システムの安全性を記述するためのモデリング言語「SafeML」

Geoffrey Biggs
産業技術総合研究所
知能システム研究部門

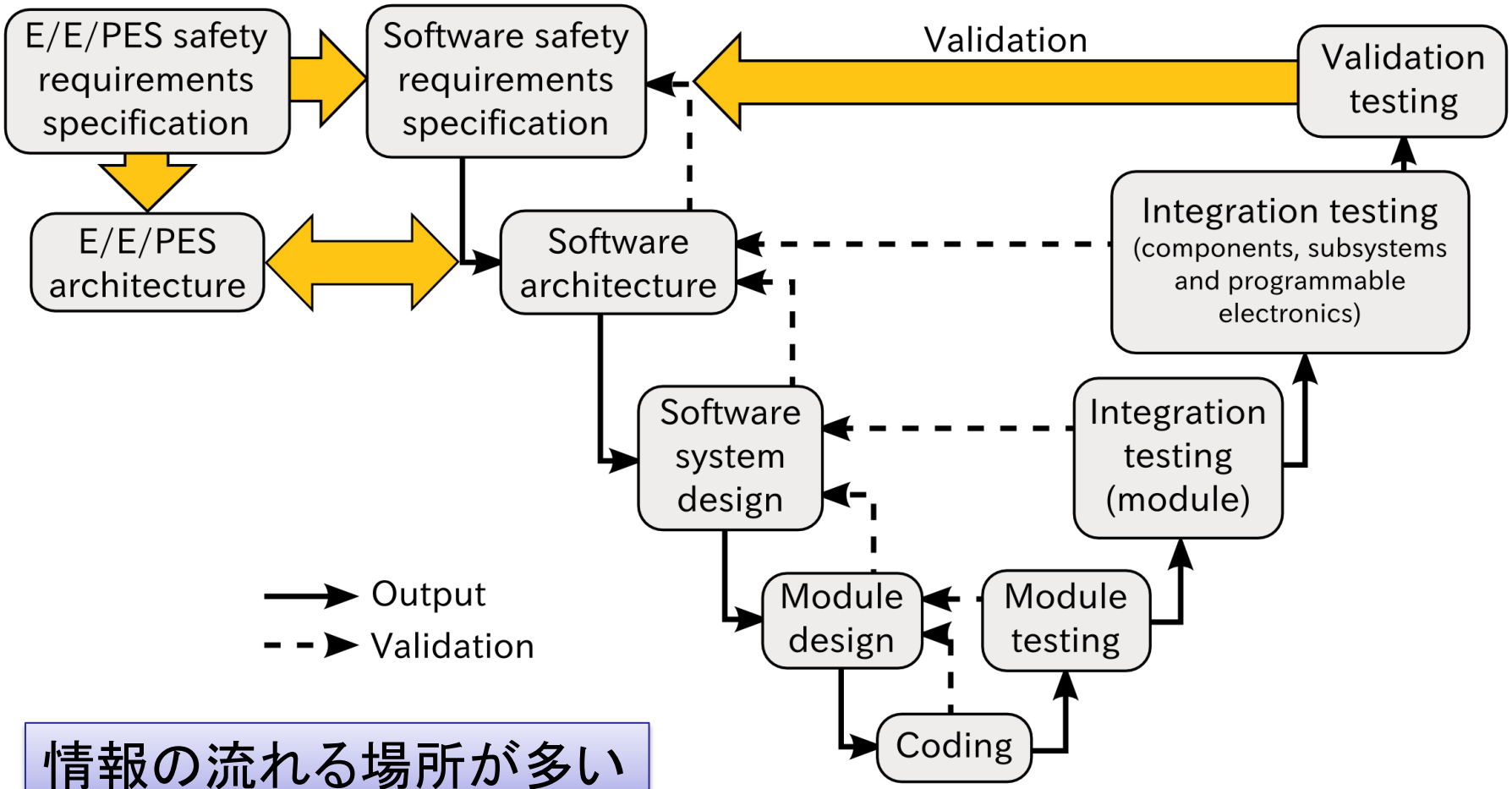
概要

- 高信頼システム開発の問題点
- モデリング言語による情報交換の必要性

- 安全モデリング言語 「SafeML」
  - 適用例
- SafeMLのツール

- まとめ

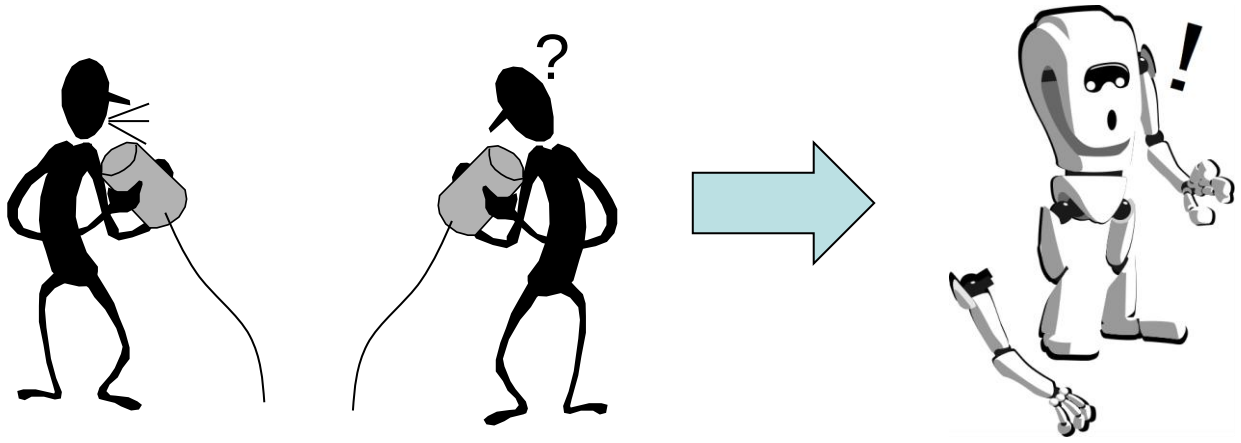
開発プロセス



情報の流れる場所が多い

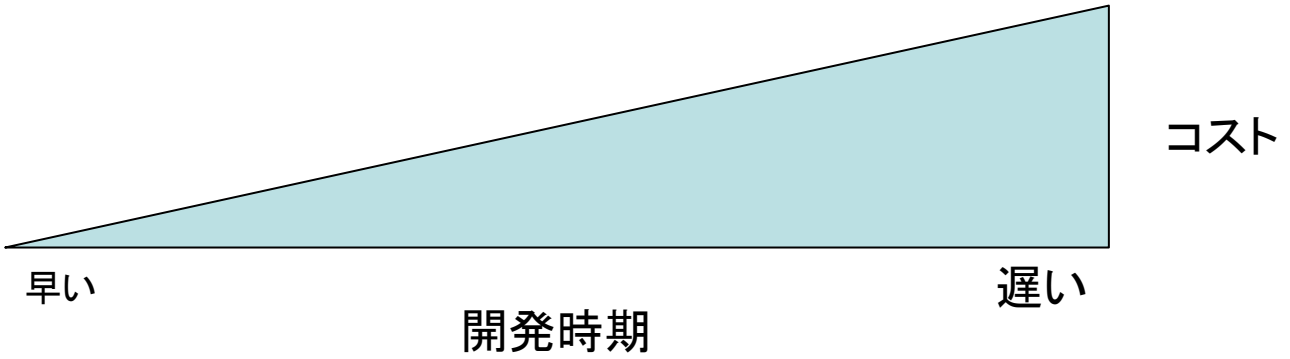
高信頼システム開発

- コミュニケーションギャップが問題
  - 要求エンジニアとソフトウェアエンジニア
  - 安全エンジニアとシステムエンジニア
- コミュニケーション不足はシステムの様々な欠陥の原因



欠陥

- 対策が必須
  - 直す
  - 防ぐ
- 対策が遅れれば遅れるほど、コストが上がる



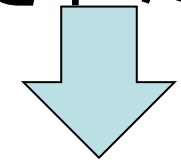
- コストが低く時期が早い段階で防ぐのが理想

コミュニケーション改善で欠陥防止

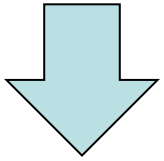
- 情報交換をより正確にすると
  - 各開発者の理解が上がる
  - 間違いが減る
- 開発プロセスの早い段階で欠陥を防げる

コミュニケーション改善で欠陥防止

- モデリング言語の適用でコミュニケーション改善は可能だと証明された

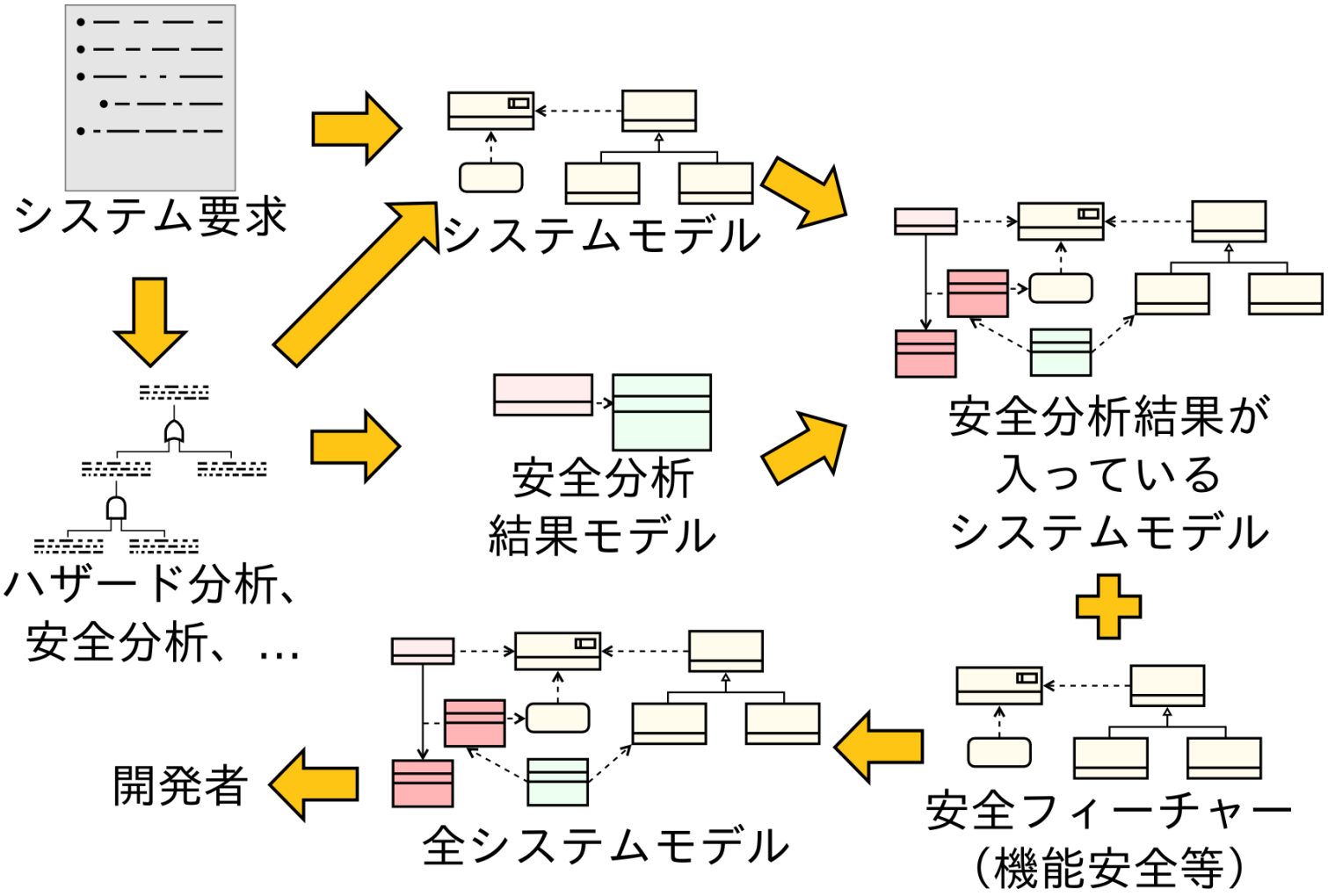


- 安全用のモデル言語で高信頼システムの安全情報を交換する

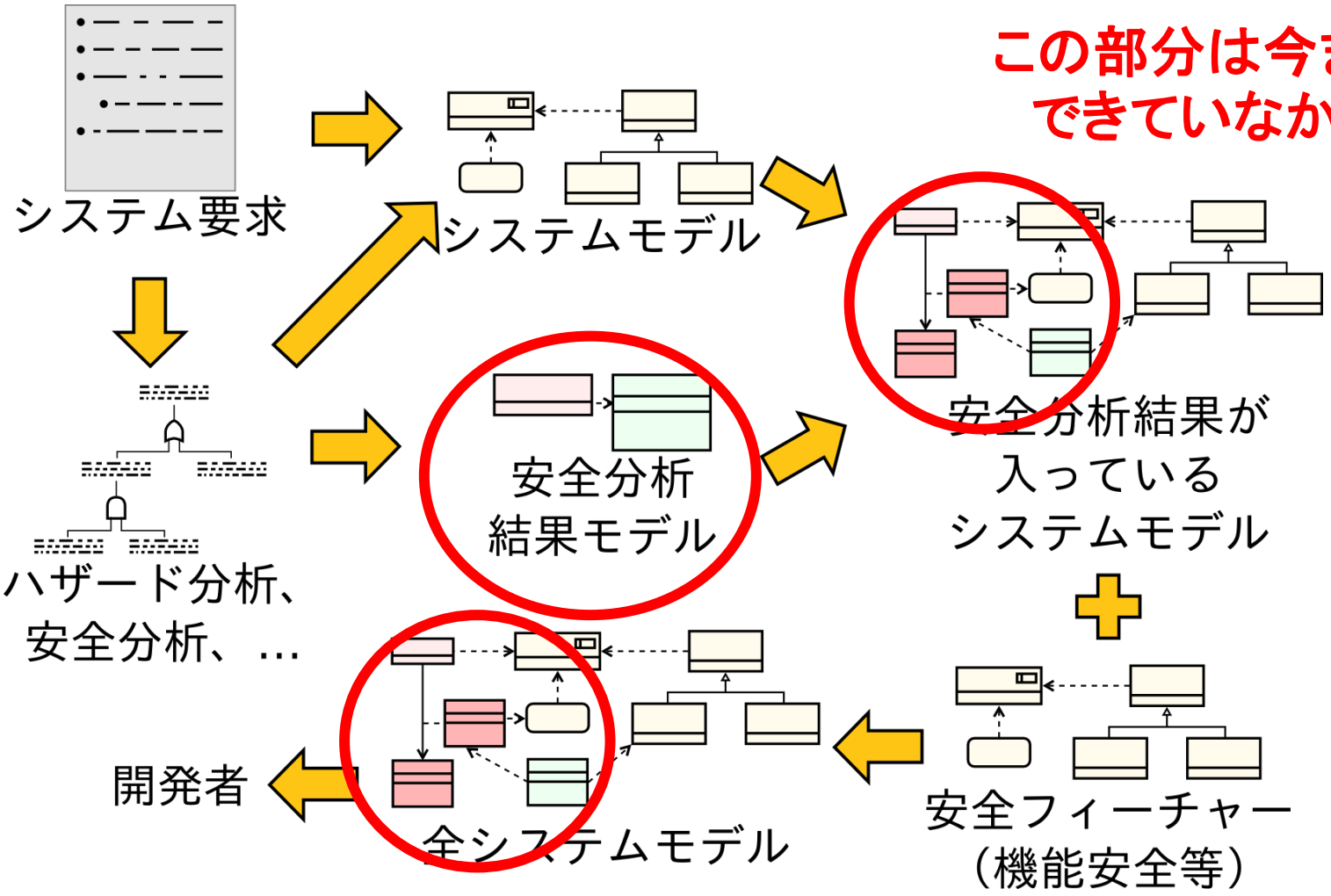


- より正確に安全情報を交換することによって欠陥を防ぐ

モデル言語で安全情報交換



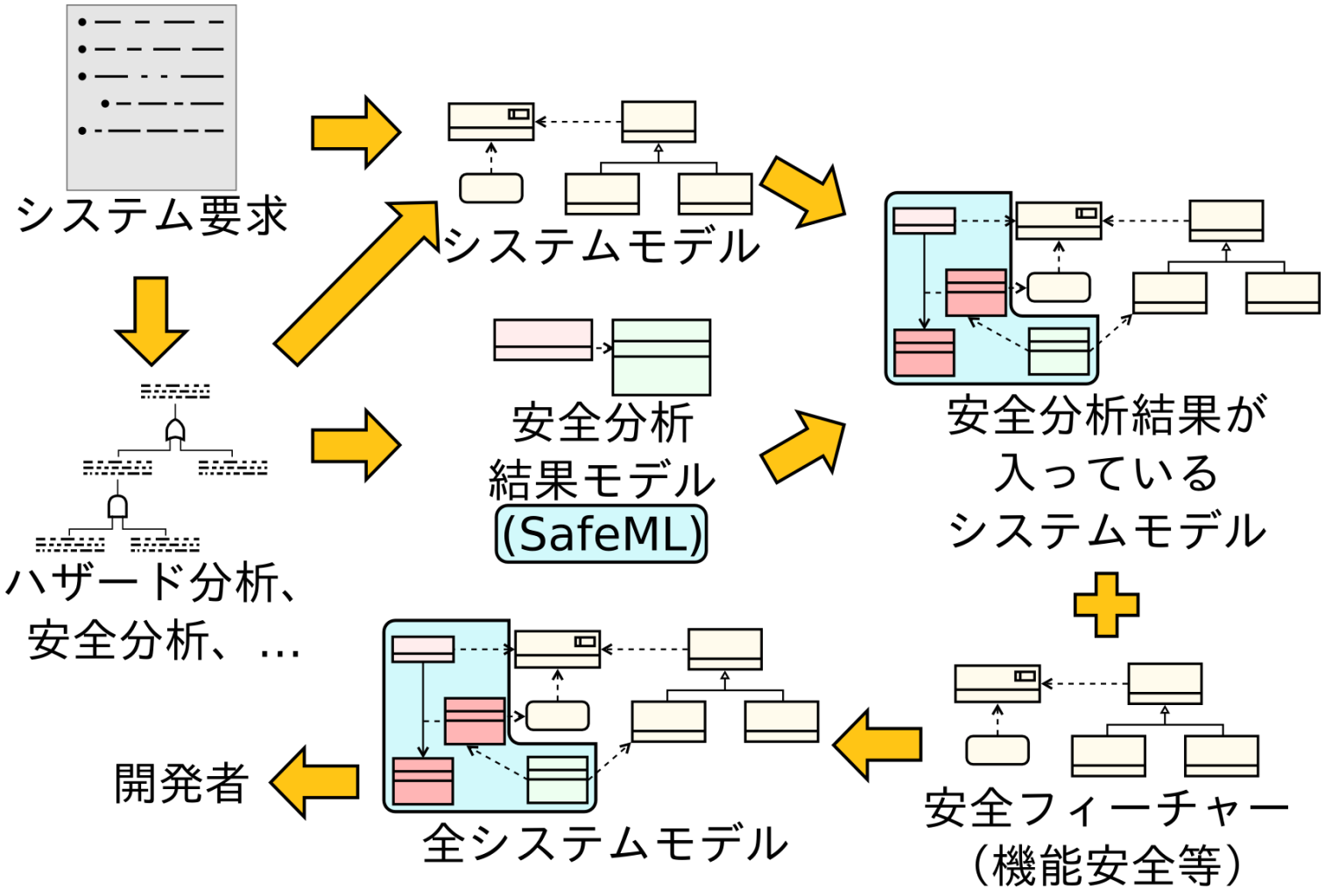
モデル言語で安全情報交換



SafeML

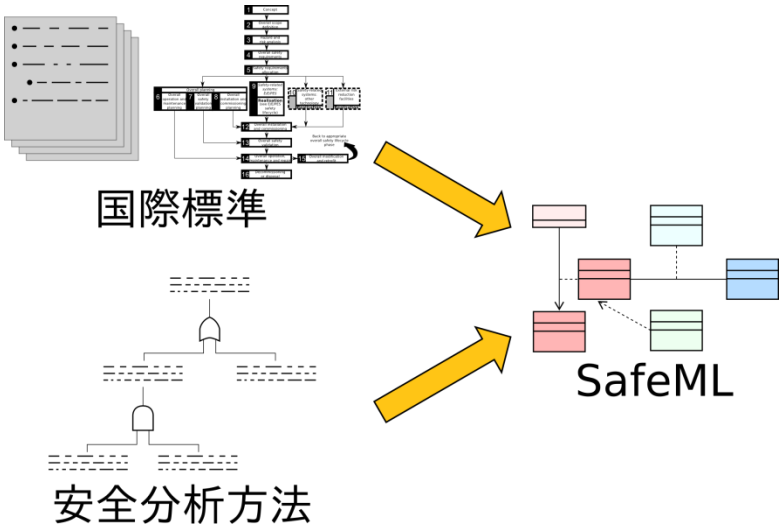
- システムの安全に関する情報を記述するためのモデリング言語
  - システムのハザード
  - 安全要求
  - 安全フィーチャー（機能安全など）

SafeMLの使う開発流れ



SafeML

- SysMLのプロファイル
- 開発チーム内のコミュニケーションの道具
- 安全標準と安全分析方法に基づいた
  - 安全分析の結果と安全機能をモデル化する



SafeMLのコアコンセプト

- ハザード情報
  - Hazard, Harm, HarmContext
  - ハザードはあるコンテキストである危害を起こす
- コンテキストの検出方法
  - ContextDetector
- 防衛方法
  - PassiveDefense, ActiveDefense, DefenseResult
  - 防衛方法によってコンテキストや危害を防ぐ

SafeMLのコアコンセプト

- Tagによって、SafeMLの要素に情報を追加する
  - コンテクストの確立
  - 危害の確立
  - Severity
  - …
- 開発支援ツールでTagを利用する
  - ある危害の合計確率の計算
  - 表や報告生成
  - …

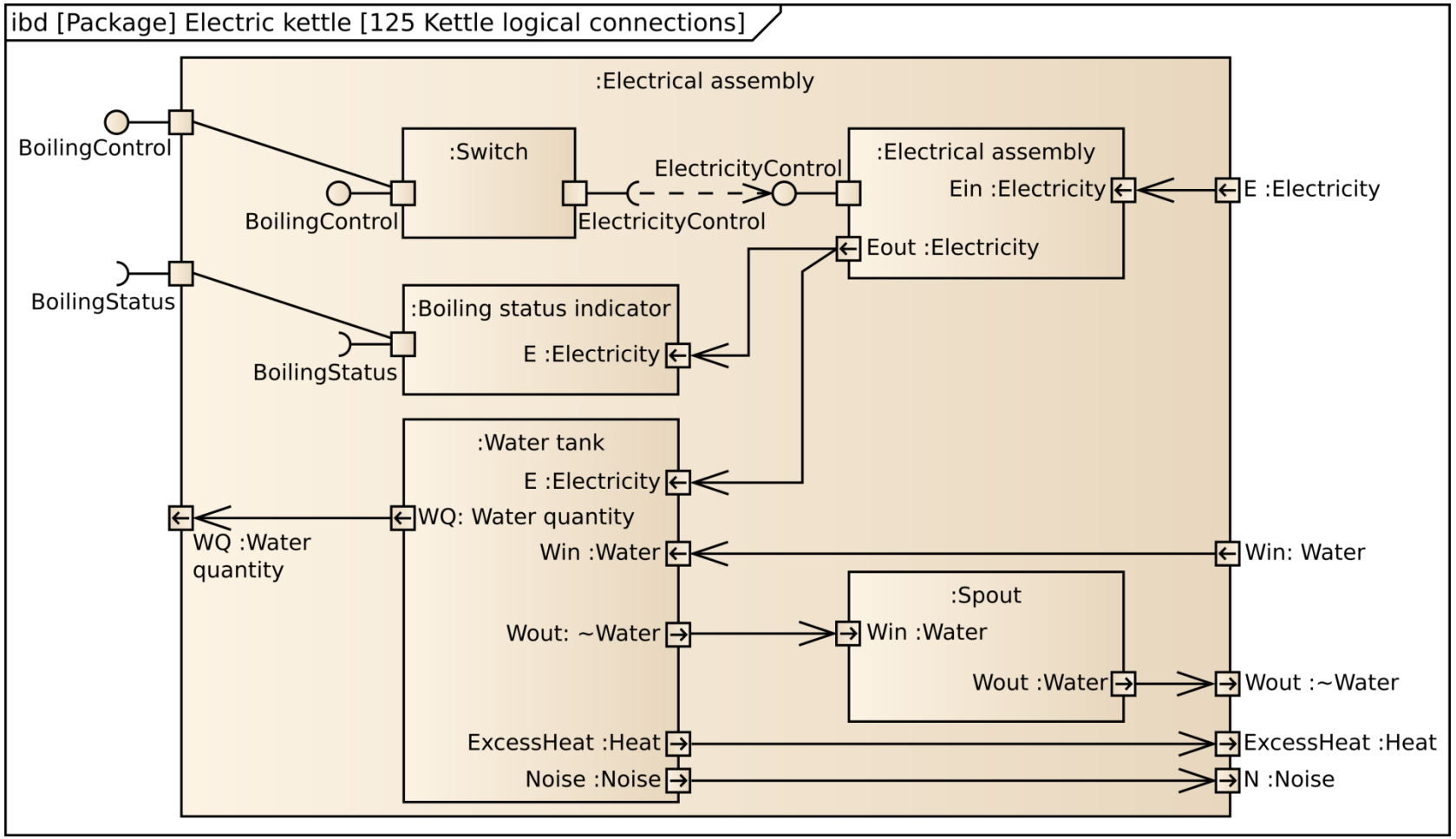
例

- 例：電気ポット
  - 簡単
  - UML勉強によく使われる例
- 既存のシステムの分析、改善

プロセスの例

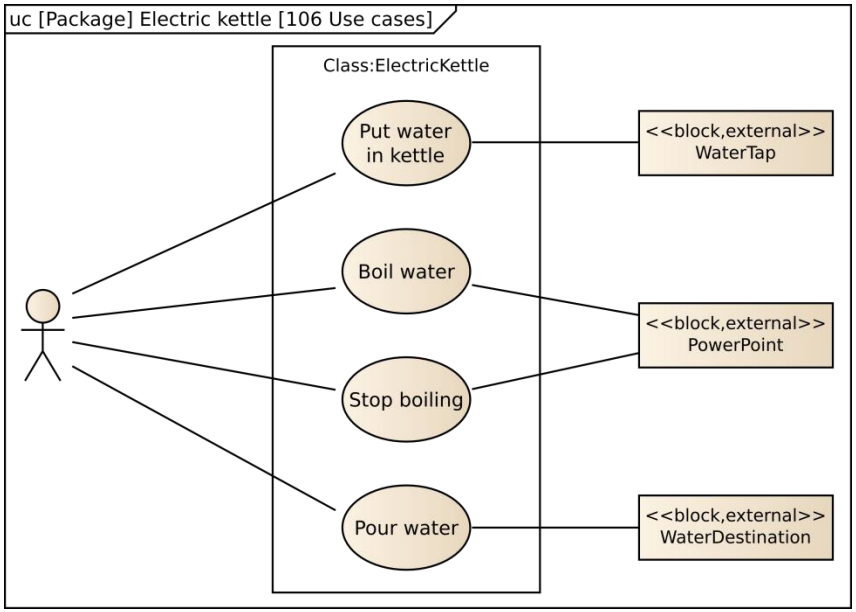
1. • 要求とユースケースの特定 → SysMLでモデル化
2. • ハザード分析、FTA → SafeMLでモデル化
3. • 安全フィーチャの設計 → SafeMLでモデル化
4. • モデルで安全フィーチャを設計に反映する
5. • 必要に応じて反復

電気ポットの例 – システム設計



（モデルの一部だけ）

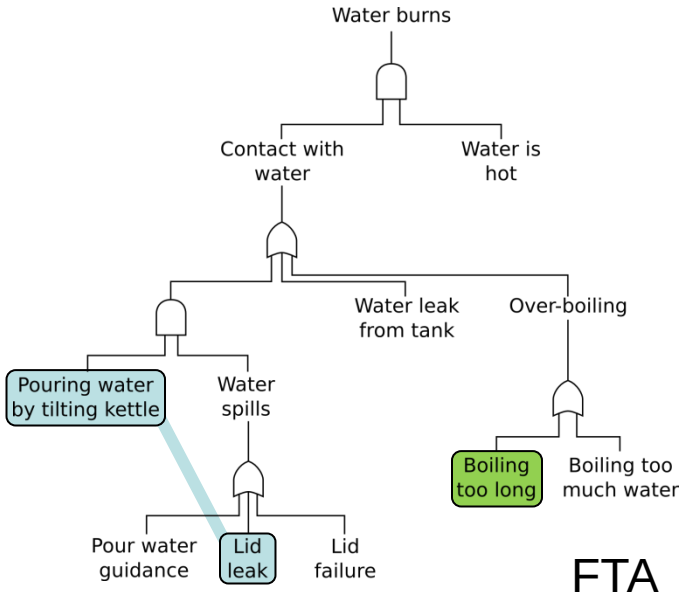
電気ポットの例 – 安全分析



ユースケース

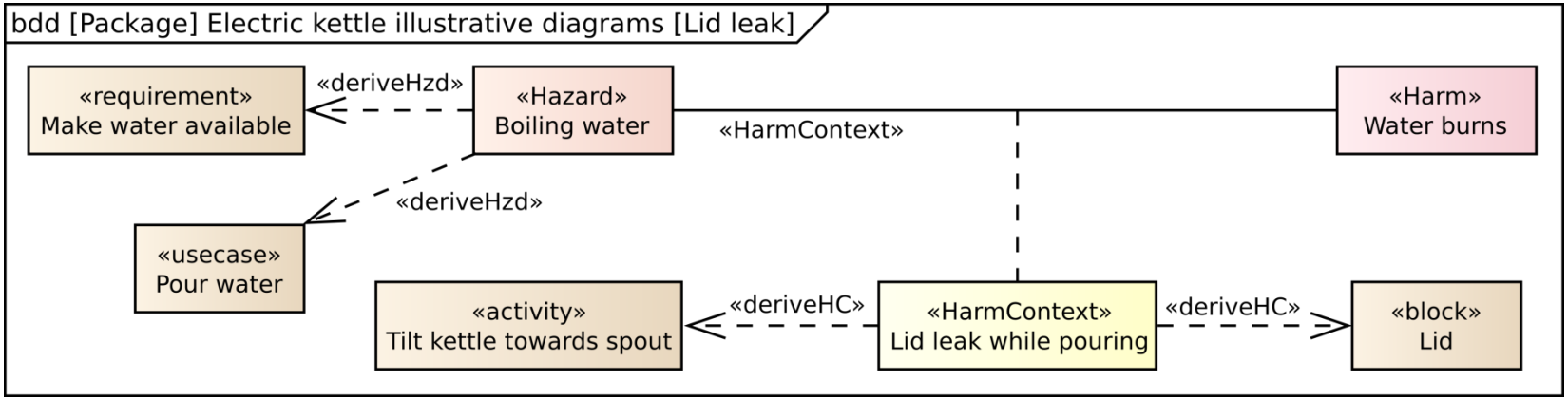
ハザード	危害
電気	感電、火事
熱	やけど、火事
蒸気	やけど
湯	やけど

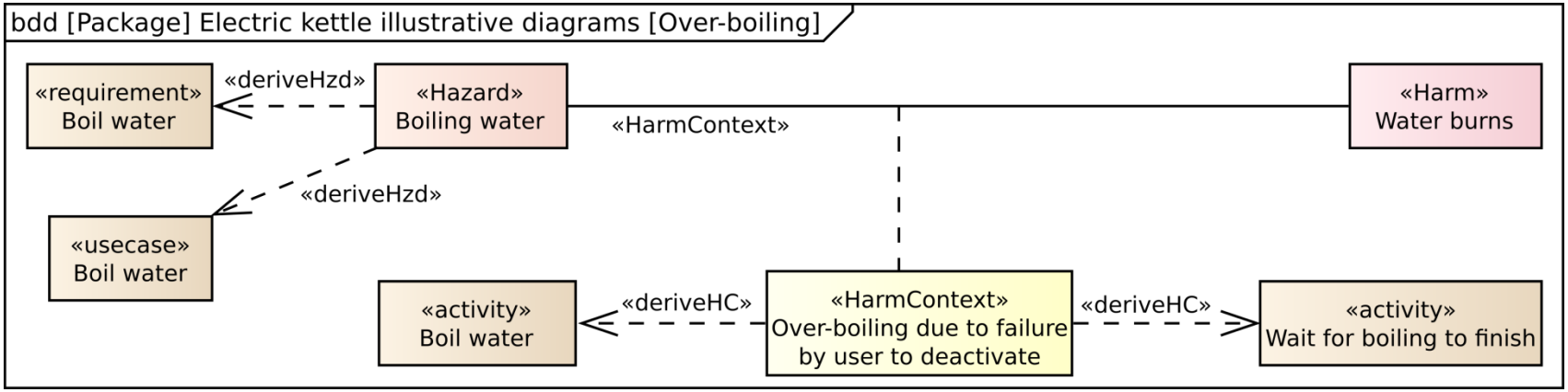
ハザード分析



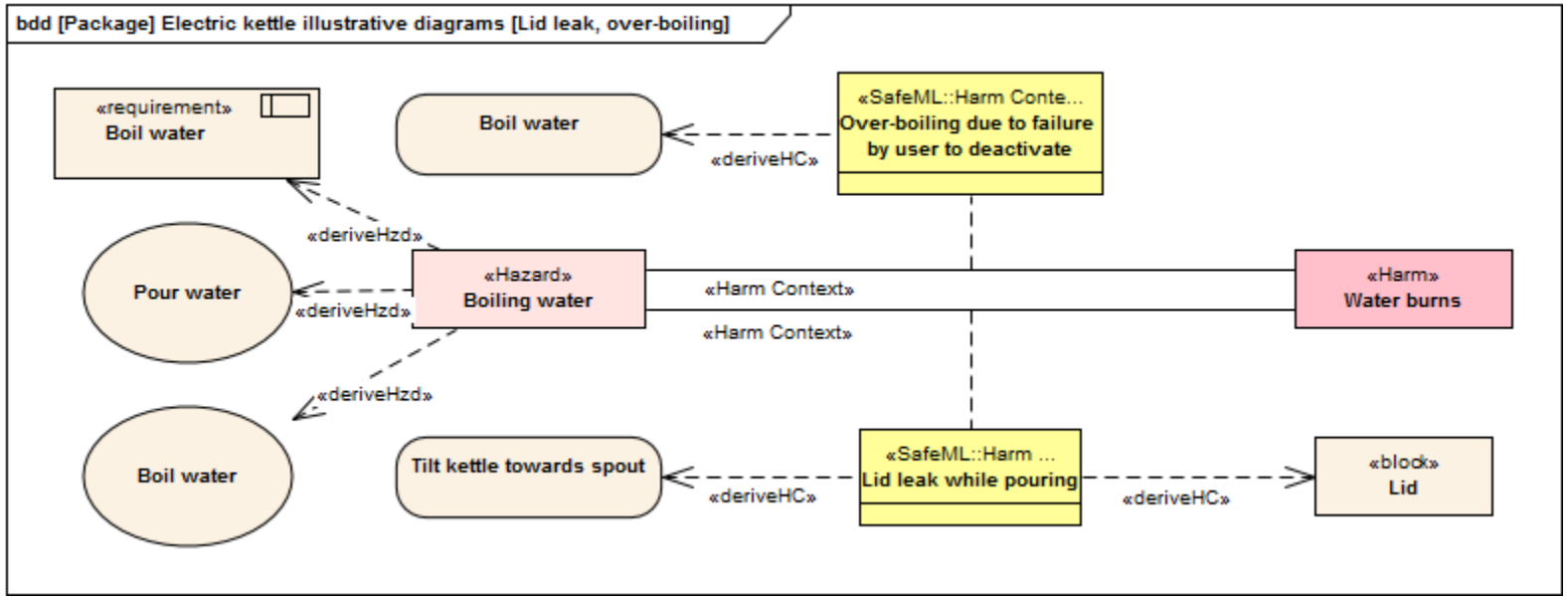
FTA

電気ポットの例 – SafeMLで安全分析結果をモデル化





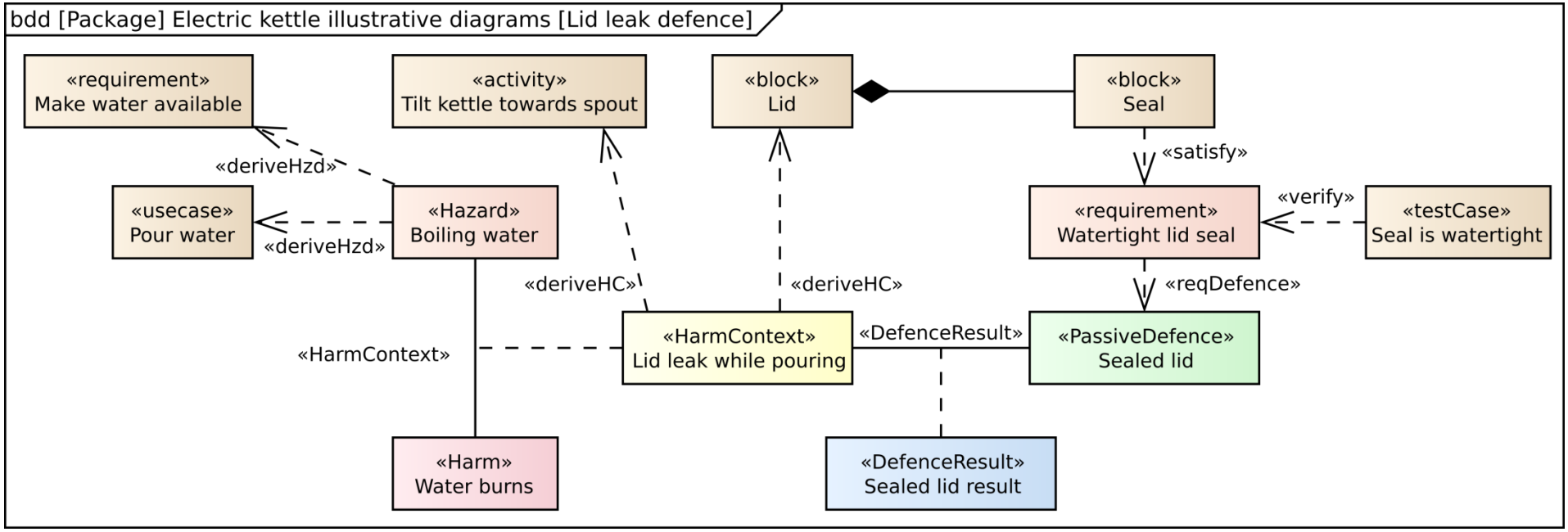
（図を分けるベネフィット）



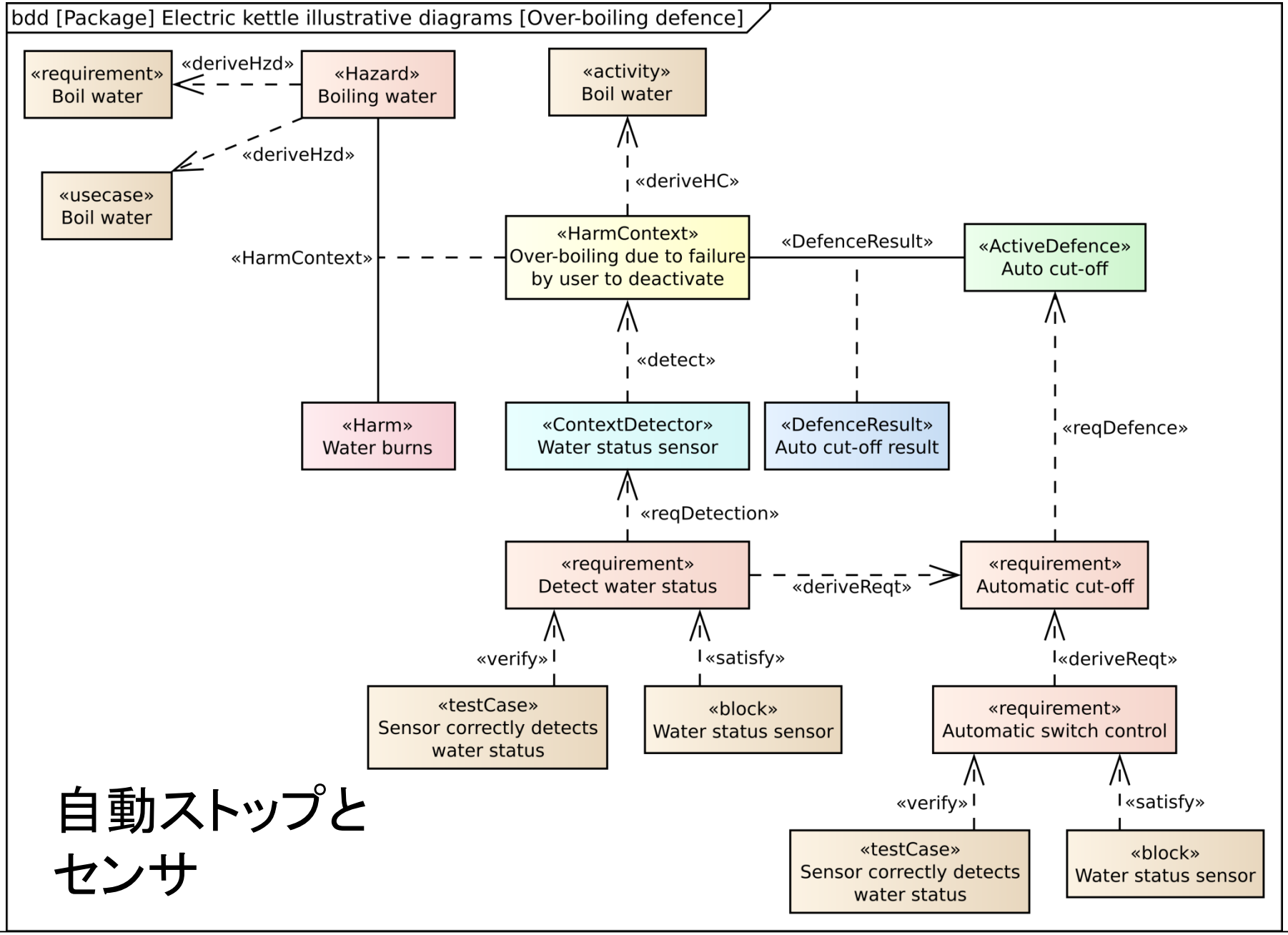
電気ポットの例 – 安全フィーチャ

- 湯漏れ
  - 防衛 : ふたにゴムパッキン
- 沸きすぎ
  - 防衛 : 自動ストップ
  - センサが必要

電気ポットの例 – 安全フィーチャのモデル化

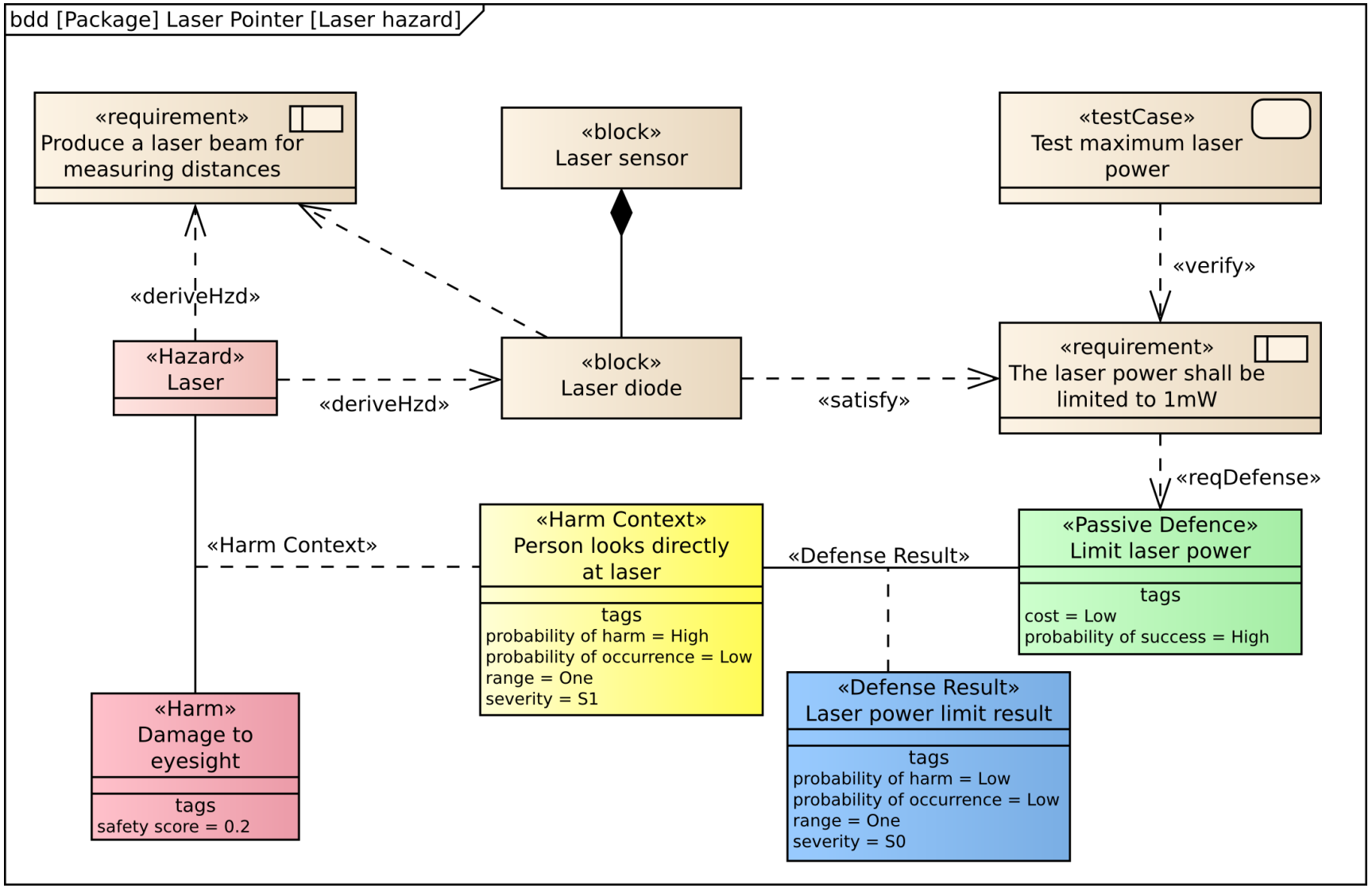


ふたのゴムパッキン

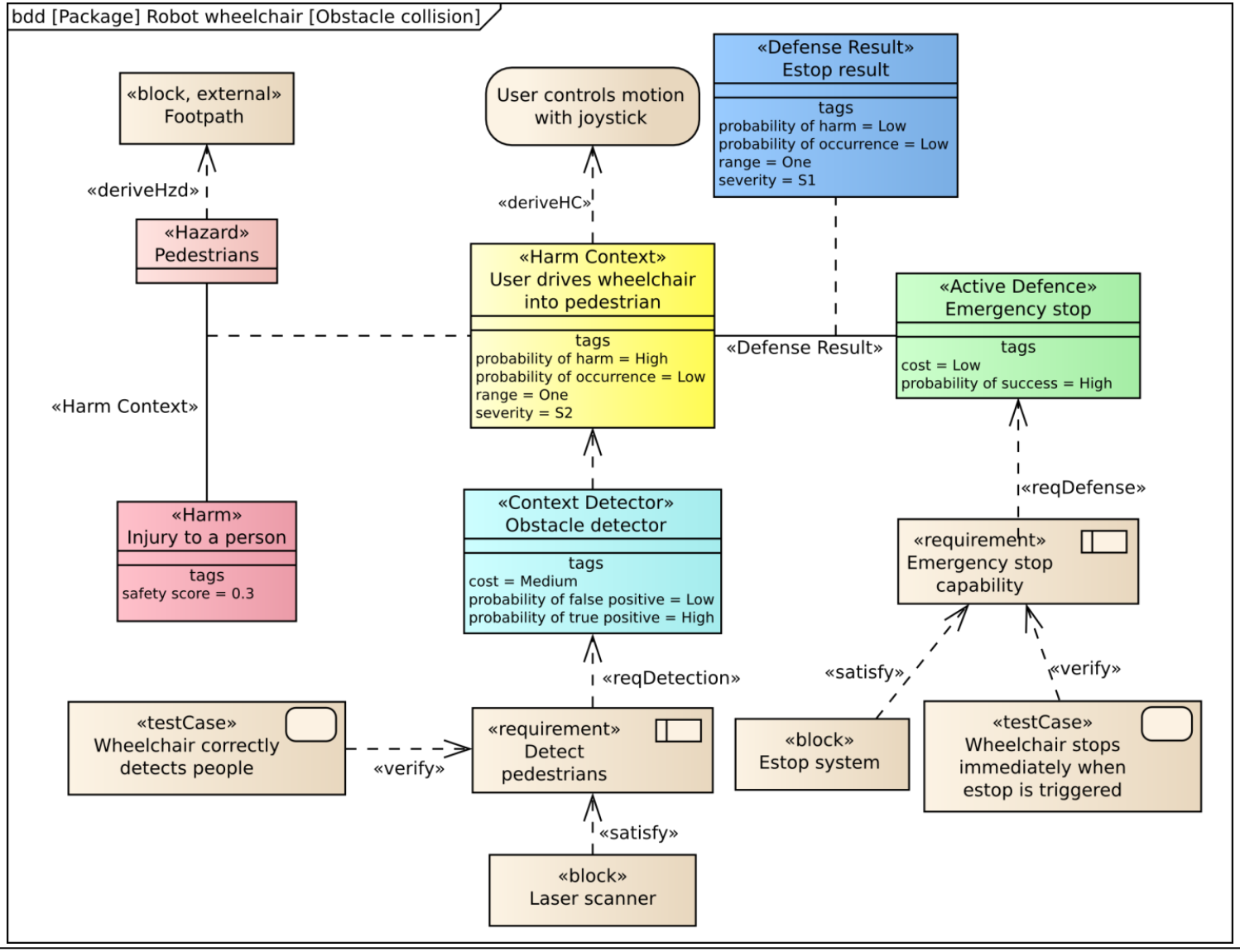

bdd [Package] Electric kettle illustrative diagrams [Over-boiling defence]
«requirement»
Boil water
«deriveHzd»
«Hazard»
Boiling water
«activity»
Boil water
«deriveHzd»
«usecase»
Boil water
«deriveHC»
«HarmContext»
«HarmContext»
Over-boiling due to failure
by user to deactivate
«DefenceResult»
«ActiveDefence»
Auto cut-off
«detect»
«Harm»
Water burns
«ContextDetector»
Water status sensor
«DefenceResult»
Auto cut-off result
«reqDefence»
«reqDetection»
«requirement»
Detect water status
«deriveReqt»
«requirement»
Automatic cut-off
«verify»
«satisfy»
«deriveReqt»
«testCase»
Sensor correctly detects
water status
«block»
Water status sensor
«requirement»
Automatic switch control
«verify»
«satisfy»
«testCase»
Sensor correctly detects
water status
«block»
Water status sensor

自動ストップとセンサ

実用例2:レーザのパワー制限



実用例3： 車椅子ロボット



開発支援ツール

- モデリング言語の利用を支援する
  - SafeMLメタモデルとの整合性を検証するモデルチェッカ
  - SafeMLの要素間の関係を表形式で示す
  - レポートの自動生成

開発支援ツール：SafeMLによる安全情報を表で表した例

Harm	HarmContext	Defence	P(Occur)	P(Harm)
Water burns	Over-boiling due to failure by user to deactivate	Undefended case	Low	Low
		Auto cut-off	Low	Low
	Over-boiling due to too much water	Undefended case	Medium	Low
		Water level sensor	Low	Low
		Water limit indication mark on tank	Low	Low
	Latch failure while pouring	***Undefended case***	***Low***	***High***
	Lid leak while pouring	Undefended case	Low	Low
		Sealed lid	Low	Low
	Poor water guidance while pouring	Undefended case	Medium	High
		Shaped spout	Low	High
Electrocution	Contact with live electrical components	Undefended case	Low	High
		Electrical insulation	Low	High
Electrical fire	Contact between water and live electrical components	Undefended case	High	Medium
		Waterproofing	Low	Medium
	Short circuit of live electrical components	Undefended case	Low	High
		Electrical insulation	Low	High
Heat burns	Contact with tank	Undefended case	Medium	High
		Heat insulation	Low	Low
	Contact with element	***Undefended case***	***Low***	***High***
Steam burns	Steam build-up from boiled water	Undefended case	High	Medium

開発支援ツール：SafeMLによる安全情報を表で表した例

	P(Occur)	P(Harm)	Range	Severity	P(Success)	SafetyScore	Cost
	Low	Low	One	S1	---	0.0123	---
	Low	Low	One	S1	High	0.0123	
	Medium	Low	One	S1	---	0.0246	---
	Low	Low	One	S1	High	0.0123	
ank	Low	Low	One	S1	Medium	0.0123	
	Low	***High***	***Few***	***S1***	---	***0.074***	---
	Low	Low	One	S1	---	0.0123	---
	Low	Low	One	S1	High	0.0123	
	Medium	High	One	S1	---	0.074	---
	Low	High	One	S1	High	0.037	
	Low	High	One	S3	---	0.1111	---
	Low	High	One	S3	High	0.1111	
	High	Medium	Many	S3	---	0.6666	---
	Low	Medium	Many	S3	High	0.2222	
	Low	High	Many	S3	---	0.3333	---
	Low	High	One	S3	High	0.1111	
	Medium	High	One	S2	---	0.1481	---

開発支援ツール：Defenses Matrix

	Over-boiling due	Over-boiling due	Latch failure whil	Lid leak while pou	Poor water guida	Contact with live	Contact between	Short circuit of li	Contact with tan	Contact with ele	Steam build-up fr
Auto cut-off	0.0123										
Water level sensor		0.0123									
Water limit indication mark on tank		0.0123									
Sealed lid				0.0123							
Shaped spout					0.037						
Electrical insulation						0.1111		0.1111			
Waterproofing							0.2222				
Heat insulation									0.0123		
Steam venting											0.0246

防衛が不十分な点などを発見することができる

開発支援ツール：レポートの自動生成

Harms Report Document

for

Electric kettle 1.0

Model date: 2013/02/15
Report date: 2013/02/15

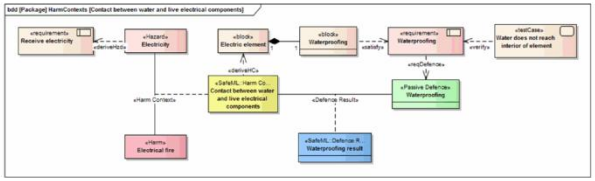
Model author(s): **Geoffrey Biggs[System engineer],Takeshi Sakamoto[System engineer]**
Model created: **2012/04/17**

Report generated by: **Geoffrey Biggs**
Position: **System engineer**

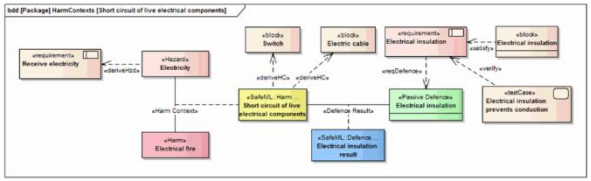
Signature: _____

1.Electrical fire
Safety score: 0.67
Hazard: **Electricity**
Sources:

1.1.Harm Context: Contact between water and live electrical components
Probability of occurrence: High
Probability of harm: Medium
Range: Many
Severity: S3
Sources:
- [Block] Electric element



1.2.Harm Context: Short circuit of live electrical components
Probability of occurrence: Low
Probability of harm: High
Range: Many
Severity: S3
Sources:
- [Block] Switch
- [Block] Electric cable



SafeMLの利点

- 安全分析とシステム設計の関連付け
  - ハザード、危害、防衛機能等の定義
- 安全機能のシステム実装への反映
  - ハザードから防衛機能実装へのトレーサビリティの確保
- 既存のSysMLツールや開発プロセスで利用できる

まとめ

- コミュニケーション不足はシステムの様々な欠陥の原因
  - 高信頼システムでは特に危険
- モデリング言語「SafeML」で安全情報をより分かりやすくする

- SafeMLで、システムの設計モデルと一緒に安全情報を記述することが可能