SysMLを拡張したSafeMLで、システムの安全性をモデル化・設計しよう

Geoffrey Biggs
産業技術総合研究所
知能システム研究部門
ディペンダブルシステム研究グループ

この発表の目的

- 安全情報をモデル化することの利点を理解する
- SafeMLを用いた安全情報管理方法とコミュニケーションを支援する方法
- SafeMLを用いた安全分析・安全設計のモデル化方法
- SafeMLの使用を支援するツールのポテンシャルについて

概要

- 安全なシステムとは
  - ハザード分析、機能安全等のコンセプト
- SafeML
  - SafeMLのコンセプトとモデリング言語の要素
- SafeML適応の例
  - 例によりSafeMLの使い方を学ぶ
- ツール
  - SafeMLモデルを処理するツールについての議論

Part 1

Safe systems
（安全なシステム）

概要

- 安全のコンセプト
  - Hazard（危険）、Harm（危害）、Context（危険状況、危険事象）
- 安全分析手法
  - FTA、FMEA等
- 安全対策
  - Intrinsic safety
  - Functional safety（機能安全）

Hazards, harms and contexts

危険：危害の潜在源

Hazard: Potential source of harm

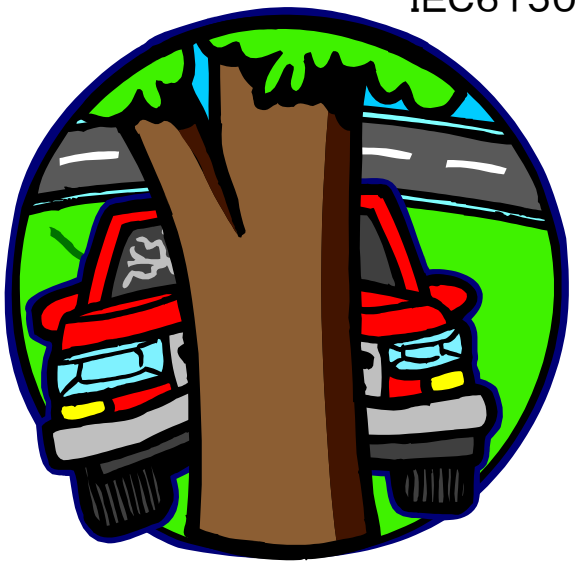
IEC61508－4より

Hazards, harms and contexts

危害：財産又は環境の損傷の結果、直接的又は間接的に与えられる、人の健康に対する肉体的傷害又は損傷

Harm: Physical injury or damage to the health of people or damage to property or the environment

IEC61508－4より

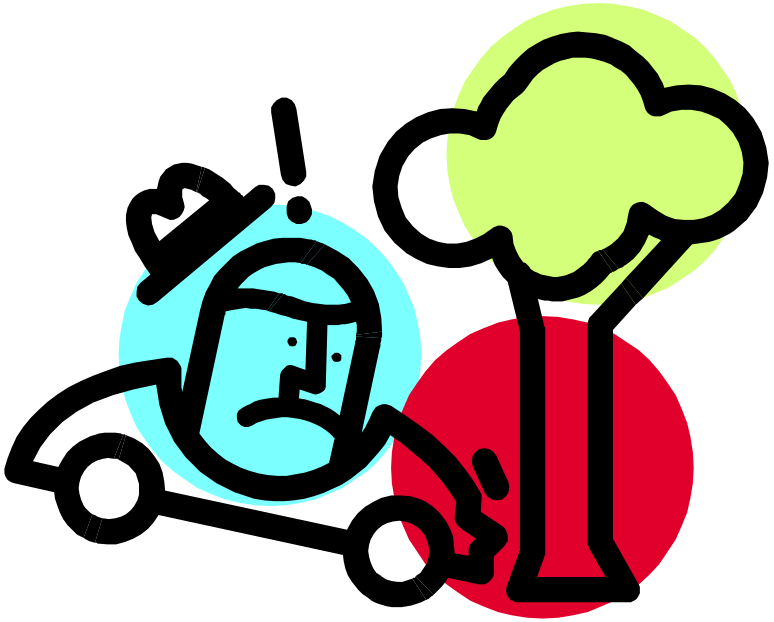


Hazards, harms and contexts

コンテキスト: 人、財産又は環境が危険にさらされる状況

Context: Circumstances in which people, property or the environment are exposed to one or more hazards

IEC61508－4より



コンテキストが重要

- 危険は常に存在する
- しかし、特定のコンテキストがないと危害は出られない
- すなわち、ハザードは特定のコンテキストで特定の危害を及ぼす

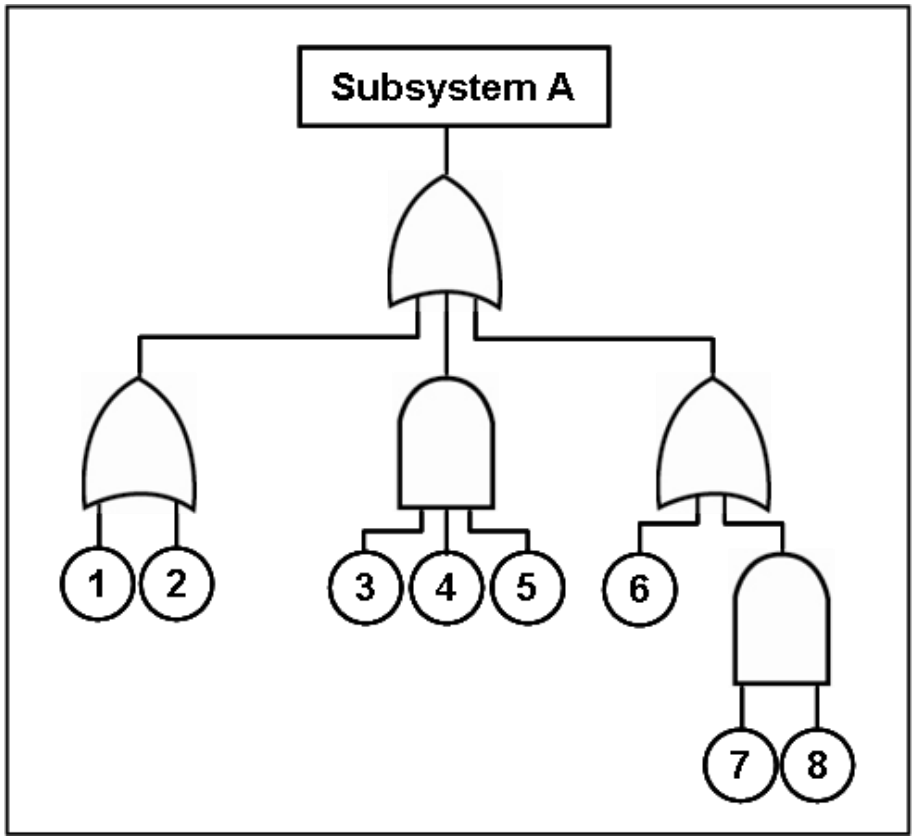
Hazard + Context = Harm

危険と危害とコンテキストを探す手法

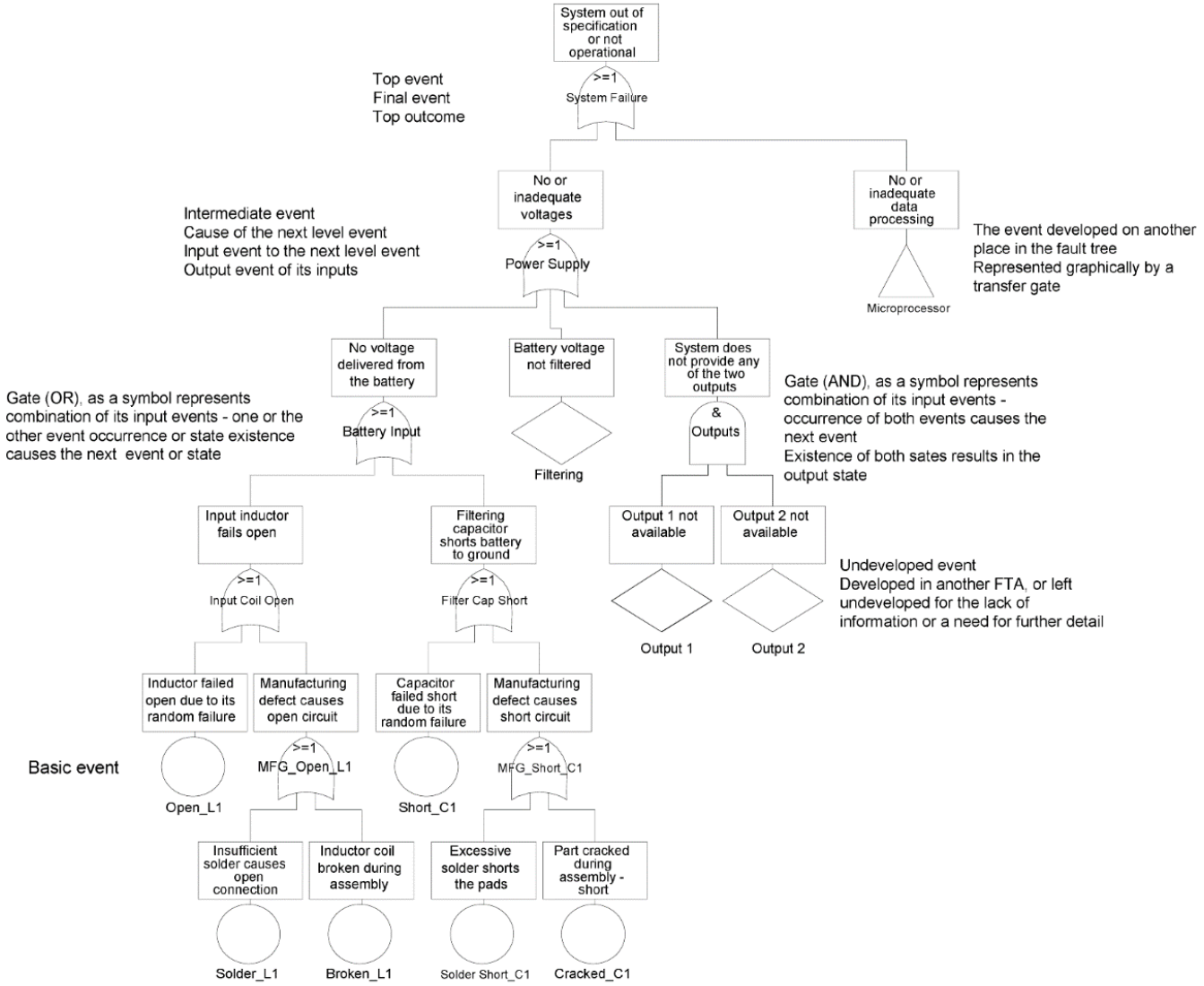
- ハザード分析はシステムにある危険などを発見する
  - 要求、設計等を分析する
- 深刻さ（severity）や確率等の解析
- 複数の手法
  - Fault Tree Analysis (FTA)
  - Failure Mode and Effects Analysis (FMEA)
  - Event Trees
  - …

Fault Tree Analysis（例）

- 望ましくない事件の原因を見つけるための分析手法



Fault Tree Analysis（例）



(IEC 61025より)

危害への対策

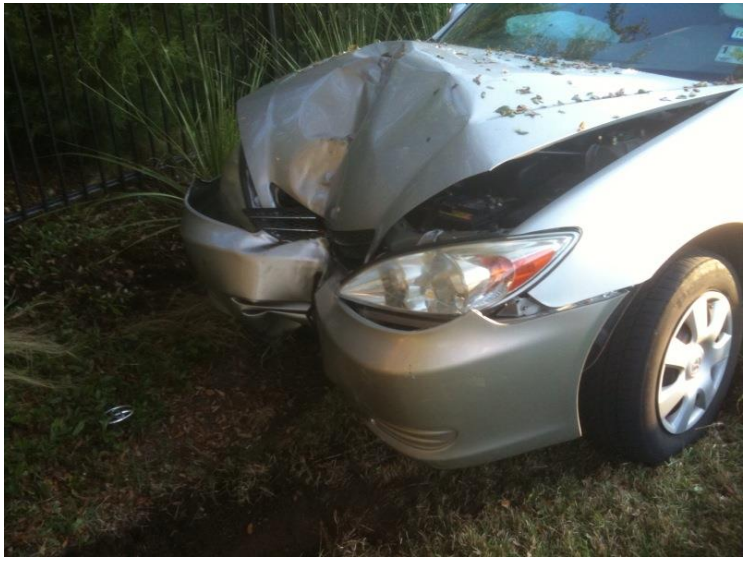
- どうやって危害が発生しないようにするか
- システムの設計を変更することによって
  - 危険が存在しないこと
  - コンテキストが現れないこと
  - コンテキストがあっても、危害が発生しない又は深刻さが減ること

危害への対策

- どうやって危害が発生しないようにするか
- システムの設計を変更することによって
  - 危険が存在しないこと
  - コンテキストが現れないこと
  - コンテキストがあっても、危害が発生しない又は深刻さが減ること

Intrinsic Safety

- システムから危険を消すシステムの特性



クラッシャブルゾーン

シートベルト

"Toyota Camry after frontal impact with tree" by Stillwaterising - Own work. Licensed under Creative Commons Attribution-Share Alike 3.0 via Wikimedia Commons - https://commons.wikimedia.org/wiki/File:Toyota_Camry_after_frontal_impact_with_tree.jpg#mediaviewer/File:Toyota_Camry_after_frontal_impact_with_tree.jpg

Functional Safety（機能安全）

- システムの機能:
  - 通常の作動中の安全性を保存する機能
  - 故障等の場合、安全性を提供する機能

エアバッグ

アンチロックブレーキシステム

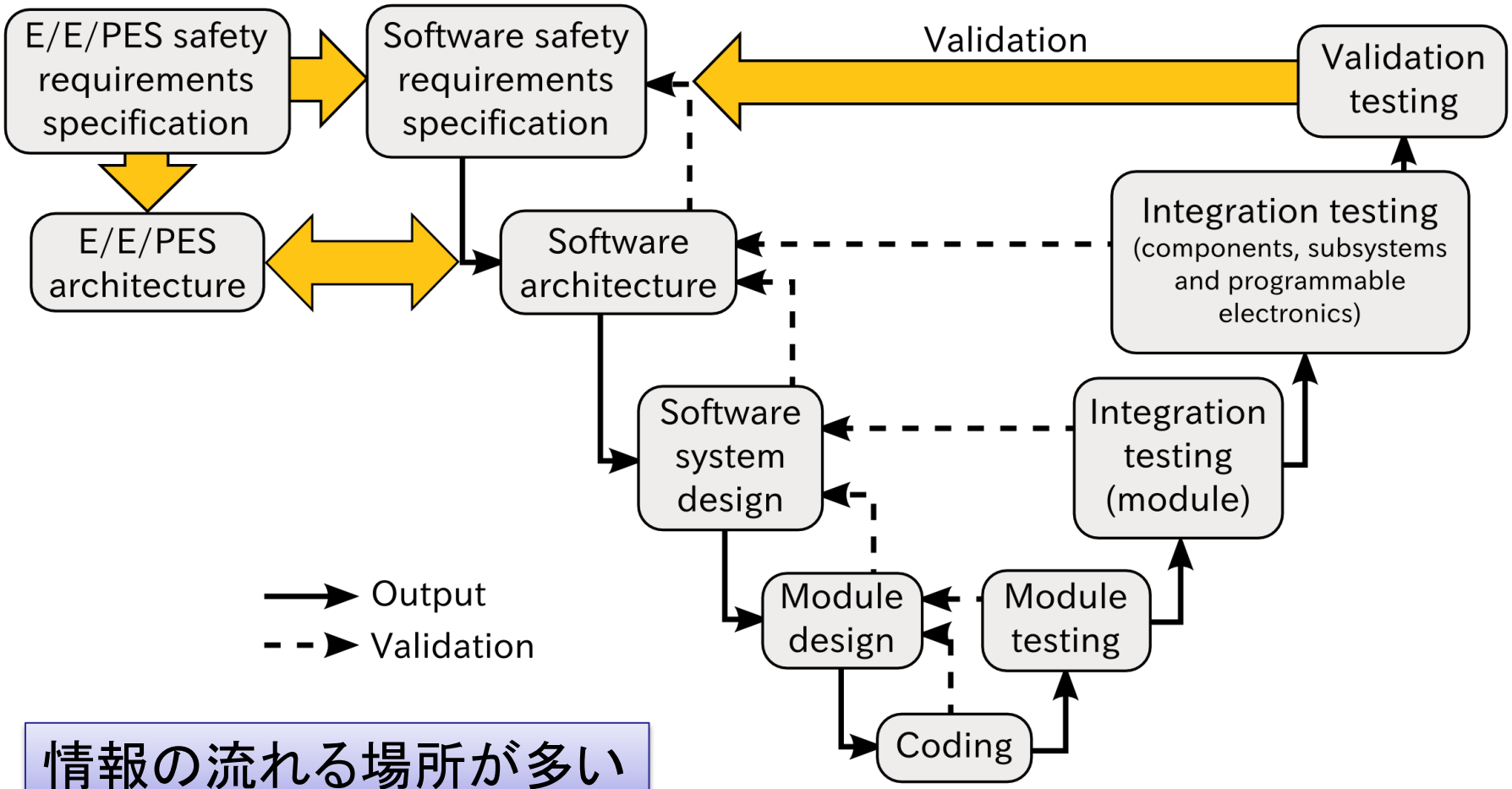
Part 2

SafeML

背景

- 開発中の情報交換
  - なぜ問題なのか
- モデリング言語により情報交換の改善
- モデルベース情報交換の利点

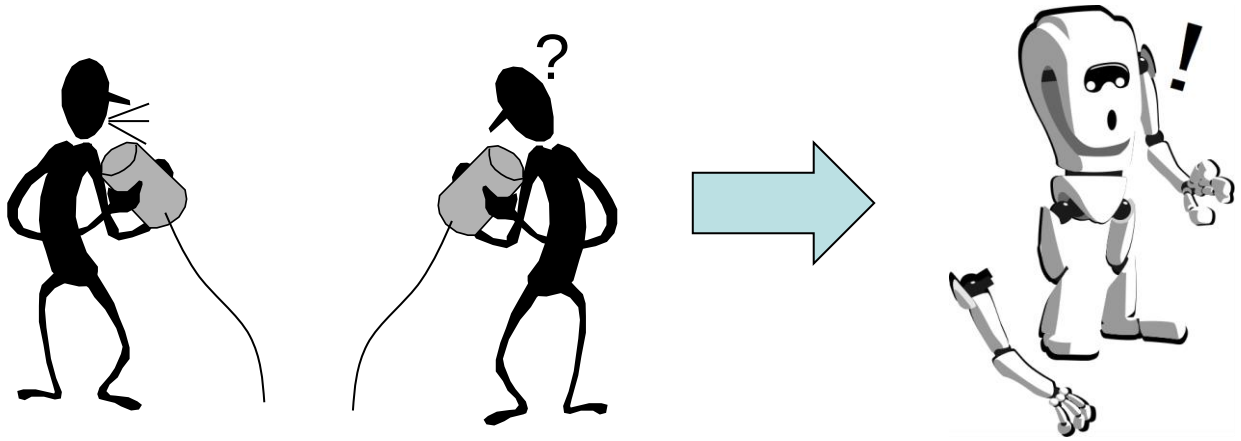
開発プロセス



情報の流れる場所が多い

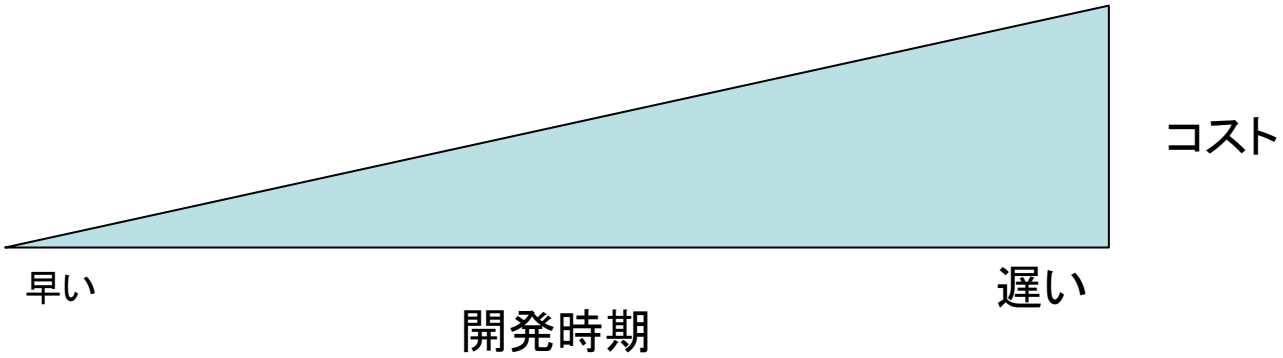
高信頼システム開発

- 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



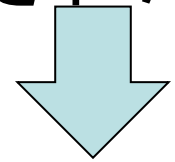
- コストが低く時期が早い段階で防ぐことが理想

コミュニケーション改善で欠陥防止

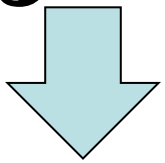
- 情報交換をより正確にすると
  - 各開発者の理解が上がる
  - 間違いが減る
- 開発プロセスの早い段階で欠陥を防げる

コミュニケーション改善による欠陥の防止

- モデリング言語の適用で、コミュニケーションが改善可能と証明された



- 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で、欠陥を防ぐ

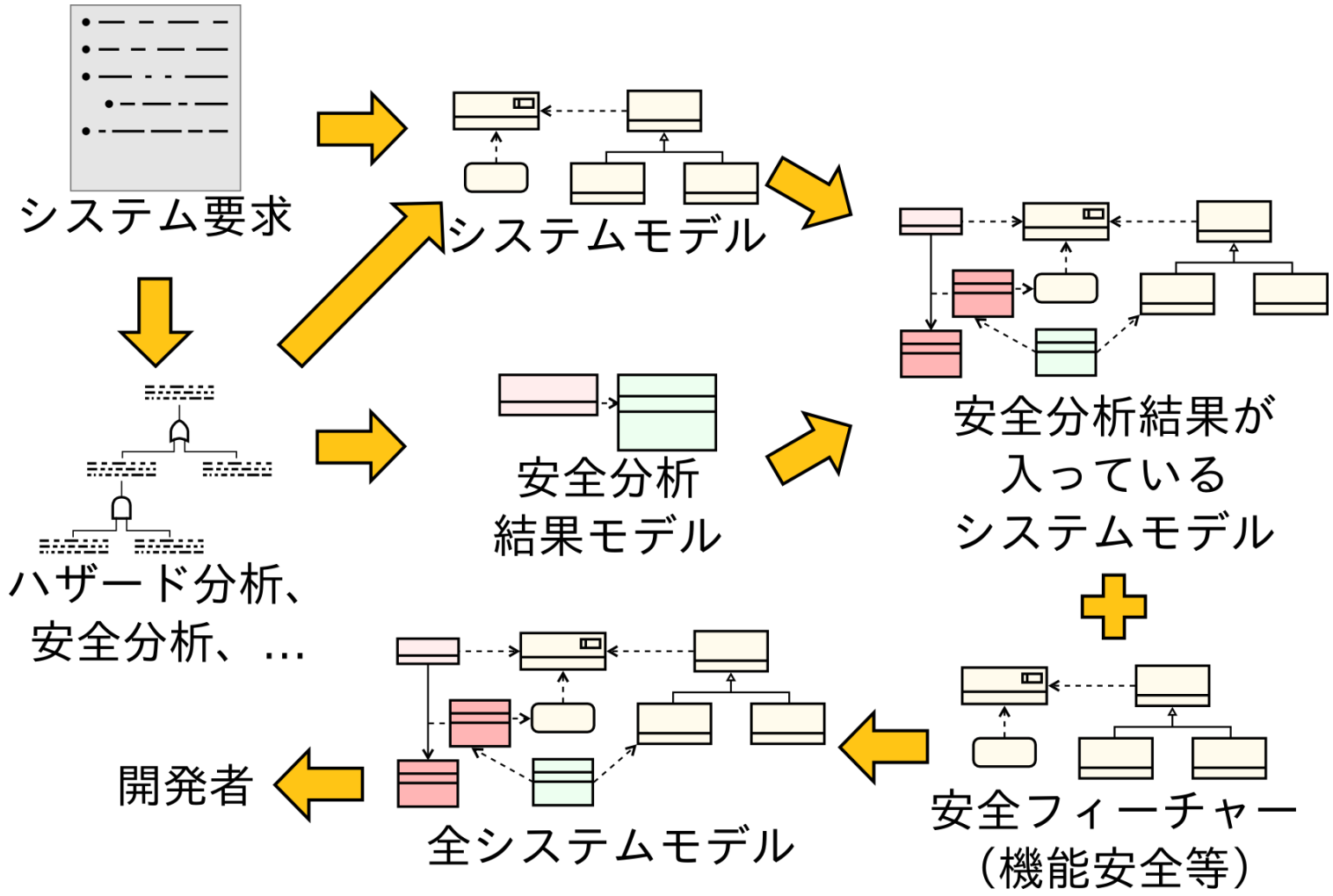
モデルベース情報交換の利点

- トレーサビリティの改良
- コンピュータは情報の構造が分かる
  - 文書で保存された情報（Word等）なら難しい
- 内容も関連も追跡できる

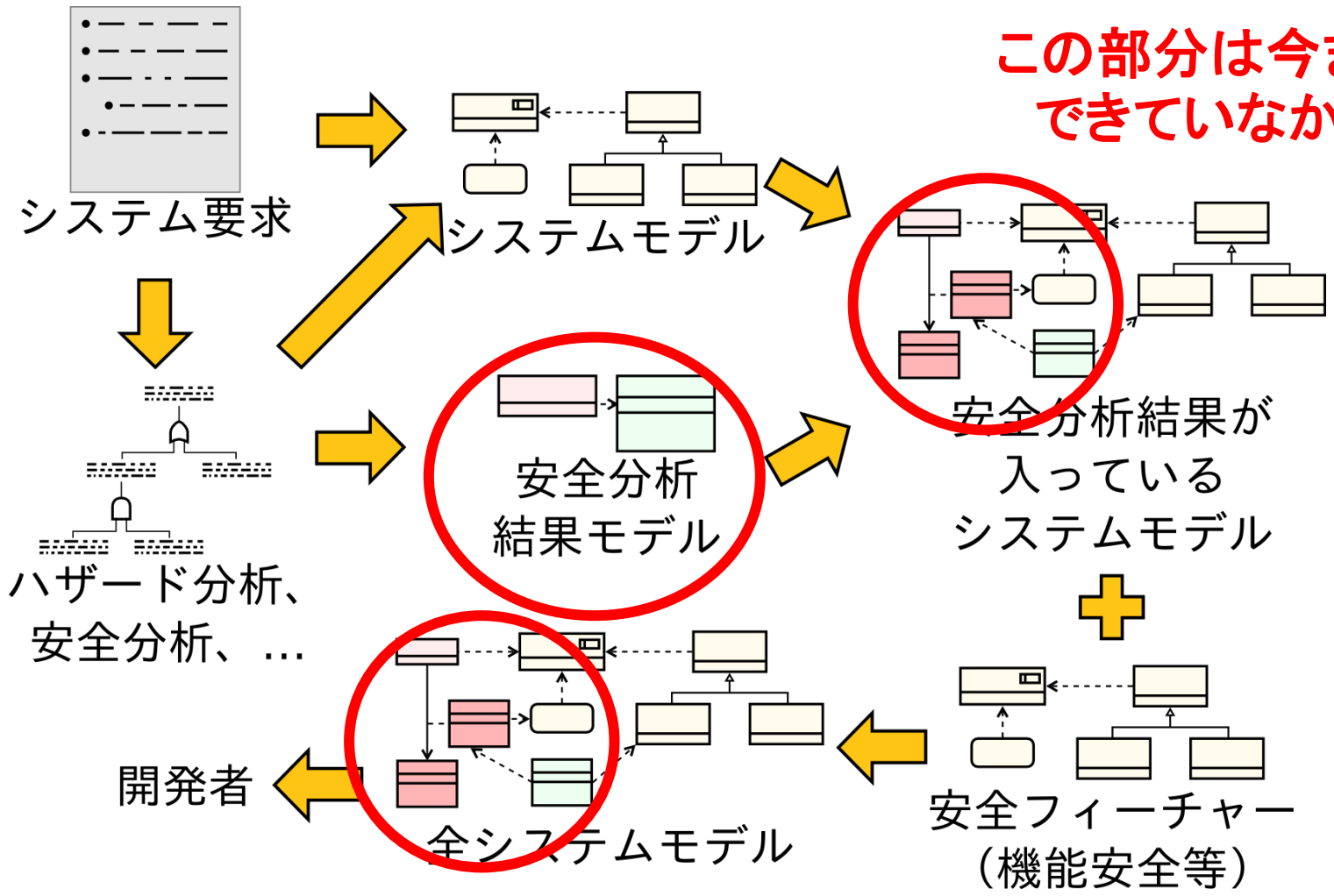
モデルベース情報交換の利点

- 自動情報処理
  - コンピュータは構造がある情報が処理できる
- 報告の自動生成
- 必要性により違う見方が可能
  - 設計レビュー
  - ギャップ探し
  - 認証

モデル言語による安全情報の交換



モデル言語による安全情報の交換

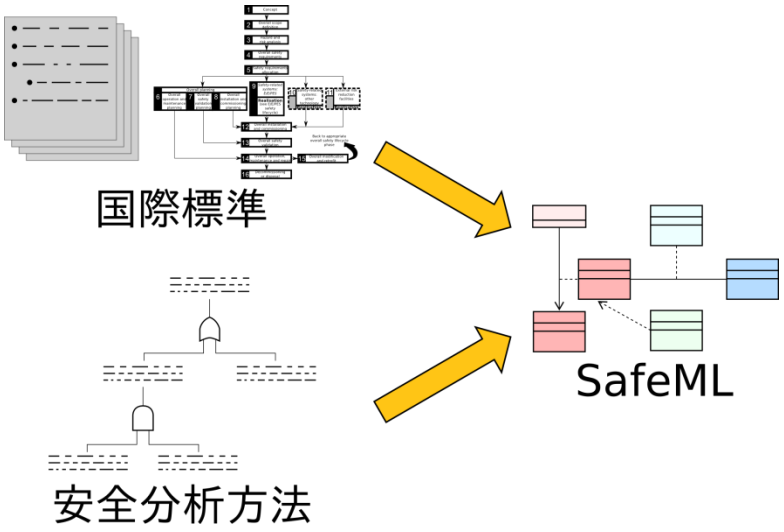


SafeML

- 安全に関するシステムの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）
- SysMLのプロファイル

SafeML

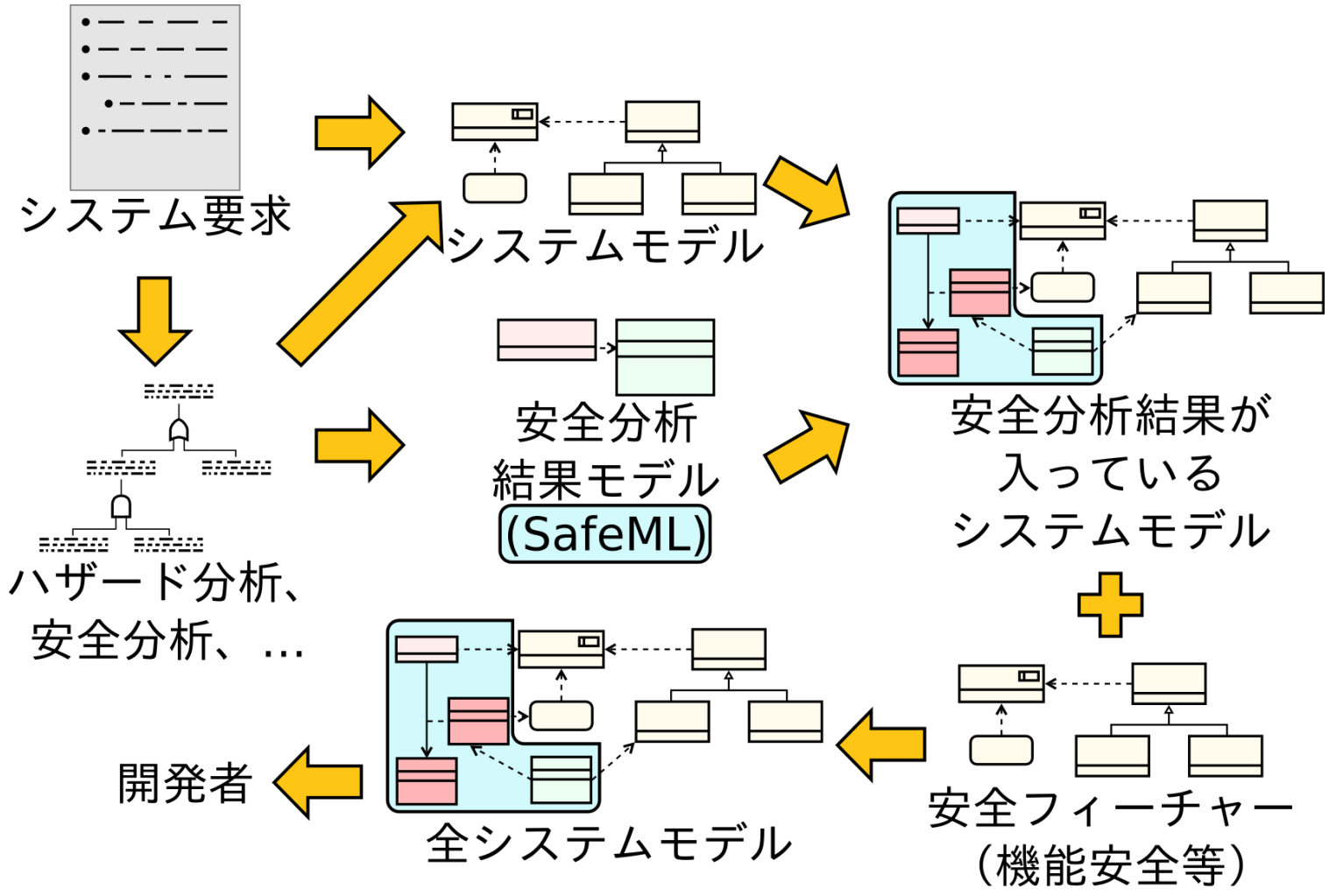
- 開発チーム内のコミュニケーションの道具
- 安全規格と安全分析手法に基づく
  - 安全分析の結果と安全機能をモデル化する



どのような機能か

- SysMLへ新しい要素と関連(「stereotypes」)の追加
- SysMLモデルにこれで
  - 要素で安全情報の表現
  - 関連で安全情報とシステム設計、要求等の関係の表現

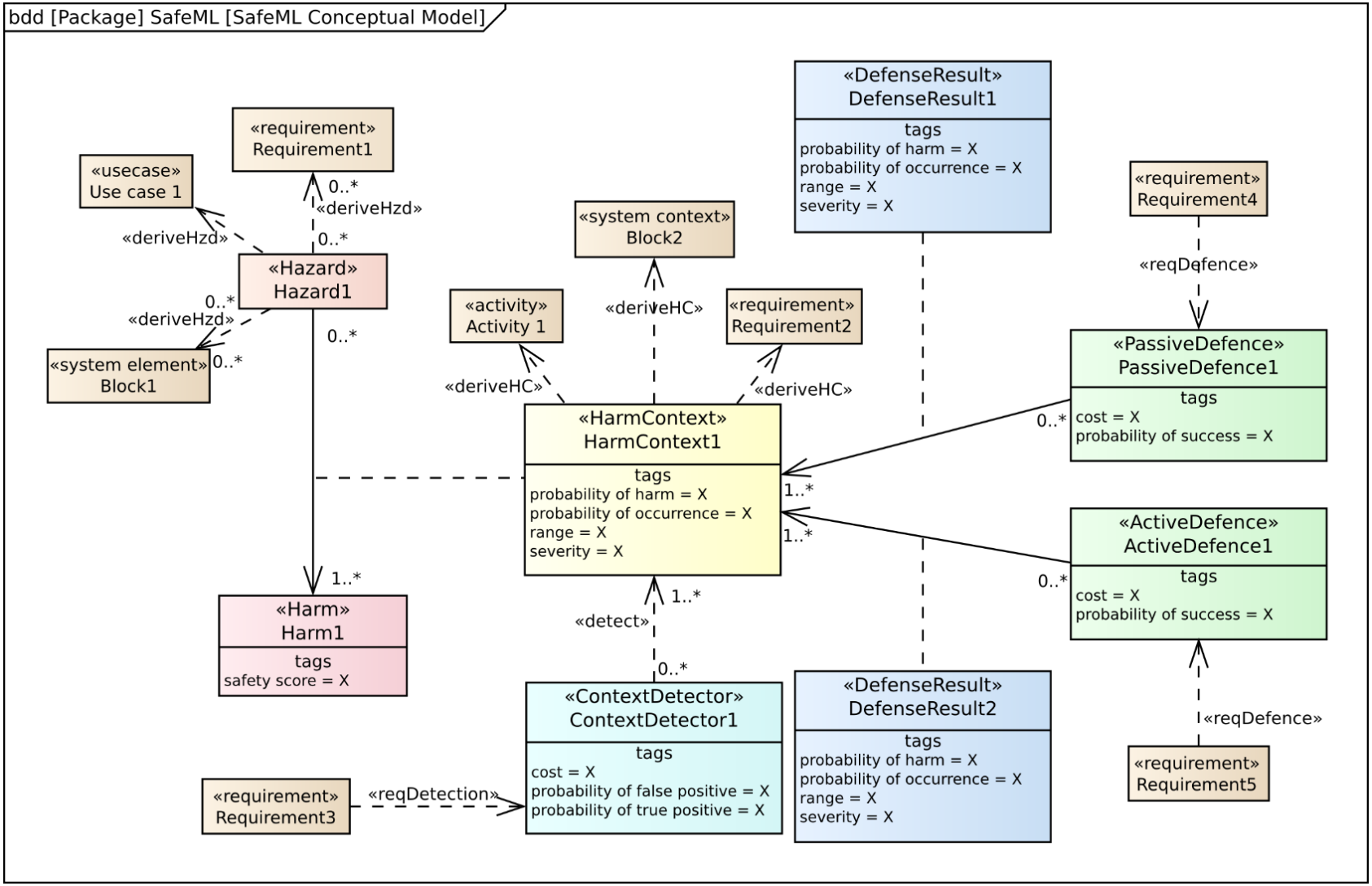
SafeMLを使った開発の流れ



SafeMLが使えるところ

- 危険から安全機能経由検証までの追跡
- 設計の時に考慮すべき危険の示すこと
  - 安全エンジニアからシステムエンジニアへ
- 設計に含まれている安全機能の示すこと
  - システムエンジニアから安全エンジニアと認証組織へ

SafeMLの要素と関連



SafeMLのコンセプト: Hazard

- Hazard＝危険
- システムにある危害の潜在源
- システムは関連する危害を引き起こす可能性が常にある
- システムのコンポーネント、設計、使われる方法（要求とユーズケース）に関連

SafeMLのコンセプト: Harm

- 財産又は環境の損傷の結果、直接的又は間接的に与えられる、人の健康に対する肉体的傷害又は損傷
- 特定のコンテキストで特定の危険により及ぼされる

SafeMLのコンセプト: Context

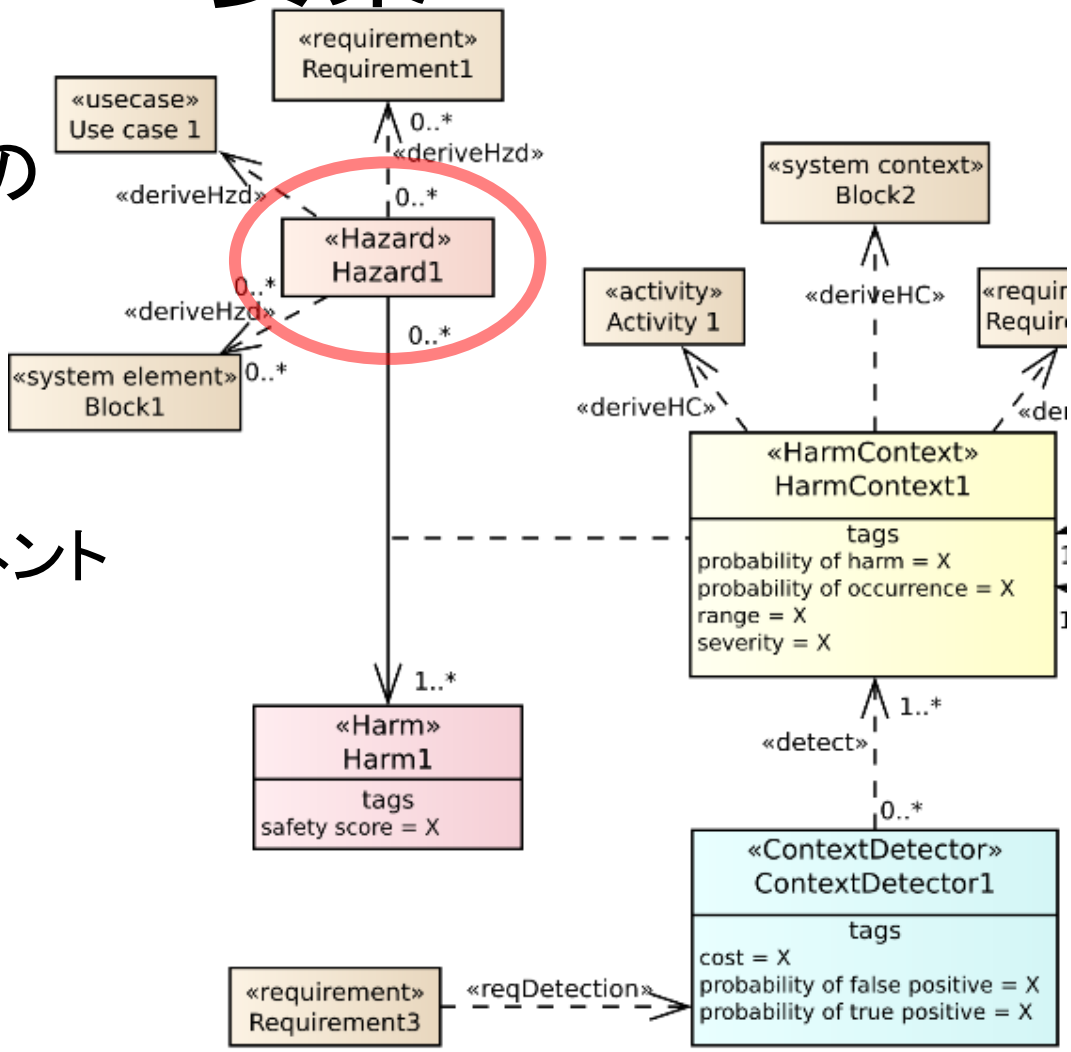
- 危険が危害を引き起こせる危険状況（コンテキスト）
- コンテキストが出ると「危険事象」（hazardous event）になる
- 危害が起こされたら「危害事象」（harmful event）になる
- **特定のコンテキストがないと危険は危害が引き起こせない**

Hazard と Context と Harm

- 危害事象にとって本質的なこと
- ハザード分析等で見つける
  - FMEA, FTA, STPA, …
- SafeMLでは3つの要素で表現する
  - Hazard
  - Harm
  - HarmContext

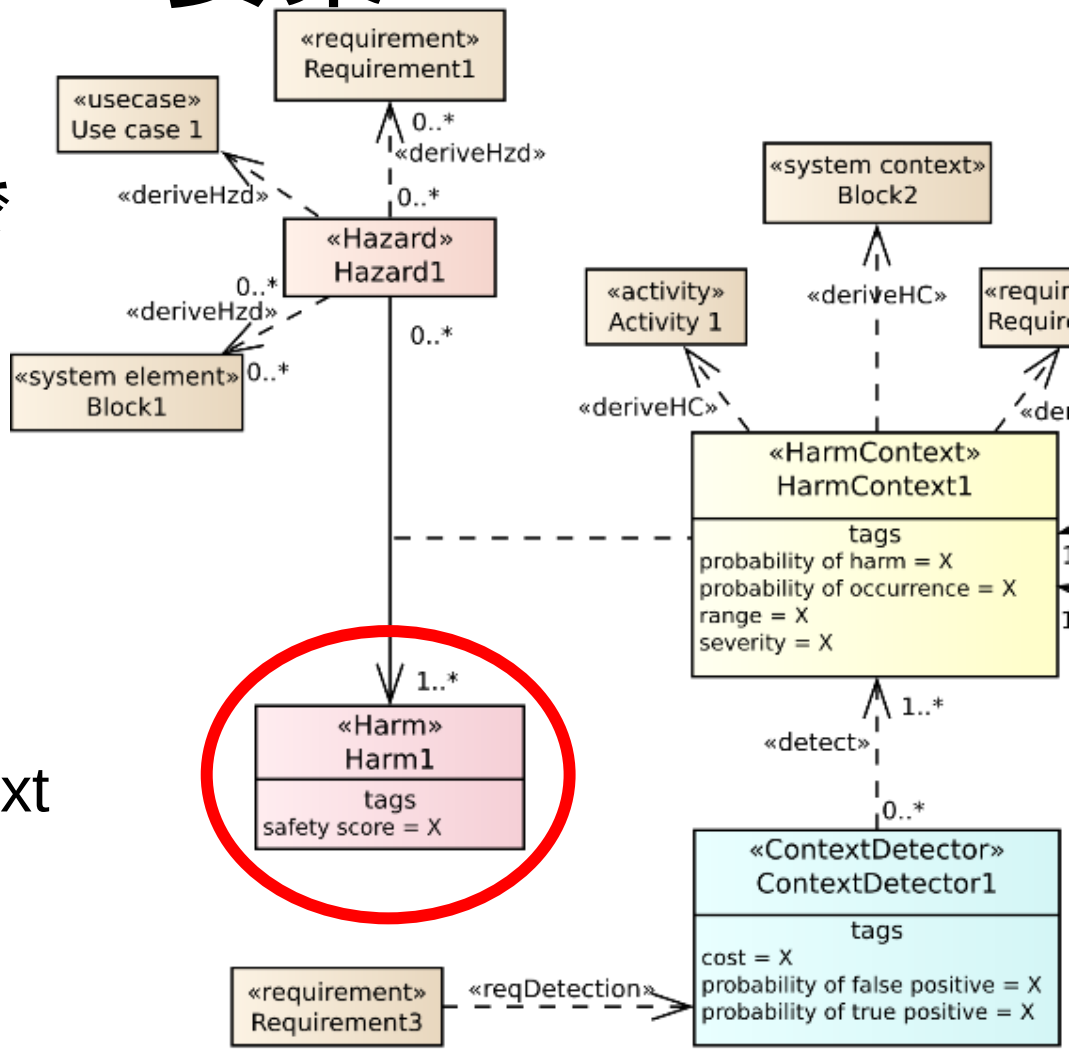
Hazard 要素

- ポテンシャルの危険の代表
- SysML 関連:
  - システム要求
  - システムのコンポーネント
  - ユーズケース
- SafeML 関連:
  - Harm (HarmContext 経由)



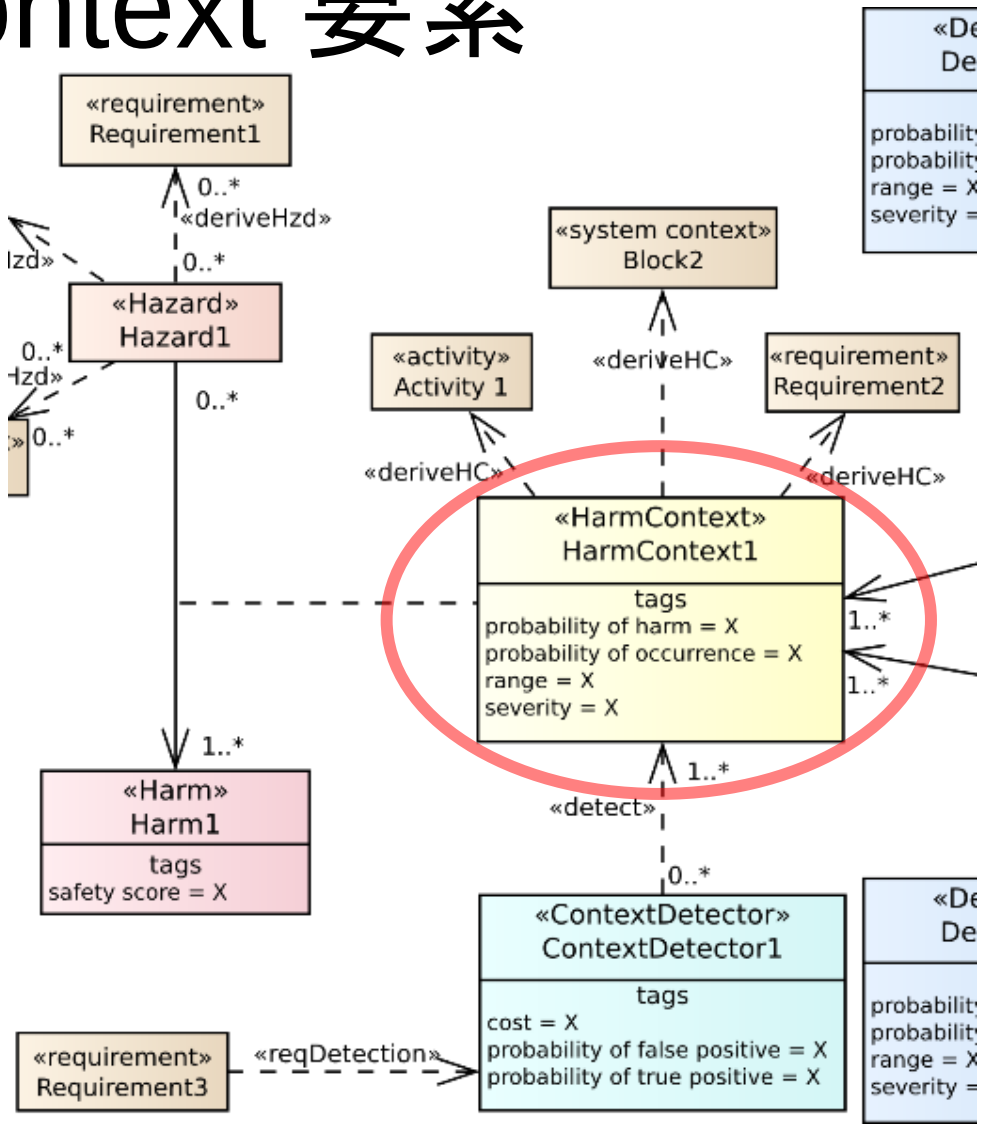
Harm 要素

- 特定のコンテキストで特定の危険が引き起こせる危害の代用
- SysML 関連:
  - なし
- SafeML 関連:
  - Hazard (HarmContext 経由)



HarmContext 要素

- 危険と危害の関連の代表
- SysML 関連:
  - システム要求
  - システムのコンポーネント
  - アクティビティ
- SafeML 関連:
  - ContextDetector



危害への対策

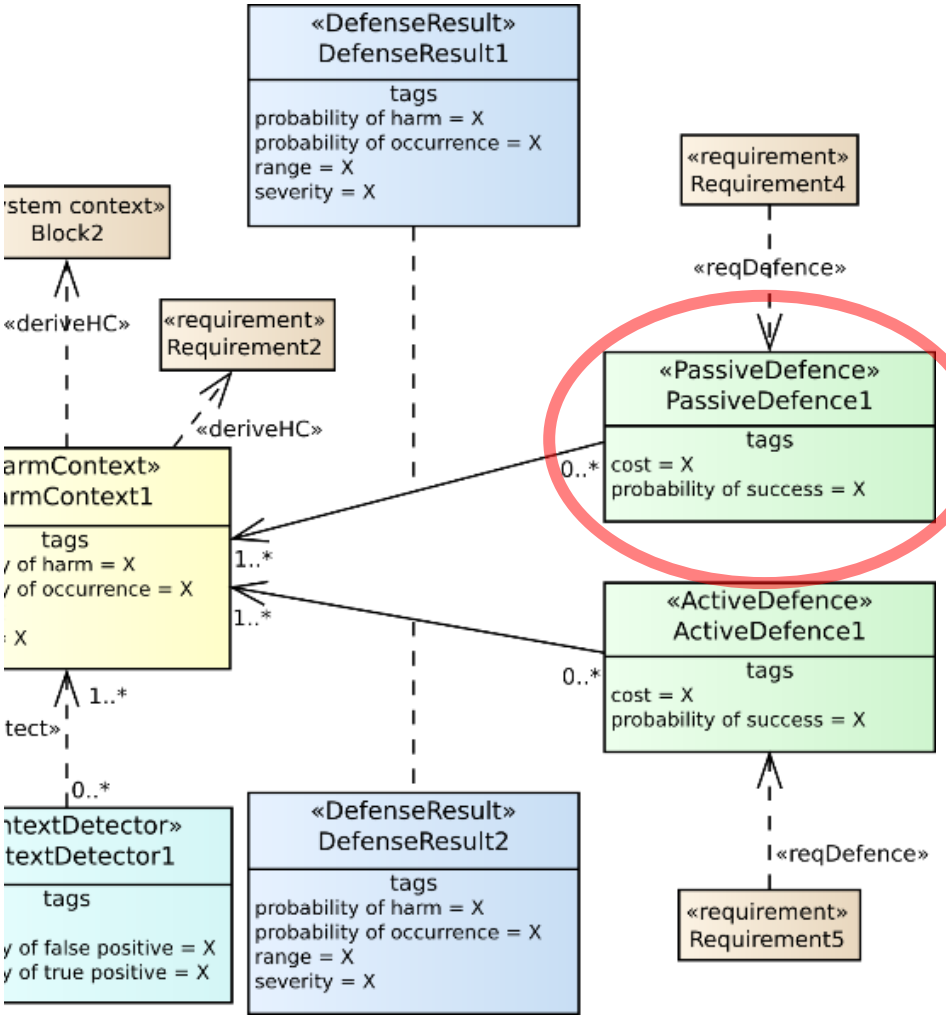
- どうやって危害が発生しないようにするか
- システムの設計を変更することによって
  - 危険が存在しないこと
  - コンテキストが現れないこと
  - コンテキストがあっても、危害が発生しない又は深刻さが減ること

SafeMLのコンセプト: Defences

- 安全を確保する手法（防衛方法）
  - コンテキストが現れないようにすること
  - 危害を緩和すること又はなくすこと
- SafeMLでは3つの要素で表現する
  - PassiveDefence and ActiveDefence
  - DefenceResult

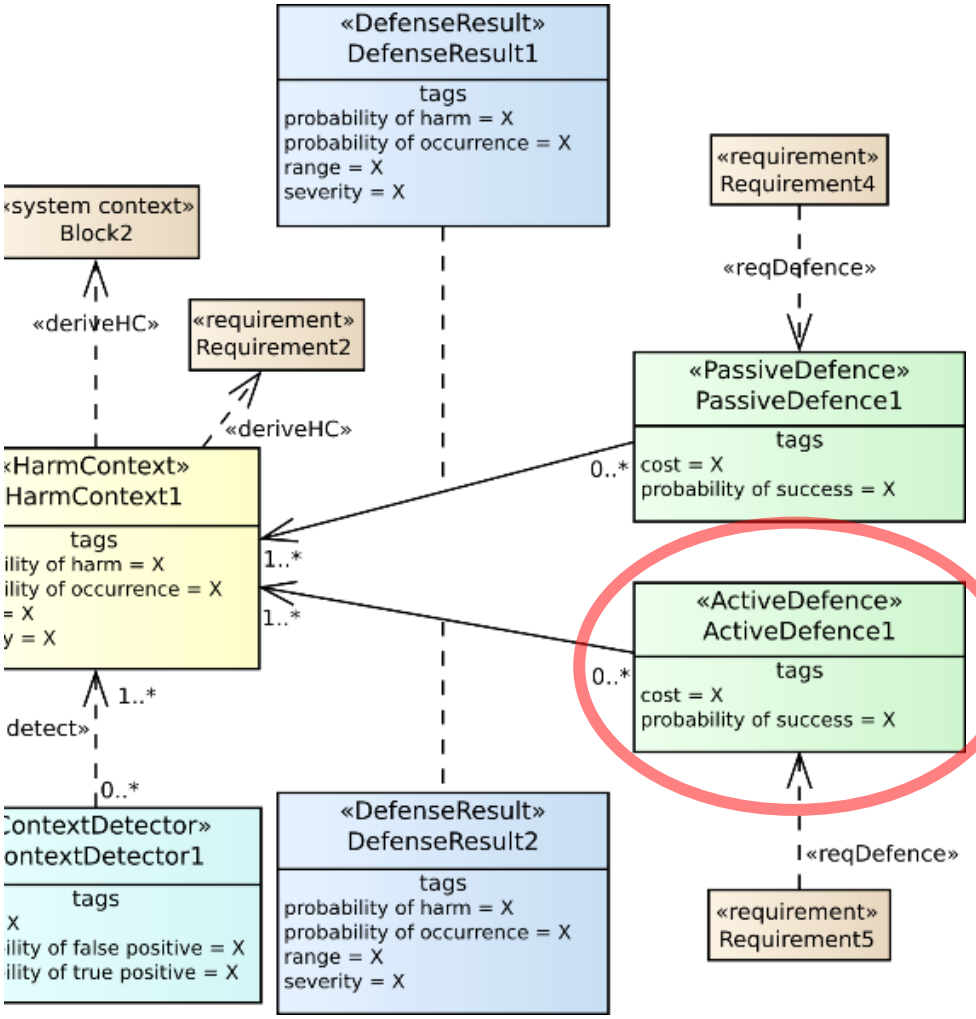
PassiveDefense 要素

- 連続的に安全を確保する安全機能（防衛方法）の代表
- 安全要求に至る
- SysML 関連：
  - システムの安全要求
- SafeML 関連：
  - HarmContext (DefenseResult 経由)



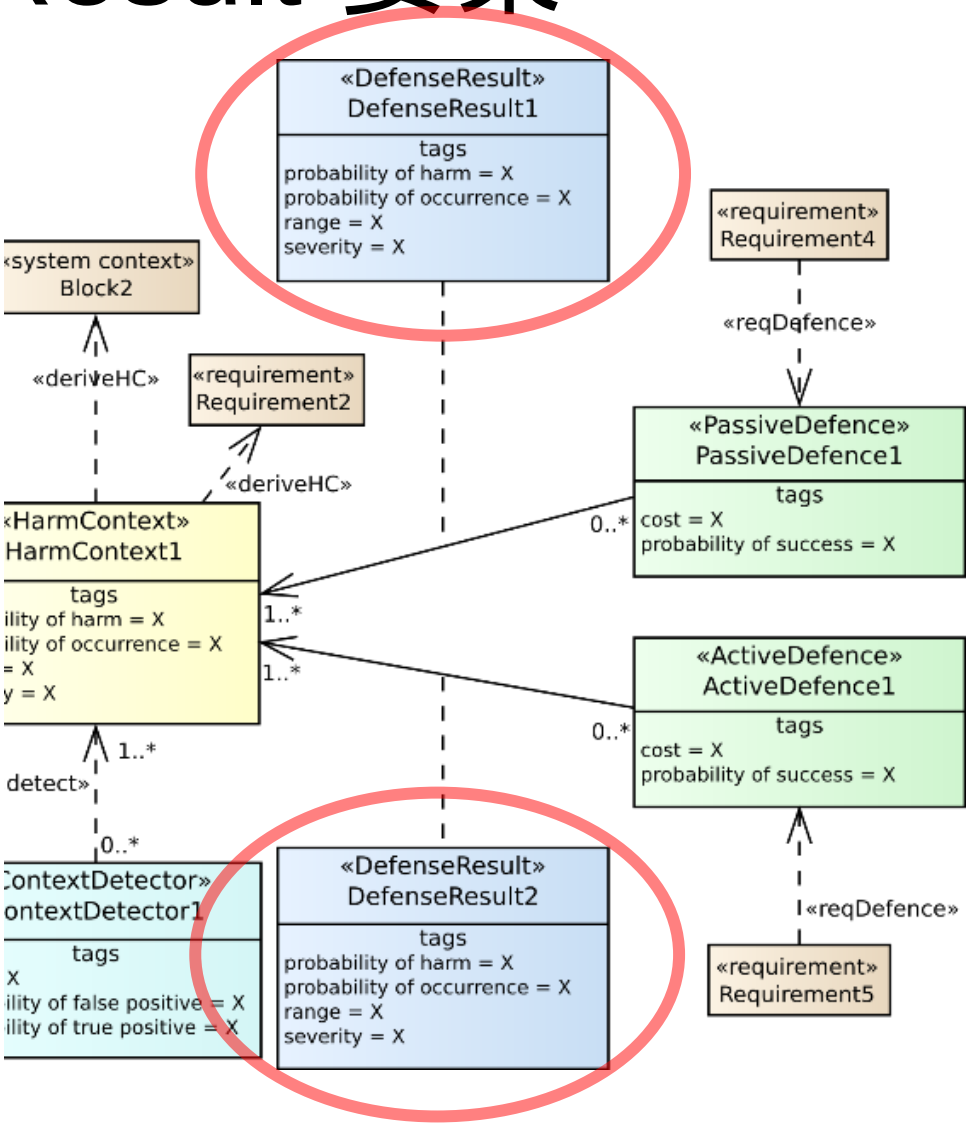
ActiveDefense 要素

- 必要がある次第安全を確保する安全機能（防衛方法）の代表
- 安全要求に至る
- SysML 関連：
  - システムの安全要求
- SafeML 関連：
  - HarmContext (DefenseResult 経由)



DefenseResult 要素

- 防衛方法の結果の代表
- 自動分析のため
- SysML 関連:
  - なし
- SafeML 関連:
  - なし

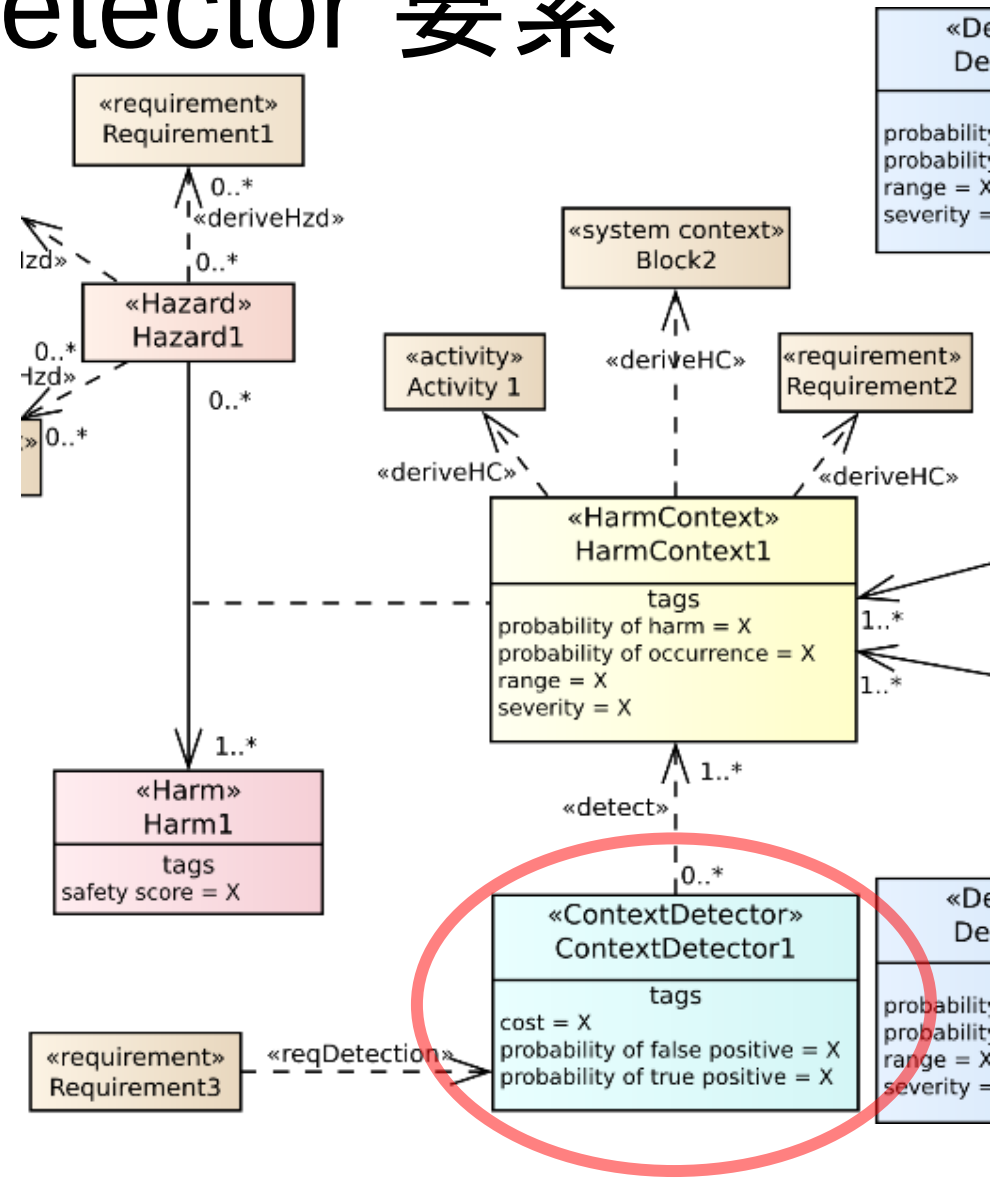


SafeMLのコンセプト: Monitoring

- 必要に応じて、安全を確保する防衛方法には、検出方法が必要
- システムの特定の機能は、危険事象の出現を検出する
- 危険事象が検出された時、防衛（安全機能）は起動する
- SafeMLで ContextDetector 要素で代表される

ContextDetector 要素

- 安全性の検出方法の代表
- 安全要求に至る
- SysML 関連:
  - システムの安全要求
- SafeML 関連:
  - HarmContext

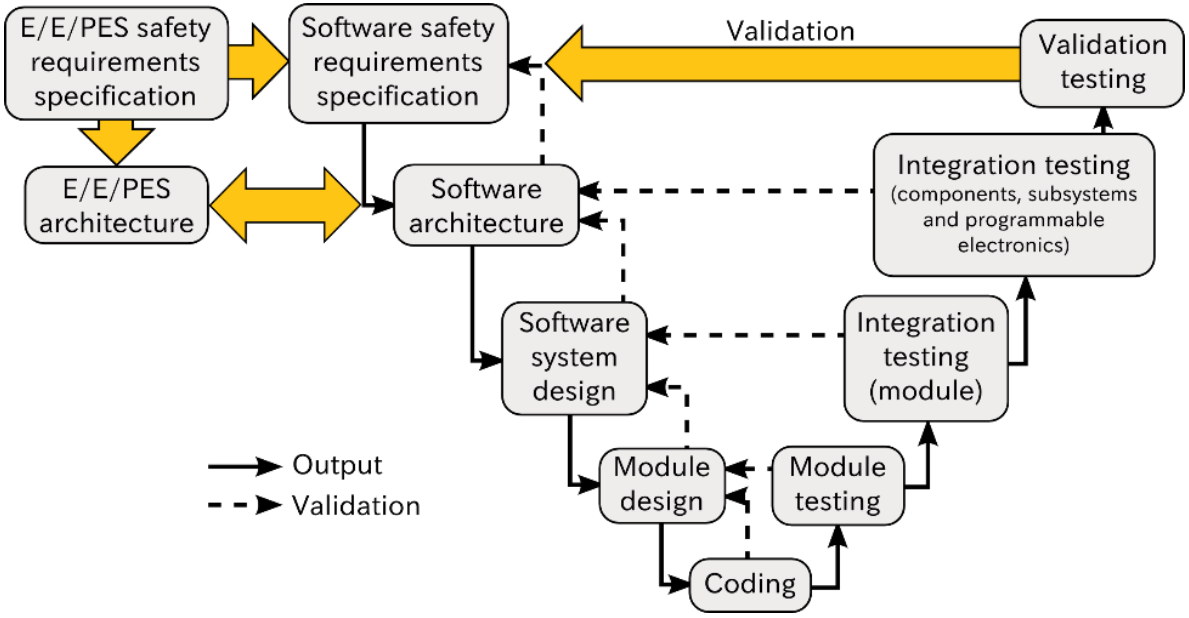


SafeMLの要素と関連

- Tagによって、SafeMLの要素に情報を追加する
  - コンテキストの確率
  - 危害の確率
  - 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
  - …

SafeMLと開発プロセス

- 安全情報が使用される状況のどのような場合でも、SafeMLは利用可能
  - ハザード分析とシステム設計の間
  - 検証（トレーサビリティ確認）

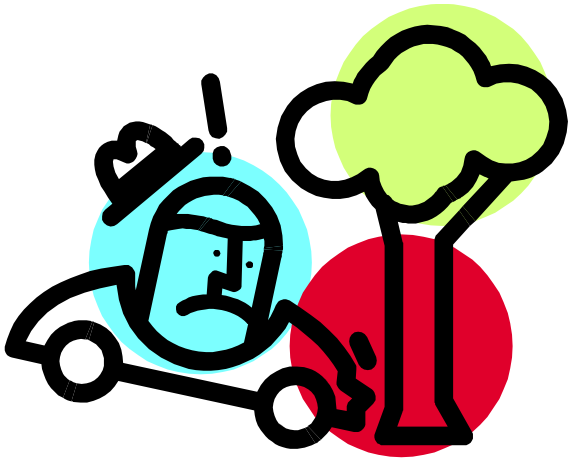


Part 3

SafeML適応の例

例:車の衝突に対しての安全性

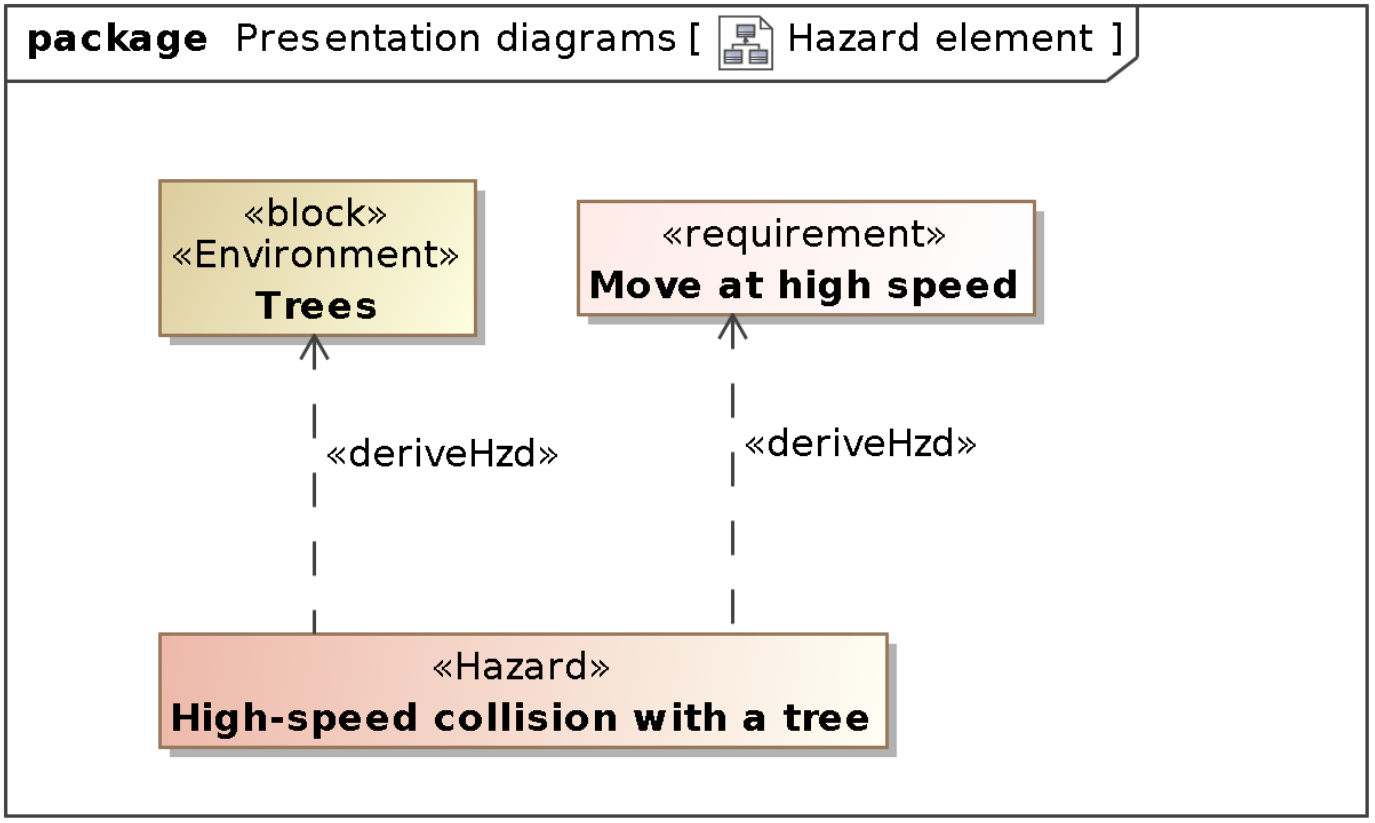
- 車が木に衝突すると運転手に危害が発生する
- 衝突の可能な原因
  - 衝突を妨げることに失敗
  - ブレーキが車の止めることに失敗
- 安全確保のための手法
  - クラッシャブルゾーン
  - アンチロック・ブレーキ・システム
  - エアバッグ



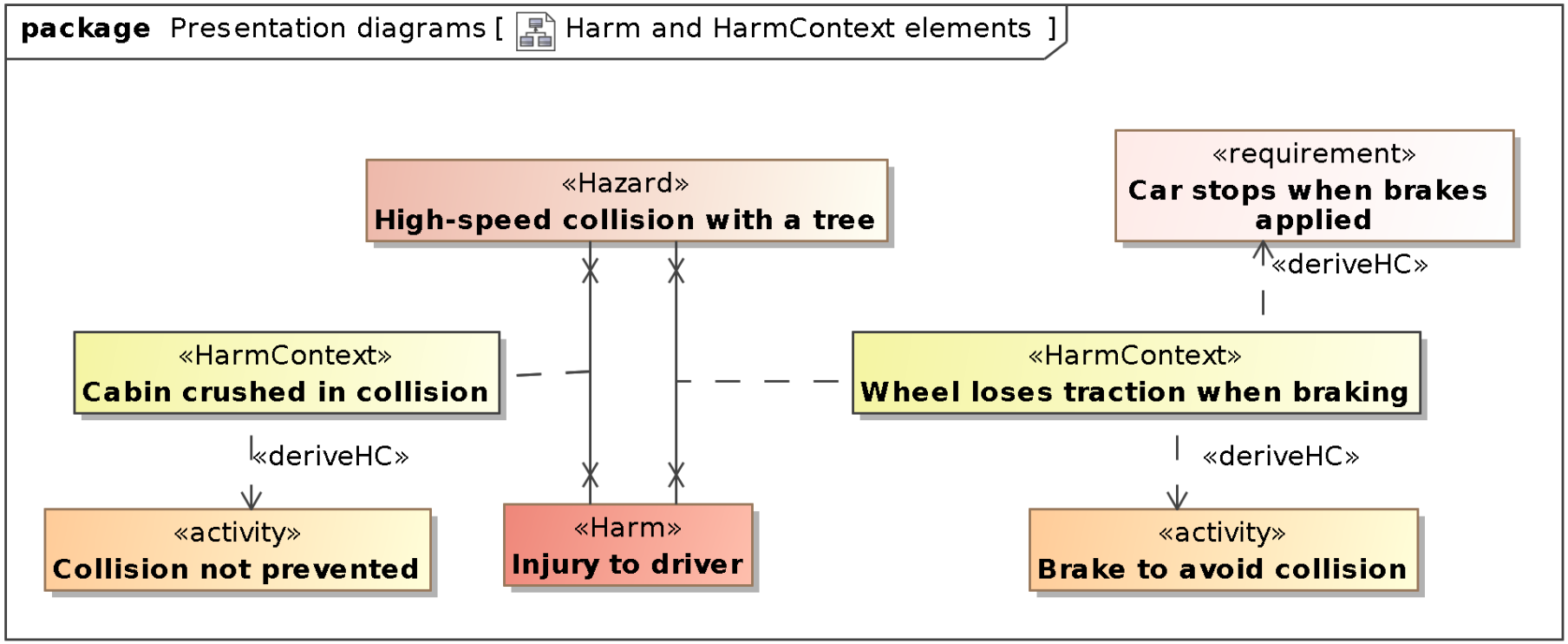
例:車の衝突に対しての安全性

- 車が木に衝突する(hazard)と運転手に危害が発生する(harm)
- 衝突の可能な原因(context)
  - 衝突を妨げることに失敗
  - ブレーキが車の止めることに失敗
- 安全確保のための手法
  - クラッシャブルゾーン(passive defence)
  - アンチロック・ブレーキ・システム(active defence)
  - エアバッグ(active defence)

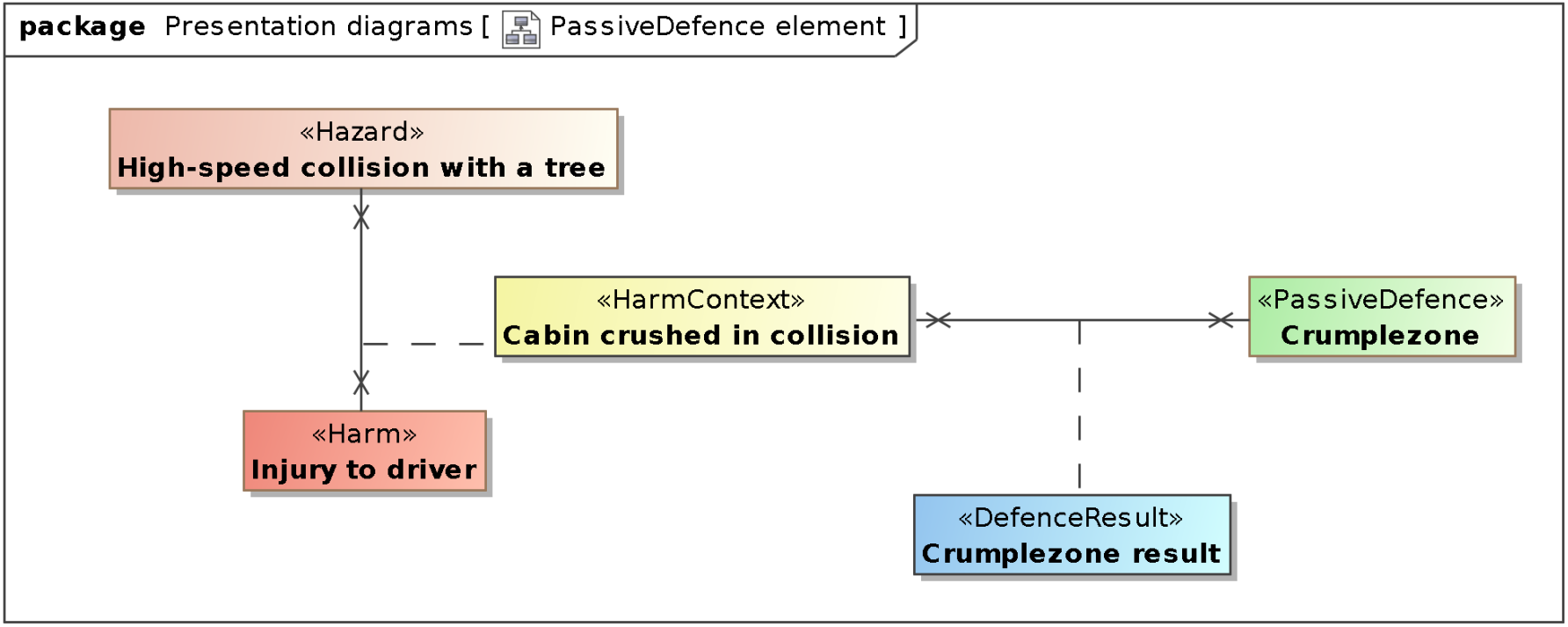
Hazard 要素


package Presentation diagrams [Hazard element]
«block»
«Environment»
Trees
«requirement»
Move at high speed
«deriveHzd»
«deriveHzd»
«Hazard»
High-speed collision with a tree

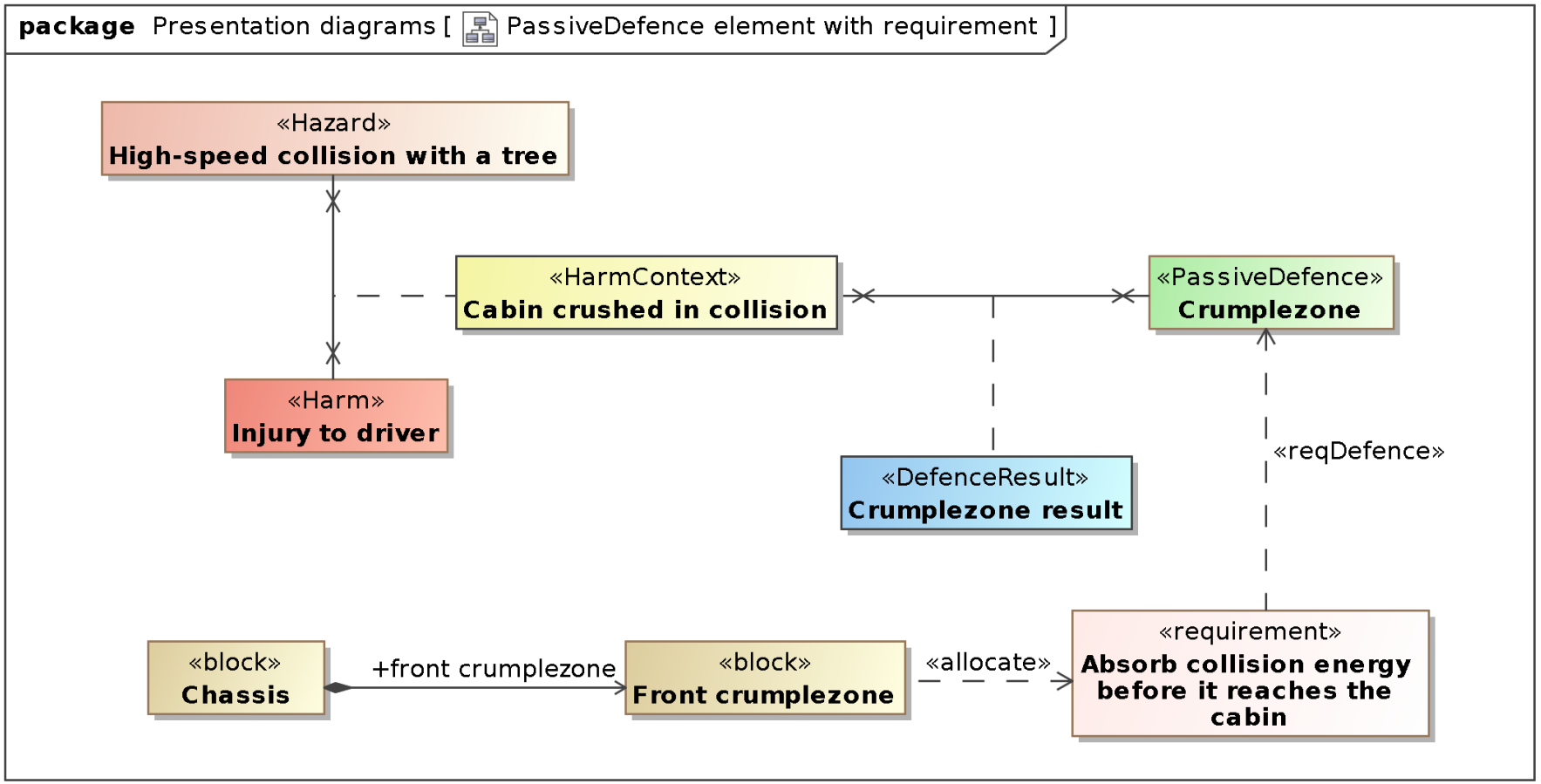
HarmContext と Harm 要素



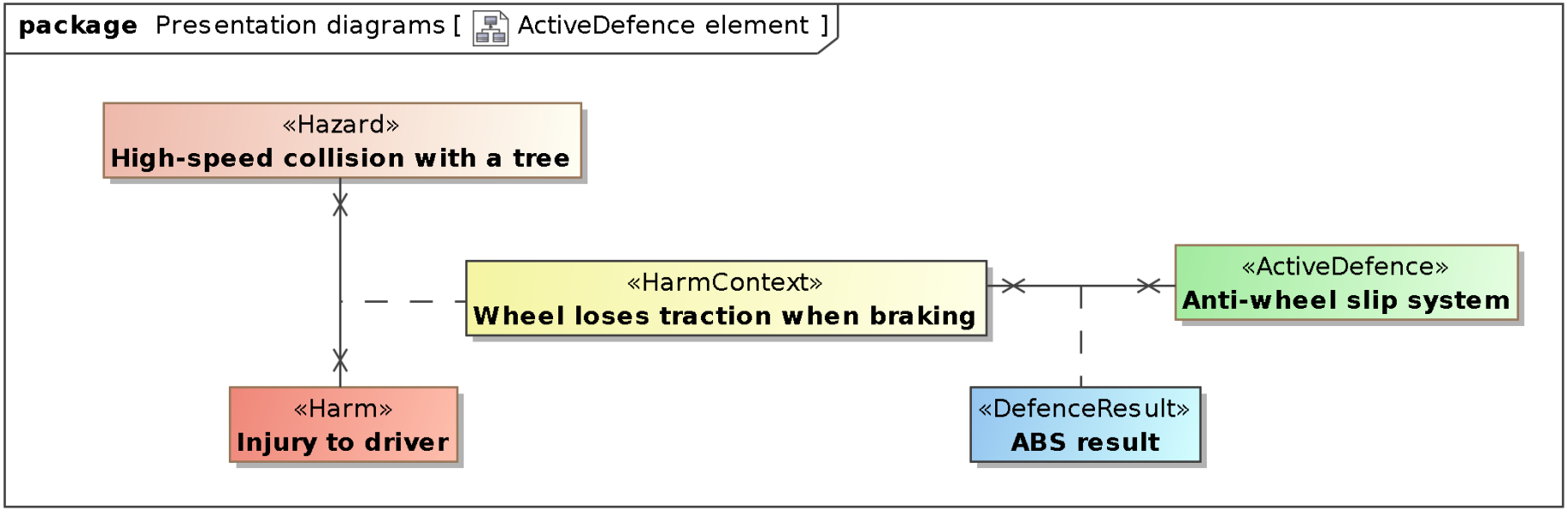
PassiveDefence 要素



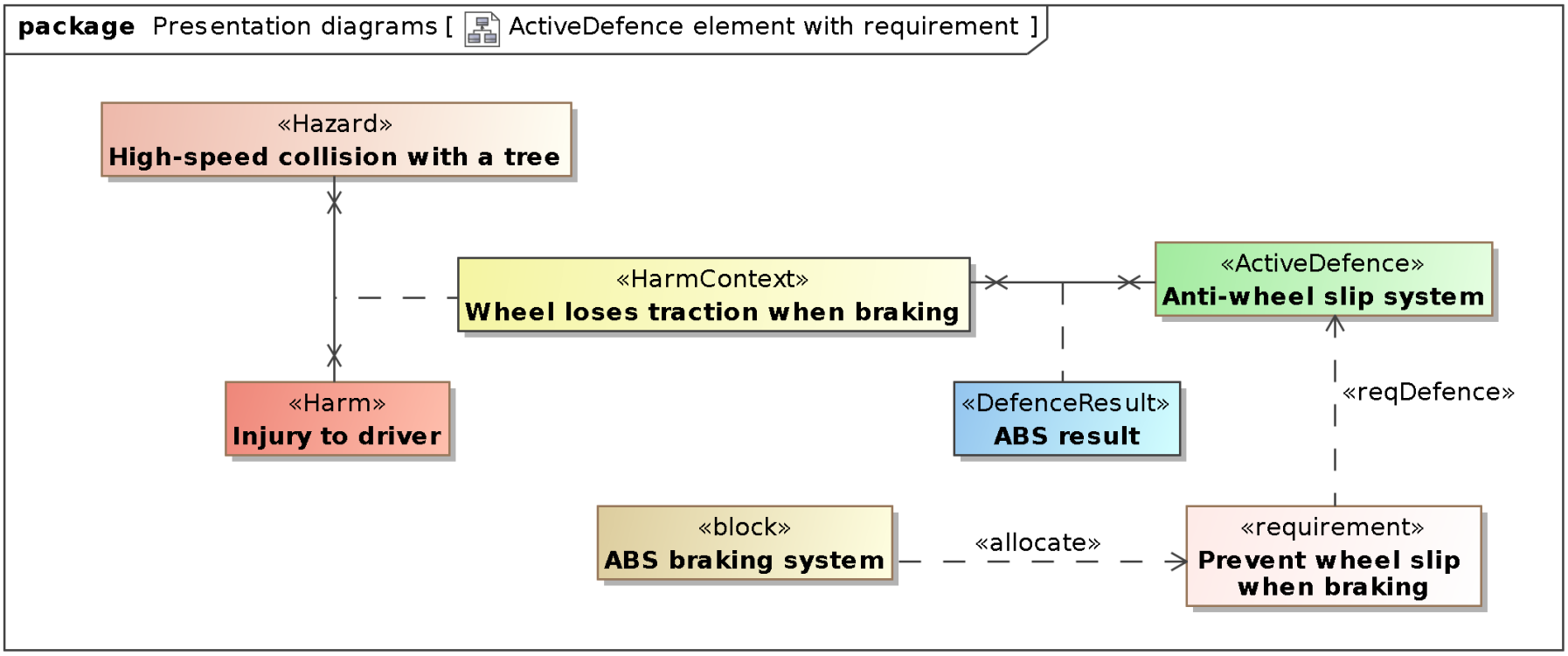
PassiveDefence 要素：安全要求



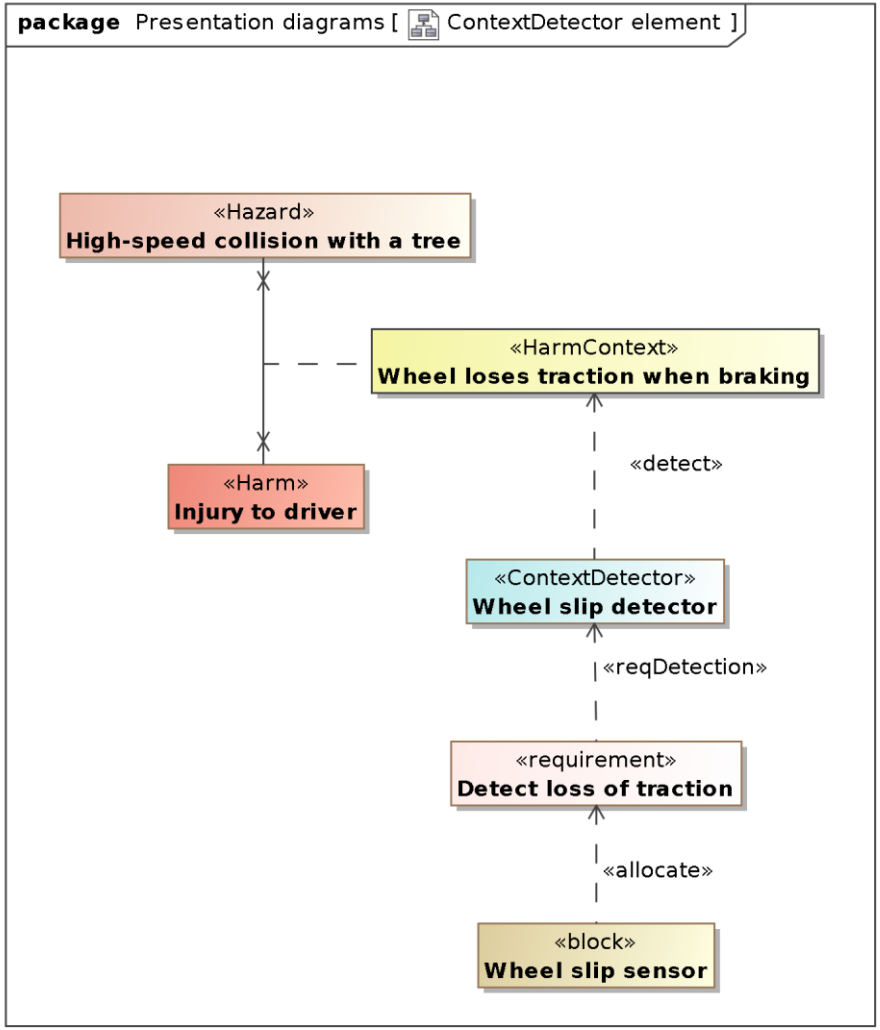
ActiveDefence 要素



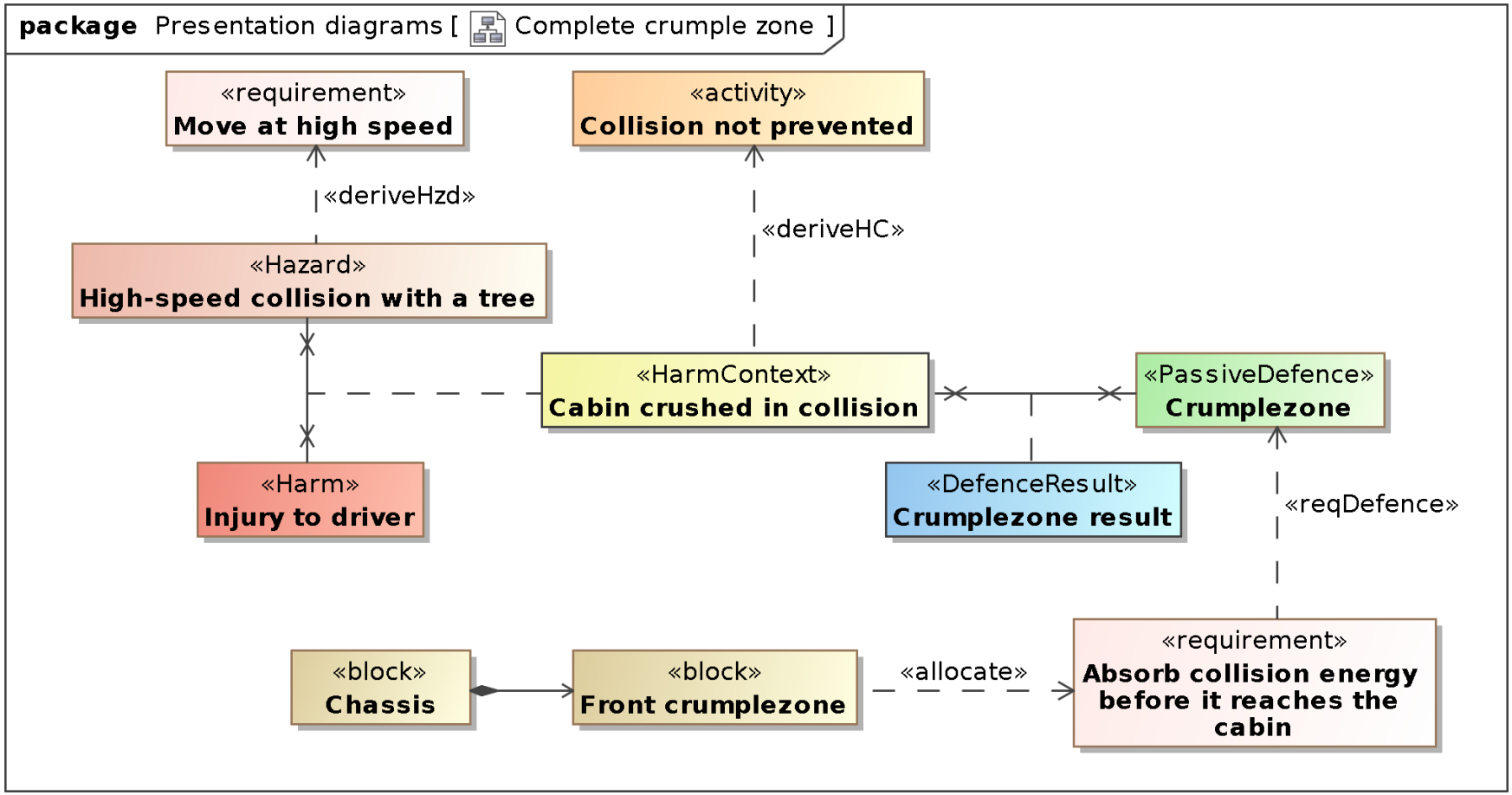
ActiveDefence要素：安全要求



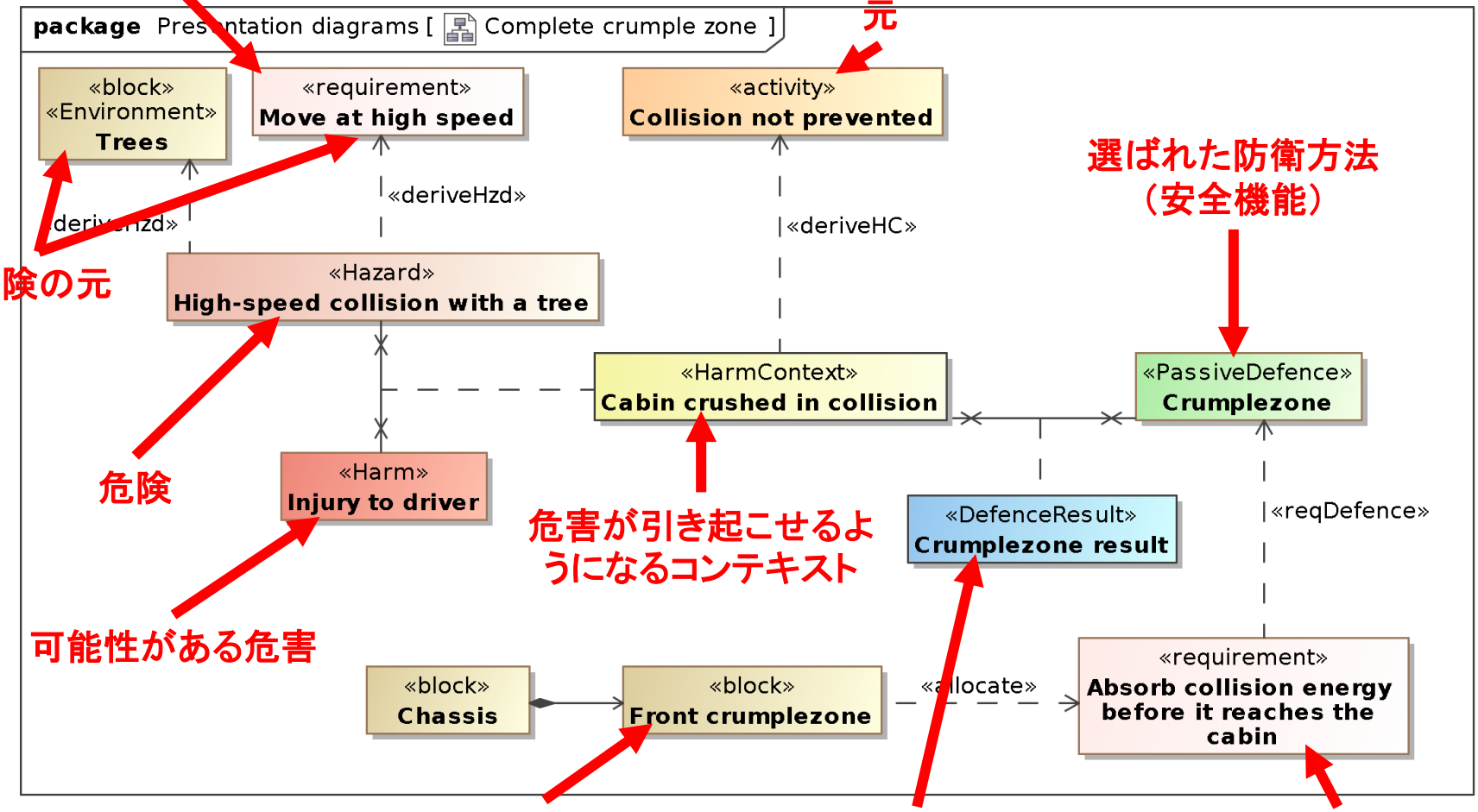
ContextDetector 要素



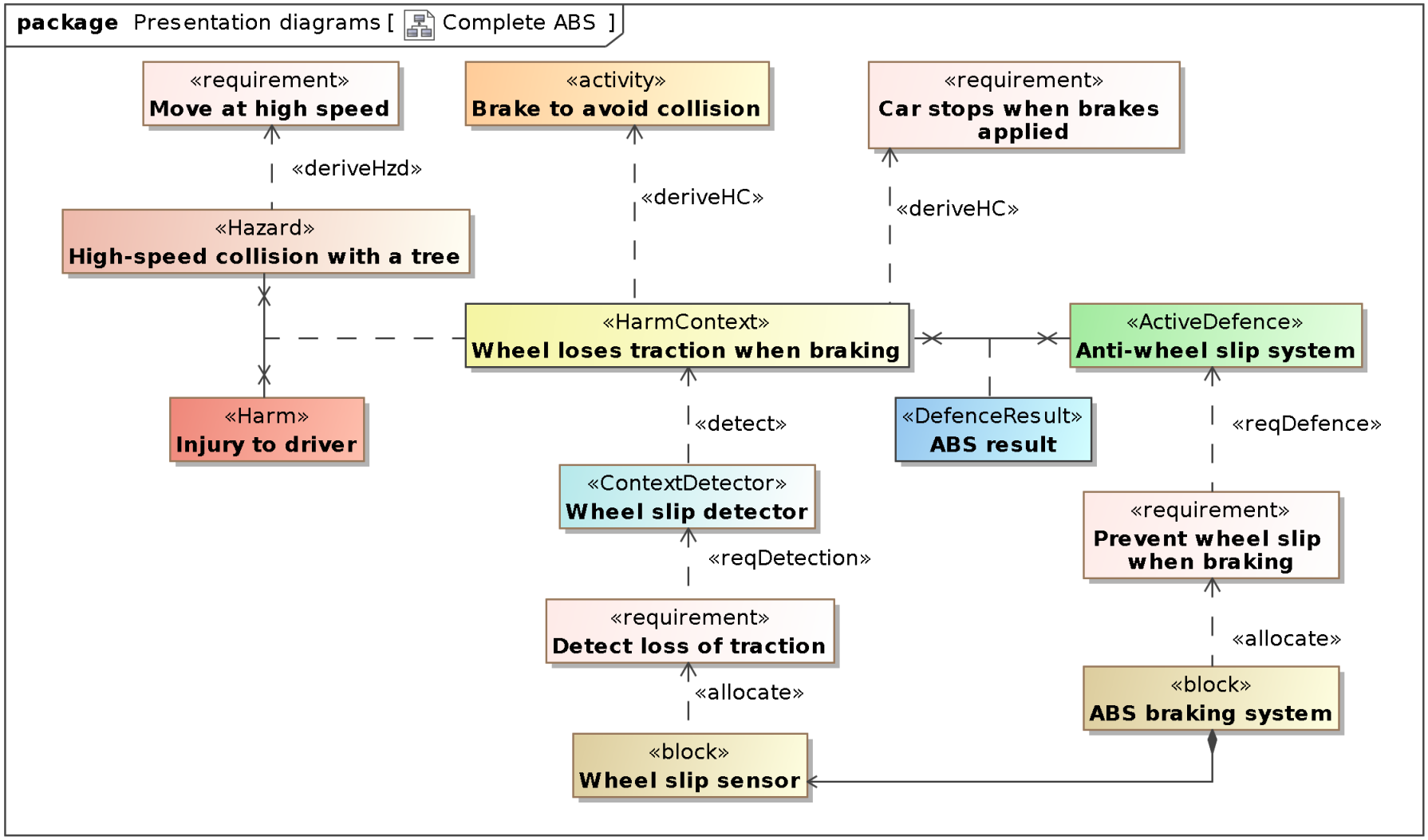
クラッシャブルゾーン安全機能



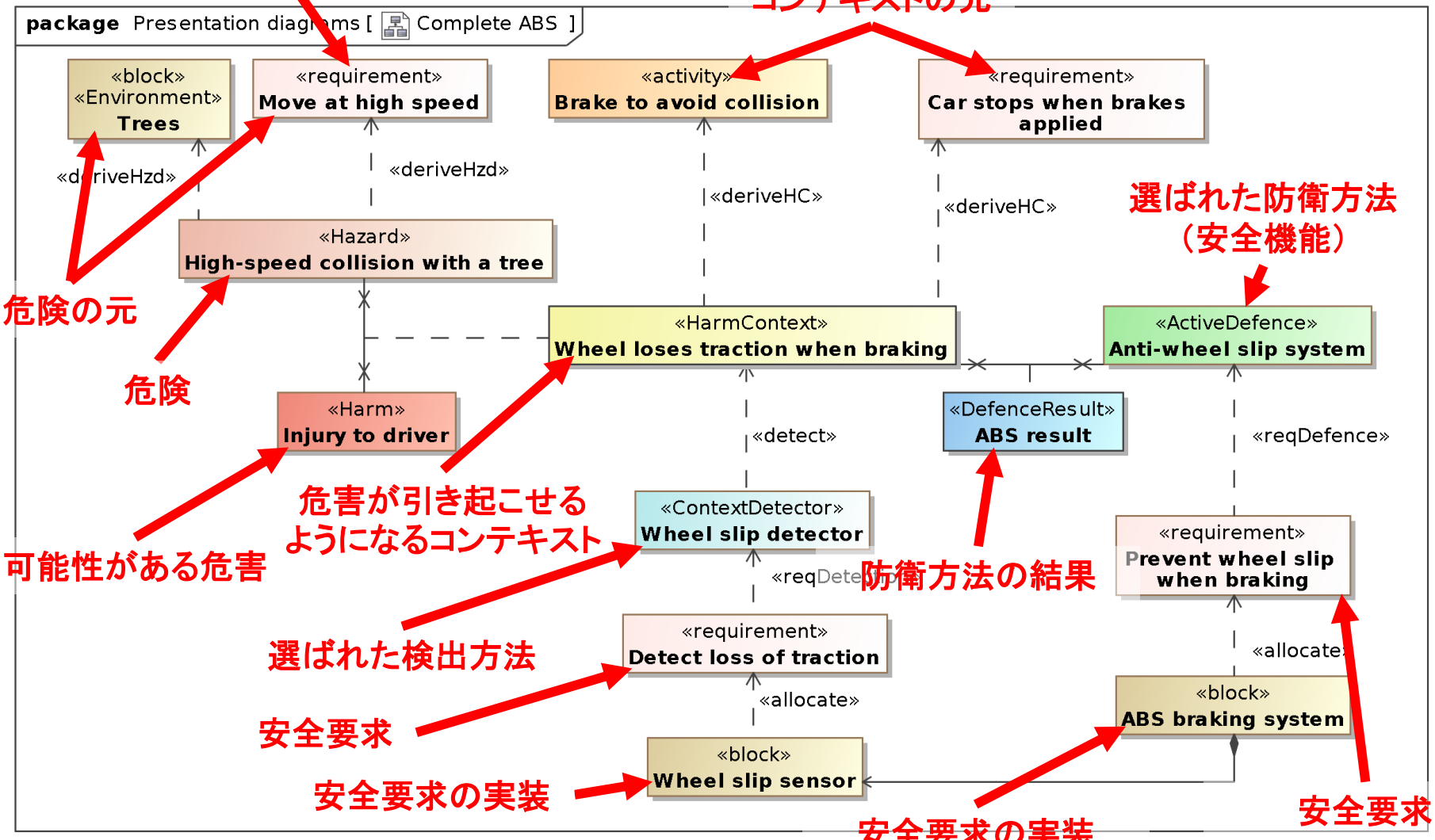
クラッシャブルゾーン安全機能



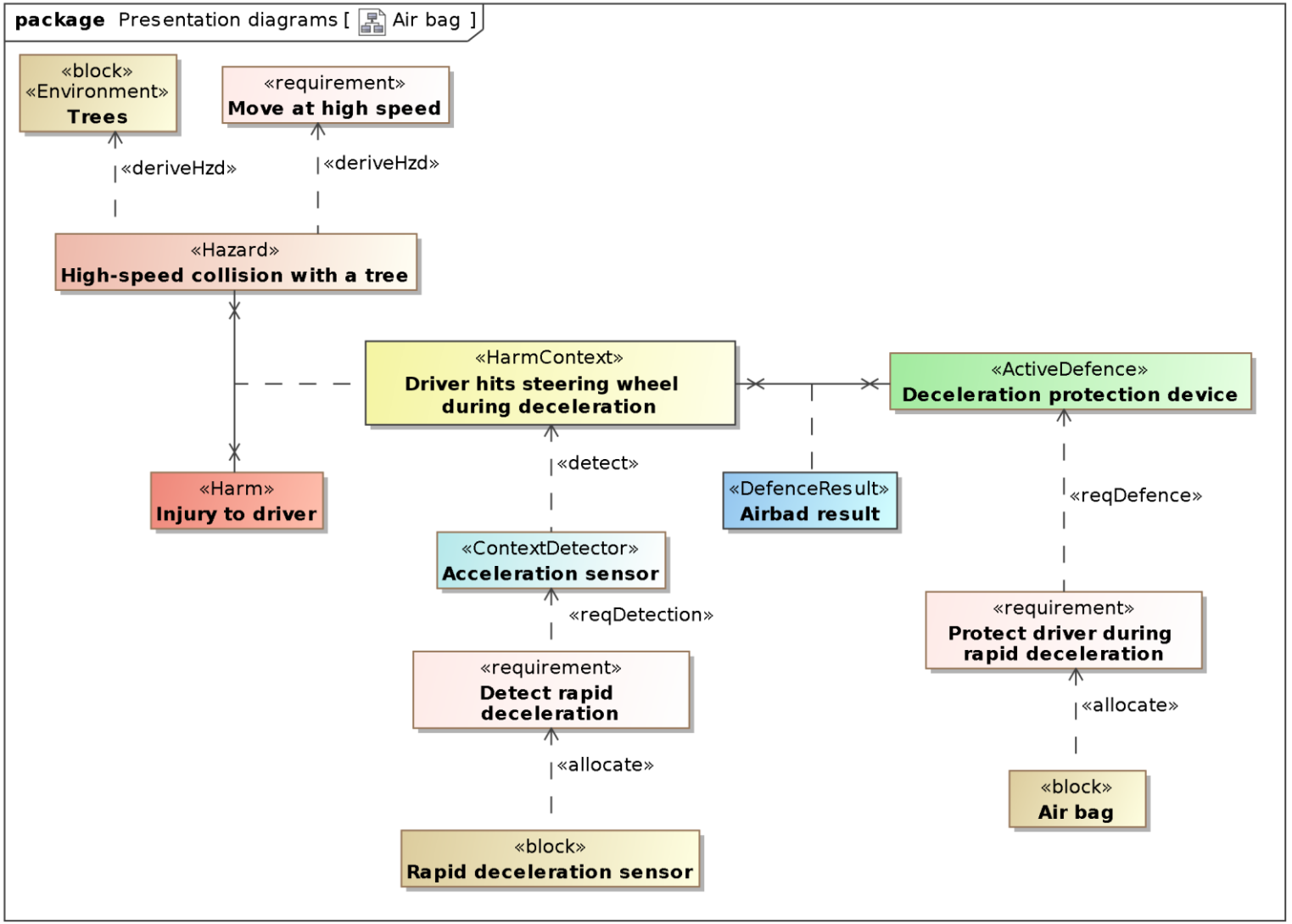
ABS 安全機能


package Presentation diagrams [Complete ABS]
«requirement»
Move at high speed
«activity»
Brake to avoid collision
«requirement»
Car stops when brakes applied
«deriveHzd»
«deriveHC»
«deriveHC»
«Hazard»
High-speed collision with a tree
«HarmContext»
Wheel loses traction when braking
«ActiveDefence»
Anti-wheel slip system
«Harm»
Injury to driver
«detect»
«DefenceResult»
ABS result
«reqDefence»
«ContextDetector»
Wheel slip detector
«requirement»
Prevent wheel slip when braking
«reqDetection»
«requirement»
Detect loss of traction
«allocate»
«allocate»
«block»
ABS braking system
«block»
Wheel slip sensor

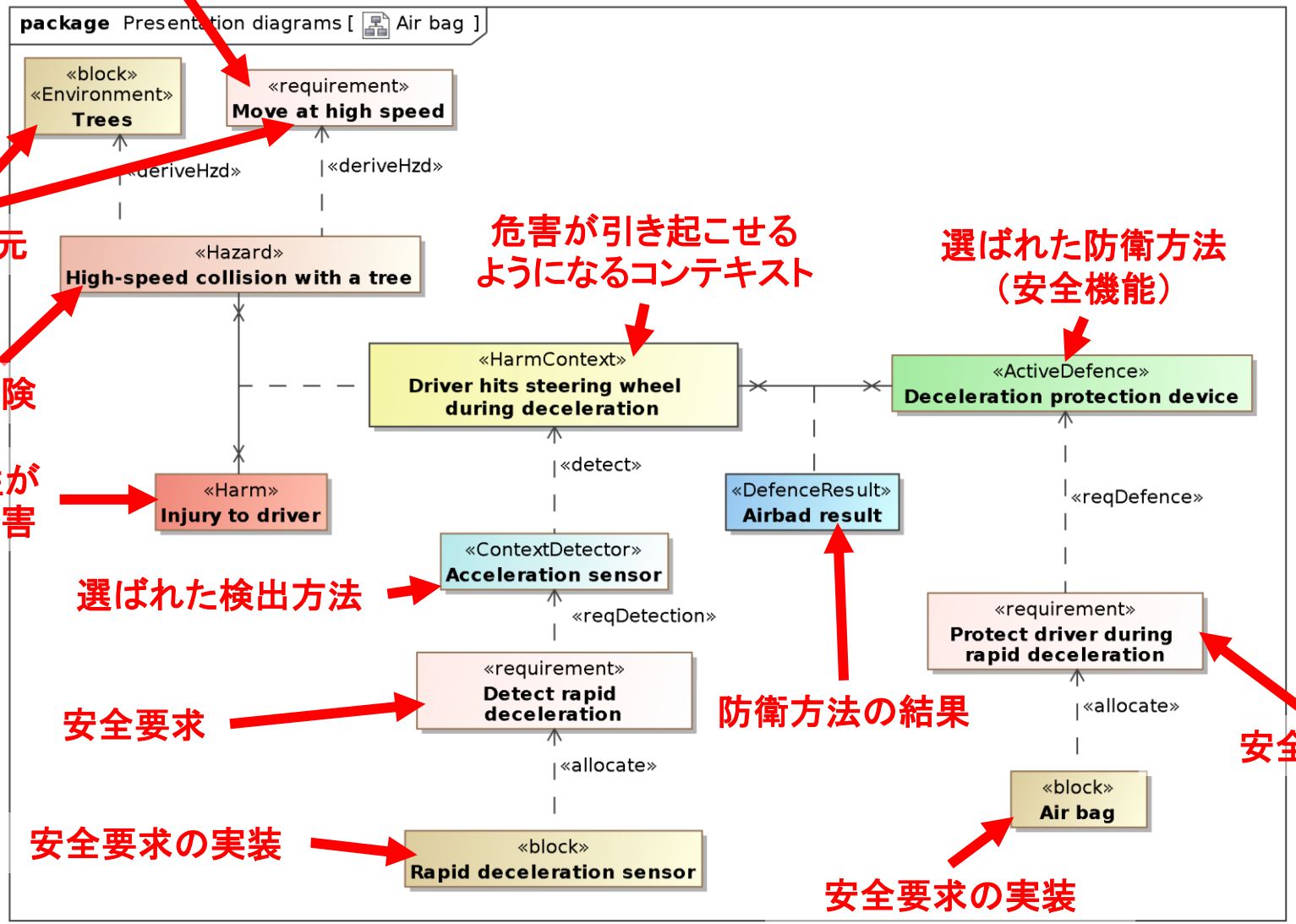
ABS 安全機能


システム要求
コンテキストの元
package Presentation diagrams [Complete ABS]
«block» «Environment» Trees
«requirement» Move at high speed
«activity» Brake to avoid collision
«requirement» Car stops when brakes applied
«deriveHzd»
«deriveHzd»
«deriveHC»
«deriveHC»
選ばれた防衛方法（安全機能）
«Hazard» High-speed collision with a tree
危険の元
危険
«HarmContext» Wheel loses traction when braking
«ActiveDefence» Anti-wheel slip system
«Harm» Injury to driver
«detect»
«DefenceResult» ABS result
«reqDefence»
危害が引き起こせるようになるコンテキスト
«ContextDetector» Wheel slip detector
可能性がある危害
防衛方法の結果
«requirement» Prevent wheel slip when braking
«reqDetection»
選ばれた検出方法
«requirement» Detect loss of traction
«allocate»
«allocate»
安全要求
«block» ABS braking system
«block» Wheel slip sensor
安全要求の実装
安全要求の実装
安全要求

Air bag 安全機能


package Presentation diagrams [Air bag]
«block»
«Environment»
Trees
«requirement»
Move at high speed
«deriveHzd»
«deriveHzd»
«Hazard»
High-speed collision with a tree
«HarmContext»
Driver hits steering wheel during deceleration
«ActiveDefence»
Deceleration protection device
«Harm»
Injury to driver
«detect»
«DefenceResult»
Airbad result
«reqDefence»
«ContextDetector»
Acceleration sensor
«reqDetection»
«requirement»
Protect driver during rapid deceleration
«requirement»
Detect rapid deceleration
«allocate»
«allocate»
«block»
Air bag
«block»
Rapid deceleration sensor

Air bag 安全機能


package Presentation diagrams [Air bag]
«block»
«Environment»
Trees
«requirement»
Move at high speed
«deriveHzd»
«deriveHzd»
«Hazard»
High-speed collision with a tree
«HarmContext»
Driver hits steering wheel during deceleration
«ActiveDefence»
Deceleration protection device
«Harm»
Injury to driver
«detect»
«DefenceResult»
Airbad result
«reqDefence»
«ContextDetector»
Acceleration sensor
«reqDetection»
«requirement»
Detect rapid deceleration
«requirement»
Protect driver during rapid deceleration
«allocate»
«allocate»
«block»
Rapid deceleration sensor
«block»
Air bag
システム要求
危険の元
危険
危害が引き起こせるようになるコンテキスト
選ばれた防衛方法（安全機能）
可能性がある危害
選ばれた検出方法
安全要求
防衛方法の結果
安全要求
安全要求の実装
安全要求の実装

Part 4

ツール

なぜツールは必要？

- モデルにある安全情報の自動処理
  - 自動ならエラーがないはずだ
- 間違えやすい処理や繰り返す処理の自動化
  - 安全性の設計にある問題探し
  - 設計変更の影響解析

ツールができること

- SafeMLメタモデルとの整合性を検証するモデルチェッカ
- 安全情報のいろいろな表示方法（ビュー）
  - 表（Excel風）
  - 文書（Word風）
  - マトリックス
- レポートの自動生成
- 新しい危険や防衛方法の影響解析

SafeMLでモデル化された安全情報の表

Harm	HarmContext	Defence	P(Occur)	P(Harm)	Range	Severity	P(Success)	SafetyScore	Cost
Water burns	Over-boiling due to failure by user to deactivate	Undefended case	Low	Low	One	S1	---	0.0123	---
		Auto cut-off	Low	Low	One	S1	High	0.0123	
	Over-boiling due to too much water	Undefended case	Medium	Low	One	S1	---	0.0246	---
		Water level sensor	Low	Low	One	S1	High	0.0123	
		Water limit indication mark on tank	Low	Low	One	S1	Medium	0.0123	
	Latch failure while pouring	***Undefended case***	***Low***	***High***	***Few***	***S1***	---	***0.074***	---
	Lid leak while pouring	Undefended case	Low	Low	One	S1	---	0.0123	---
		Sealed lid	Low	Low	One	S1	High	0.0123	
	Poor water guidance while pouring	Undefended case	Medium	High	One	S1	---	0.074	---
		Shaped spout	Low	High	One	S1	High	0.037	
Electrocution	Contact with live electrical components	Undefended case	Low	High	One	S3	---	0.1111	---
		Electrical insulation	Low	High	One	S3	High	0.1111	
Electrical fire	Contact between water and live electrical components	Undefended case	High	Medium	Many	S3	---	0.6666	---
		Waterproofing	Low	Medium	Many	S3	High	0.2222	
	Short circuit of live electrical components	Undefended case	Low	High	Many	S3	---	0.3333	---
		Electrical insulation	Low	High	One	S3	High	0.1111	
Heat burns	Contact with tank	Undefended case	Medium	High	One	S2	---	0.1481	---
		Heat insulation	Low	Low	One	S1	High	0.0123	
	Contact with element	***Undefended case***	***Low***	***High***	***Few***	***S1***	---	***0.074***	---
Steam burns	Steam build-up from boiled water	Undefended case	High	Medium	One	S1	---	0.074	---
		Steam venting	Low	Medium	One	S1	High	0.0246	

SafeMLでモデル化された安全情報の表

HarmContext	Defence	P(Occur)	P(Harm)
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
Contact with live electrical components	Undefended case	Low	High
	Electrical insulation	Low	High
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
Contact with tank	Undefended case	Medium	High
	Heat insulation	Low	Low

マトリックスビュー

	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

安全性にあるギャップの簡単な検索

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**

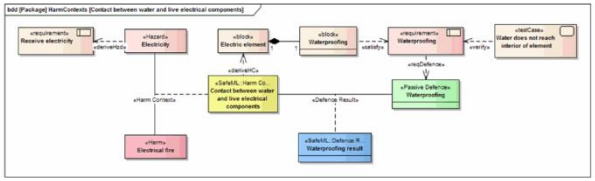
Signature: _____

1.Electrical fire

Safety score: 0.67
Hazard: **Electricity**
Sources:

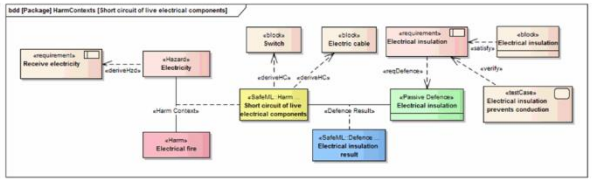
1.1.Harm Context: Contact between water and live electrical components

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が可能
  - 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を支援する
  - 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となる
  - 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安全情報を統合的に記述することが可能

More information

- SafeML ホームページ
  - https://staff.aist.go.jp/geoffrey.biggs/safeml/
- SafeMLプロファイルを**無料**で配布している
  - Enterprise Architect versions 9, 10 and 11
  - MagicDraw versions 17 and 18
- 講習会は可能

Appendix

Additional examples

応用の例

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

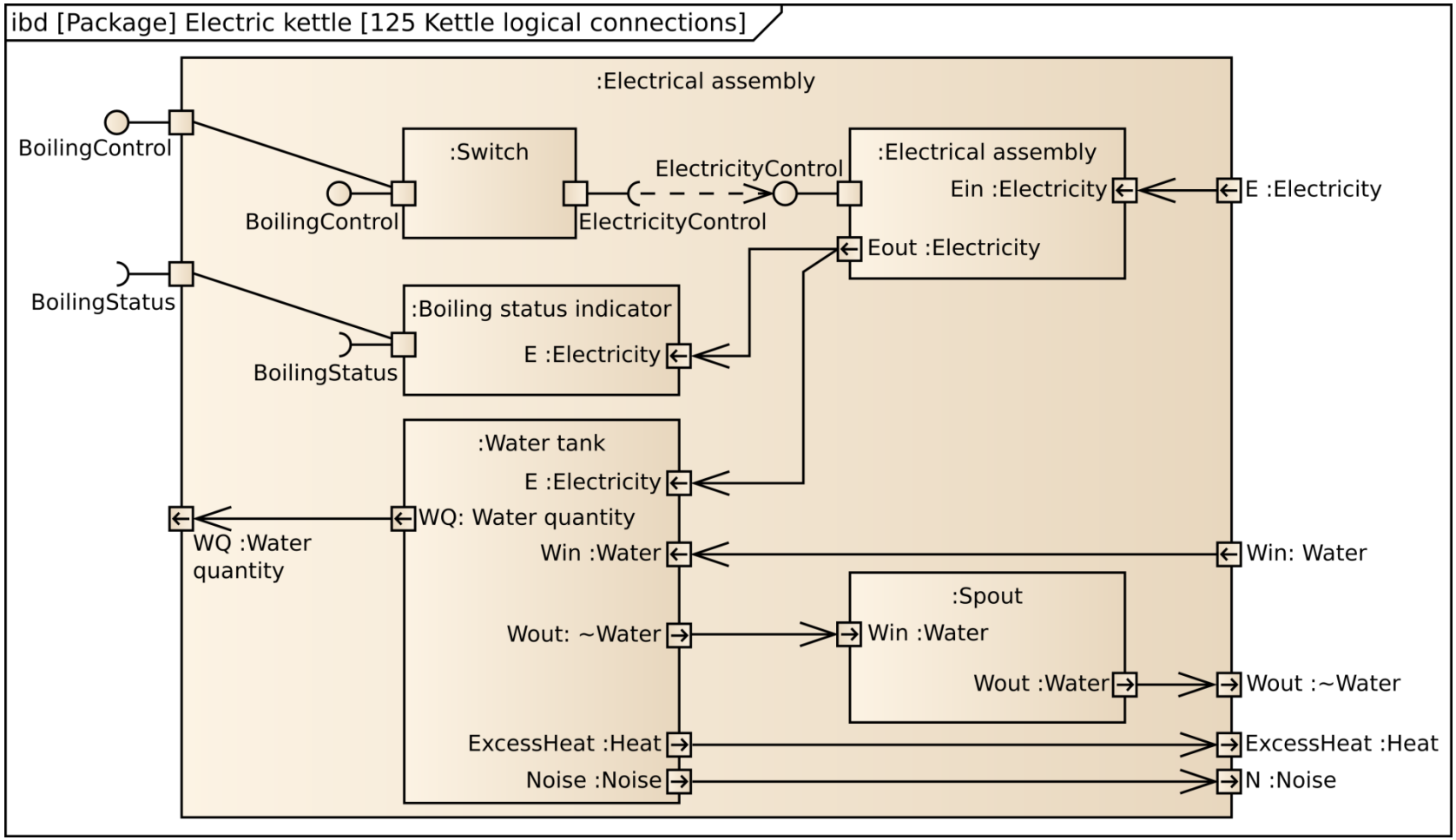
例

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

プロセスの例

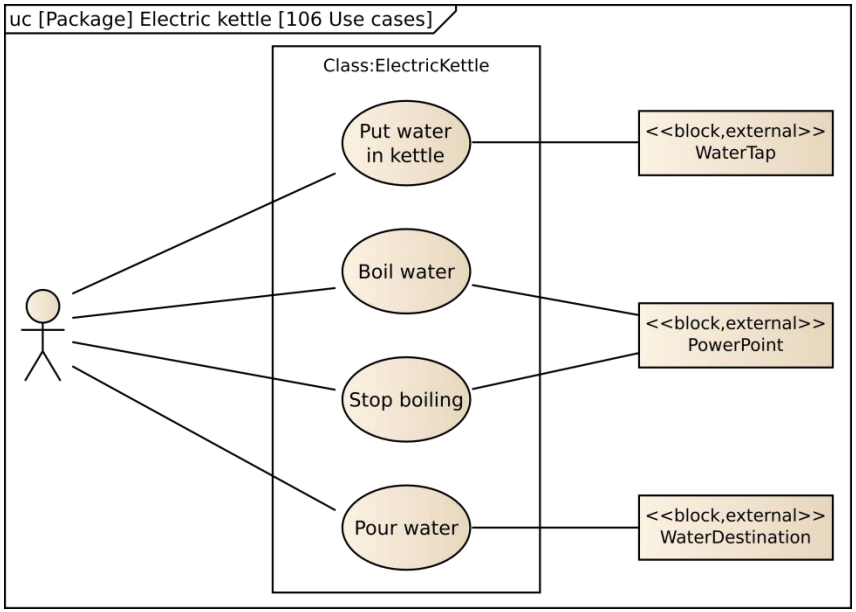
1. • 要求とユースケースの特定 → SysMLでモデル化
2. • ハザード分析、FTA → SafeMLでモデル化
3. • 安全フィーチャの設計 → SafeMLでモデル化
4. • モデルで安全フィーチャを設計に反映する
5. • 必要に応じて反復（1.に戻る）

電気ポットの例 – システム設計



（モデルの一部だけ）

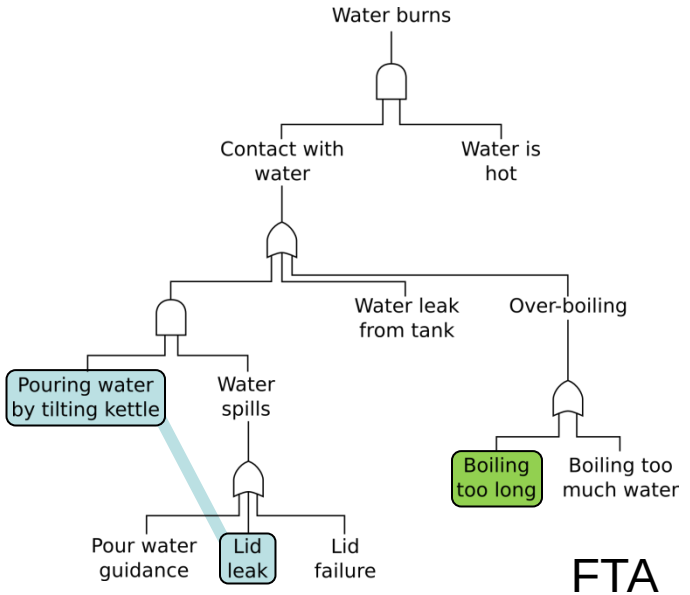
電気ポットの例 – 安全分析



ユースケース

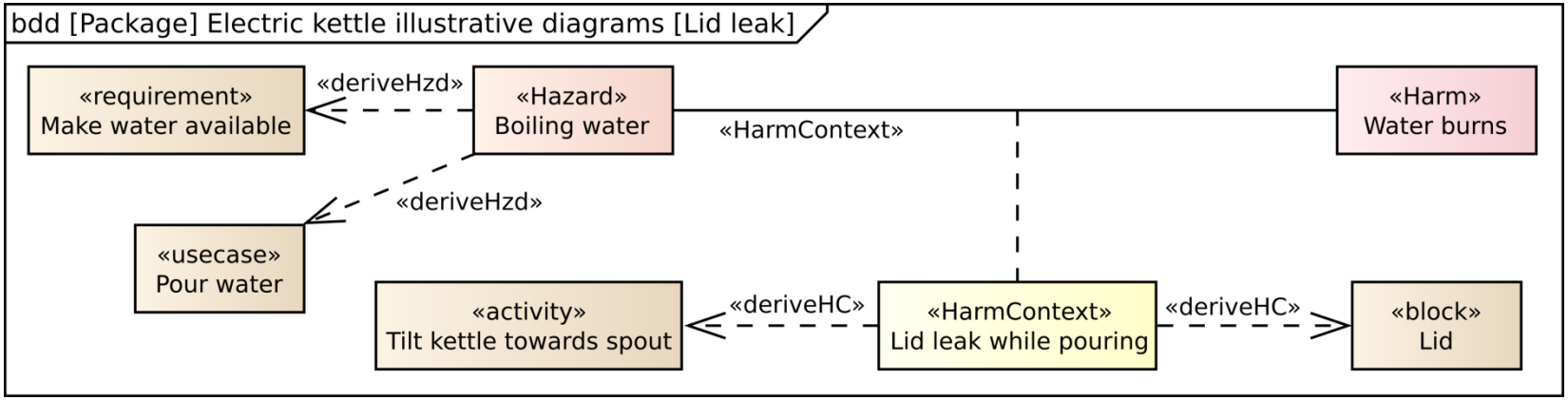
ハザード	危害
電気	感電、火事
熱	やけど、火事
蒸気	やけど
湯	やけど

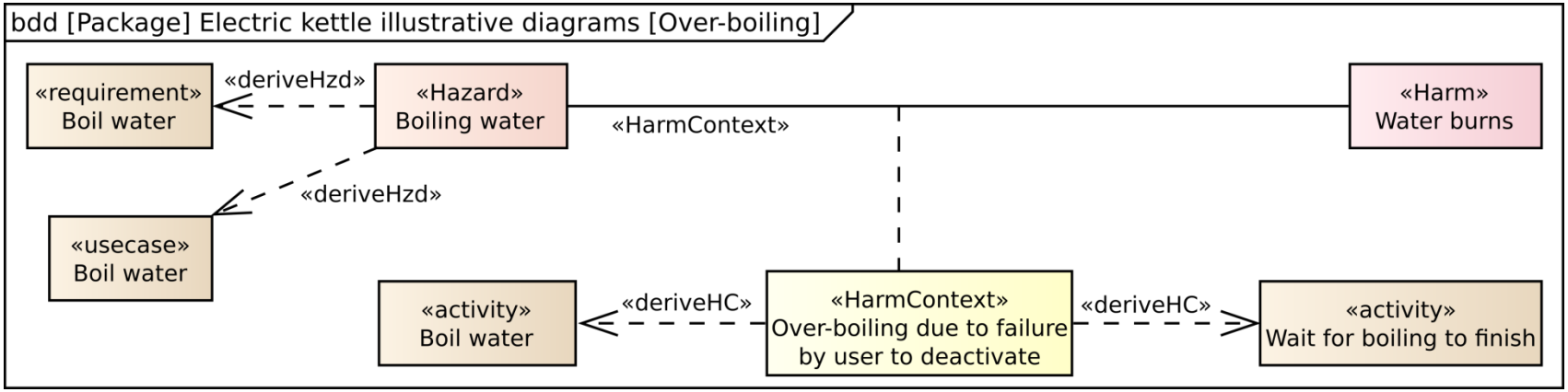
ハザード分析



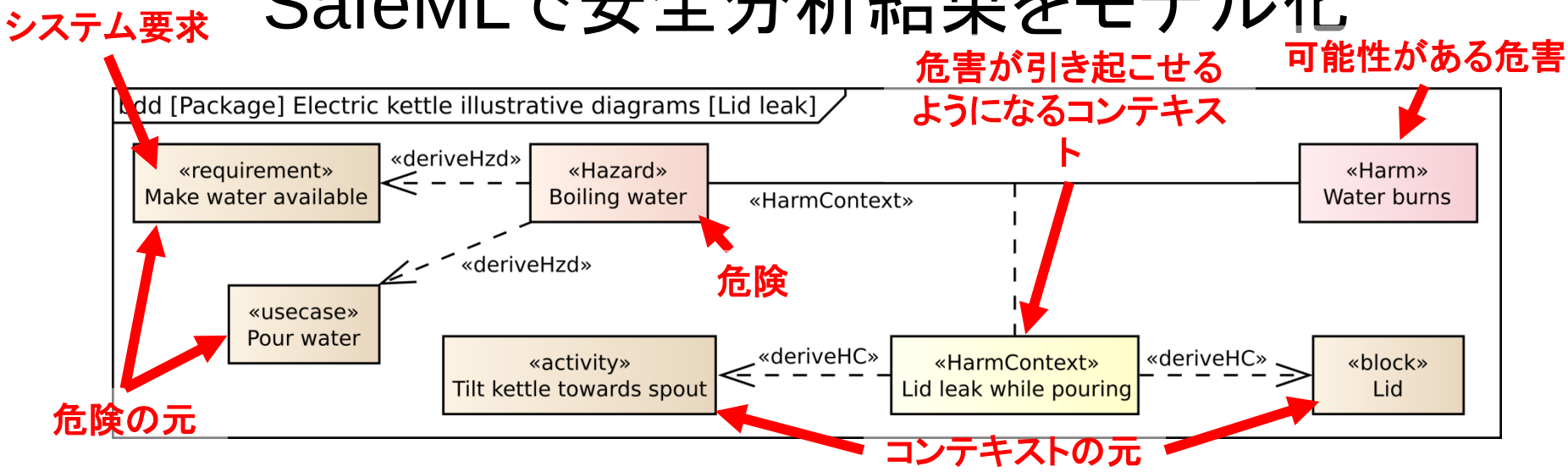
FTA

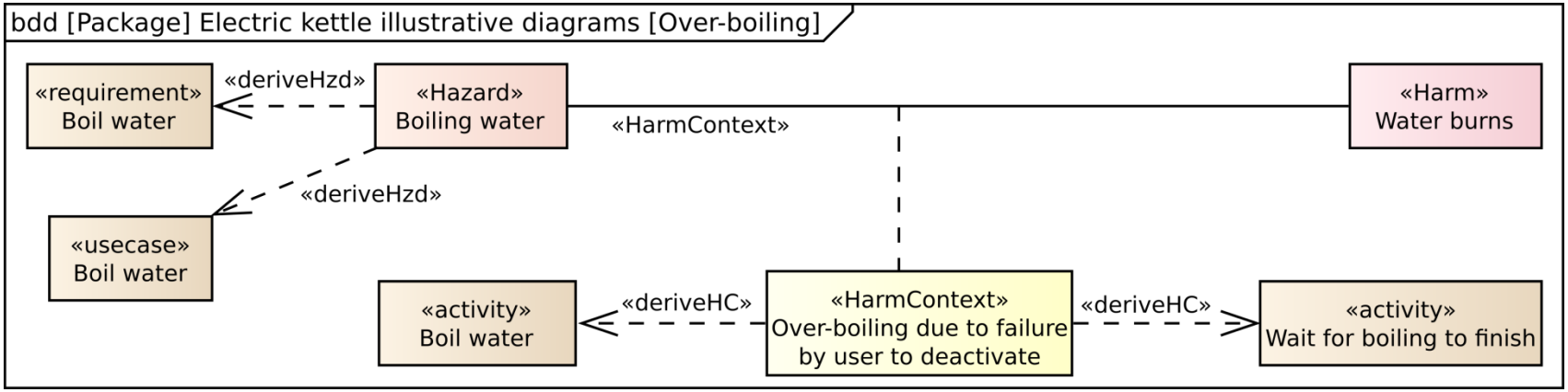
電気ポットの例 – SafeMLで安全分析結果をモデル化



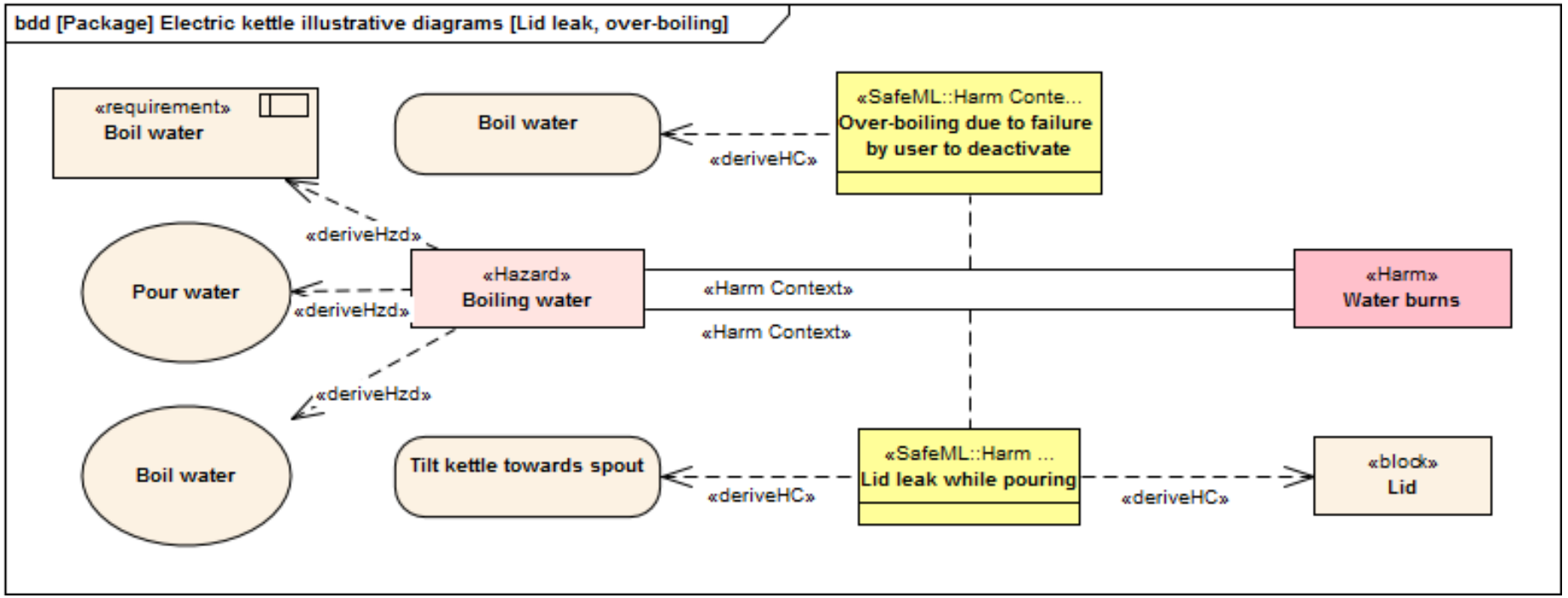


電気ポットの例 – SafeMLで安全分析結果をモデル化





Tip: Hazard/harm/contextによって図を分ける

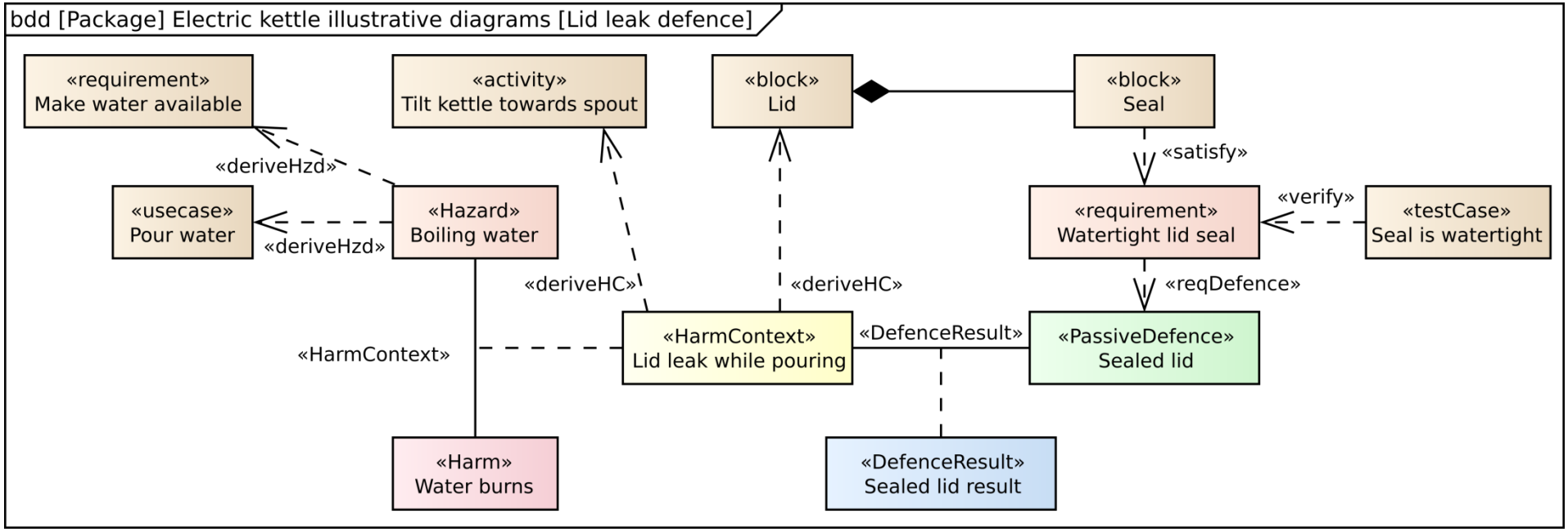


これは前スライドと
同じ内容だ

電気ポットの例 – 安全フィーチャ

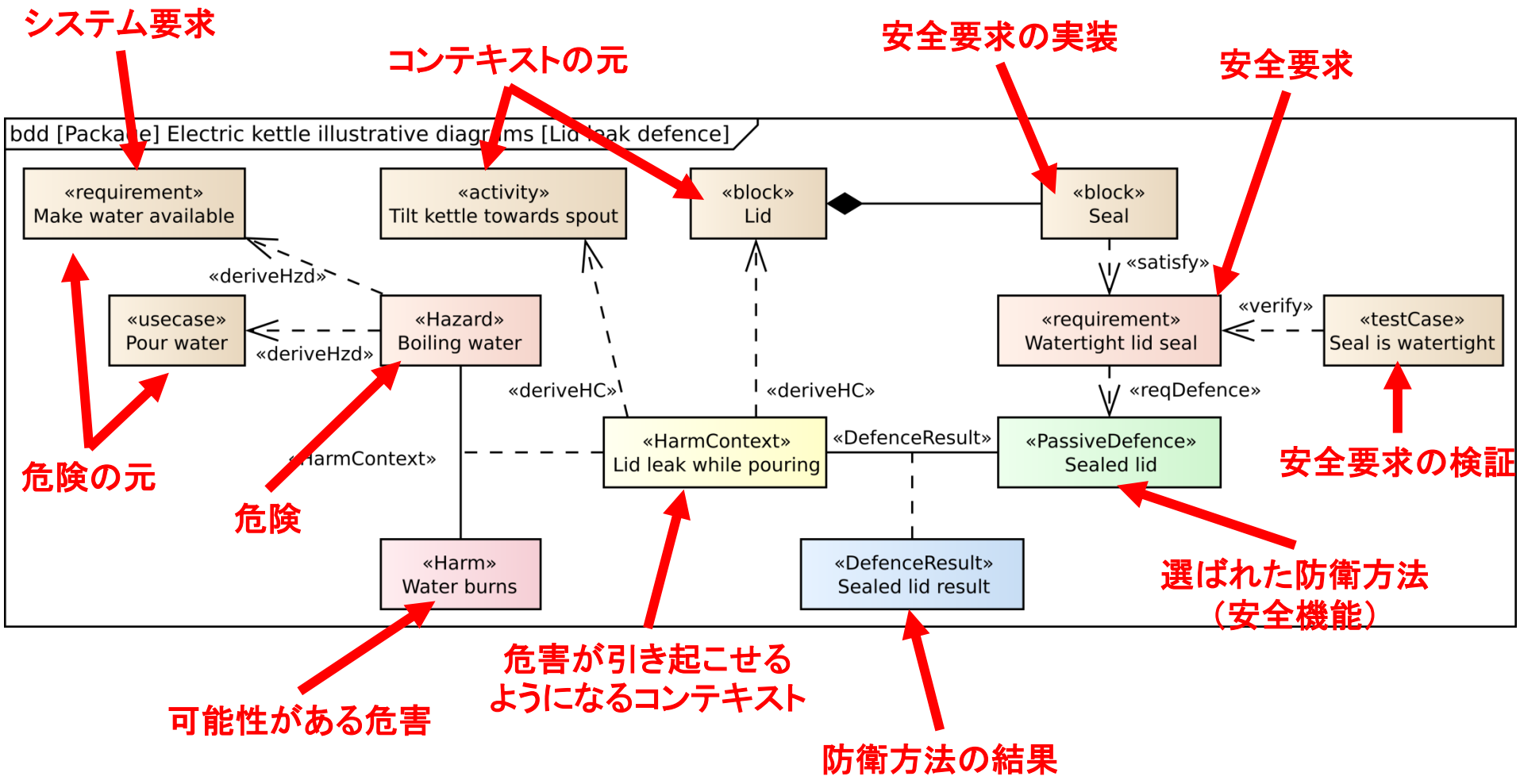
- 湯漏れ
  - 防衛：ふたにゴムパッキンを使う
- 沸かしすぎ
  - 防衛：自動ストップ
  - センサが必要

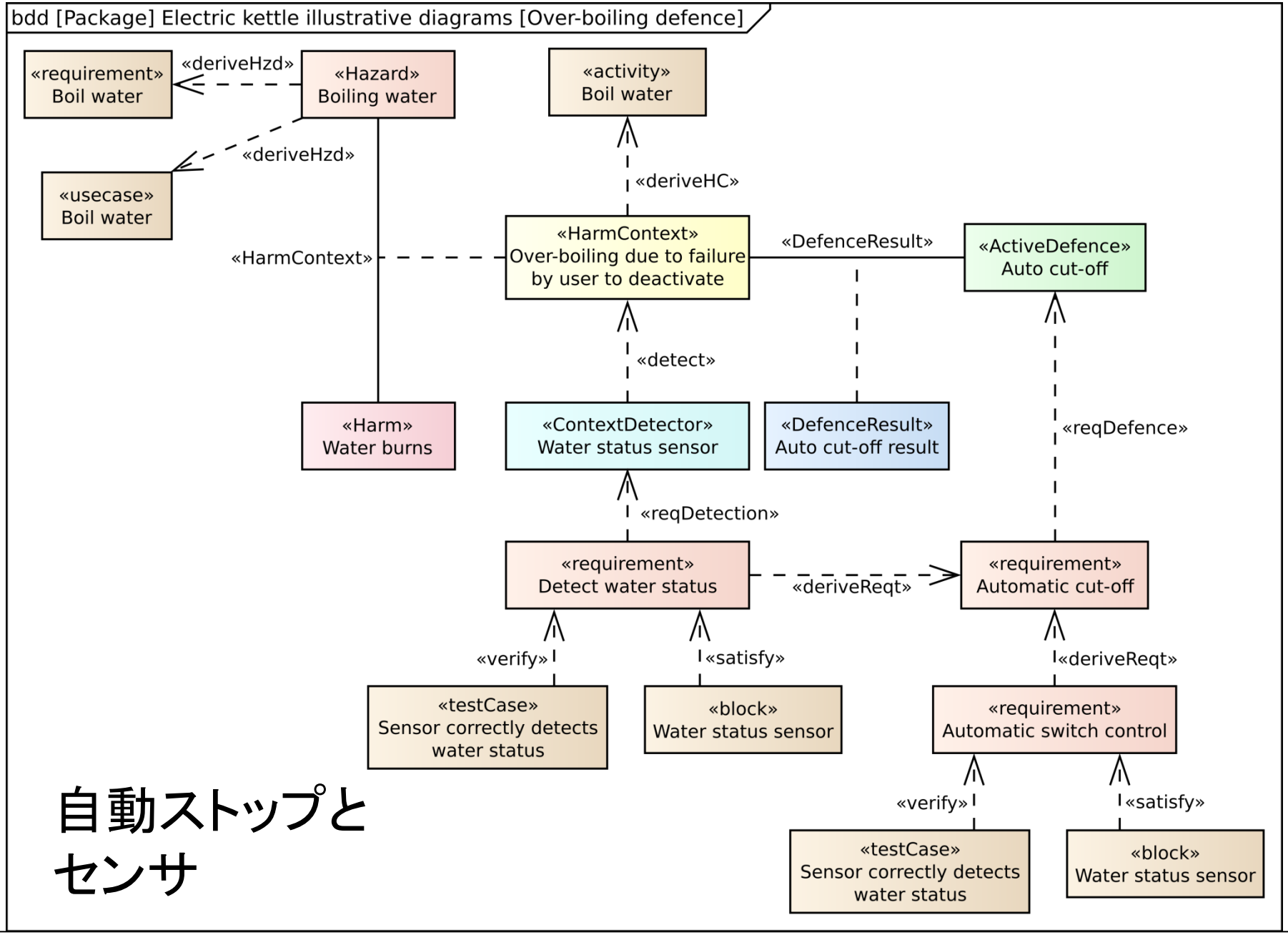
電気ポットの例 – 安全フィーチャのモデル化



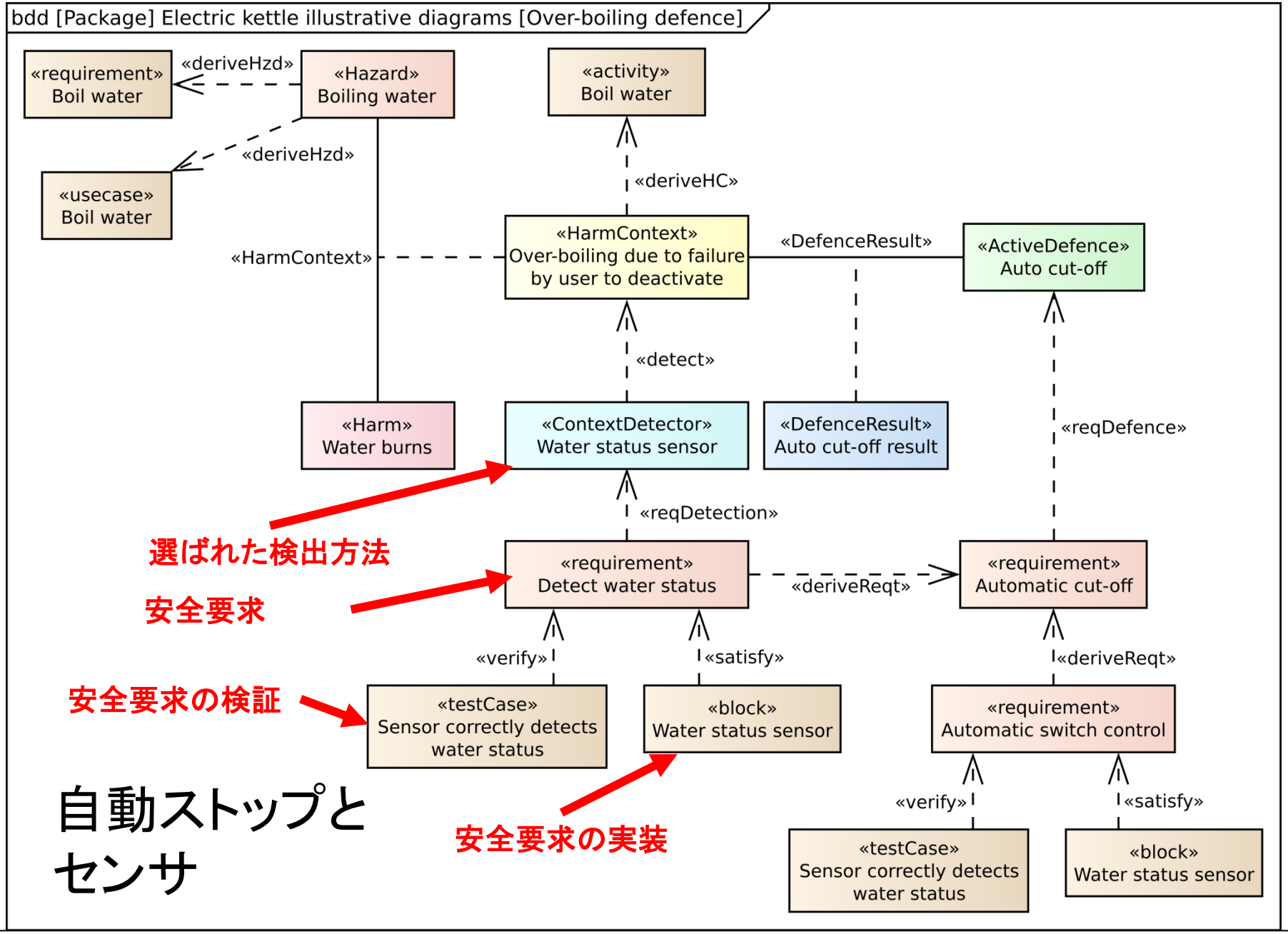
ふたのゴムパッキン

電気ポットの例 – 安全フィーチャのモデル化




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

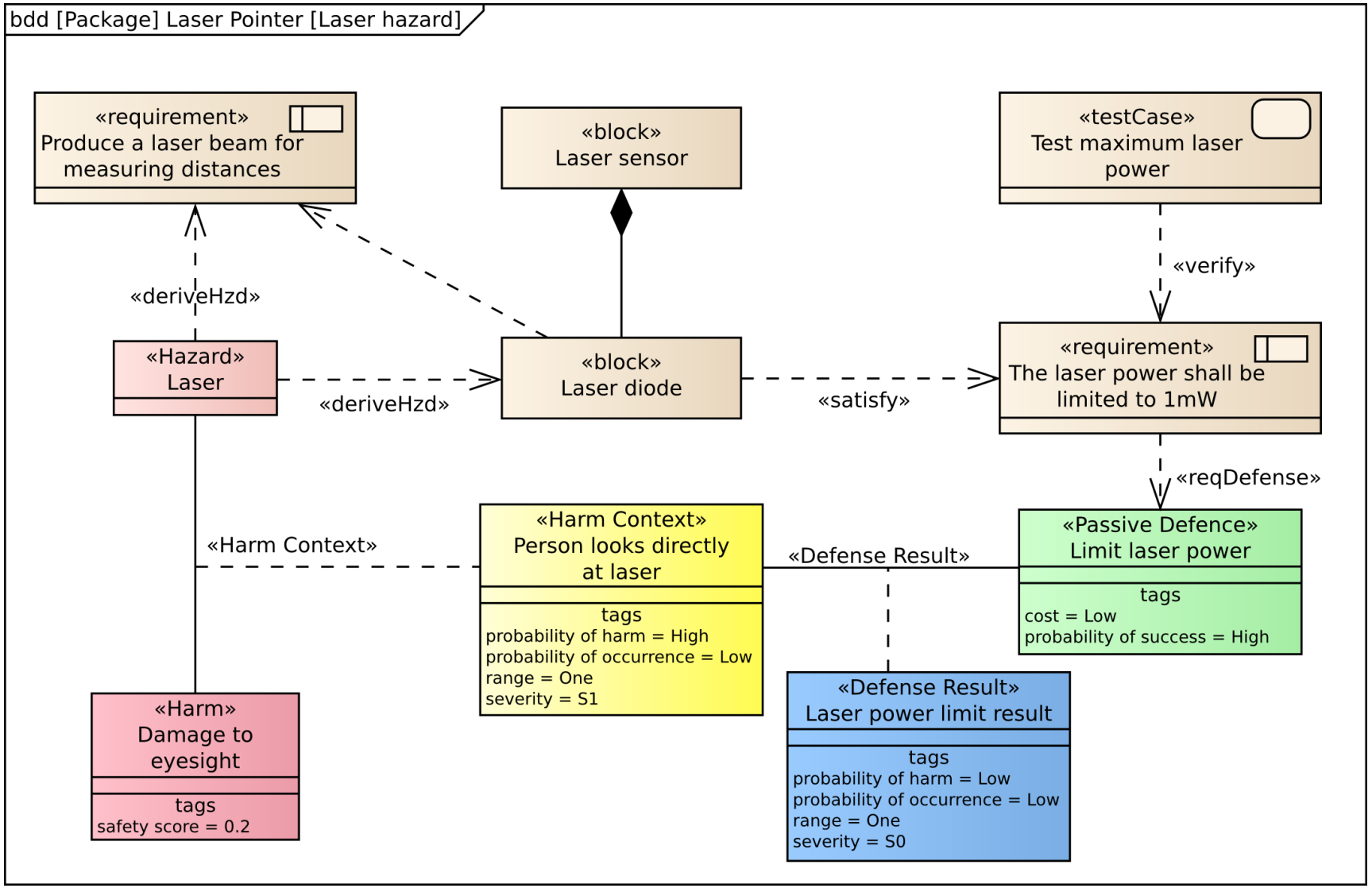
自動ストップとセンサ


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»
選ばれた検出方法
安全要求
«requirement»
Detect water status
«deriveReqt»
«requirement»
Automatic cut-off
«verify»
«satisfy»
«deriveReqt»
安全要求の検証
«testCase»
Sensor correctly detects
water status
«block»
Water status sensor
«requirement»
Automatic switch control
自動ストップと
センサ
安全要求の実装
«verify»
«satisfy»
«testCase»
Sensor correctly detects
water status
«block»
Water status sensor

応用例2:レーザのパワー制限

- レーザポインタはレーザを出すことが必要
- レーザは危険
  - ある程度力があると視力へ危害を
- *Hazard*: レーザ
- *Harm*: 視力への危害
- *Context*: 人がレーザを放射するところに見る
- *Defense*: レーザの力に制限を使う

応用例2:レーザのパワー制限



応用例3： 車椅子ロボット

- ロボット車椅子は歩道者との衝突を防ぐことが必要
- 衝突はけがを起こす可能性がある
- *Hazard*: 歩道者
- *Harm*: けが
- *Context*: 車いすは歩道者と衝突する
- *Defense*: 自動緊急停止
- *Monitoring*: 歩道者の存在と位置の検出

応用例3： 車椅子ロボット