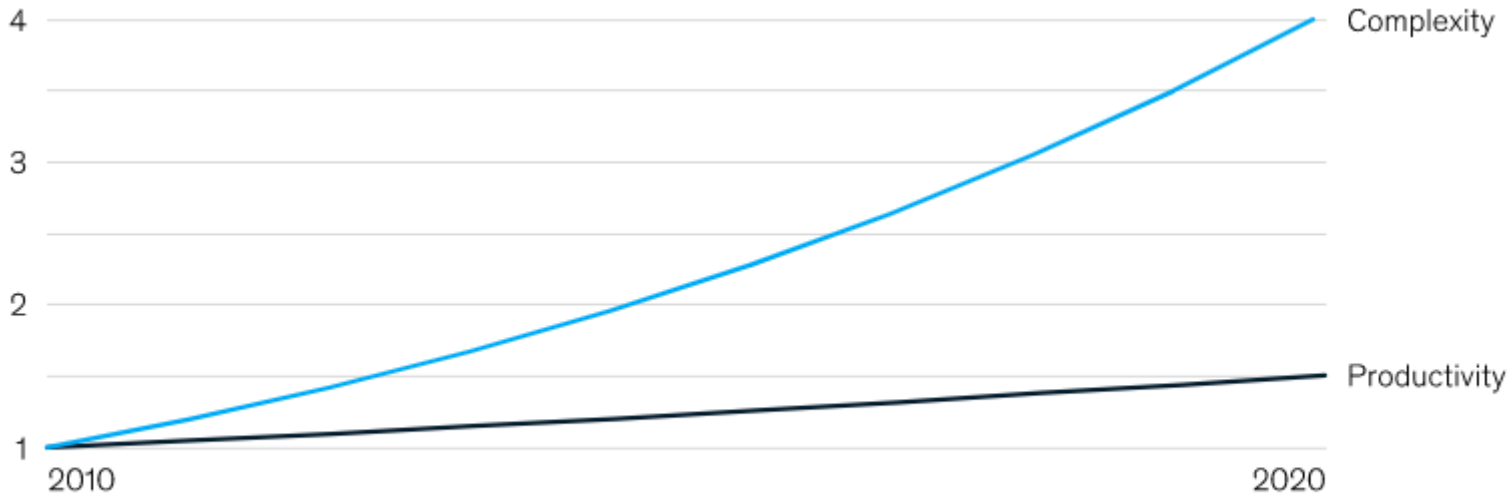
Agenda

- 車載システム開発の傾向と課題
- ドメイン固有モデリング言語
- ドメイン固有にMBSEを強化
  1. 同時コラボレーション
  2. Simulink 統合
  3. 様々な成果
  4. 安全性とセキュリティ
- まとめ

Continuous growth of functionality

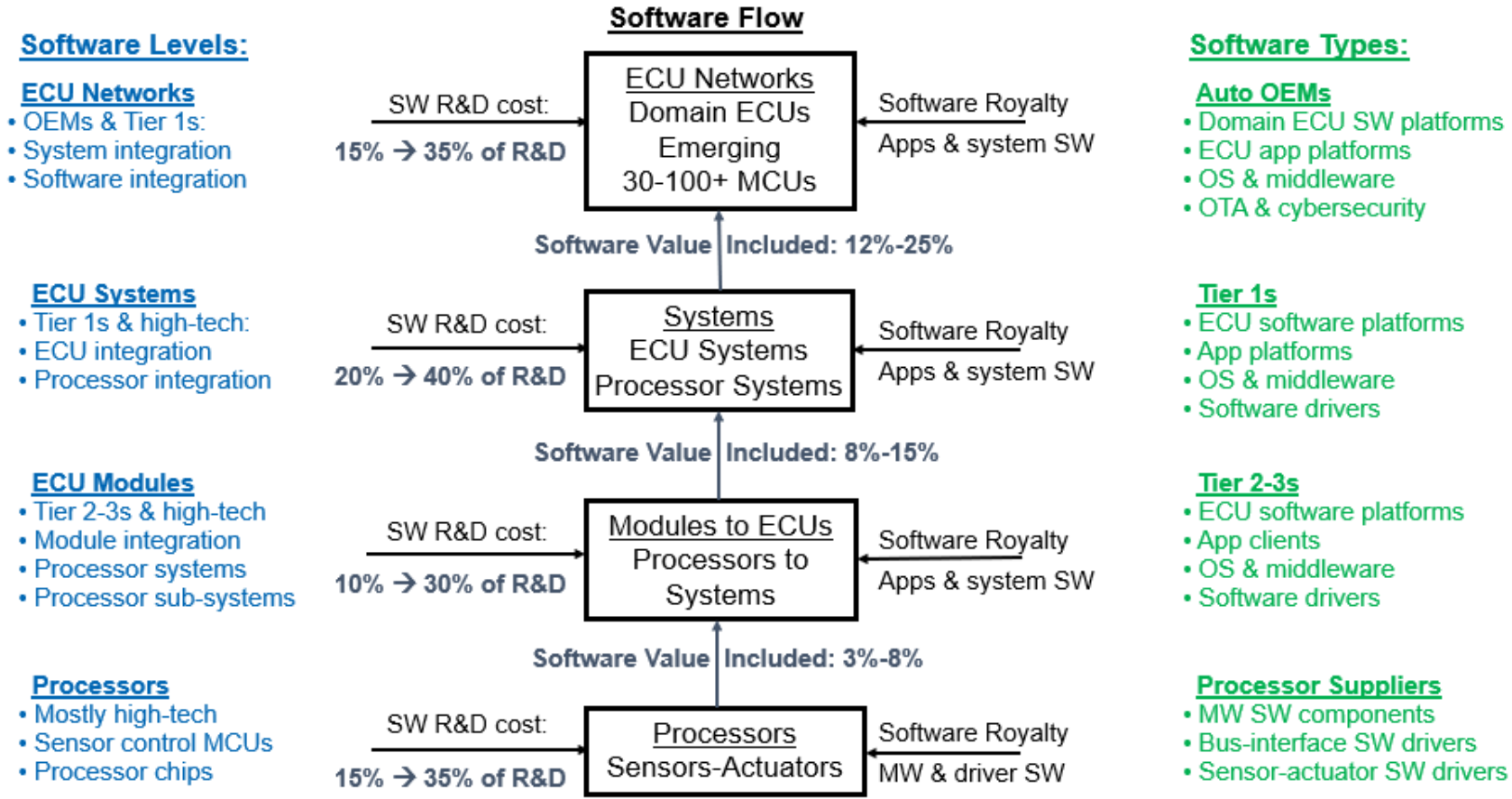


Complexity is growing faster than development productivity



Source: McKinsey's SoftCoster embedded software project database

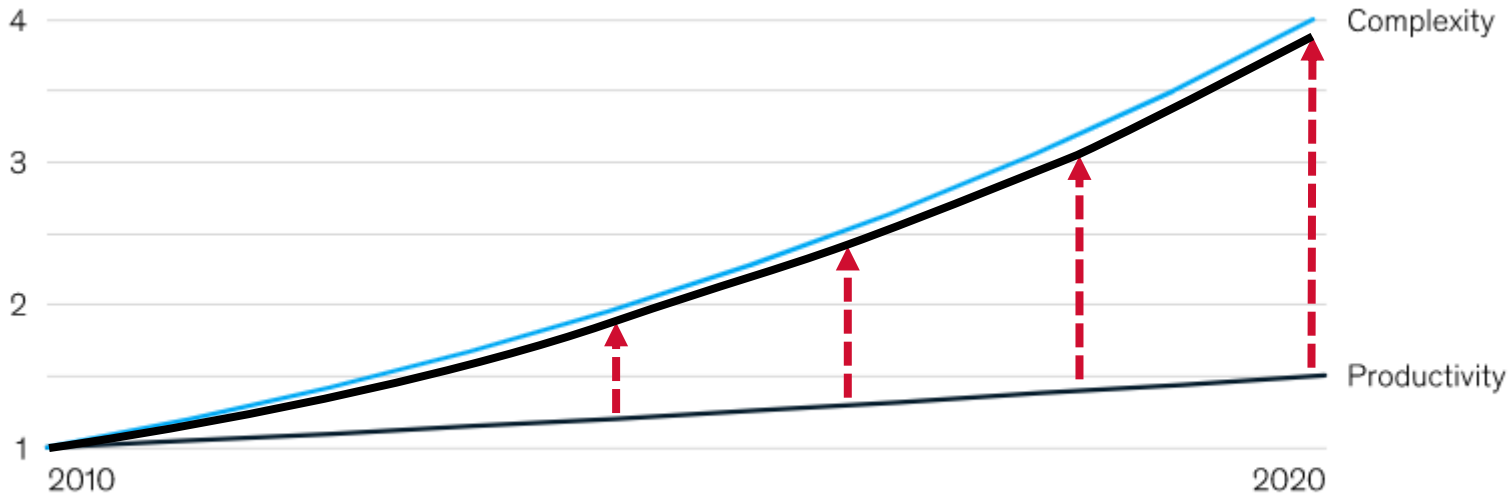
Auto Software Cost & Value Flow



Source: Egil Juliussen; May 2021

ECU=Electronic Control Unit; MW=Middleware; OS=Operating System; SW=Software

How to improve productivity?



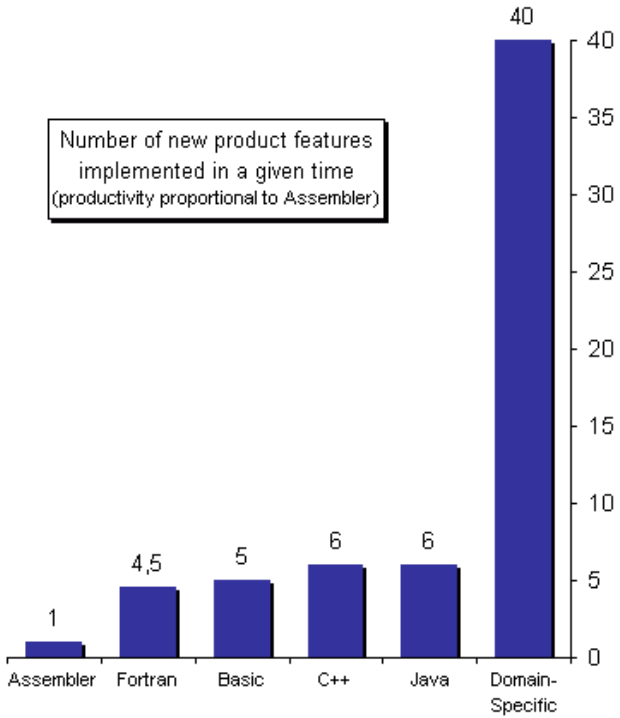
Source: McKinsey's SoftCoster embedded software project database

Agenda

- 車載システム開発の傾向と課題
- ドメイン固有モデリング言語
- ドメイン固有にMBSEを強化
  1. 同時コラボレーション
  2. Simulink 統合
  3. 様々な成果
  4. 安全性とセキュリティ
- まとめ

How has productivity improved?

- 過去、言語の抽象度を上げることで、開発作業の生産性と品質が向上した
- もはや汎用プログラミング言語では、生産性向上は期待できない
- ドメイン固有言語なら抽象レベルを上げて生産性を向上させる



***Software Productivity Research & Capers Jones, 2002**

Industry experiences

Elektrobit

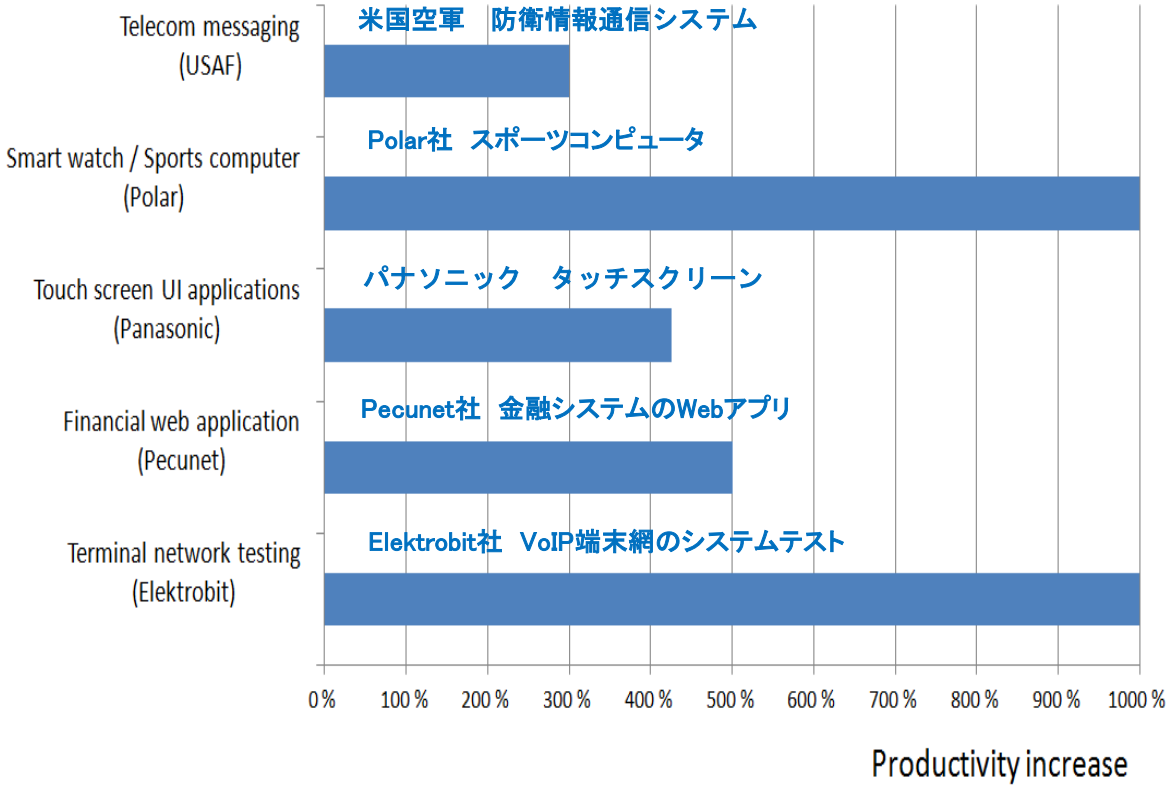
"provided more than ten times faster speed in comparison to previous experience"

AIRBUS

"The quality is clearly better, simply because the modeling language rules out errors"

POLAR

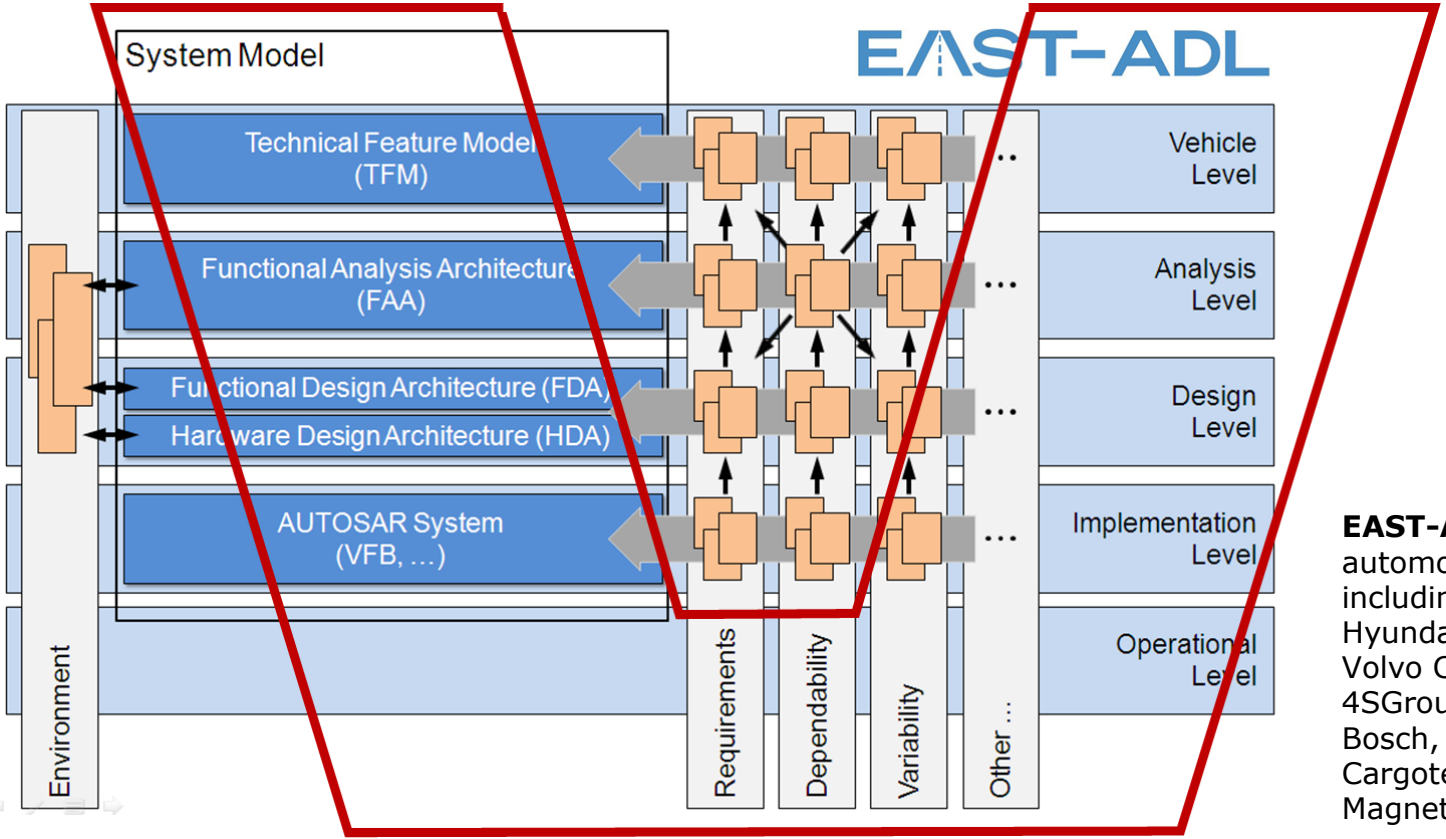
"750% increase in productivity, and greatly improved quality in the code and development"



Generic vs Specific Language (in automotive)

Agenda

- 車載システム開発の傾向と課題
- ドメイン固有モデリング言語とメタモデリングツール
- ドメイン固有にMBSEを強化
  1. 同時コラボレーション
  2. Simulink 統合
  3. 様々な成果
  4. 安全性とセキュリティ
- まとめ



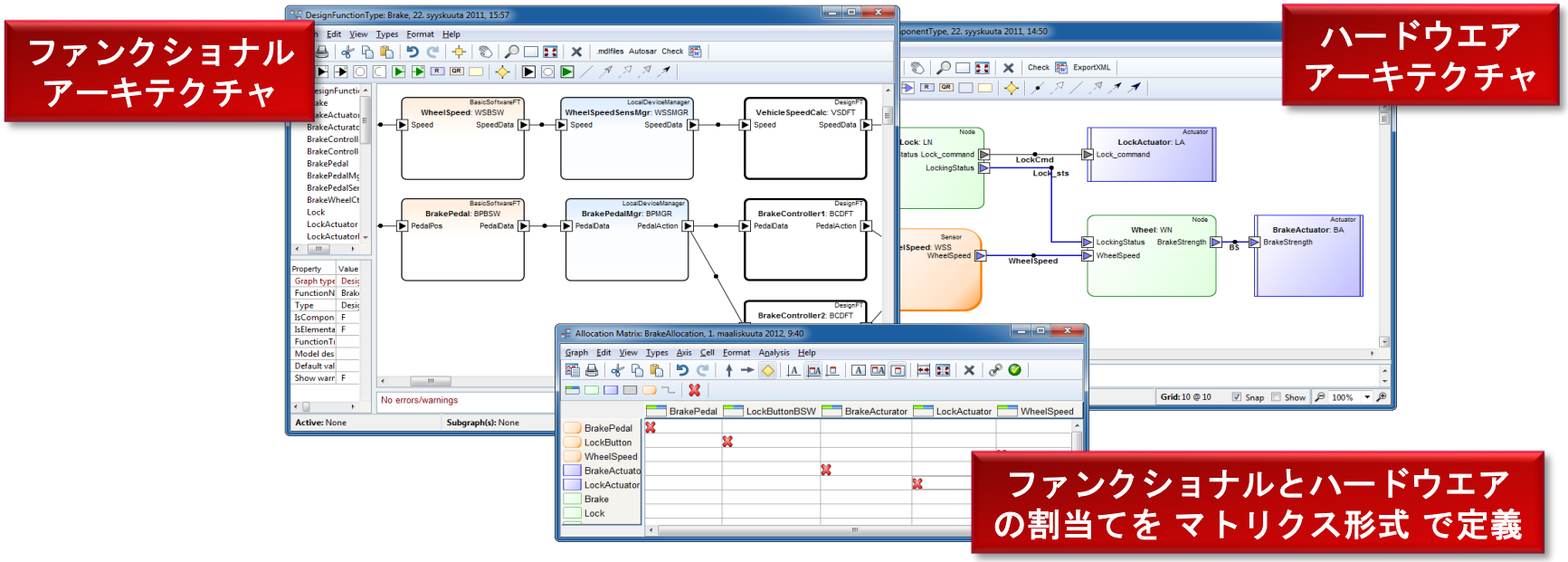
EAST-ADL is developed by automotive companies, including Carmeq/VW, FIAT, Hyundai, McLaren, Volvo, Volvo Cars, 4SGroup, Autoliv, AVL, Bosch, Brace Automotive, Cargotec, Continental, FEV, Magneti Marelli, Mecel/Delphi

EAST-ADL and MetaEdit+

- EAST-ADL
  - 車両の機能、機能アーキテクチャ、およびハードウェアアーキテクチャの仕様をカバーして、これらはすべて追跡可能な要件に関連する
  - 検証と検証、ディペンダビリティとエラー モデリング、タイミングなど、自動車システム内のさまざまな問題に焦点を当てた拡張機能がある
  - EAST-ADL のモデルは抽象化レベルで構造化されており、各サブモデルは関連する詳細レベルで完全な組み込みシステムを表現する
    - 車両レベル、解析レベル、設計レベル、実装レベル
  - EAST-ADL は、仕様の作成に関するルールと制約を定義し、仕様の一貫性と追跡可能性を維持する
- MetaEdit+ Tool
  - MetaEdit+ は、EAST-ADL 言語を特定のニーズに適応させることを可能にするメタモデリングツール。モデリングのサポートと、さまざまな分析、統合、およびコードを生成するジェネレーターを変更したり、新しいものを作成したりできる。
  - モデリングツールとして、EAST-ADLで作成された各種アーキテクチャ仕様書の作成・編集・閲覧が可能
  - MetaEdit+ を使用すると、モデルの分析とチェック、ドキュメントの生成、EAST-ADL モデルからの他のモデルとコードの生成、および要件ツールやさまざまな分析およびシミュレーション ツールへのエクスポートが可能になる。

EAST-ADLをMetaEdit+で実装・拡張

Volvo, Fiat, Daimler, Continental, Bosch



https://www.fuji-setsu.co.jp/products/MetaEdit/SystemModeling.html#EAST-ADL

EAST-ADL and MetaEdit+

- EAST-ADL
  - 車両の機能、機能アーキテクチャ、およびハードウェアアーキテクチャの仕様をカバーして、これらはすべて追跡可能な要件に関連する
  - 検証と検証、ディペンダビリティとエラー モデリング、タイミングなど、自動車システム内のさまざまな問題に焦点を当てた拡張機能がある
  - EAST-ADL のモデルは抽象化レベルで構造化されており、各サブモデルは関連する詳細レベルで完全な組み込みシステムを表現する
    - 車両レベル、解析レベル、設計レベル、実装レベル
  - EAST-ADL は、仕様の作成に関するルールと制約を定義し、仕様の一貫性と追跡可能性を維持する
- MetaEdit+ Tool
  - MetaEdit+ は、EAST-ADL 言語を特定のニーズに適応させることを可能にするメタモデリングツール。モデリングのサポートと、さまざまな分析、統合、およびコードを生成するジェネレーターを変更したり、新しいものを作成したりできる。
  - モデリングツールとして、EAST-ADLで作成された各種アーキテクチャ仕様書の作成・編集・閲覧が可能
  - MetaEdit+ を使用すると、モデルの分析とチェック、ドキュメントの生成、EAST-ADL モデルからの他のモデルとコードの生成、および要件ツールやさまざまな分析およびシミュレーション ツールへのエクスポートが可能になる。

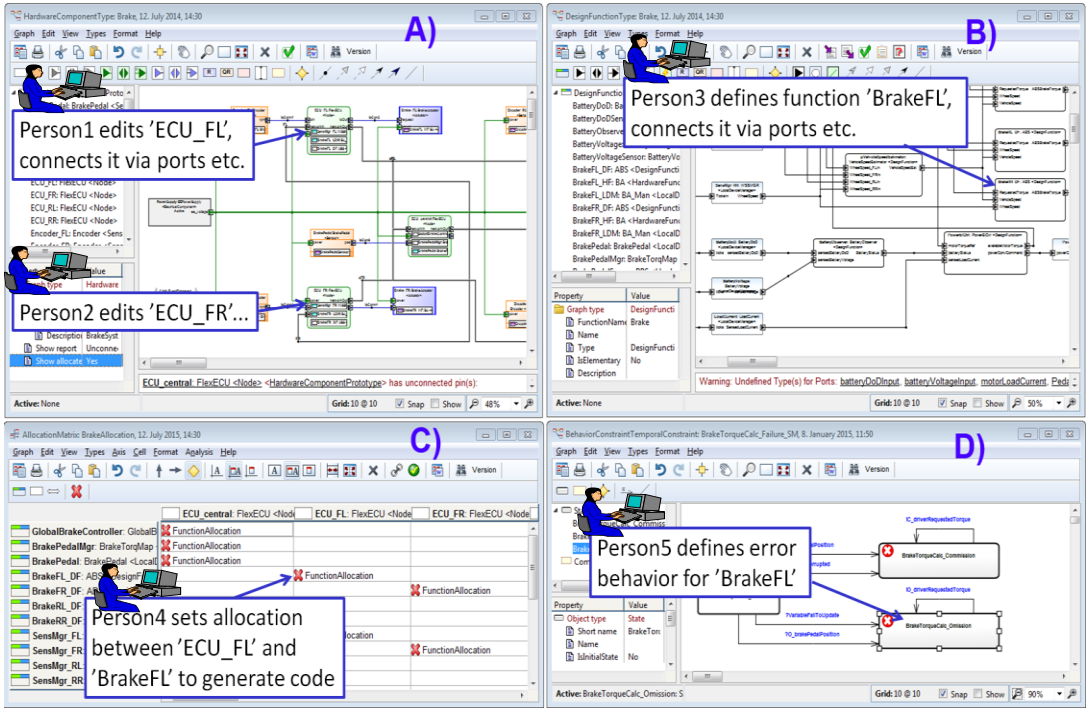
同時コラボレーション開発

同じモデル A) で異なるハードウエア部品を２人で編集する中、

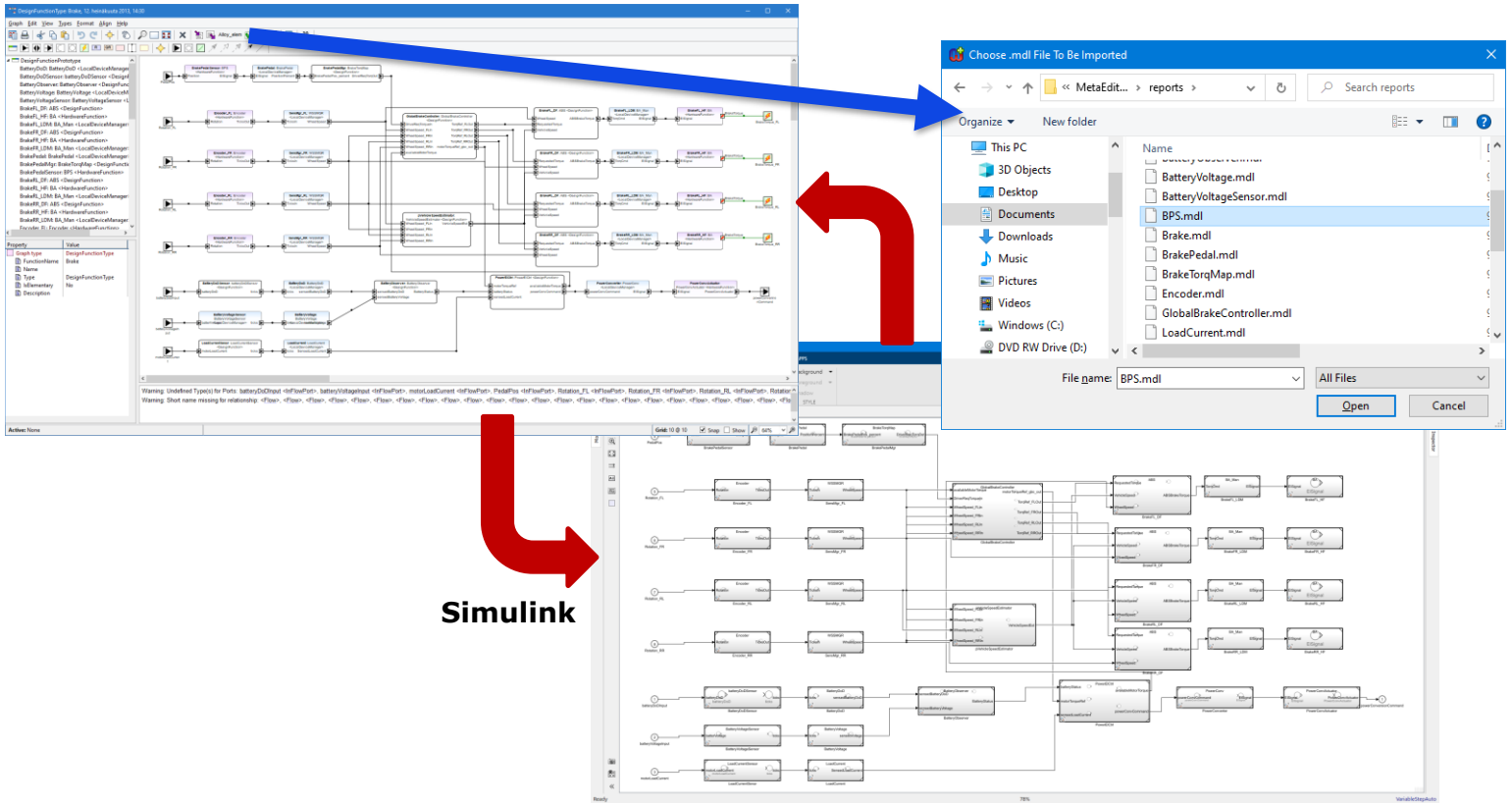
同時にモデル B)でシステムの論理コンポーネントを別の担当者が定義していて、

モデル C)ではコードを生成させるために、A)とB)で編集中のハードウエアと論理アーキテクチャ間のアロケーションを定義している

そしてモデル D)では、機能安全担当がB)で定義される論理コンポーネントのエラーモデルを定義している



Simulinkモデル生成、インポート



動画デモ： https://www.fuji-setsu.co.jp/demo/EASTADL_Simulink.wmv

Simulink 統合

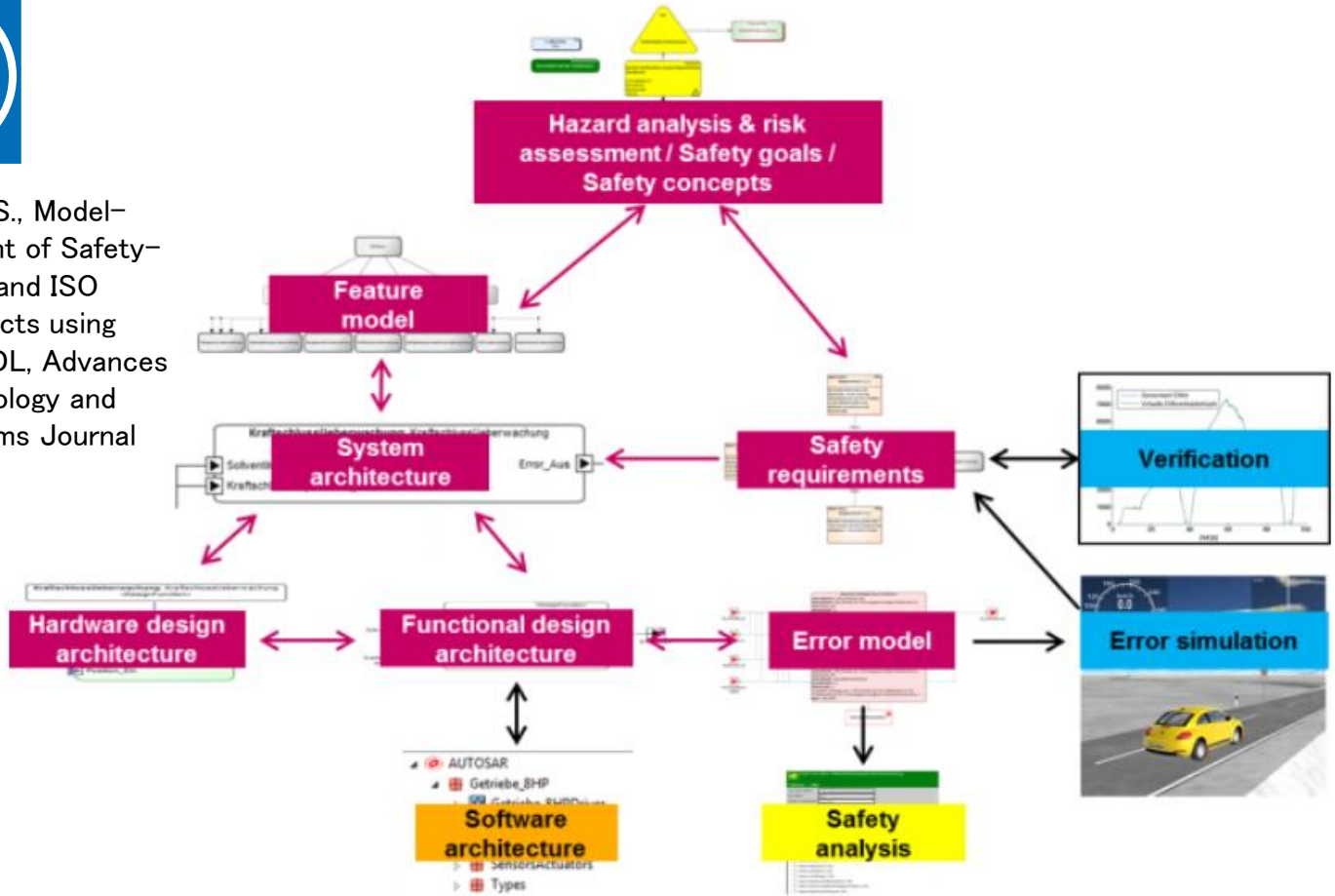
- MetaEdit+ のジェネレータ機能を用いてMDL ファイルを生成
  - + no need for Simulink
  - - mdl format changes between Simulink versions
- MetaEdit+ からSimulink API を呼び出してモデルを生成
  - + stable interface
  - - needs Simulink installed to get models

- インポートはmdl ファイルを使用
  - For reuse, use the model reference command of Simulink
    - Subsystem block is unique inside a file, can’t reuse between files

成果：ドメイン固有にMBSEを強化

- 同時コラボレーション開発
  - 比較（diff）とマージの必要無く、共同開発を円滑化
- 重複作業の排除
  - 変更は全ての成果物に同時反映され通達もされる
- 双方向のトレーサビリティを全プロセスに渡って
  - 要件からコードまで全ての成果物で双方向に
- モデル変換、コード生成、ツール連携
  - Simulinkモデル、Cコード、各種ドキュメント
- 組織ごとの需要に応じてカスタマイズ

Sari, B., Reuss, H.S., Model-based Development of Safety-critical Functions and ISO 26262 Work Products using modified EAST-ADL, Advances in Science, Technology and Engineering Systems Journal Vol. 2, No. 3, 2017



モデルベース開発を効率化

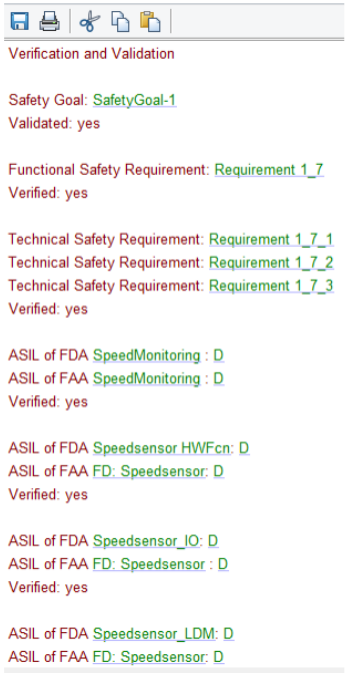
- システムおよび安全性アーキテクチャの開発をEAST-ADLで統合
- ASILを考慮した安全目標と機能安全のトレーサビリティの自動化
- 機能安全コンセプトと技術安全コンセプトのシミュレーション検証を早期開発段階で実施
- ハザード分析およびリスク評価から安全要件に至る、ISO 26262作業成果物をモデルベースで自動化
- 体系的なエラーおよびエラーのシステムへの影響の早期検出

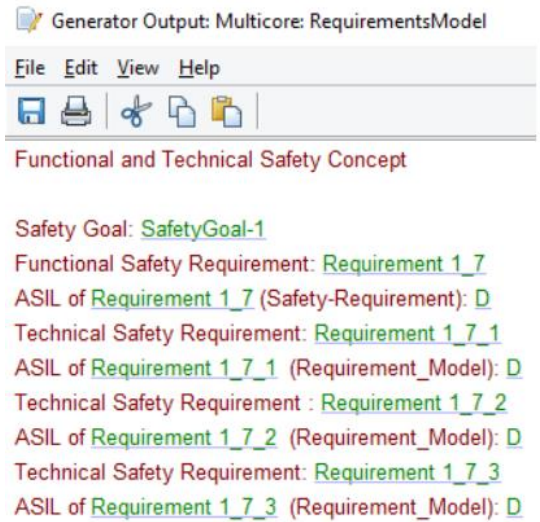
https://www.fuji-setsu.co.jp/products/MetaEdit/SystemModeling.html#case

Reporting, e.g. ZF*

- Sari, B., Fail-operational Safety Architecture for ADAS/AD Systems and a Model-driven Approach for Dependent Failure Analysis, Springer, 2020

https://link.springer.com/book/10.1007/978-3-658-29422-9





Integrating Security and Safety with Systems Engineering: a Model-Based Approach

Matthias Bergler
Technische Hochschule Nürnberg
Nürnberg, Germany
matthias.bergler@th-nuernberg.de

Juha-Pekka Tolvanen
MetaCase
Jyväskylä, Finland
jpt@metacase.com

Ramin Tavakoli Kolagari

Abstract—**Development of reliable systems requires that safety and security concerns are acknowledged during system development. Adding them afterwards is risky as many concerns are missed if not elicited together with the system requirements. Unfortunately, languages for systems engineering, like SysML, typically ignore security and safety forcing development teams to split the work into different formats, languages and tools without easy collaboration, with limited traceability, separate versioning and restricted use of automation that tools can provide. We present a model-based approach targeting automotive that integrates safety and security aspects with other system development practices. This is achieved via a comprehensive domain-specific modeling language that is extendable by language users. We demonstrate this approach with practical examples on how security and safety concerns are recognized along with traditional system design and analysis phases.**

Keywords—model-based development; security, safety, domain-specific language; system engineering; software engineering

Our work on integ
collaborative work
companies, researcher
the approach with prac
concerns are recogn
traceability and aut
safety-related method
FMEA [13], and secu
scores [3]. The integra
that safety and security
being developed and a
can provide, covering
analysis and reporting

We start by prese
modeling languages
languages. This is fol
approach with practic
existing automotive

https://www.fuji-setsu.co.jp/products/MetaEdit/SystemModeling.html#SystemModelingISO26262

TECHNISCHE HOCHSCHULE NÜRNBERG
GEORG SIMON OHM

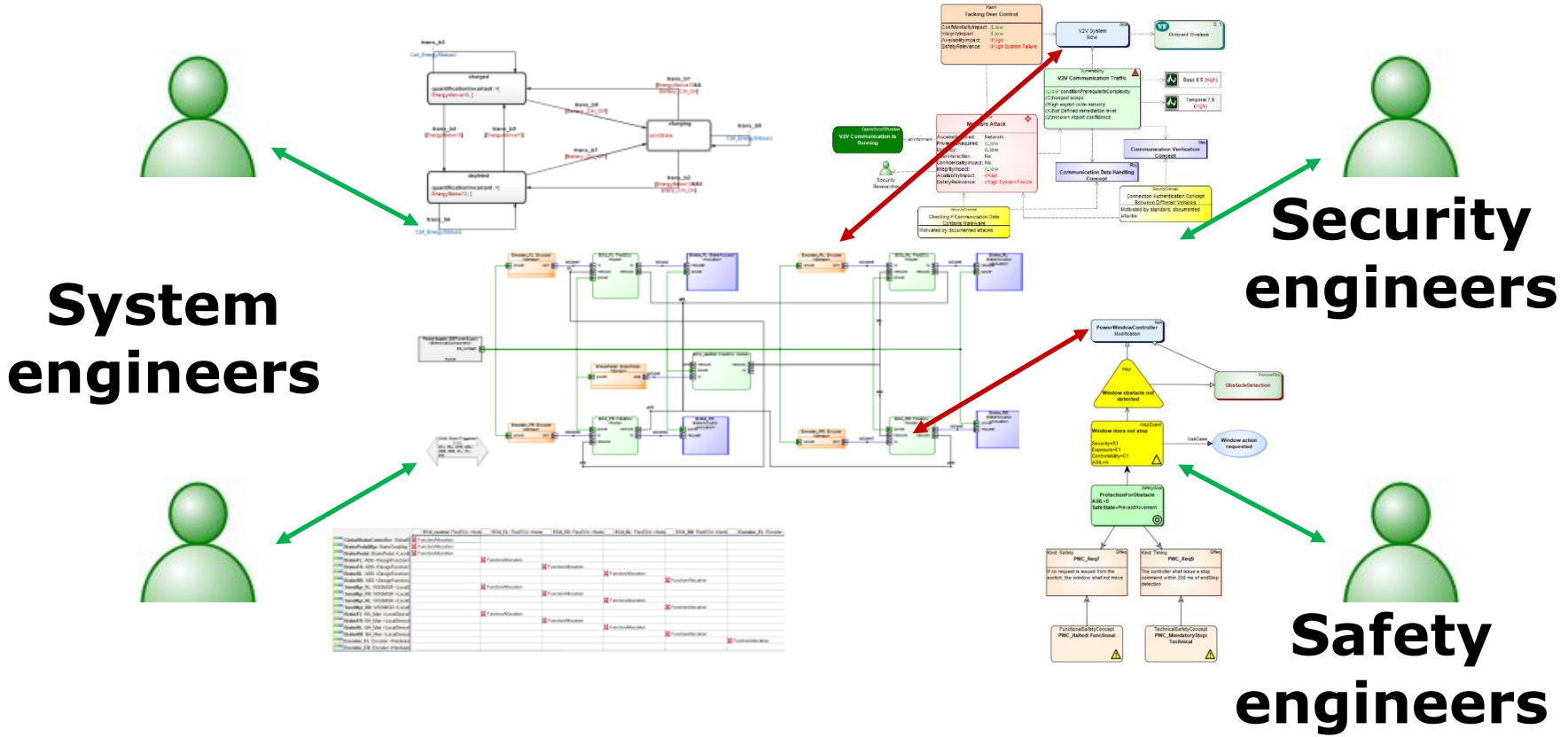
MetaCase

Integrating Security and Safety with Systems Engineering: a Model-Based Approach

embeddedworld2022 Exhibition&Conference ... it's a smarter world

Matthias Bergler, Technische Hochschule Nürnberg
Juha-Pekka Tolvanen, MetaCase
Ramin Tavakoli Kolagari, Technische Hochschule Nürnberg

共通言語で同時コラボレーション



動機：安全性もセキュリティも同時に、、

- SysML等では安全性とセキュリティの側面は考慮されない
  - これらは異なる形式、言語、ツールで定義
  - システム設計との明確なトレーサビリティは無く
  - 個別のサブセクションに分断される
  - それらモデルは個別にバージョン管理される
  - ツールによる自動化やサポートは期待できない
  - 担当者間の意思疎通が損なわれて共同作業に支障をきたす

➢ **利害関係者の重要な懸念が失われて難しくなりエラーの基に、、**

対策：安全性とセキュリティの側面を統合

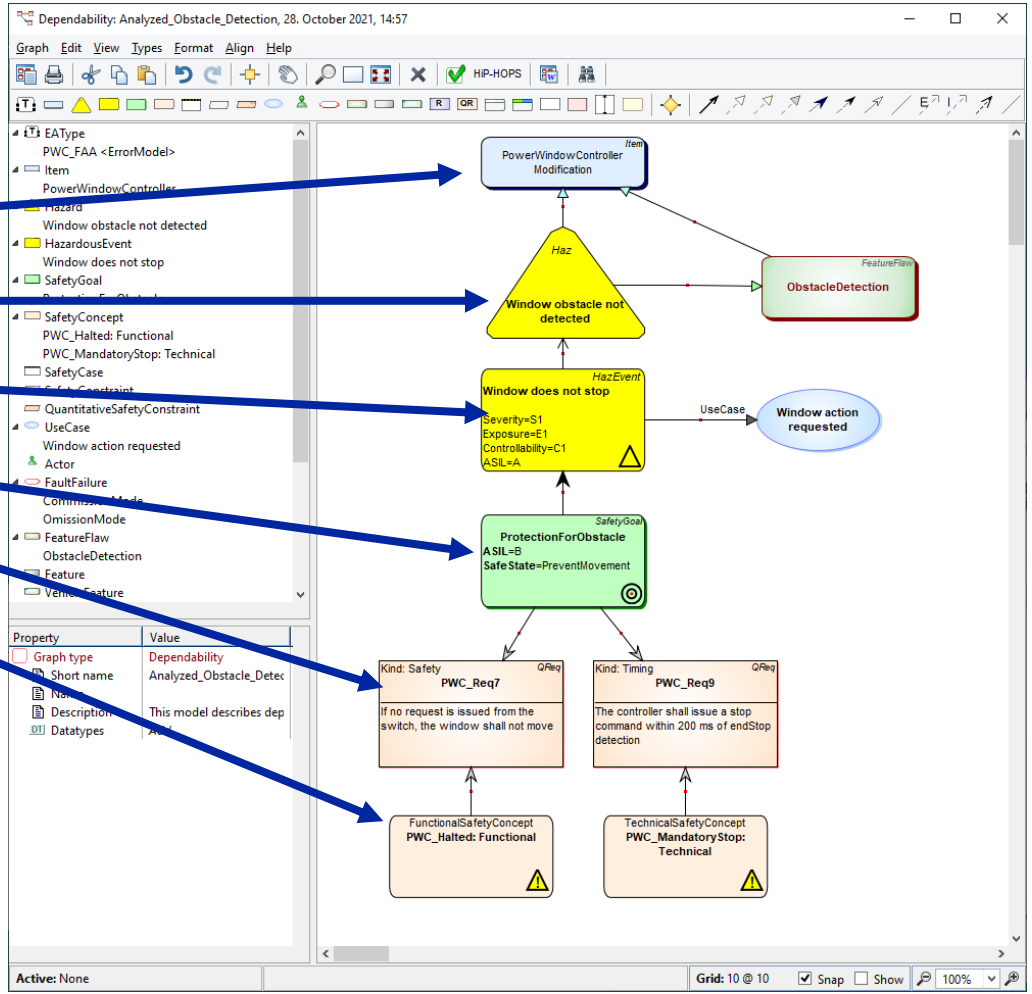
- 安全性とセキュリティの側面でMBSEを強化する
  - 風通しを良くしてコラボレーションを強化
  - 継続的な統合と高度なトレーサビリティ
    - 早期段階で分析、シミュレーションを可能にする
  - モデルチェック、検証の自動化支援
  - バージョン管理も統合

➢ **設計とテストのサイクルを短縮し、早期段階の分析とシミュレーションを可能にする**

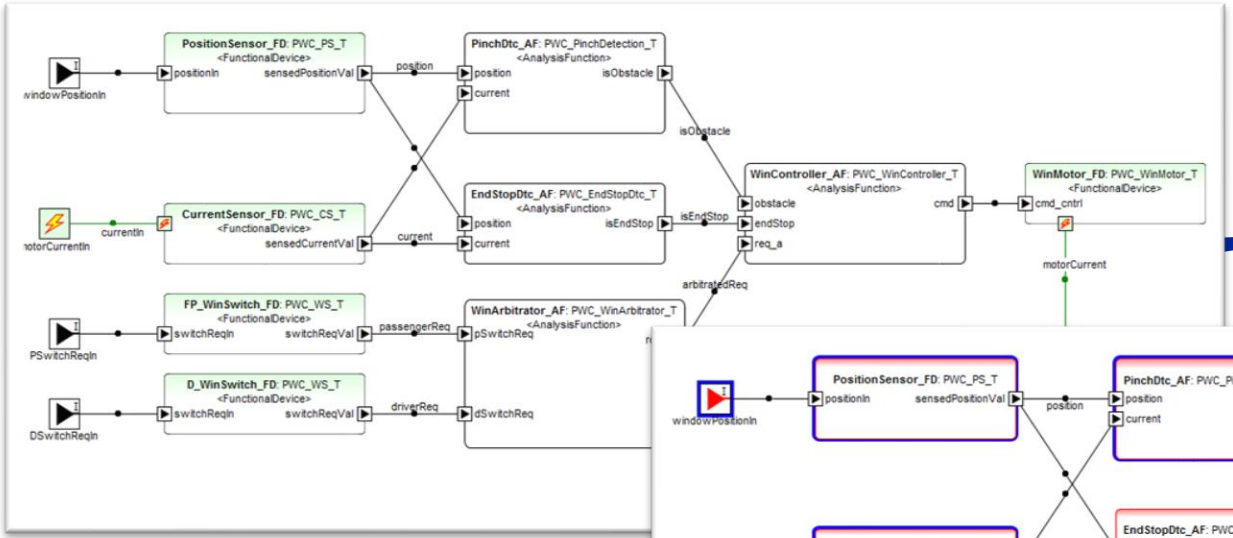
Modeling safety

- 安全性の関心事でモデル化
  - 安全性に固有の概念をモデリングに採用
    - 安全基準（ISO 26262）に固有の専門用語で
- 安全性モデルをシステムの設計とメタモデルレベルで統合
  - システムが開発される時点で問題を把握
  - 設計から安全設計者向けの初期エラーモデルを生成
  - 設計の修正を支援する双方向のトレーサビリティ
- 分析ツールの活用
  - モデルから様々な解析ツールへの入力を生成する
  - 様々な用途に応じて多様なツールを活用

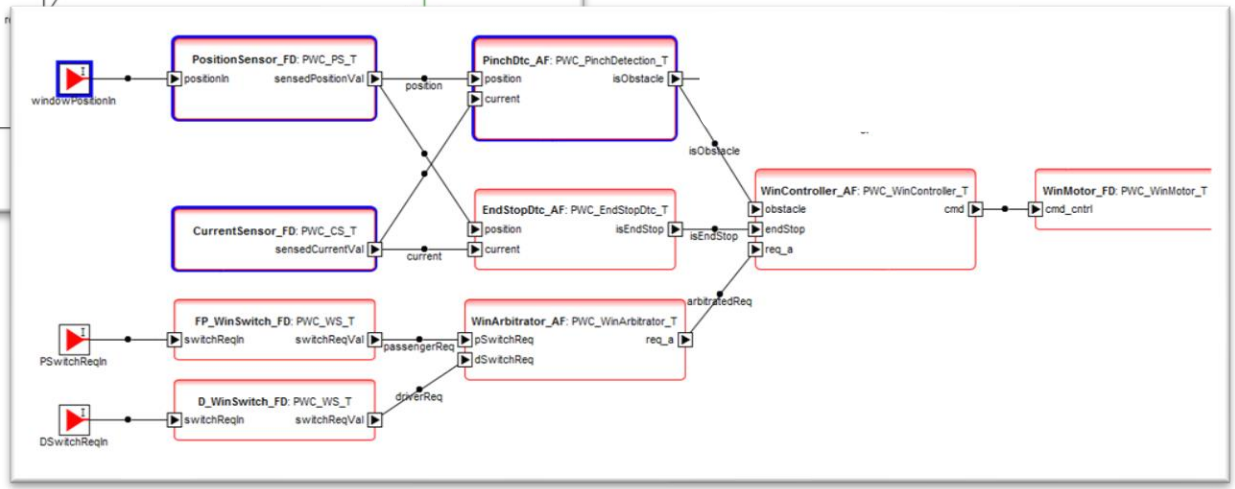
- ISO26262
  - Item
  - Hazard
  - HazardEvent
  - SafetyGoal
  - Requirement
  - SafetyConcept
  - Operational situation
  - Use case
  - …



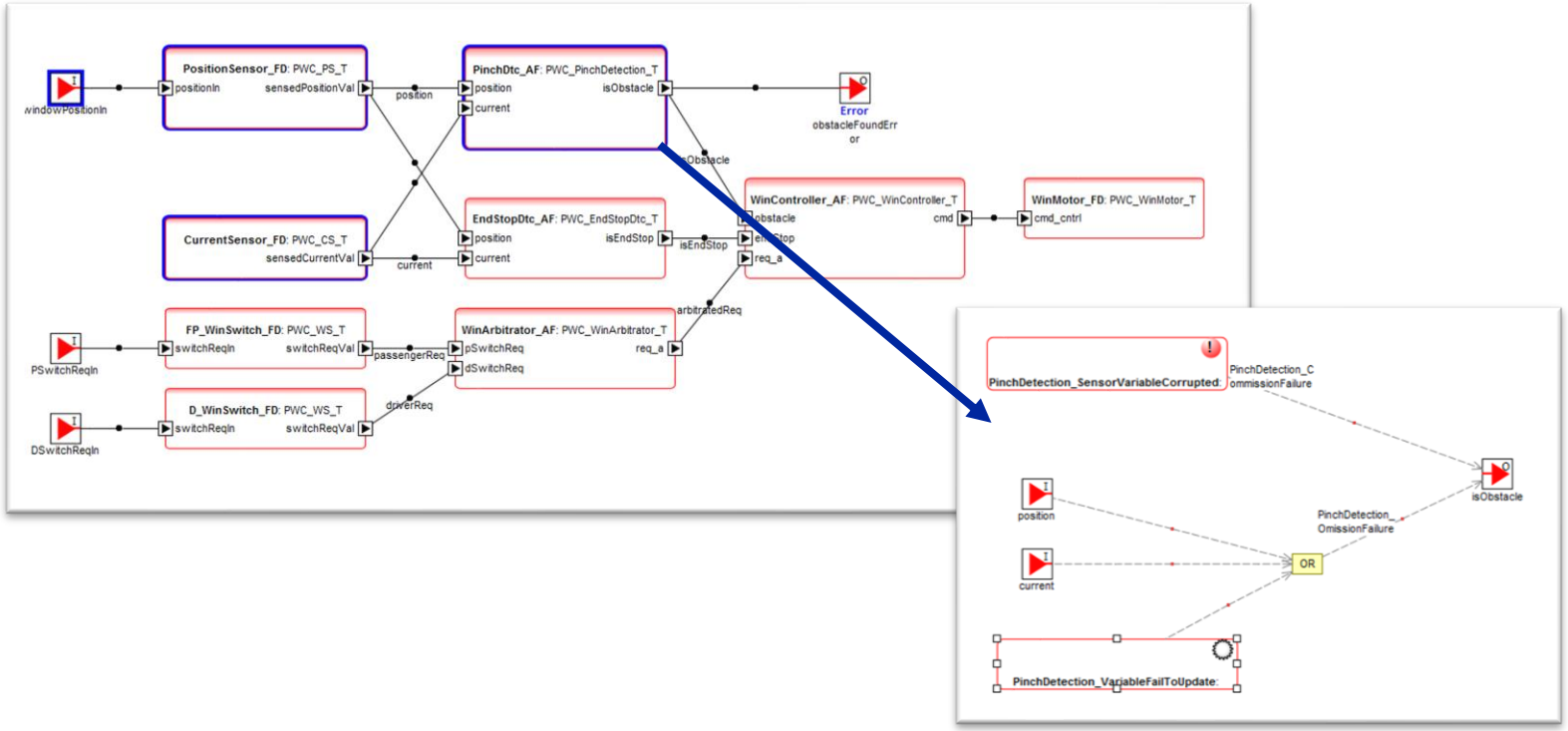
Integrate error models with system designs



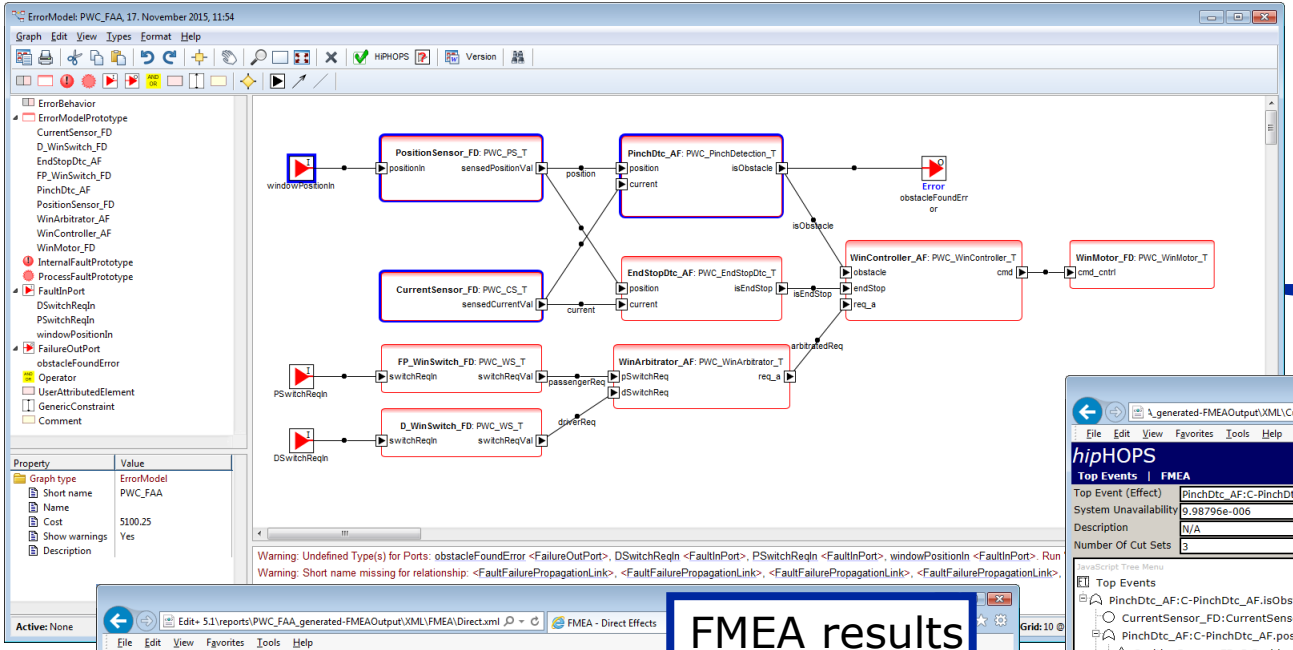
Model transformation



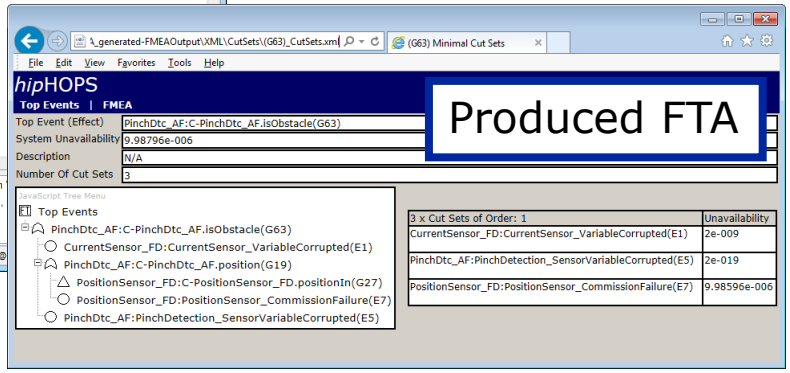
Error logic – partly generated

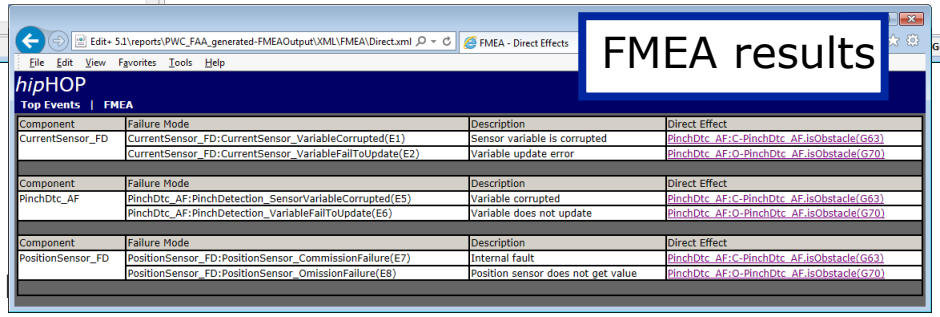


Link with analytical tools



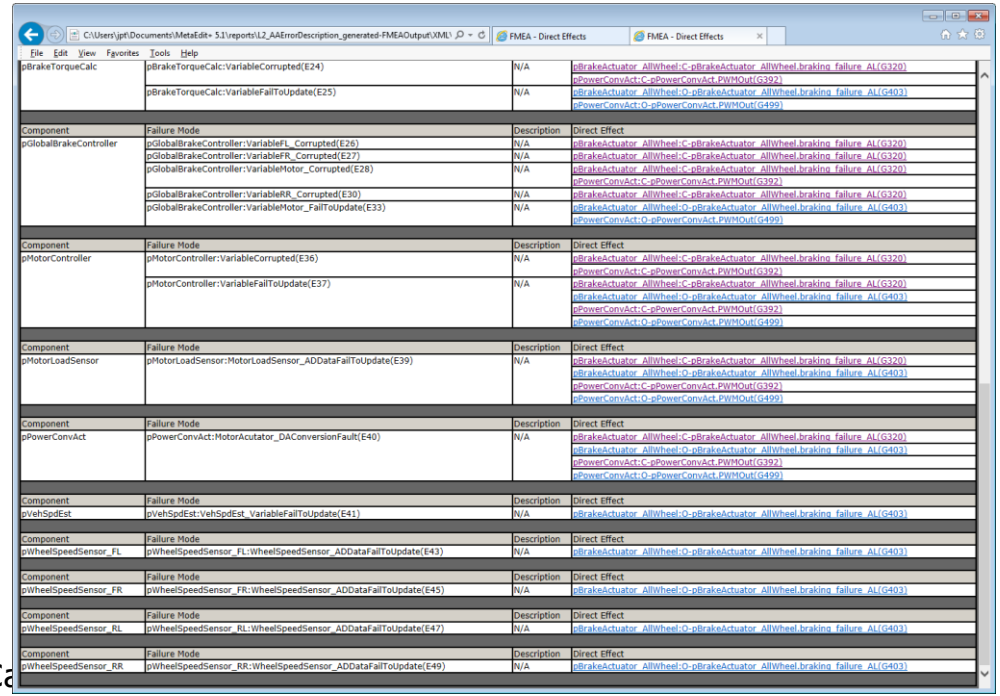
Exports the error model to HiP-HOPS tool



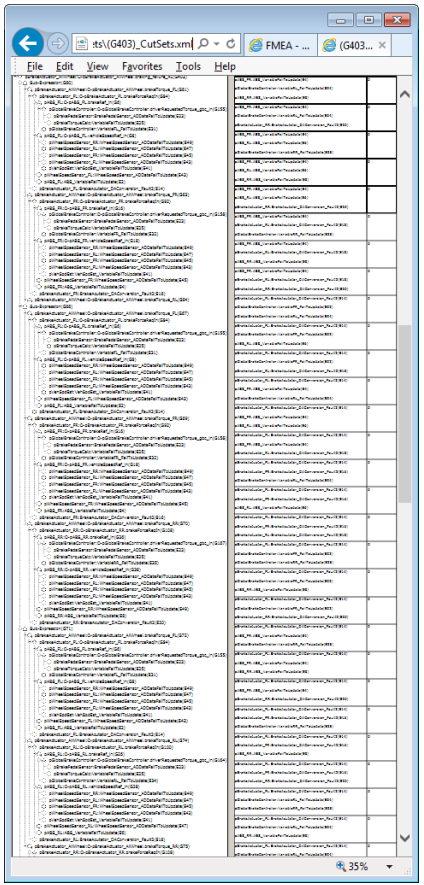


Scaled for larger systems

- FTA/FMEA with cut sets, unavailability, costs, failure rates, repair rates



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)

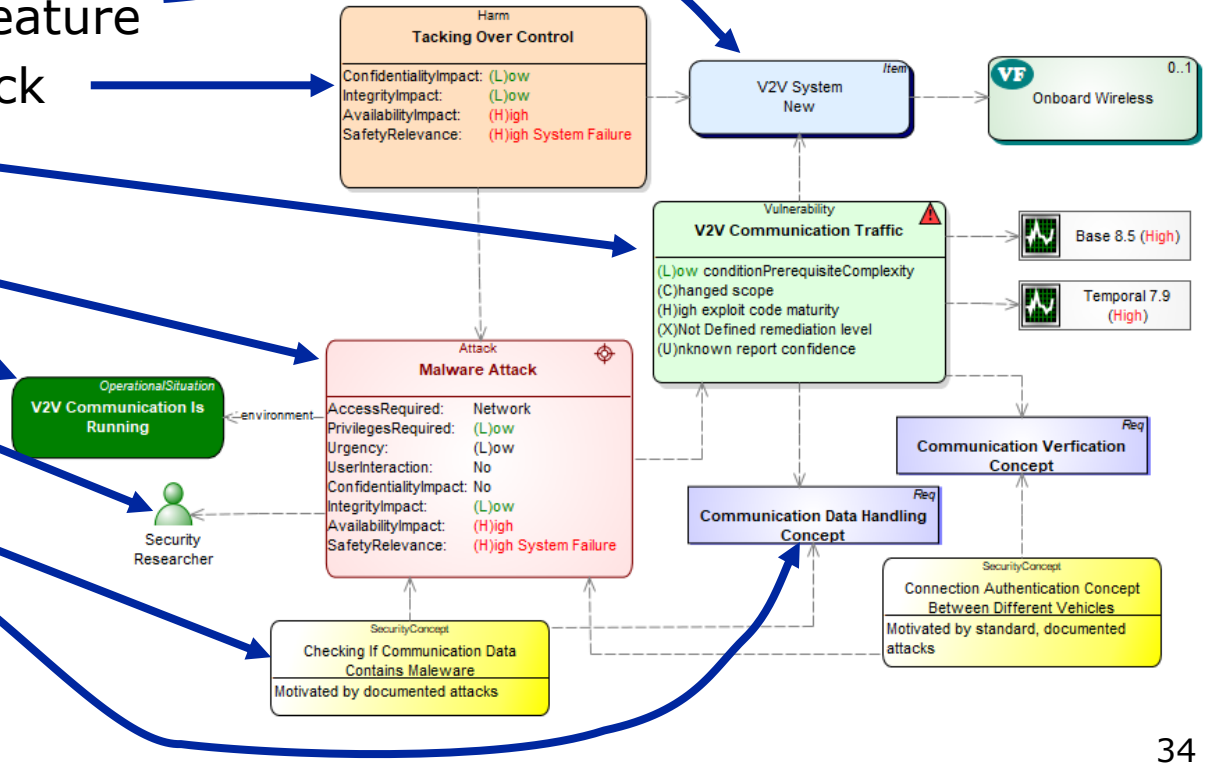


Modeling security

- セキュリティの関心事でモデル化する
  - E.g. Attack, Vulnerability, Adversary,
- セキュリティモデルをシステムの設計とメタモデルレベルで統合
  - 攻撃や脆弱性と車両の機能
  - 設計の修正を支援する双方向のトレーサビリティ
- 脅威分析の自動化
  - 脆弱性スコアの計算
  - セキュリティ脅威分析のレポートを生成

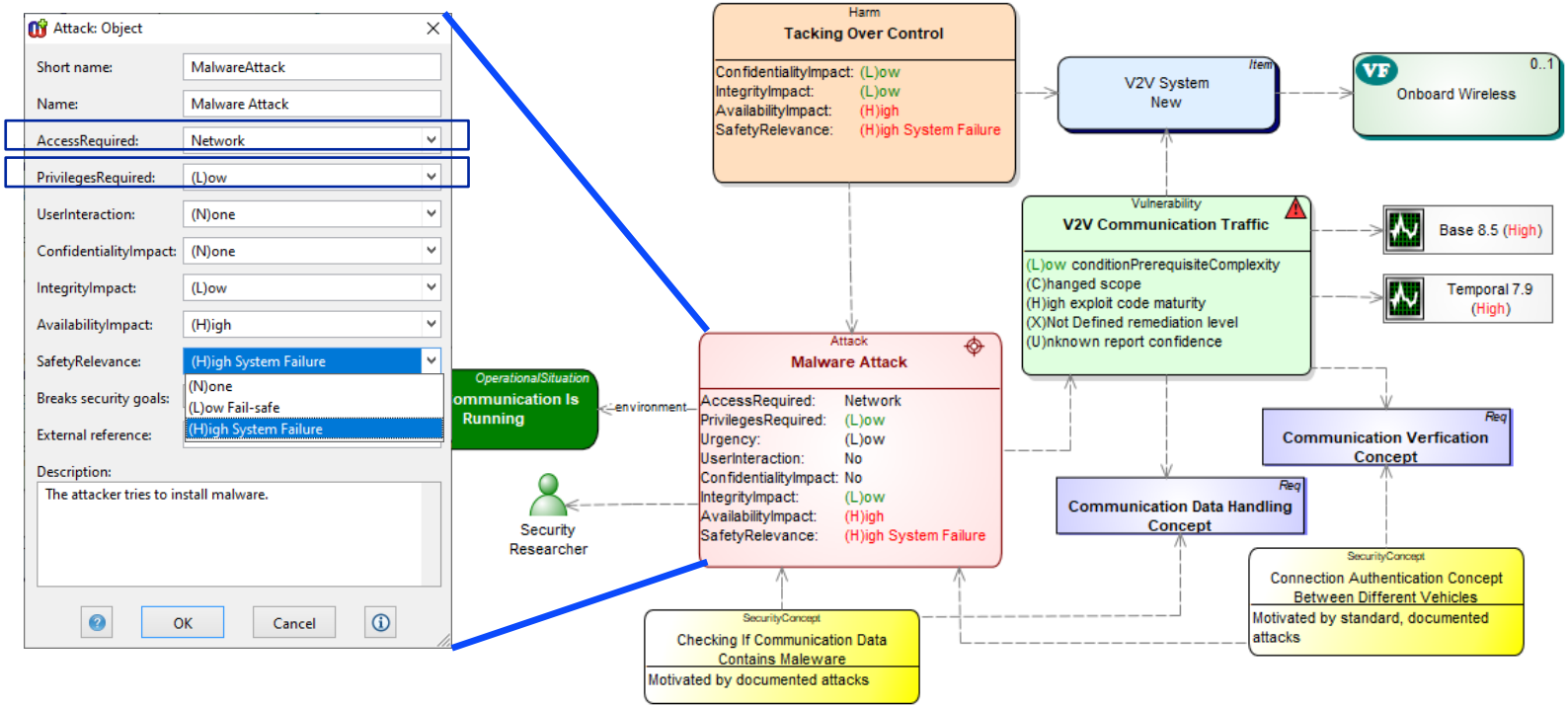
セキュリティモデルの例：車車間通信

- Aspects of security
  - Item and vehicle feature
  - Motivation for attack
  - Vulnerability
  - Attack
  - Situation
  - Adversary
  - Security concept
  - Requirement

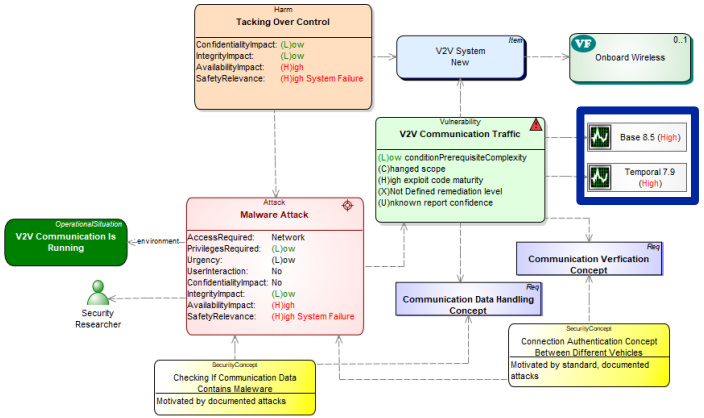


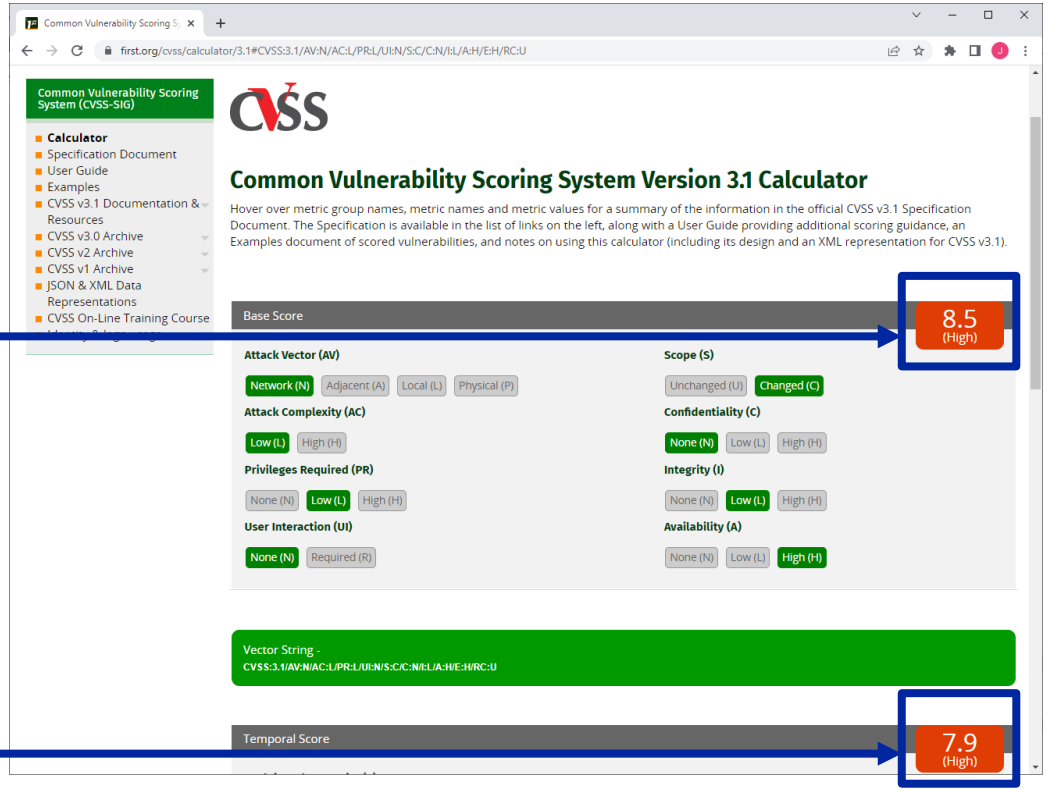
Security aspects details

- E.g. properties of attack:



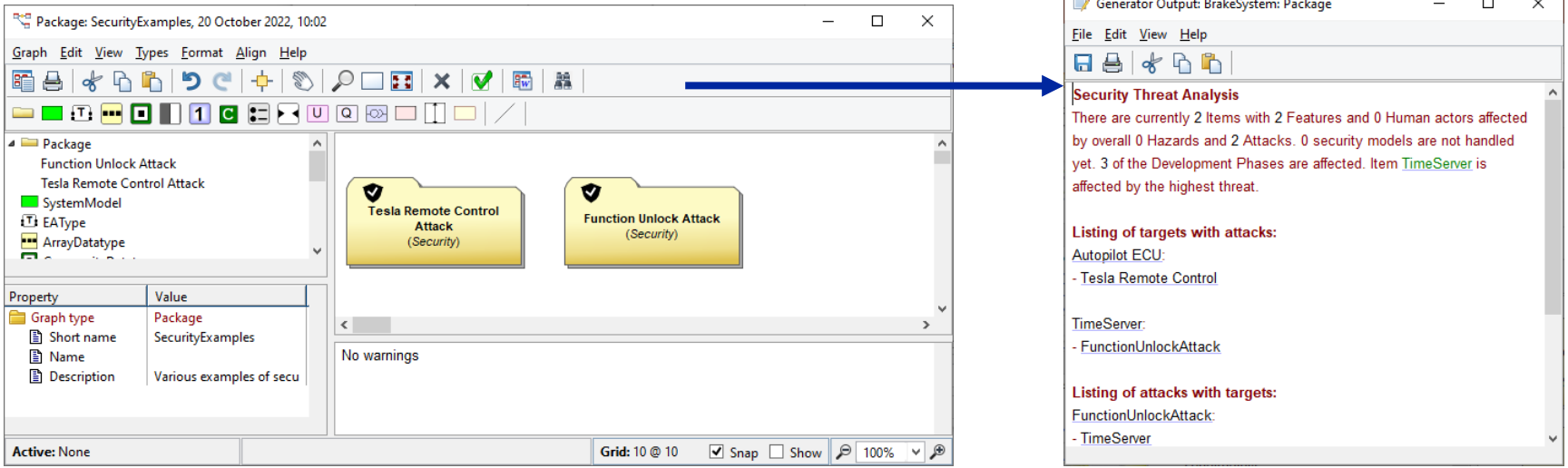
Calculate Vulnerability Scores (CVSS)



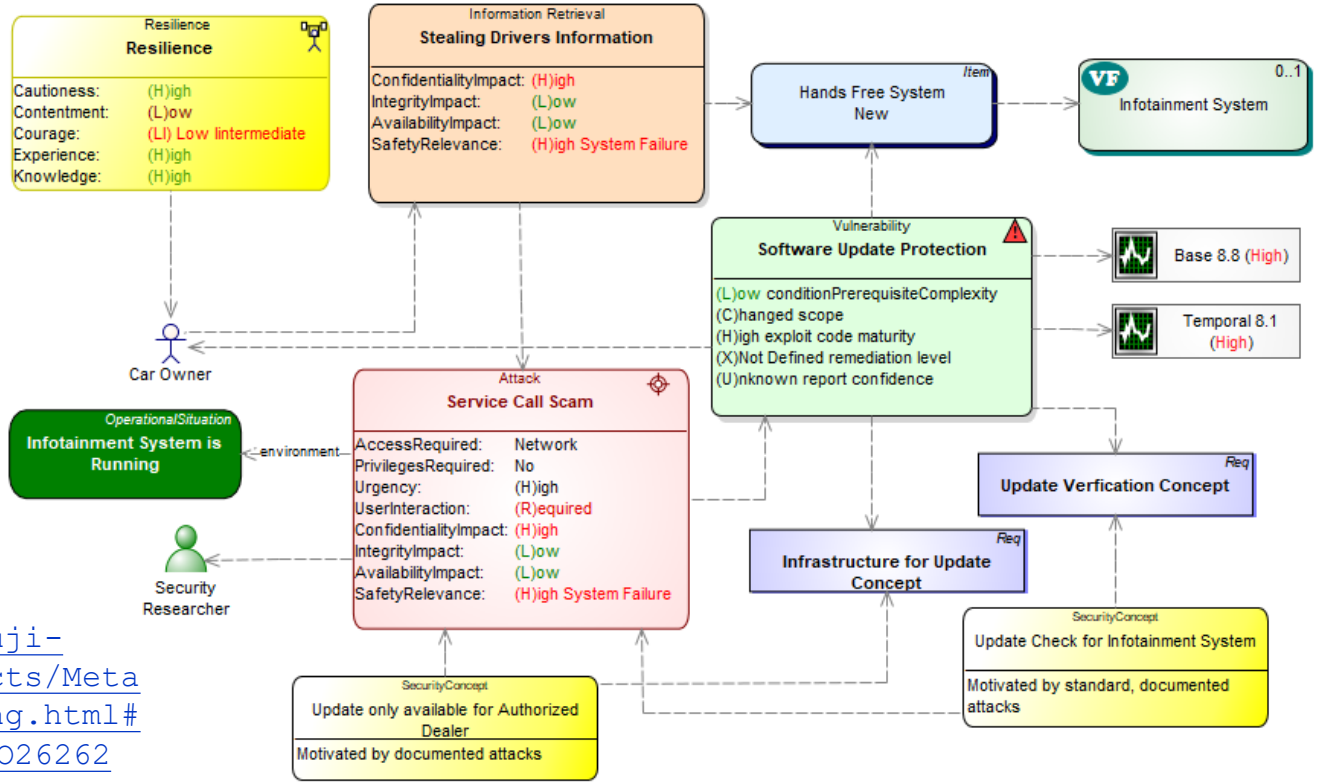


Reporting

- Security models can be reported with rest of the system designs



Modeling social engineering attacks



https://www.fuji-setsu.co.jp/products/MetaEdit/SystemModeling.html#SystemModelingISO26262

Agenda

- 車載システム開発の傾向と課題
- ドメイン固有モデリング言語
- ドメイン固有にMBSEを強化
  1. 同時コラボレーション
  2. Simulink 統合
  3. 様々な成果
  4. 安全性とセキュリティ
- まとめ

成果：安全性とセキュリティでMBSEを強化

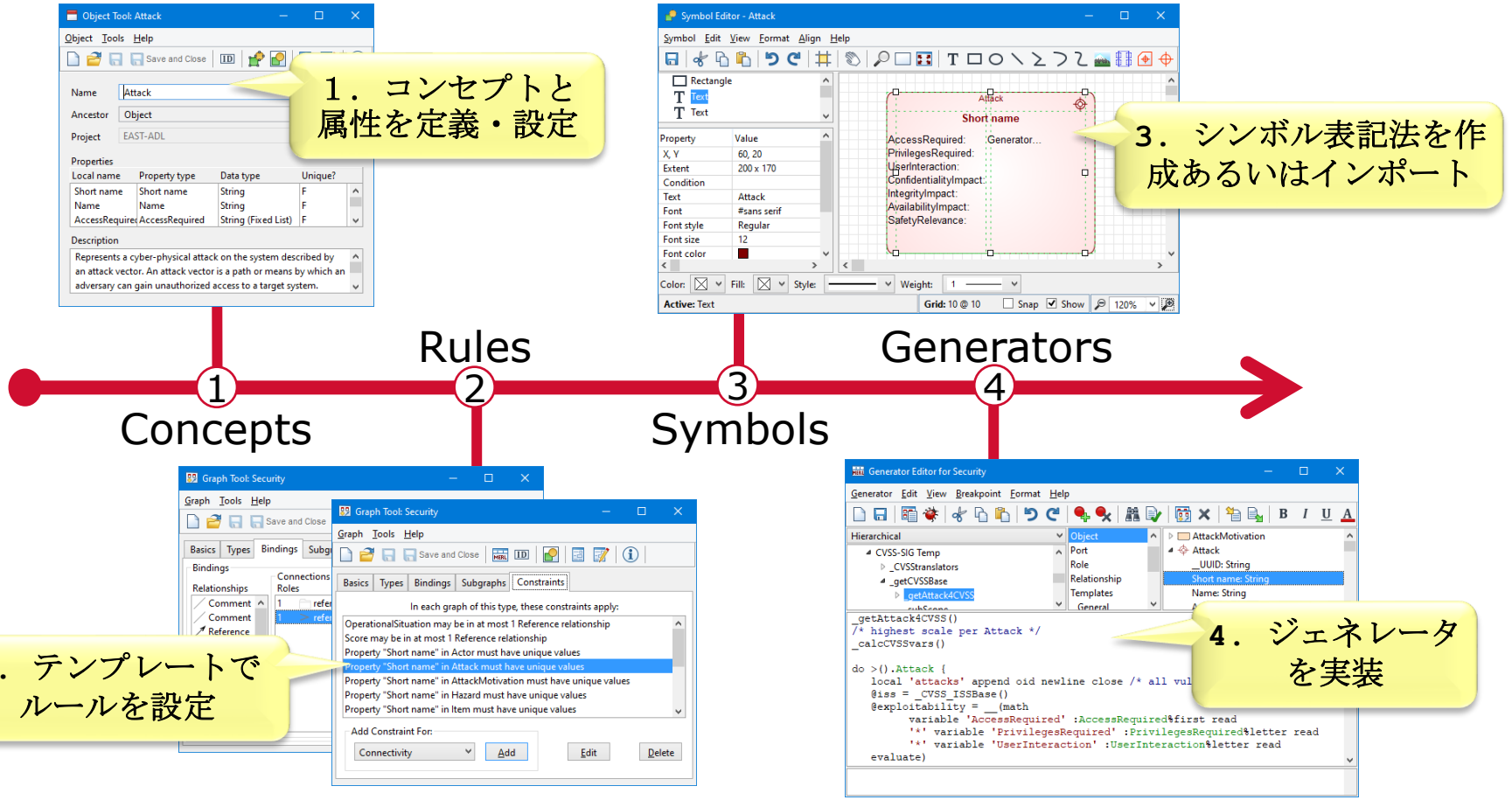
- あらゆる利害関係者で双方向にトレースして分析
  - システム設計または安全性/セキュリティモデルの変更を双方向にトレースして分析
- 同時コラボレーション
  - 両方の設計にアクセスして、高速フィードバックループを実現
- バージョン管理
  - 全てのモデルを一緒にバージョン管理
- 自動モデルチェックとモデル変換
  - フォールトツリー分析（FTA）、故障モード影響分析（FMEA）
  - 脆弱性スコア、セキュリティ脅威分析レポート

➢ **安全性やセキュリティの懸念がシステム設計と密接に関連する**

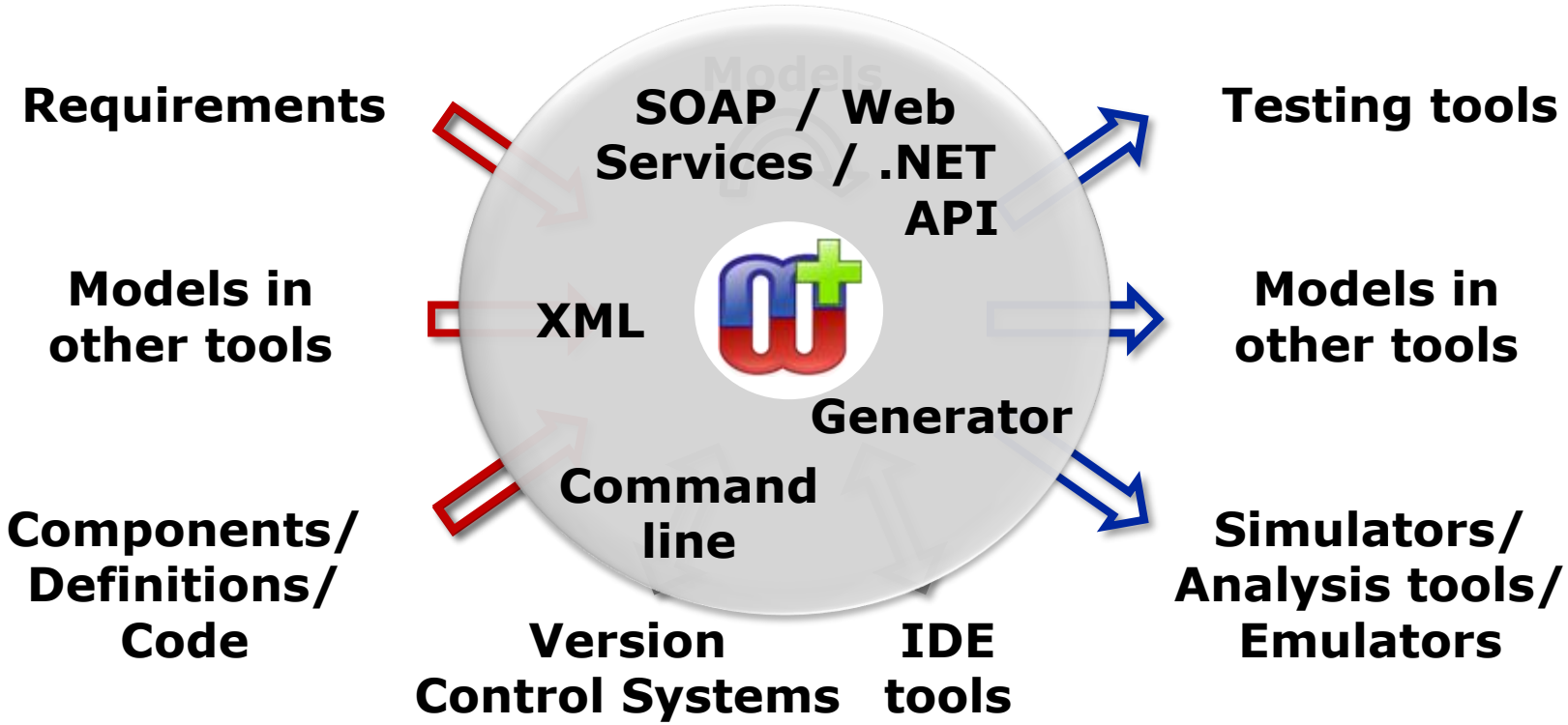
Experiences on language definition

- 手順:
  1. メタモデルを定義
  2. ルールや制約を設定して形式化を図る
  3. ドメインに適した表記法を定義
  4. ジェネレータを実装
     - モデルチェック, トレース, Simulink, FMEA and CVSS
- 高度なメタモデリングツールで柔軟かつ迅速に
  - セキュリティモデリングの拡張は、メタモデル、脆弱性スコアの計算等をカバーし、1 人の実働 1 週間で済みました

➢ **言語拡張は特定のモデリング言語に限定されない**



Integration scenarios



Examples

Imported:

- XML
- XMI
- XSD
- Java
- Python
- Simulink
- CDIF
- Conformiq
- C
- Excel
- DOORS
- PDF
- SVG

Simulators, Analyzers, Testing:

- Simulink
- QEMU
- Stateflow
- Modelica
- Labview
- SPIN
- UPPAAL
- Autosar
- FMU
- HIPHOPS
- ReqIF
- VLAB etc.

Generated:

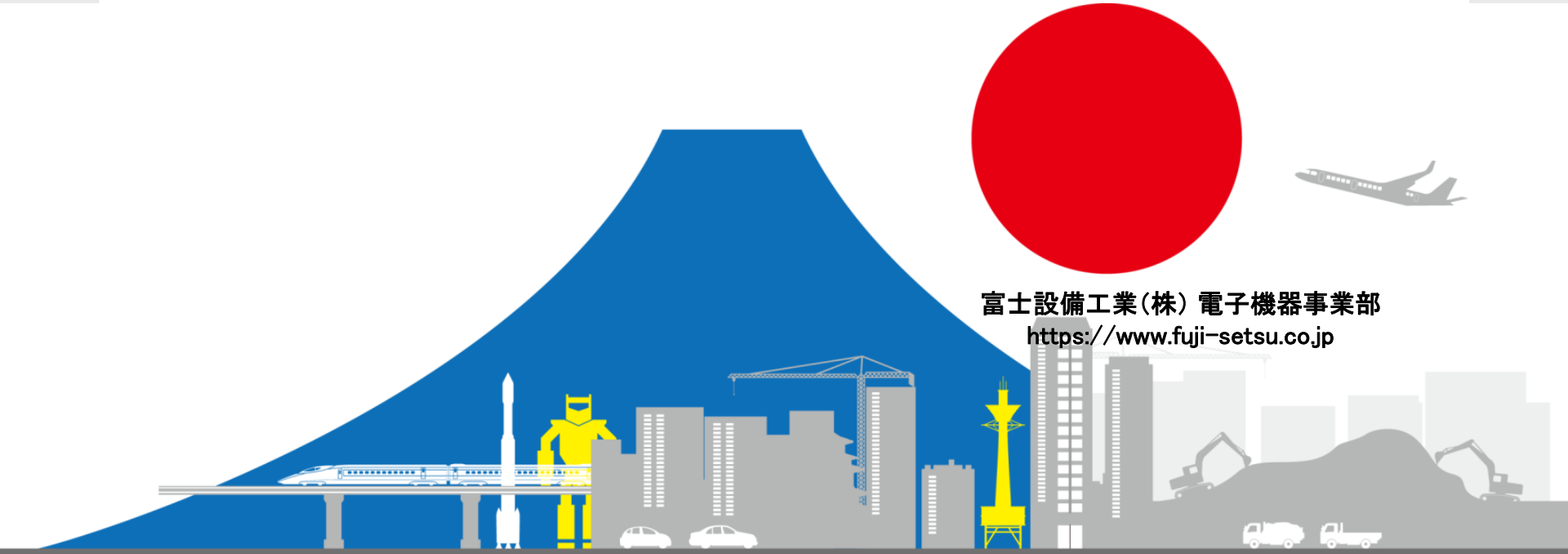
- Assembler
- C
- C++
- C#
- Corba
- Delphi
- Go
- HTML
- IEC 61131-3
- Java
- JavaScript
- MATLAB
- Pascal
- Perl
- Promela
- Python
- RDF
- Ruby
- Smalltalk
- SQL
- XMI
- XML, XSD
- …

MetaCase

FUJI SETSUBI

Domain-Specific Modeling について

https://www.fuji-setsu.co.jp/products/MetaEdit/tutorials.html



富士設備工業(株) 電子機器事業部
https://www.fuji-setsu.co.jp