GDAŃSK UNIVERSITY OF TECHNOLOGY

ARGEVIDE

Use of Evidence Based Arguments in Standard Compliance

Managing Safety Case Relations to System Models

Andrzej Wardziński
Gdańsk University of Technology, Poland
ARGEVIDE sp. z o.o., Gdańsk, Poland

SCSSS 2017
23 May 2016, Stockholm

Agenda

Part 1: Use of Evidence Based Arguments in Standard Compliance

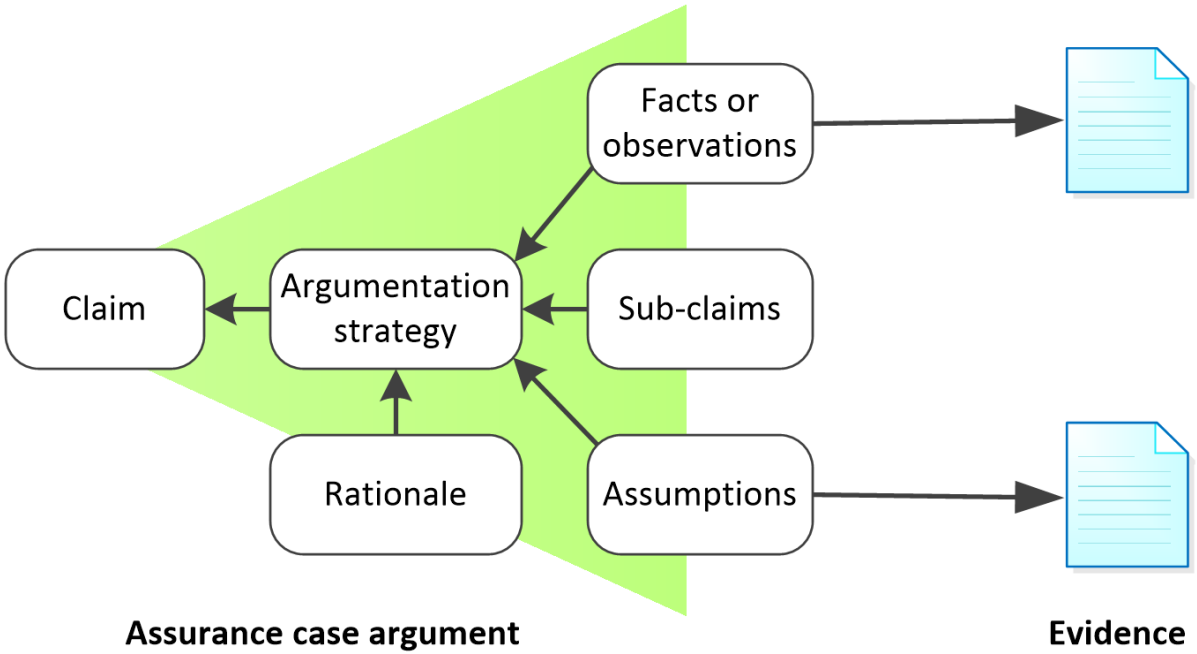
1. Evidence based arguments
2. Standard structure and requirements
3. Demonstrating compliance and making assessment
4. Managing standards

Part 2: Managing Safety Case Relations to System Models

1. References to the system context
2. System model
3. Establishing and maintaining relations

Evidence based arguments

- Argument structure based on Toulmin's argument model
  - comply with ISO 15026 and OMG SACM
- Argument premises may be supported by evidence



Argument notations

Available argument notations:

- **graphical** notations (GSN, CAE)
- **tabular** notation
- **hierarchical** textual notation
  - **TCL** – Trust Case Language
    - developed at Gdańsk University of Technology in 2007

Prescriptive vs. goal based standards

Use of evidence based arguments (assurance cases) is already required by some goal based standards

	Prescriptive standards	**Goal based standards**
Requirements of a standard	specify precisely what should be demonstrated	specify goals and allow different ways how it is achieved
How to demonstrate compliance	Provide evidence the requirement is satisfied	1. Define strategy how the goal is achieved 2. Justify the strategy is effective 3. Provide evidence the strategy is followed

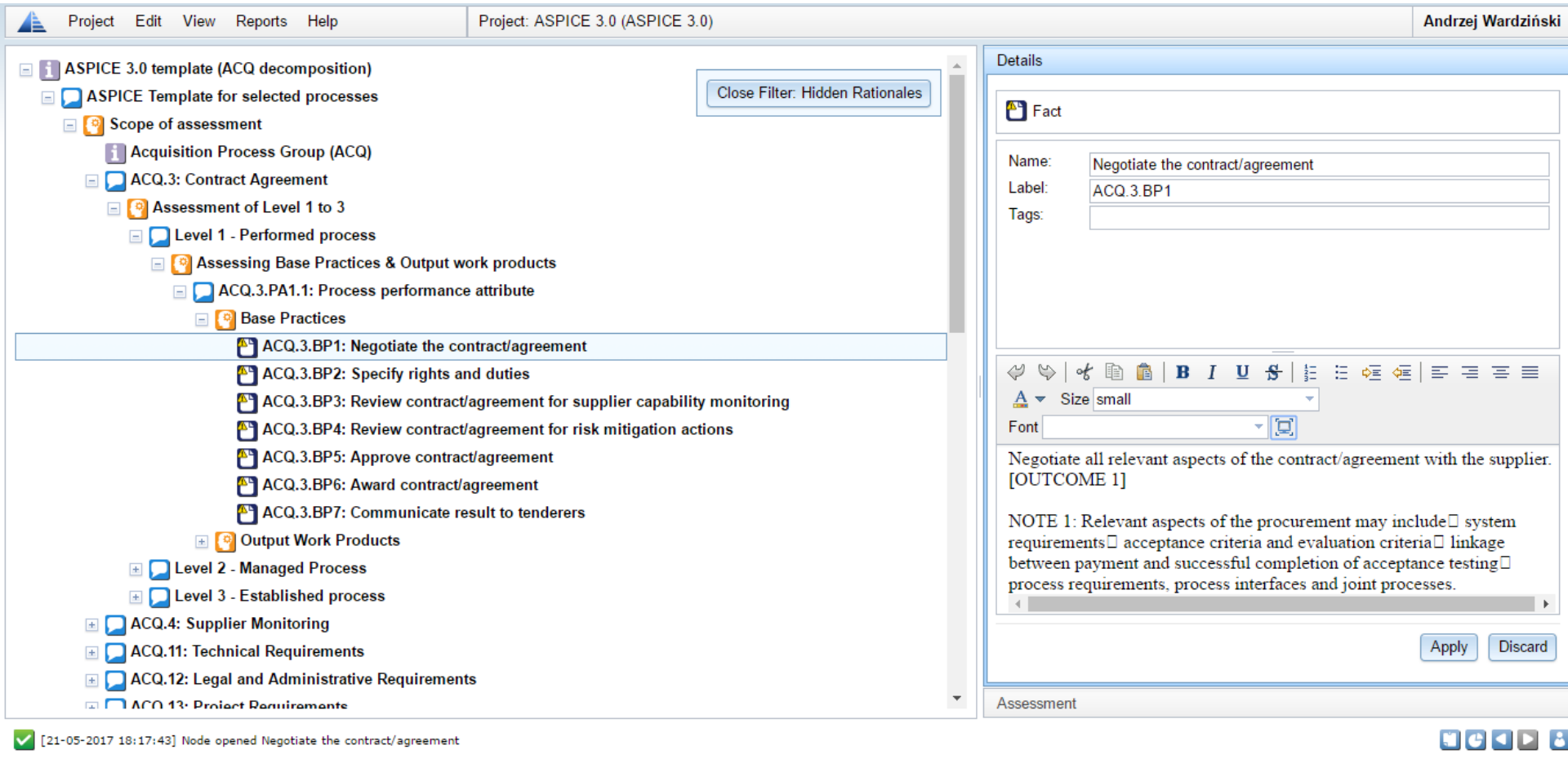
Conformance case

- Argument hierarchy can represent **structure** of a standard
  - directly or with mapping
- Leaves of the argument represent **requirements** of the standard
- Users can provide **evidence** to demonstrate compliance
- Argument can be extended with **additional information** like:
  - guidance for standard users
  - assessment procedures and criteria

GDAŃSK UNIVERSITY OF TECHNOLOGY

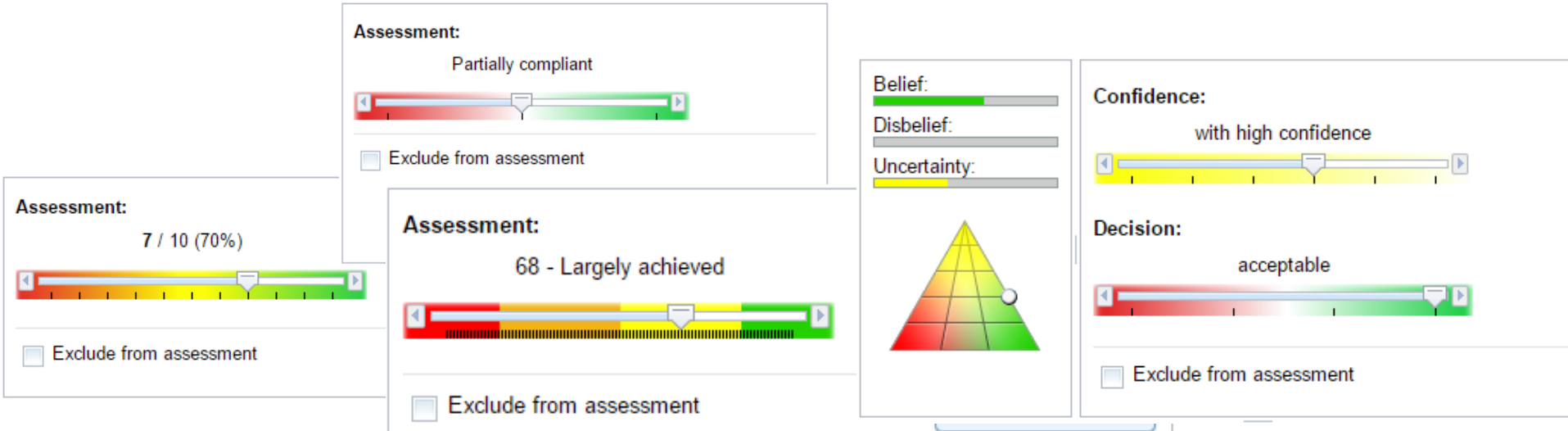
Argument structure example

ARGEVIDE



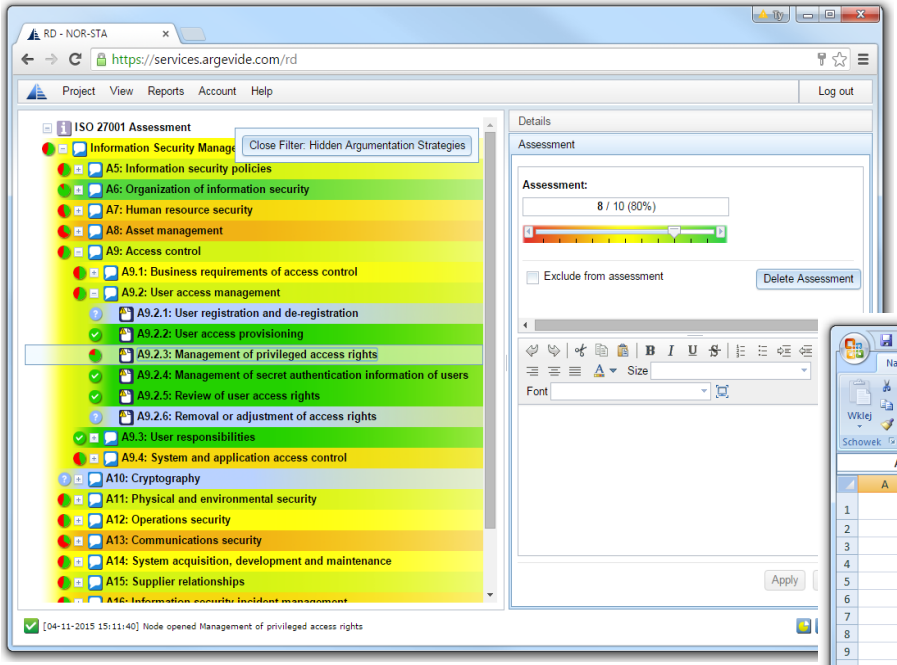
Assessment scales

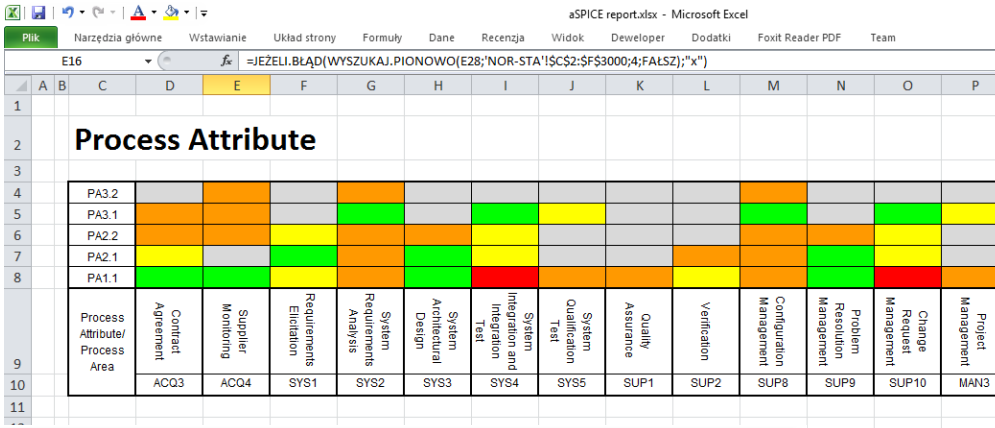
- Compliance for each requirement of the standard can be evaluated separately
- Different assessment methods can be used, for example:
  - Dempster-Shafer method permits to represent uncertainty (e.g. missing information)
  - SPICE is using 0..100 scale with four levels of compliance (N-P-L-F)
  - Rating scale is using number for evaluation
  - 3-value scale (noncompliant, partially compliant, compliant)

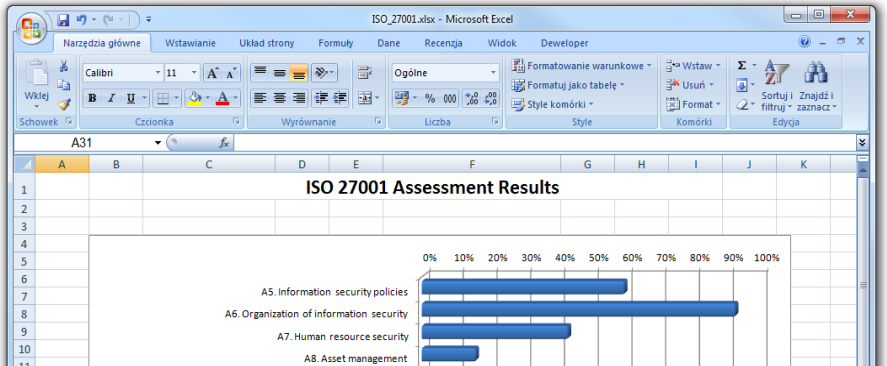


Assessment reporting

- Assessment results can be
  - represented with a color scale
  - reported to MS Excel, XML, PDF







The Compliance Process Supported with Arguments

1. Define structure of a standards (conformance case template)
2. Plan your compliance project (start with an empty compliance case)
3. Provide evidence and compliance argument
4. Make assessment (self assessment, certification assessment)
5. Report progress and level of the compliance
6. Maintain compliance

Applications

The approach has been applied by commercial users for standards:

- Hospital Accreditation Standards (NCQA, Poland)
- ISO 9001 Quality management systems
- ISO 14001 Environmental Management Systems
- OHSAS 18001 Occupational Health and Safety Management
- ISO 27001 Information Security Management
- IEC 62443 Security for industrial automation and control systems
- EN/IEC 61511 Functional safety – Safety instrumented systems for the process industry sector
- ISO 26262 Road vehicles – Functional safety
- ISO/IEC 17065 Conformity assessment — Requirements for bodies certifying products, processes and services

Experiences

- Argumentation structure is easier to comprehend that traditional documentation of standards
  - users better understand the standard requirements
- You can create an integrated compliance environment consisting of:
  - requirements of the standard
  - guidance, best practices, evidence samples
  - compliance evidence and descriptions
  - assessments and comments
- The approach helps to maintain consistency in conformance projects
- Online cooperation improves communication between organizations

Standards going electronic

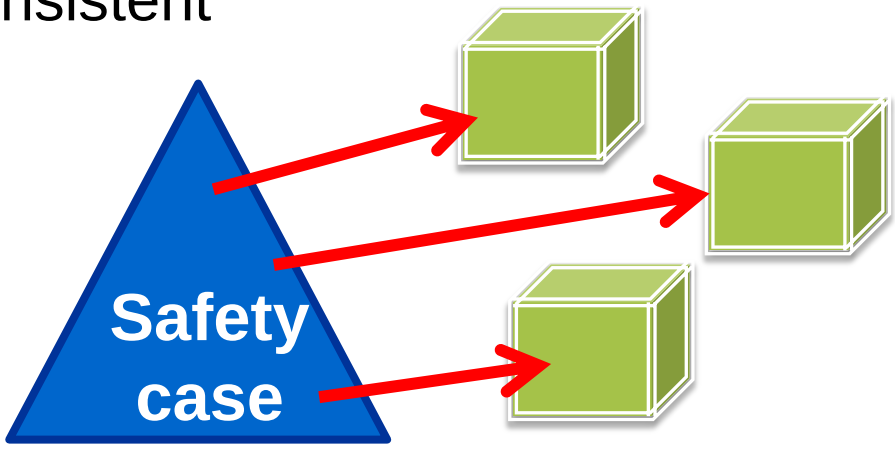
- Traditional document structure of standards is
  - optimal for technical publication (and will not disappear)
  - not optimal for using it and for managing
- Standards logical structure and dependencies become more and more complex
  - maturity levels, SILs, EALs, process areas, practices, etc.
- Argumentation structure is a step in the right direction to represent logical structure of a standard
  - More advanced data structures may also be useful
- It helps to manage complex standards
- XML representation makes possible exchange of compliance information between systems and organizations

Part 2

Managing Safety Case Relations to System Models

Safety argument in the context

- Argument context includes…
  - System structure, elements and their properties
  - Behaviour (events, processes)
  - Risk model (hazards, causes, safety requirements)
  - Environment structure and properties
  - System life cycle activities and artefacts
- A valid safety argument needs the context to be correct and consistent

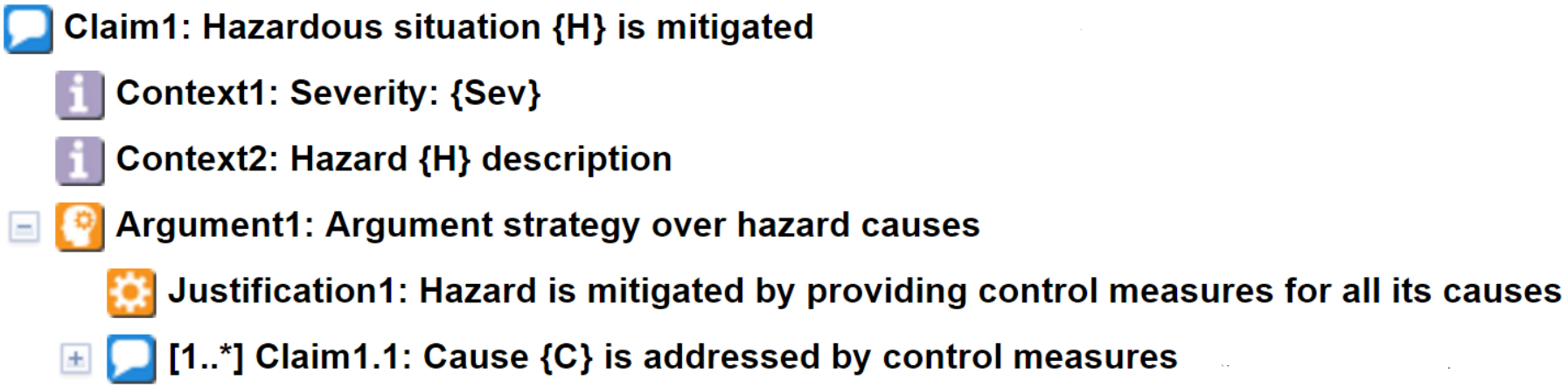


How can the context be managed?

- Informal references
  - Use context names in argument elements
    - Example claim: Speed sensor S17 failure rate is below 10-6
- Distinct context elements
  - GSN Standard specifies a Context element
    - *A context, presents a contextual artefact. This can be a reference to contextual information, or a statement.*
- Model generated argument
  - Automatic safety argument generators ensure argument consistency with system models used.
- Direct references to system model elements

Example

For the presented fragment of an argument:



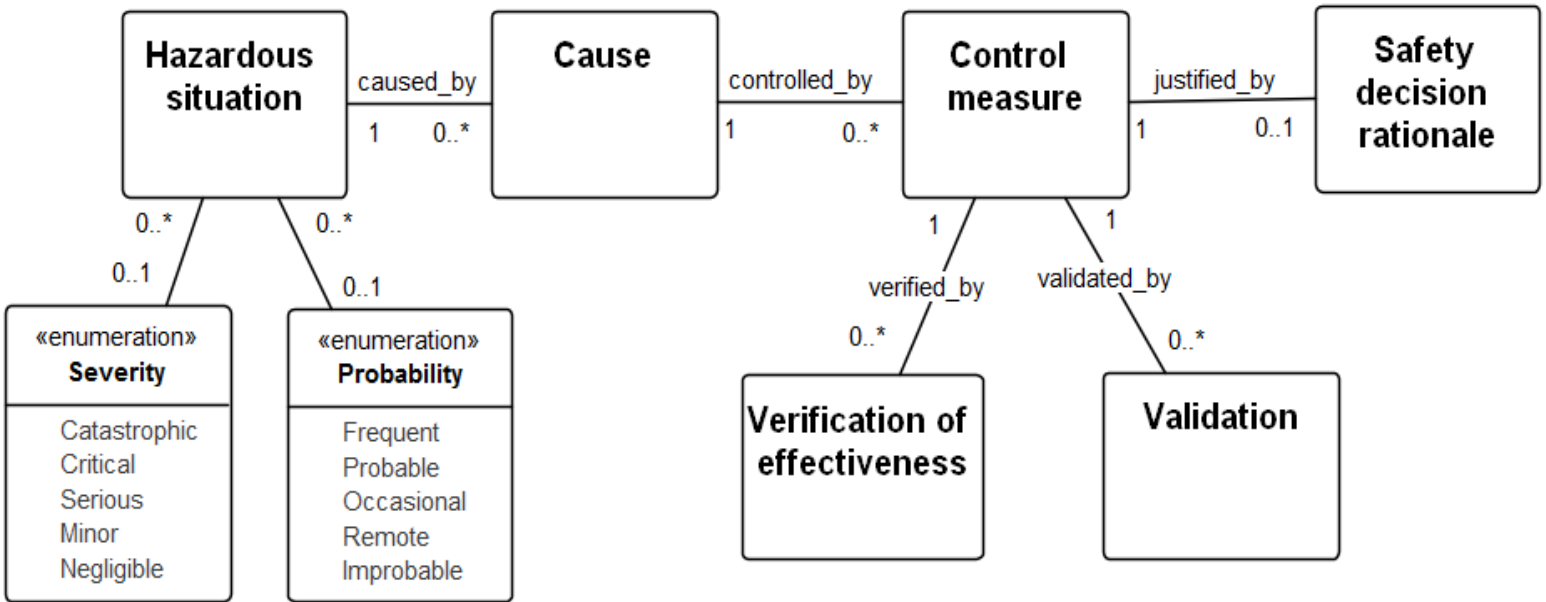
The goal is:

- to establish references to valid elements of the risk model
- to ensure referenced elements relations hold
  (e.g. we refer to causes of the hazard specified in the parent claim)
- to maintain correctness of the references and to be informed
  when it is challenged (e.g. elements of the risk model are modified)

The system metamodel

- System metamodel defines an abstract schema for system models
  - It defines entities, attributes and relations
- UML class diagram can be used to present a metamodel

Example:



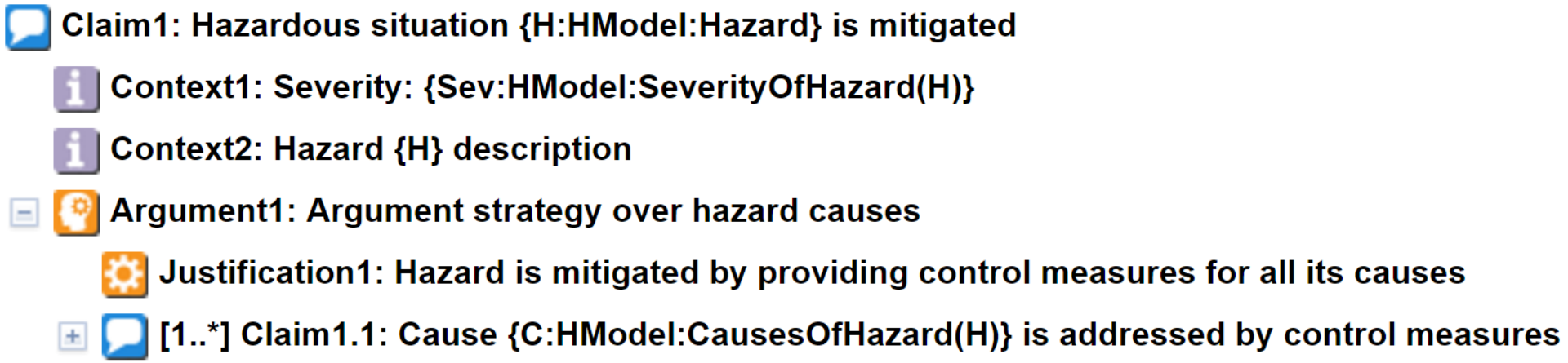
Reference model

System metamodel enables establishing references to:

- elements of a given type
- elements in a specified relation with context elements

We extend the safety argument parameters with:

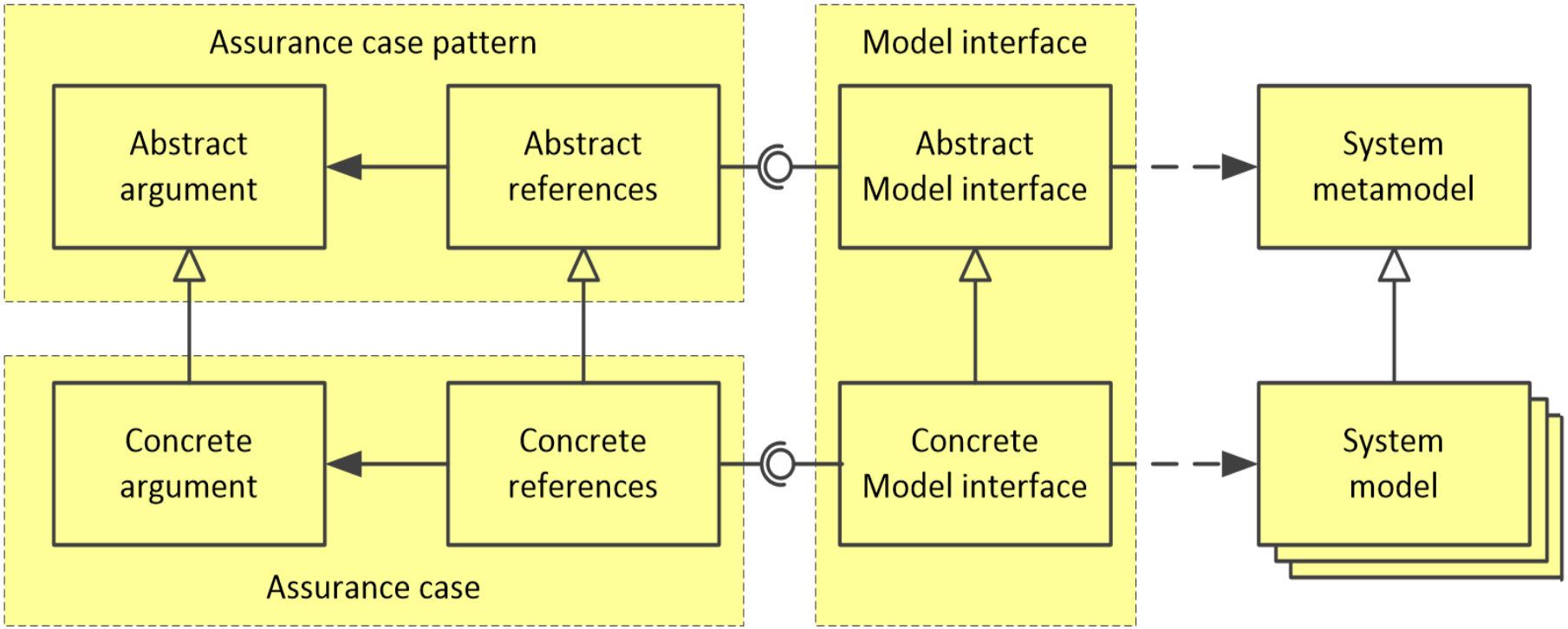
- a model type
- a selector which specifies an element type or relation



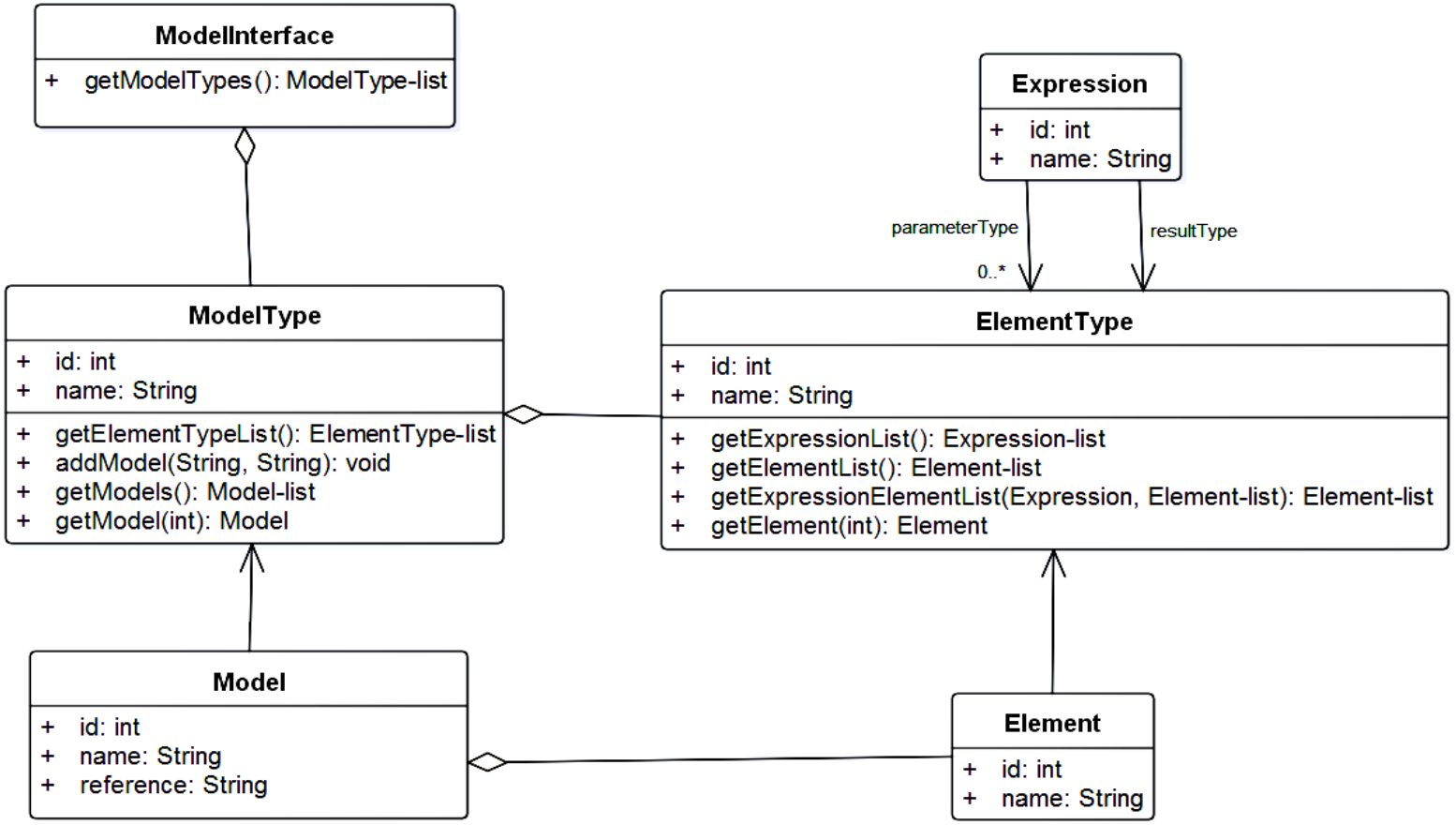
Interfacing safety argument to system models

An intermediary named *Model interface* can:

- provide information about system metamodel classes and relations
- give lists of elements which satisfy the reference requirement
- verify if a given element or relation is up to date



Uniform model interface



The minimal model of a model interface which permits to establish and maintain references to system models.

Integration process

Pre-development phase steps

1. System metamodel specification
2. Model interface development
3. Argument pattern development

Development phase steps

4. System modeling
5. Assurance case development (instantiation)
6. System models and assurance case maintenance (iteration of steps 4 and 5)

Relations data

The relations data are maintained in:

- abstract reference table

Pattern element id	Reference name	Model type	Element selector
Claim1 Context2	H	HModel (the risk model)	Hazard
Context1	Sev	HModel (the risk model)	SeverityOfHazard(H)
Claim1.1	C	HModel (the risk model)	CausesOfHazard(H)

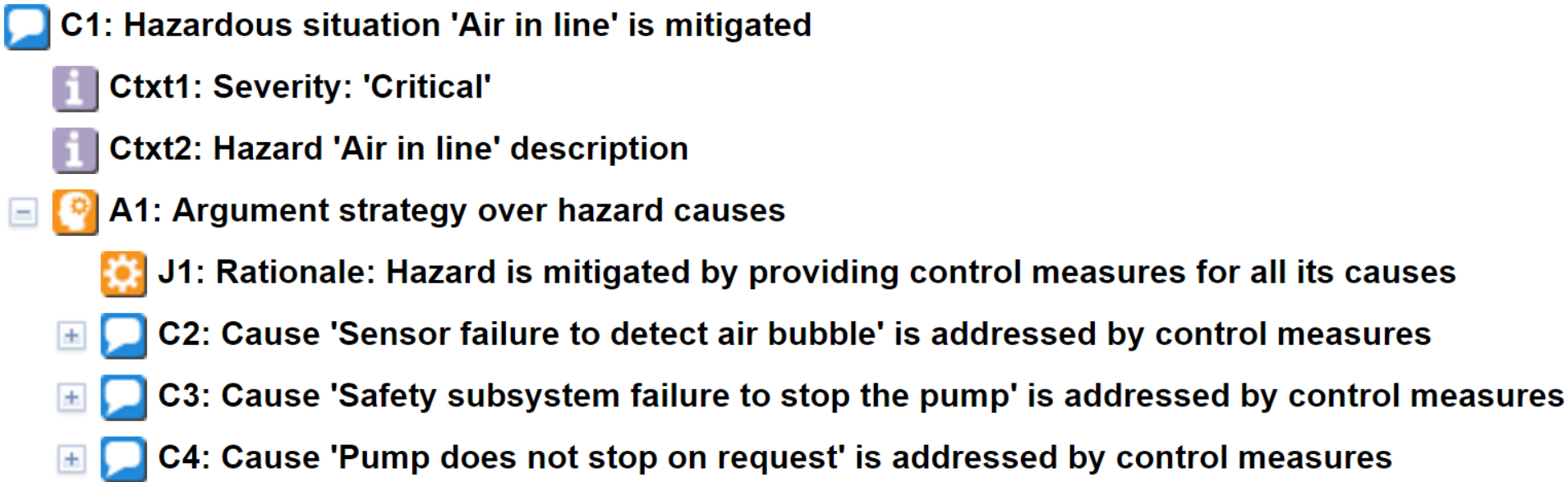
- concrete(instantiation) reference table

Argument element id	Reference name	Model name	Model element id	Element name
C1 Ctxt2	H	PCAHazardTable.xml	H1	Air in line
Ctxt1	Sev	PCAHazardTable.xml	S1	Critical
C2	C	PCAHazardTable.xml	C1	Sensor failure to detect air bubble
C3	C	PCAHazardTable.xml	C2	Safety subsystem failure to stop the pump
C4	C	PCAHazardTable.xml	C4	Pump does not stop on request

Prototype solution

Prototype solution

- Manual specification of argument pattern parameters
- Prototype instantiation tool reads / writes SACM 1.1 arguments
- The model interface implemented for XML risk model and OSATE AADL models (partially)



Experiences

- Conclusions
  - Uniform model interface is sufficient for establishing and maintenance of assurance case relations to system model
  - Use of GUIDs in system models is effential for references maintenance
- Further work
  - Case studies for other types of models
  - Verification function to detect model changes
  - Maintenance of the instantiation reference tables
  - Integration with SACM 2.0 (Terminology package)

Summary

- Uniform model interface will facilitate establishing and maintaining assurance case relation to system models
  - We expect this to be easier for safety engineers
- The established relations are:
  - correct as they rely on directly on existing models
  - up to date (this can be verified at any moment of time)
- System model changes can be propagated to the safety argument

GDAŃSK UNIVERSITY OF TECHNOLOGY

ARGEVIDE

Thank you for your attention