SEVENTH FRAMEWORK PROGRAMME

Large-scale integrating project (IP)

OPENCOSS

Open Platform for EvolutioNary Certification Of Safety-critical Systems

The role of the safety-case lexicon in cross-domain translation: the OPENCOSS Project

Transferable Safety Seminar, London
5th December, 2013

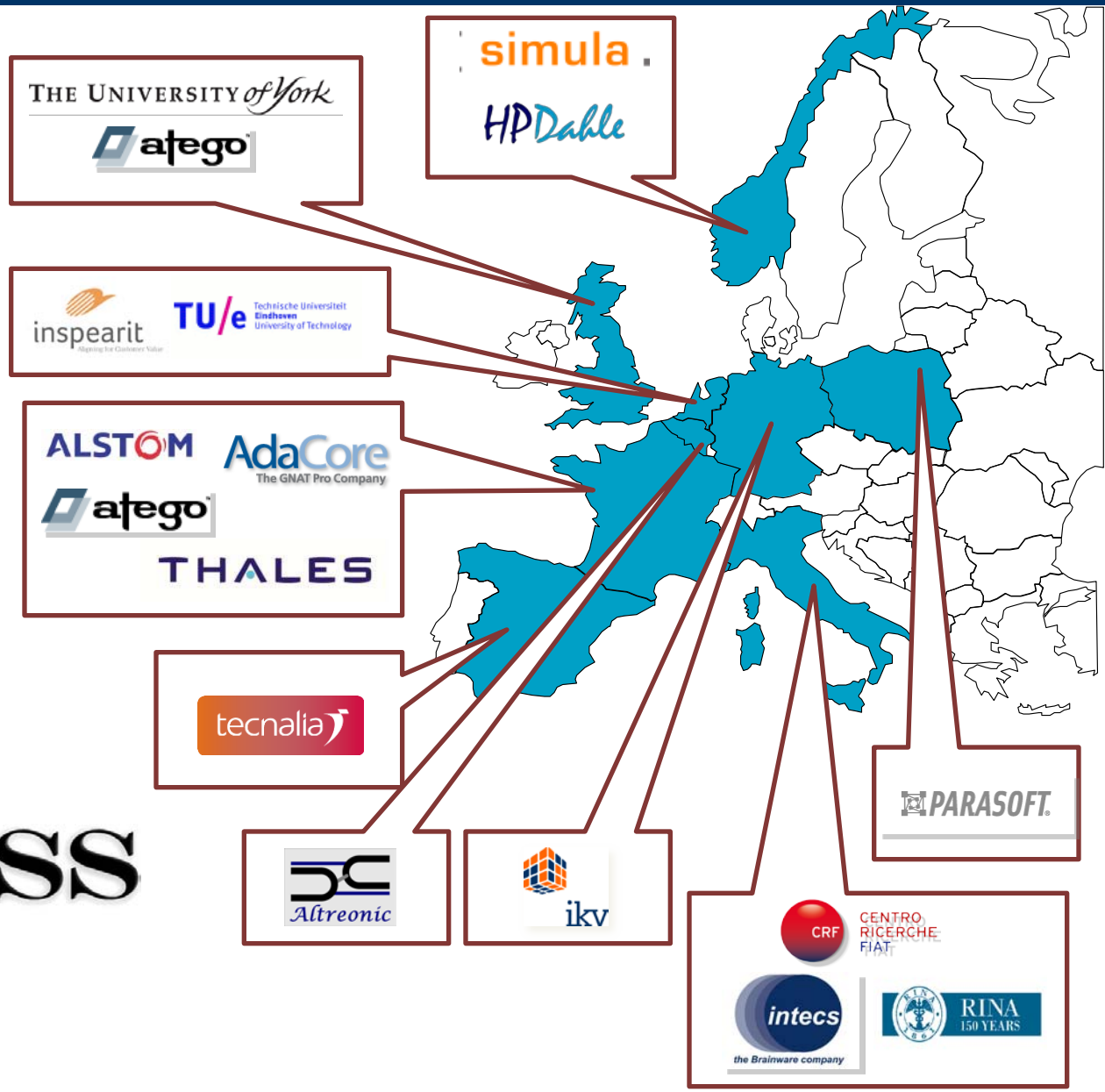
Katrina Attwood
University of York

Outline

- Aims of the OPENCOSS Project with respect to Transferable Safety
- Challenges and Opportunities
- Models and Vocabulary - Common Certification Language
- Simple Example

OPENCOSS at a Glance

Main Figures	
Duration:	42 months
Budget:	~M€ 11,79
EC contrib:	~M€ 8,44
Total Effort:	968 PMs
Consortium:	17 partners - 9 countries →3 Manufacturers →1 Certif. Body →5 Solution Providers →3 Consultancy →2 Universities →2 Research



OPENCOSS

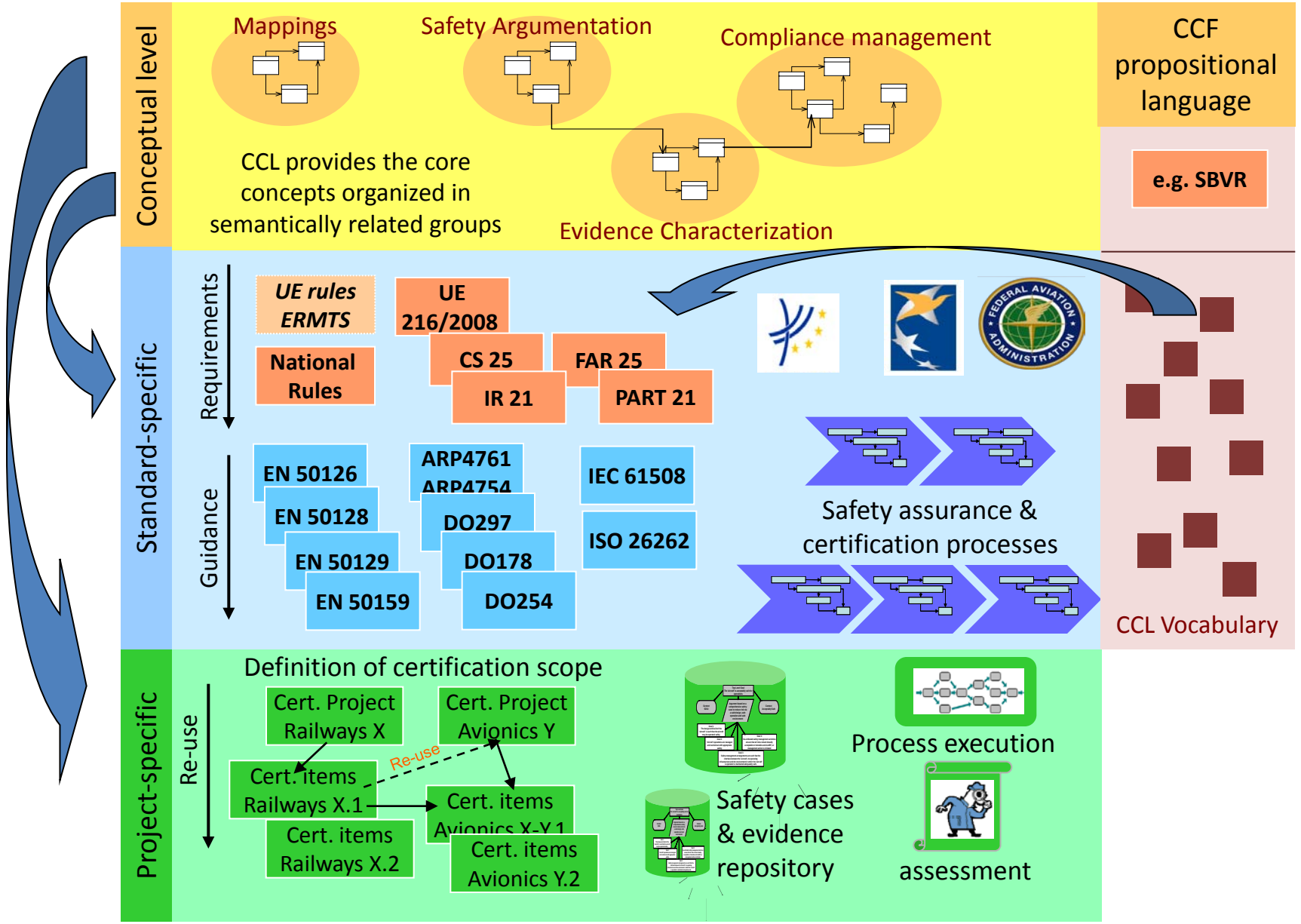
OPENCOSS Mission Statement

- *Conceptual Certification Framework*
  - The common certification language (CCL)
    - Target domains: Railway, Avionics, Automotive
  - A compositional certification approach.
    - Reuse safety arguments, safety evidence and contextual information about system components
    - "in a way that makes certification more cost-effective, precise and scalable"
- *Safety Certification Management Infrastructure*
  - Management of evolutionary evidential chain
  - Management of metrics for a transparent certification process
  - Management of a compliance-aware process
- *Open-Source Tooling*

Challenges and Opportunities

- *Reuse is the backbone*
  - Of assurance assets – evidence, argument, context
- *Claims made for a particular component in a defined context*
  - Specific evidence to support the claim
  - Switching context may invalidate the evidence and thus undermine the claim
- *Argument made against domain-specific standards*
  - If we are re-using argument and evidence what is required in the new context?
  - Do the standards require the same things?

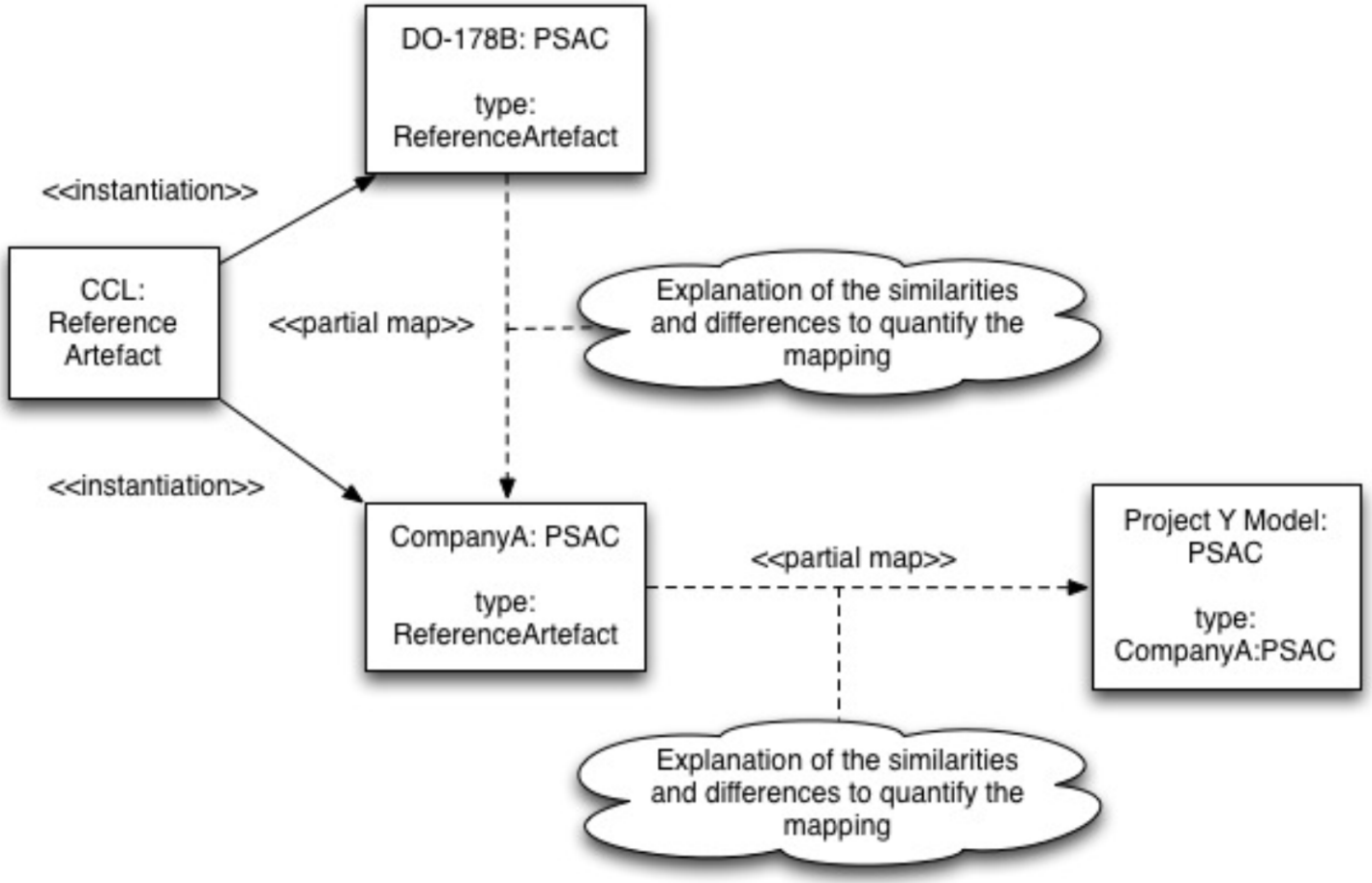
Common Certification Language



Common Certification Language

- *Exploit commonalities between concepts of certification across the domains*
  - Model-based approach
  - Metamodel identifying concepts common to assurance across the domains
  - e.g. Activity, Objective, Artefact
- *Domain-, Standard- and Project-Specific glossaries to support the concepts*
  - More detailed definitions required to indicate commonalities and differences
  - Language used to characterise: concepts, assurance assets, activities, objectives, requirements, argument claims, contextual assumptions
  - To establish whether there is sufficient similarity to consider reuse
- *"Mapping" concept used to indicate similarities and differences*
  - Information used to support user decisions

Mappings

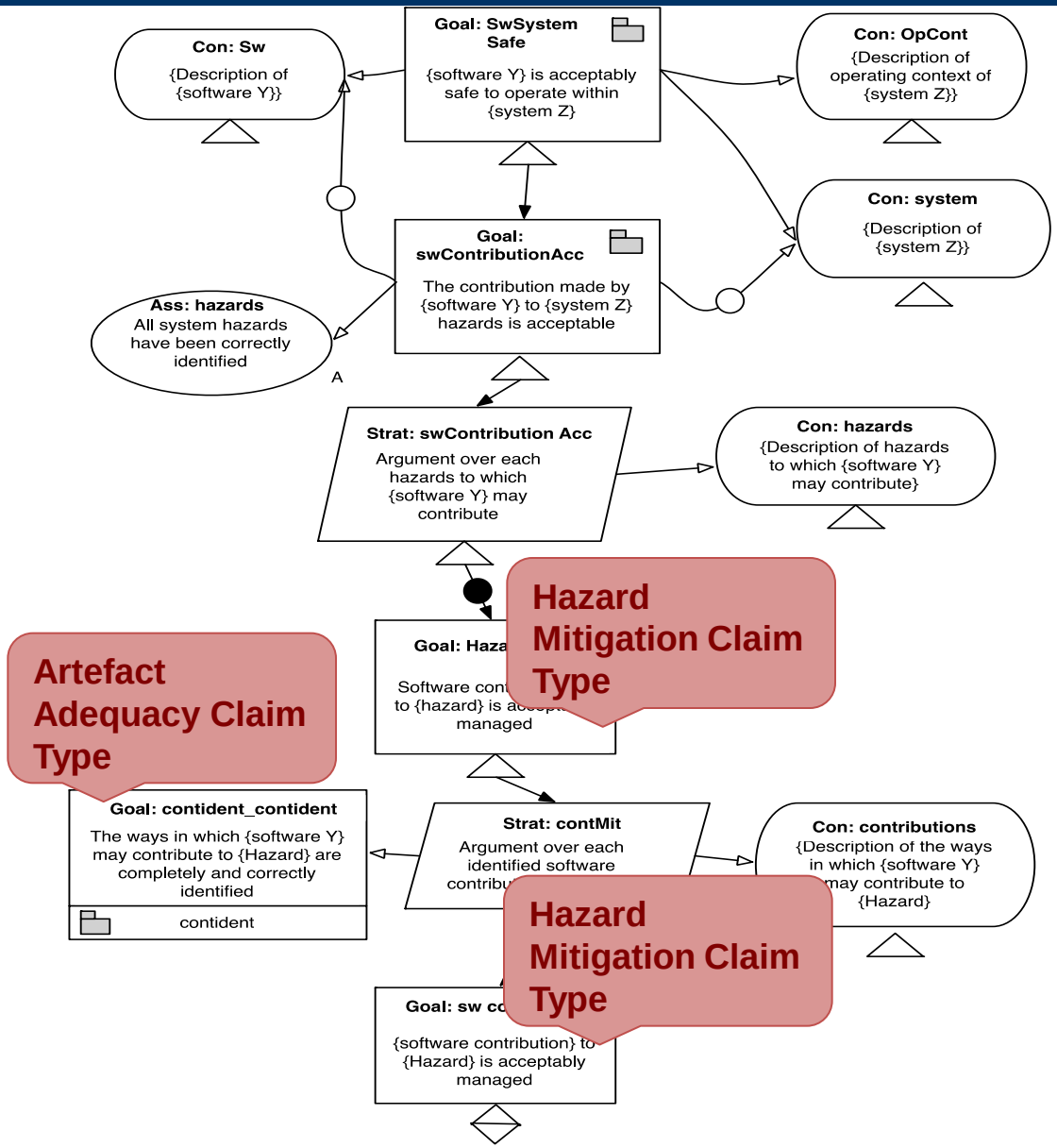

DO-178B: PSAC
type:
ReferenceArtefact
<<instantiation>>
CCL:
Reference
Artefact
<<partial map>>
Explanation of the similarities and differences to quantify the mapping
<<instantiation>>
CompanyA: PSAC
type:
ReferenceArtefact
<<partial map>>
Project Y Model:
PSAC
type:
CompanyA:PSAC
Explanation of the similarities and differences to quantify the mapping

Vocabulary Representation

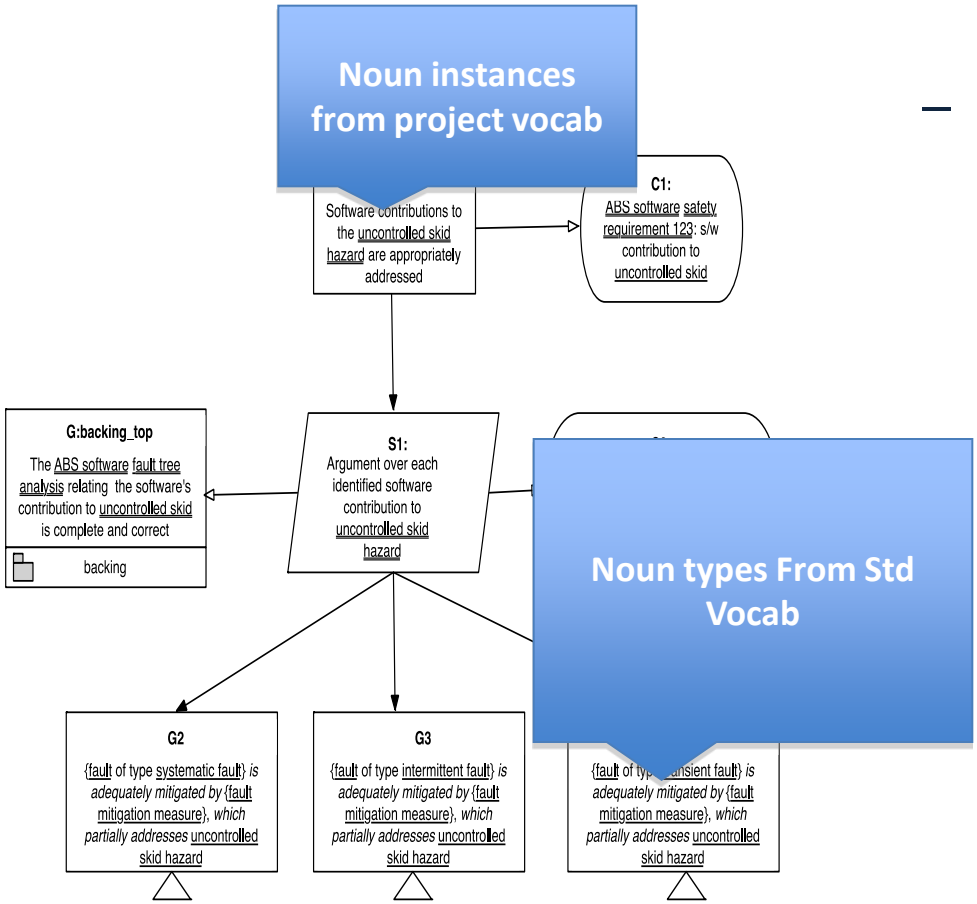
- *3-Layered Structure*
  - Vocabulary from Standard, Company Standard and Project
  - Mapping types to indicate relationships

- *Using SBVR (Semantics of Business Vocabulary and Business Rules)*
  - Controlled, potentially closed vocabulary
  - Some automated checking possible
  - Concept definitions and fact types
  - Concept types provide a possible basis for mappings
  - In some cases, SBVR fact types enable us to specify a generic claim type and populate from the vocabularies

Single-Domain Example (1)

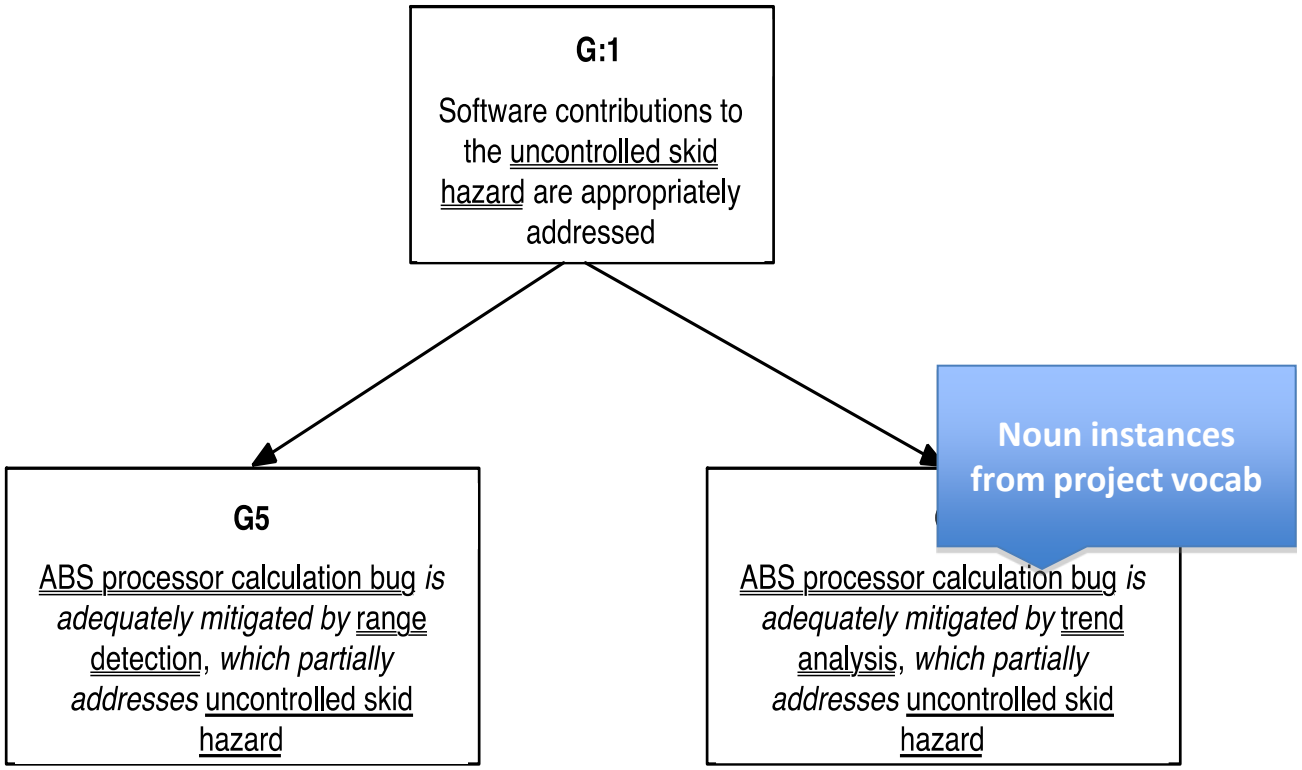


Single-Domain Example (2)



- G1 has two generic claim types: Fault Accommodation Claim and Hazard Mitigation Claim
  - Parameterized with generic noun types from SBVR model of ISO 26262 § 3
  - SBVR gives us underlying conceptual model, which we use to generate fact types:
    - fault *causes* at least one failure behaviour
    - failure behaviour *may lead to* hazard
    - safety measure *mitigates* fault

Single-Domain Example (3)



Cross-Domain Example (1)

- *Reuse of a software component*
  - Generic health monitoring component
  - Developed in US military aerospace context where it is used to monitor whether landing gear has descended
    - Applicable standards include Mil Std 882d
  - Reused in European automotive context, to monitor whether brakes have failed to operate
- *Differences in the application domains need to be considered*
  - Timing more of an issue in automotive
  - Superficial similarities between terms (see next slide) …
  - … but some subtle differences may have important consequences for assurance effort

Cross-Domain Example (2)

US Mil Std 882d (system safety)

hazard
Definition: real or potential condition that *can cause* injury, illness or death to personnel, damage or loss to a system, equipment or property, or damage to the environment
Source: Mil Std 882d, § 3.2.3
Dictionary Basis: Mil Std 882d,
General Concept: condition
Possibility: hazard *causes* mishap

mishap
Definition: unplanned event or series of events that *results in* injury, illness or death to personnel, damage or loss to a system, equipment or property, or damage to the environment
Source: Mil Std 882d, § 3.2.6
Dictionary Basis: Mil Std 882d,
General Concept: event
Necessity: (1) mishap *has* one or more consequences
(2) mishap *has* mishap risk
(3) mishap *is assigned to* mishap severity category
(4) mishap *has* probability of occurrence
(5) mishap *is caused by* hazard

ISO 26262

hazard
Definition: potential source of harm caused by malfunctioning behaviour of an item
Dictionary Basis: ISO 26262 Part 1, § 1.56
General Concept: condition
Necessity: (1) hazard *has* cause
(2) hazard *has* effect
(3) hazard *is assigned to* severity class
Possibility: hazard *causes* hazardous event

hazardous event
Definition: event that results from a combination of a hazard and an operational situation
Dictionary Basis: ISO 26262 Part 1, § 7.1
General Concept: event
Necessity: (1) hazardous event *has* one or more consequences

Mil Std 882d Mishap Severity Categorisation

TABLE A-I. **Suggested mishap severity categories.**

Description	Category	Environmental, Safety, and Health Result Criteria
Catastrophic	I	Could result in death, permanent total disability, loss exceeding $1M, or irreversible severe environmental damage that violates law or regulation.
Critical	II	Could result in permanent partial disability, injuries or occupational illness that may result in hospitalization of at least three personnel, loss exceeding $200K but less than $1M, or reversible environmental damage causing a violation of law or regulation.
Marginal	III	Could result in injury or occupational illness resulting in one or more lost work days(s), loss exceeding $10K but less than $200K, or mitigatible environmental damage without violation of law or regulation where restoration activities can be accomplished.
Negligible	IV	Could result in injury or illness not resulting in a lost work day, loss exceeding $2K but less than $10K, or minimal environmental damage not violating law or regulation.

ISO 26262 Concepts Contributing to ASIL Calculation

Table 1 — Classes of severity

	Class			
	S0	**S1**	**S2**	**S3**
Description	No injuries	Light and moderate injuries	Severe and life-threatening injuries (survival probable)	Life-threatening injuries (survival uncertain), fatal injuries

Table 2 — Classes of probability of exposure regarding operational situations

	Class				
	E0	**E1**	**E2**	**E3**	**E4**
Description	Incredible	Very low probability	Low probability	Medium probability	High probability

Table 3 — Classes of controllability

	Class			
	C0	**C1**	**C2**	**C3**
Description	Controllable in general	Simply controllable	Normally controllable	Difficult to control or uncontrollable

Conclusion

- *No common approach to safety in different domains*
  - "Harmonisation" of concepts, vocabulary and standards very difficult
  - Politically, technically and academically
  - A "common language" for safety is a very long way off
    - But we propose a mappings mechanism for pairwise comparison
- *Need for a clear understanding of similarities and differences to inform reuse*
  - Combination of model-based and vocabulary-based approaches may help provide guidance to engineers/argument developers
  - Clear understanding of the context implicit in claims made in assurance arguments
  - "Push-button reuse" of argumentation is not possible
    - Or desirable?