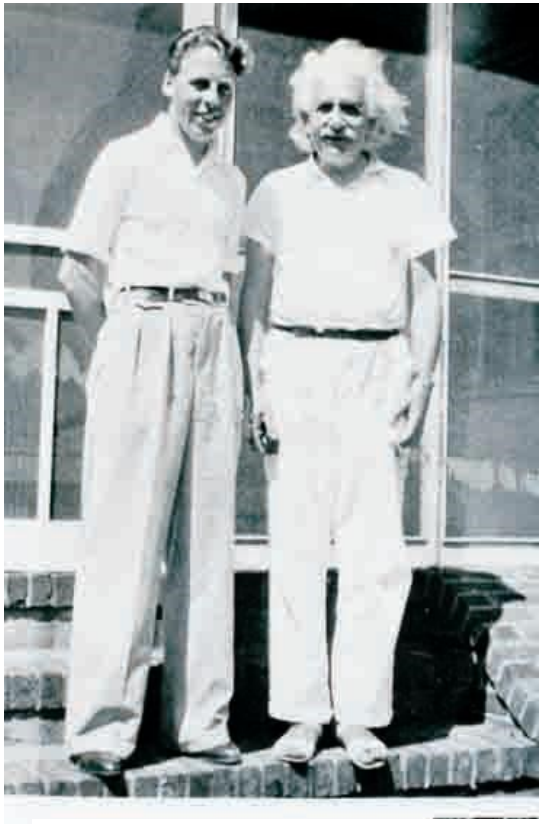
Using traceability information for building safety cases

Vikash Katta

Institute for Energy Technology, Halden, Norway

Institute for Energy Technology (IFE)

- Independent foundation established in 1948
- Norway's second largest research institute
- Hosting the OECD Halden Reactor Project
- International nuclear industry and Nordic transportation, process, energy and petroleum industry



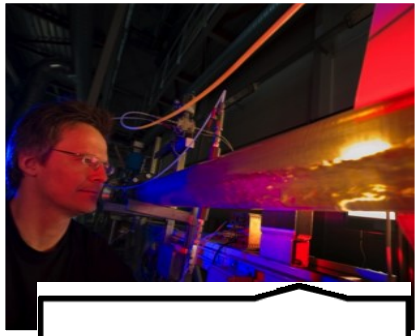
G. Randers, IFEs founder

OECD Halden Reactor Project

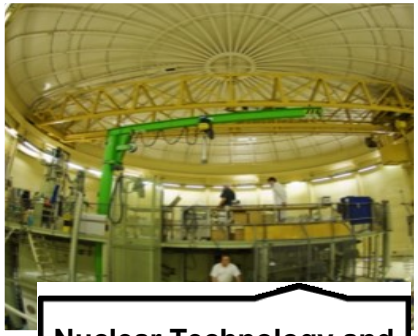
- Established in 1958
- Joint undertaking - OECD NEA
- Three year program periods
  - Current period: 2015-2017
- 20 member-countries and more than 100 organisations
- Experimental programs
  - HBWR, HAMMLAB , VR-lab, Integrated operations lab

IFEs main activity areas….

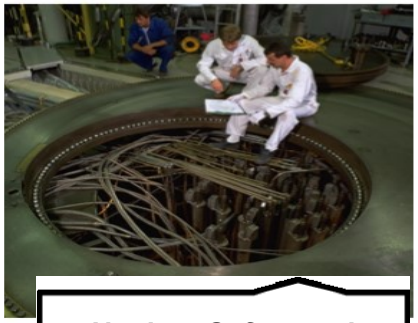
Energy and Environmental Technology



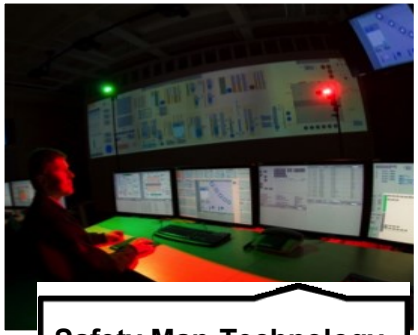
Petroleum Technology



Nuclear Technology and Physics



Nuclear Safety and Reliability



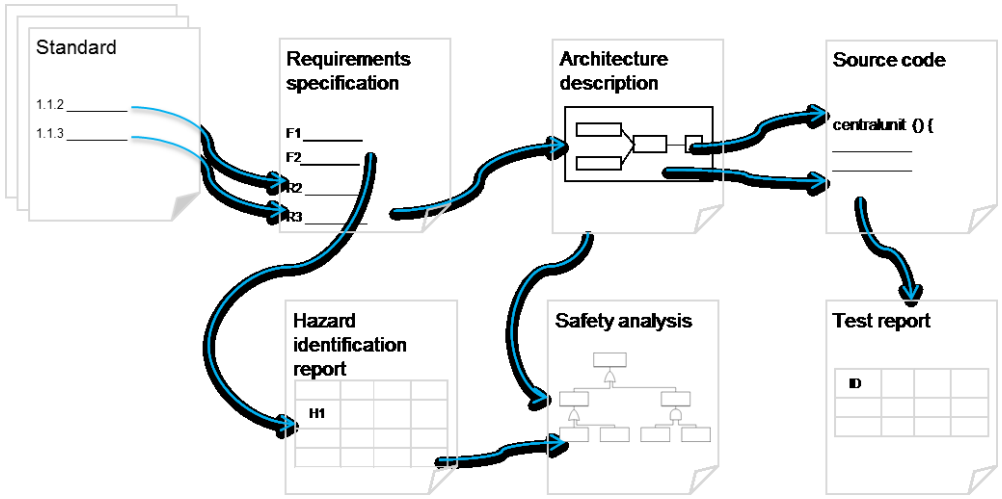
Safety Man-Technology-Organisation (MTO)

Contents

- Background
- SaTrAp traceability approach
- Case: ATM Remote Tower
  - traceability
  - safety argumentation
- Observations

Traceability and needs

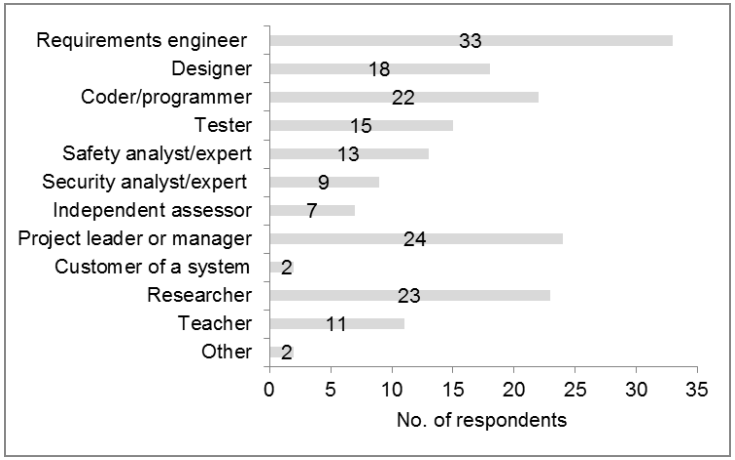
- A mechanism to relate artefacts/elements.



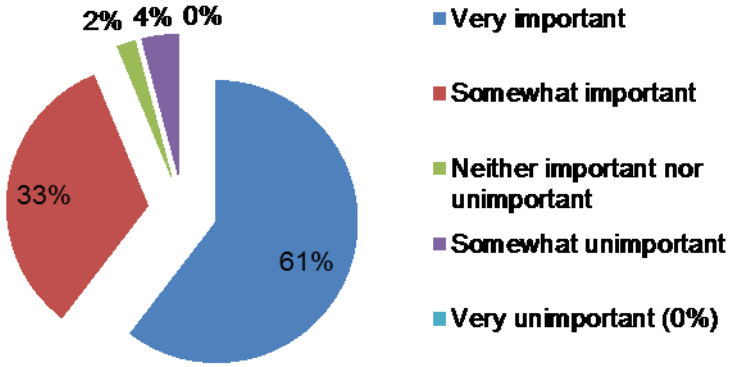
- Different stakeholders have different uses of traces
  - Requirements engineer: estimate change impact
  - Designer : all requirements are considered in the design, synchronising models (MDD)
  - Tester: coverage of tests
  - Safety analyst: manage hazard log, validate safety requirements

Problem statement

- Survey: traceability during development of systems with safety and security implications - importance, tools, and challenges

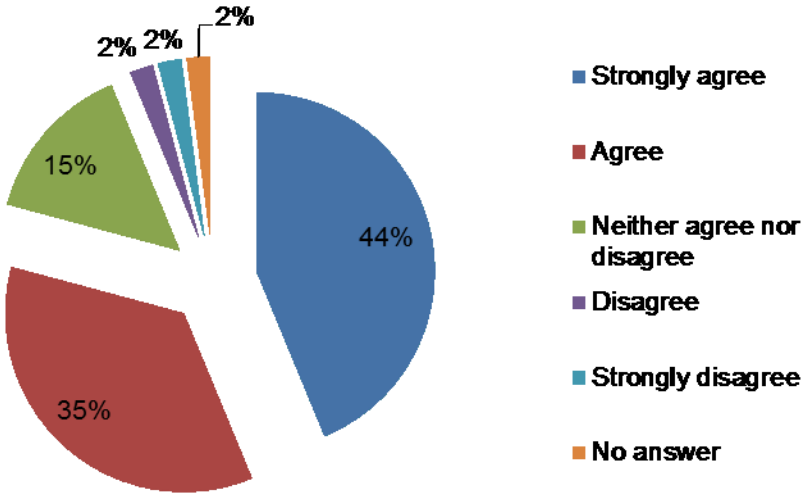


- Importance of implementing traceability in projects



Problem statement

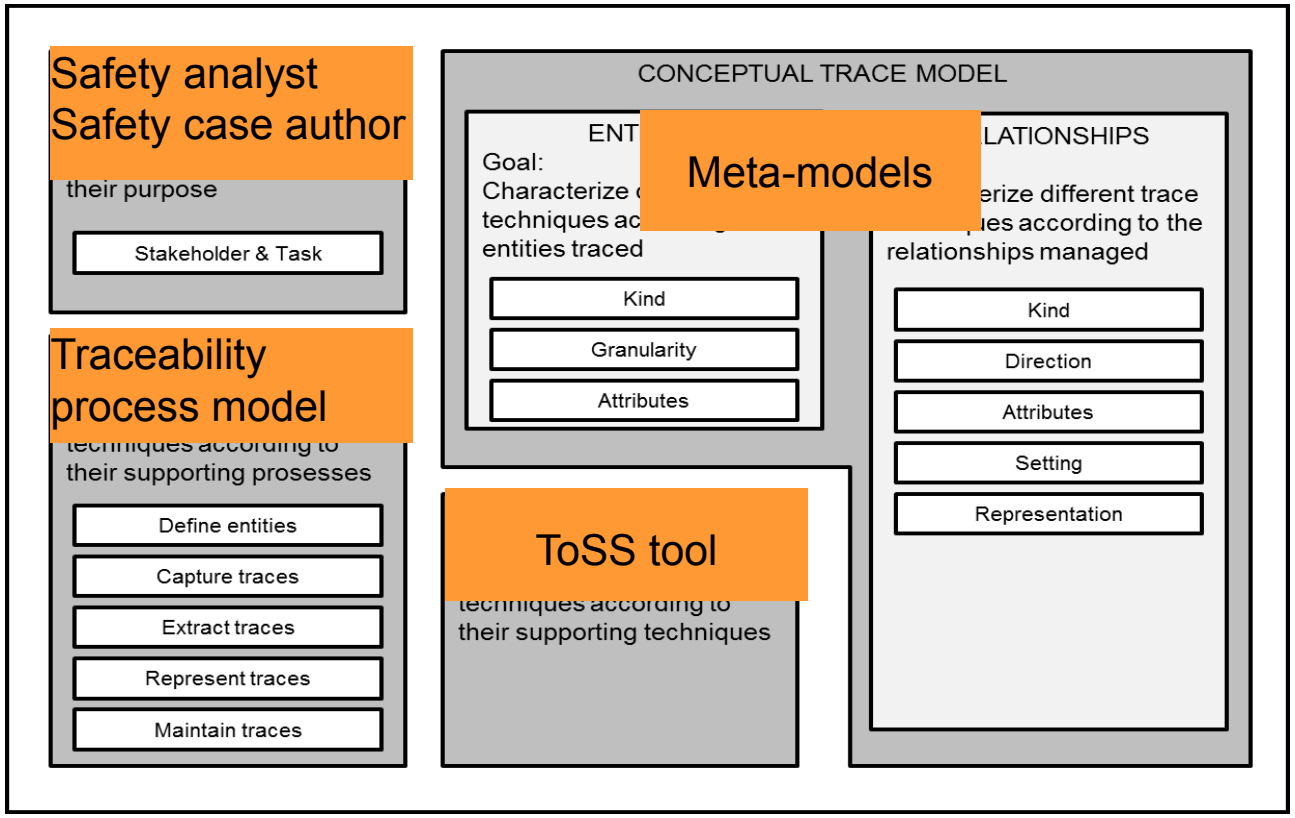
- Need for better guidance on traceability



- Challenges for implementing traceability
  - lack of understanding of the use traceability
  - lack of guidance on implementing
  - not easy to use tools
    - effort to tailor to project specific needs

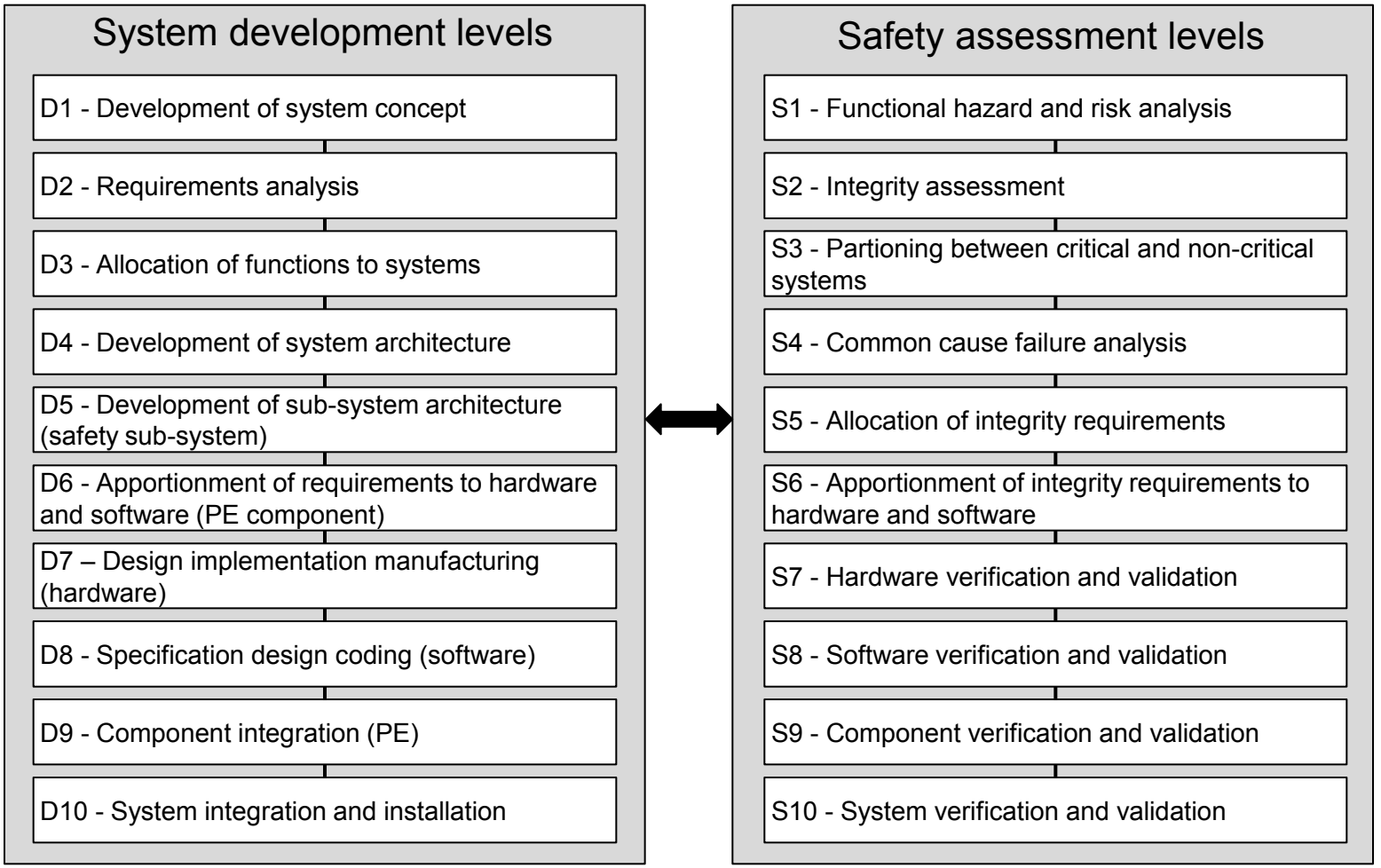
Safety Traceability Approach (SaTrAp)

- Consists of four main concepts, as defined in [A. V. Knethen, 2002]



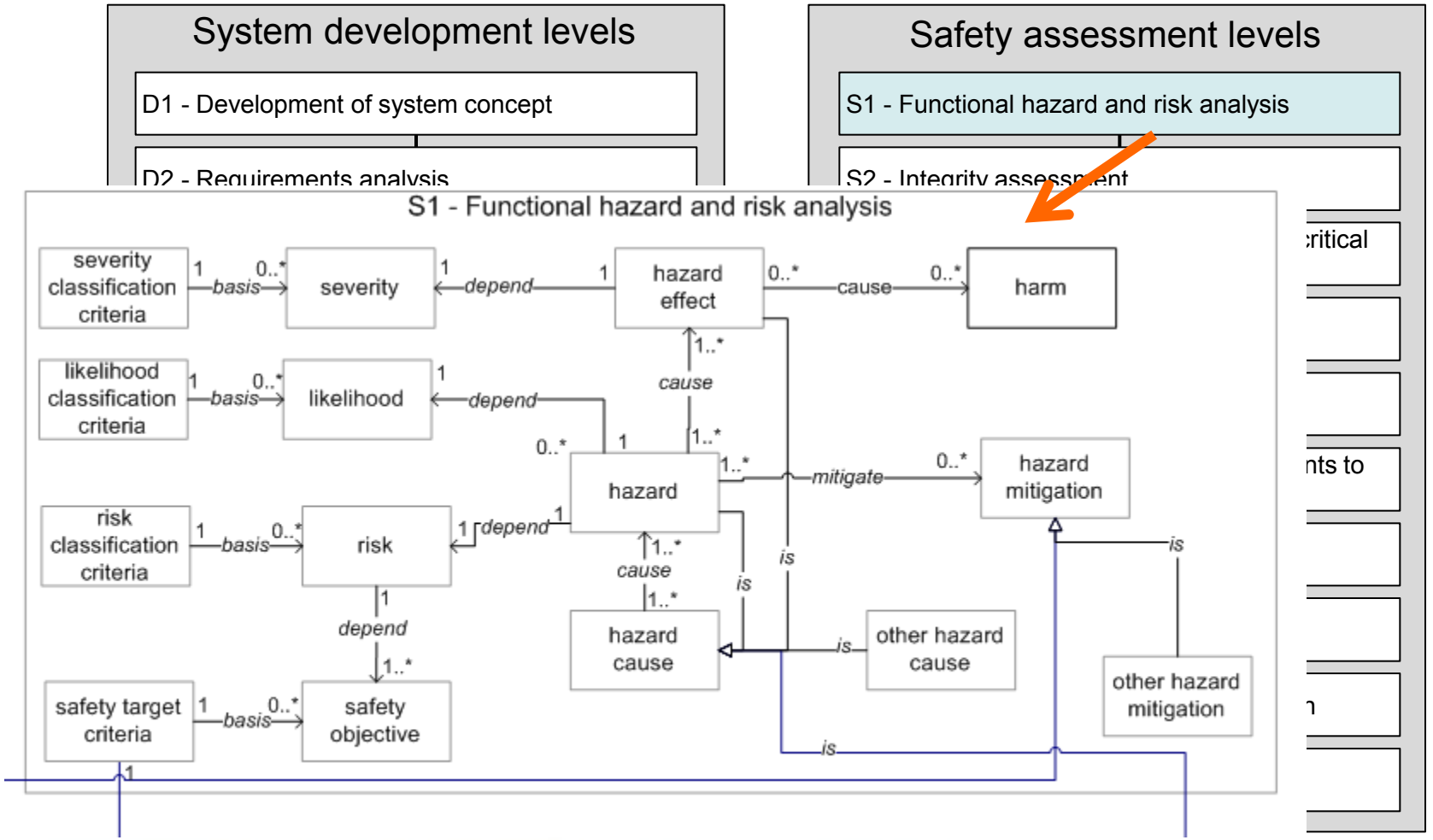
Traceability process model

- Blueprint describing a process to capture traces (what, when, how)



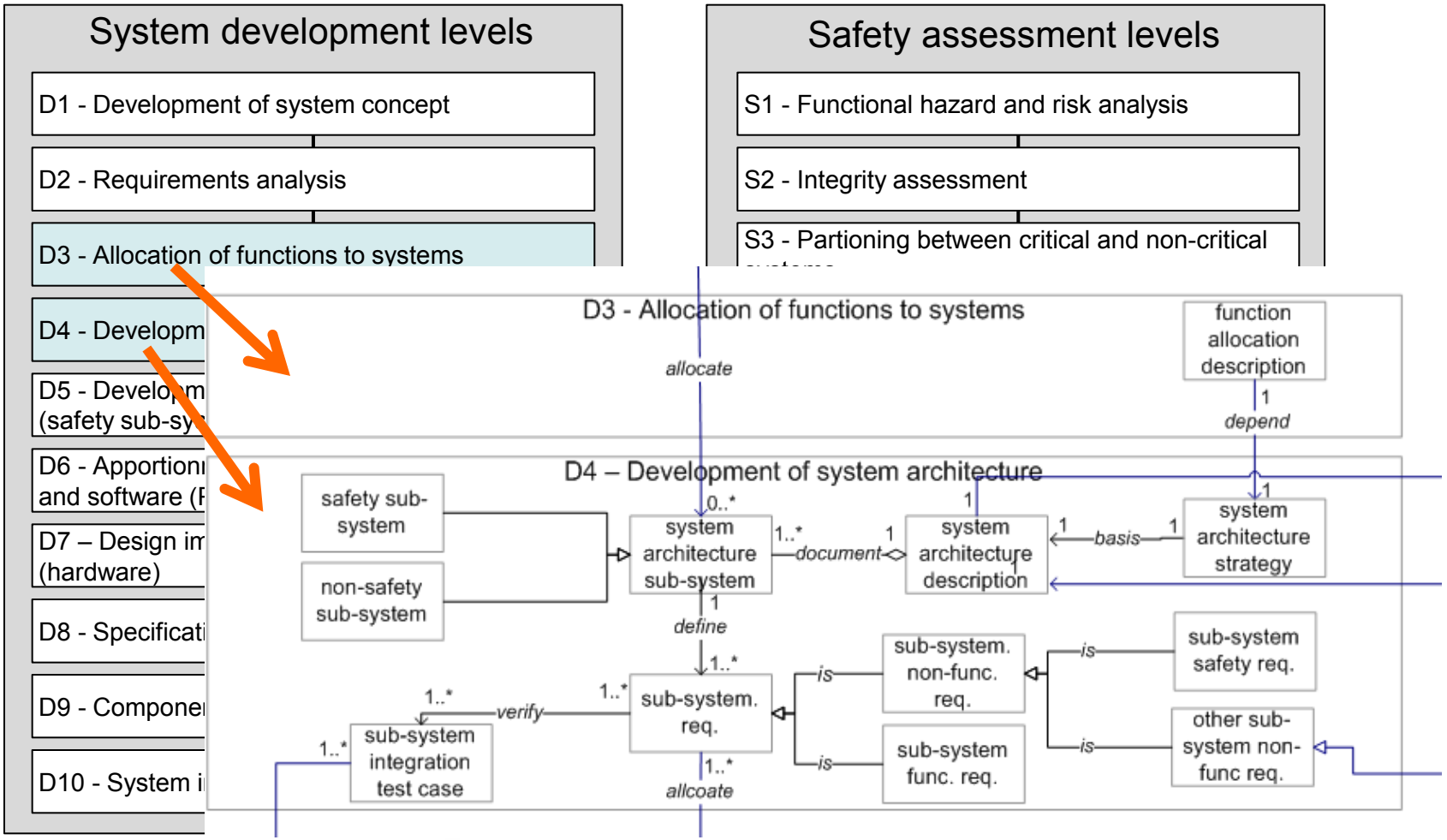
Traceability process model

- Blueprint describing a process to capture traces (what, when, how)

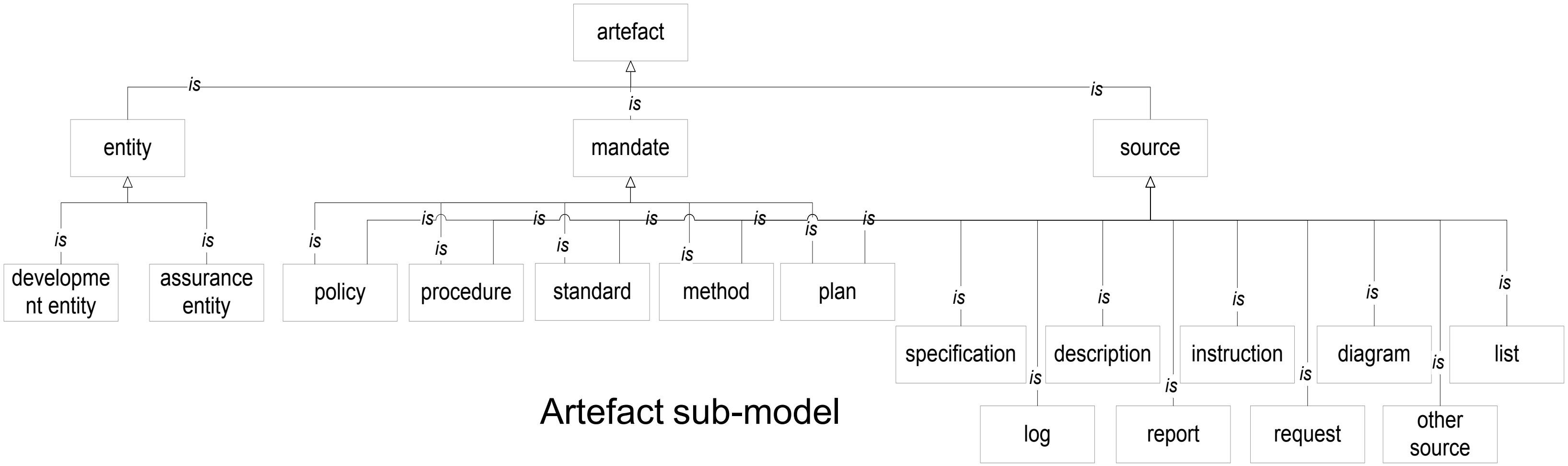


Traceability process model

- Blueprint describing a process to capture traces (what, when, how)

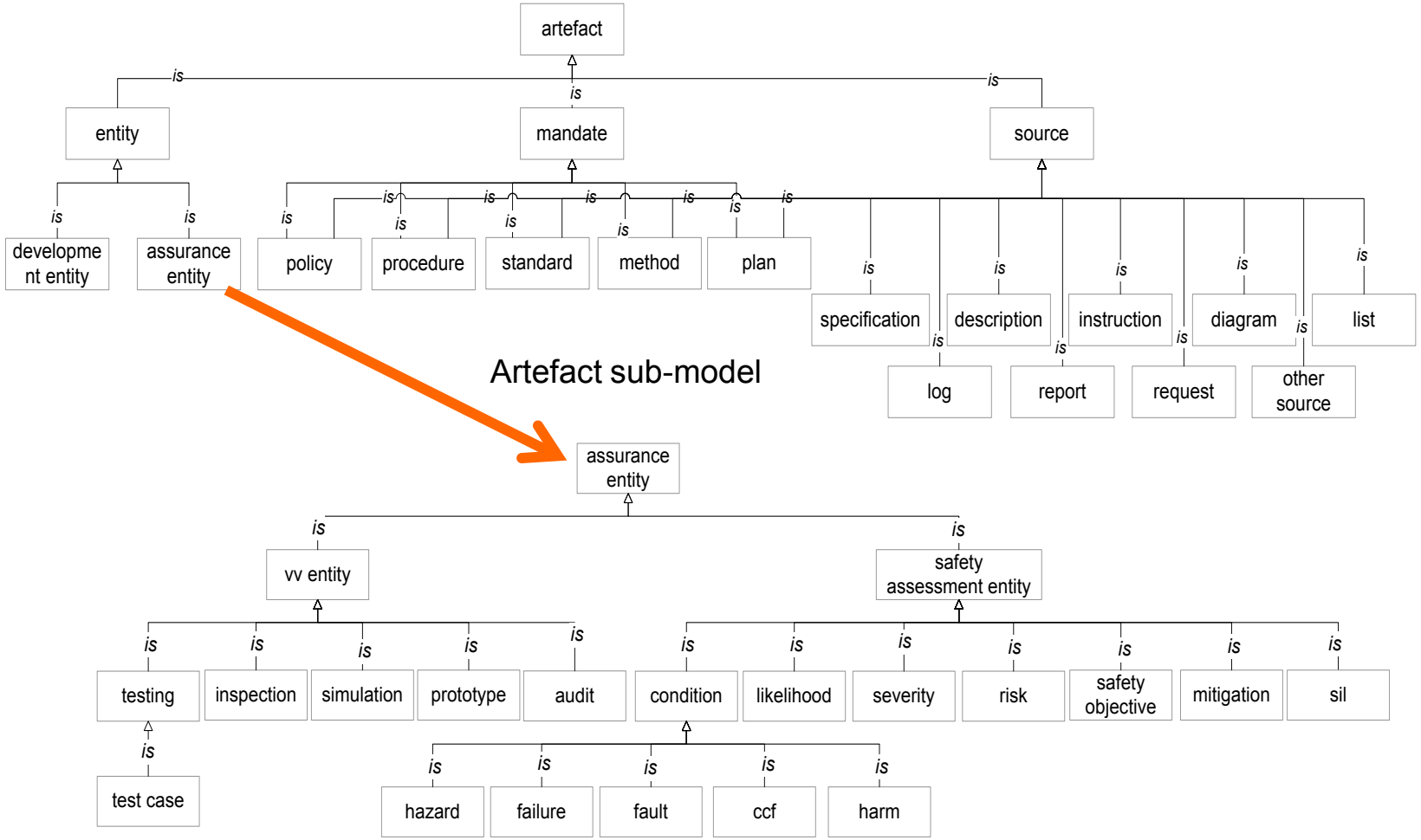


Meta-models



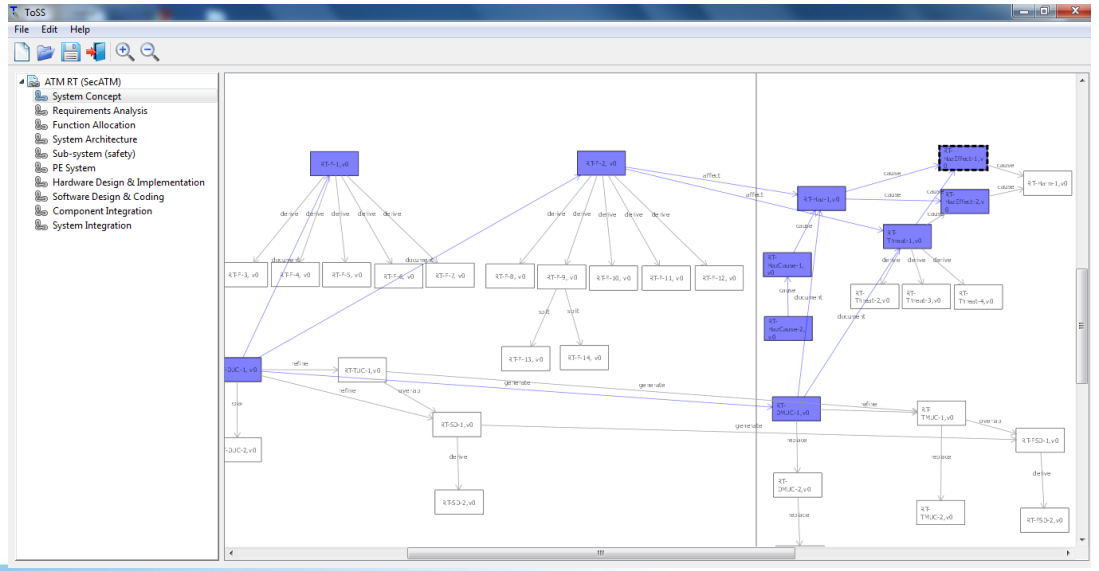
Artefact sub-model

Meta-models (cont.)



ToSS tool (prototype)

- Implements the process model and meta-models.
  - Qt, Prolog, and C++
- Applied on:
  - S18 Aircraft desktop example (ARP4761, AIR6110)
  - Remote Tower desktop example (ATM BN project)
  - Multi-sensor tracking system (part of ATM system)



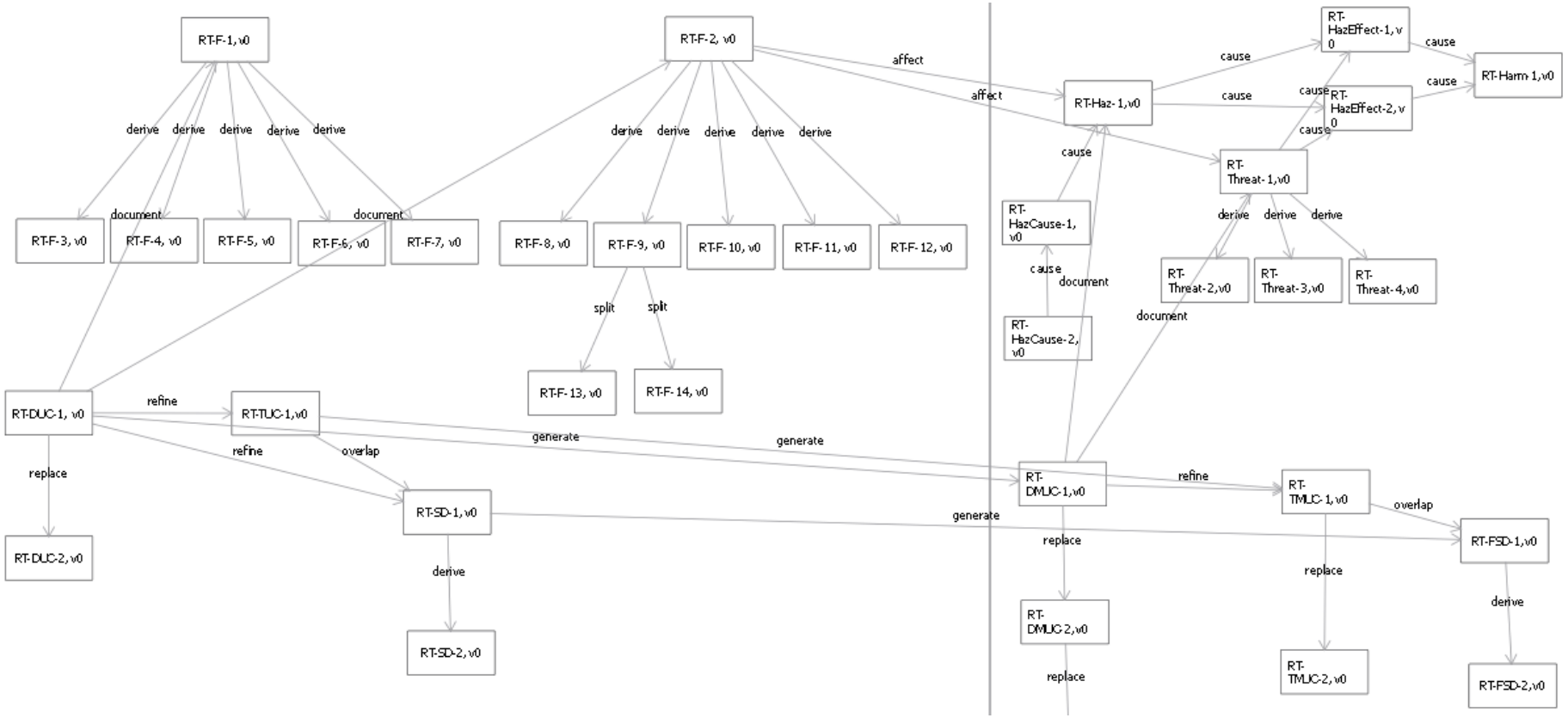
Application - ATM Remote Tower

- Participate in ATM industry network (ATM BN) project
  - 10 companies within the ATM domain
  - desktop example –Remote Tower (RT)
  - investigate possible improvements of their safety & security processes
  - shadow case using CHASSIS method for safety and security assessment
    - produced different diagrams and descriptions
    - ToSS tool was used to capture traces

RT example

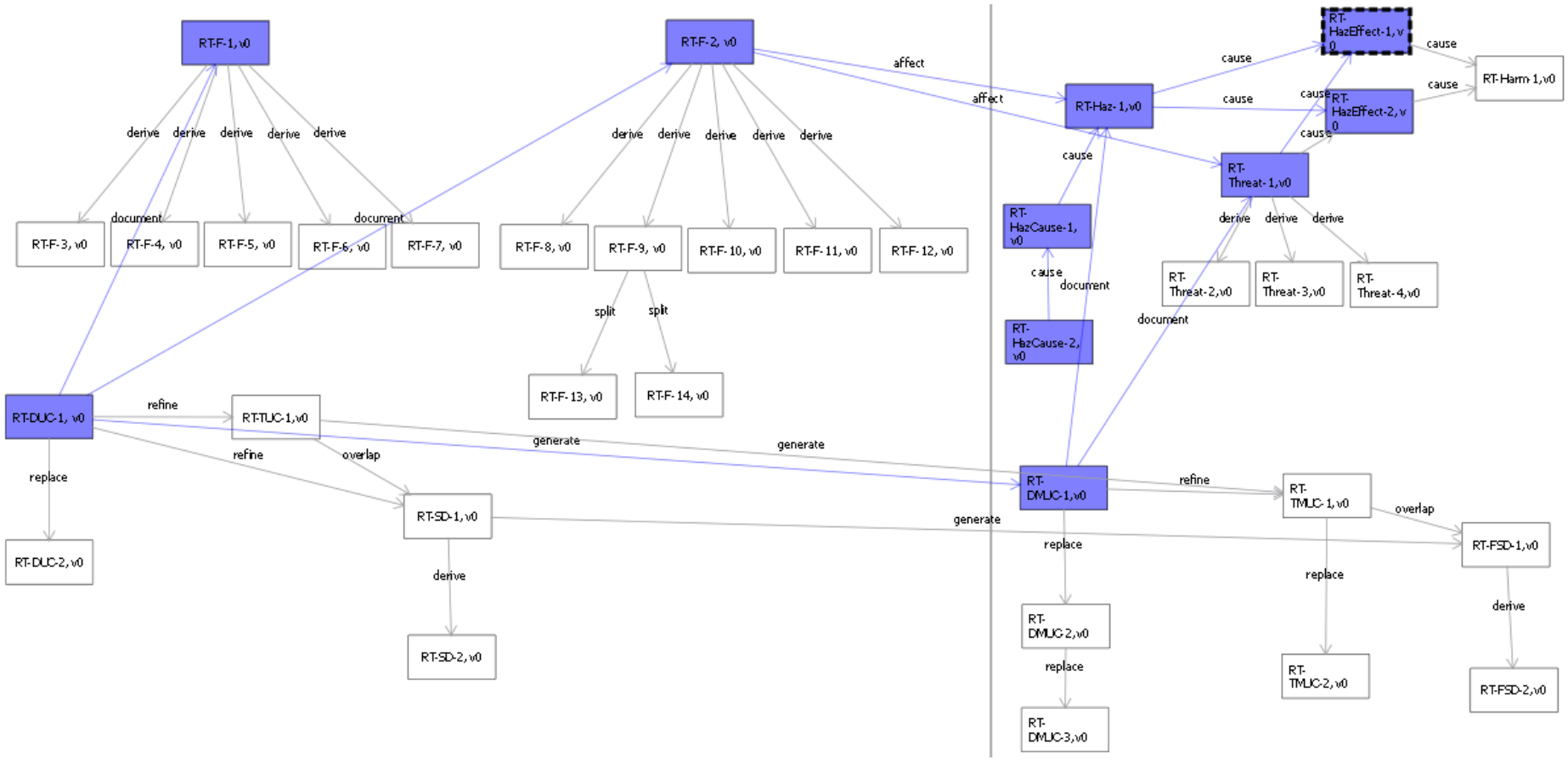
ID	Type	Description (simplified)
RT-F-1	Function	Monitoring
RT-F-2	Function	Providing clearance
RT-F-10	Function	Providing take-off clearance
RT-Haz-1	Hazard	Flight crew has wrong clearance
RT-Threat-1	Threat	Fabrication of false clearance
RT-HazEffect-1	Hazard effect	Delayed take-off clearance for flight crew-1
RT-HazCause-1	Hazard	Communication channel delays the ATCO clearance
RT-HazCause-2	Hazard	NW1 has hang up
RT-SFM-1	SW Failure Mode	Router fails to send take-off clearance
RT-SFM-2	SW Failure Mode	Routing CPU fails to send packets
RT-NF-1	Non-func. Req.	Broadcast clearance to all aircrafts to recognise wrong clearance
RT-NF-2	Non-func. Req.	Make clearance available only for the targeted aircraft

Traceability graph – RT example

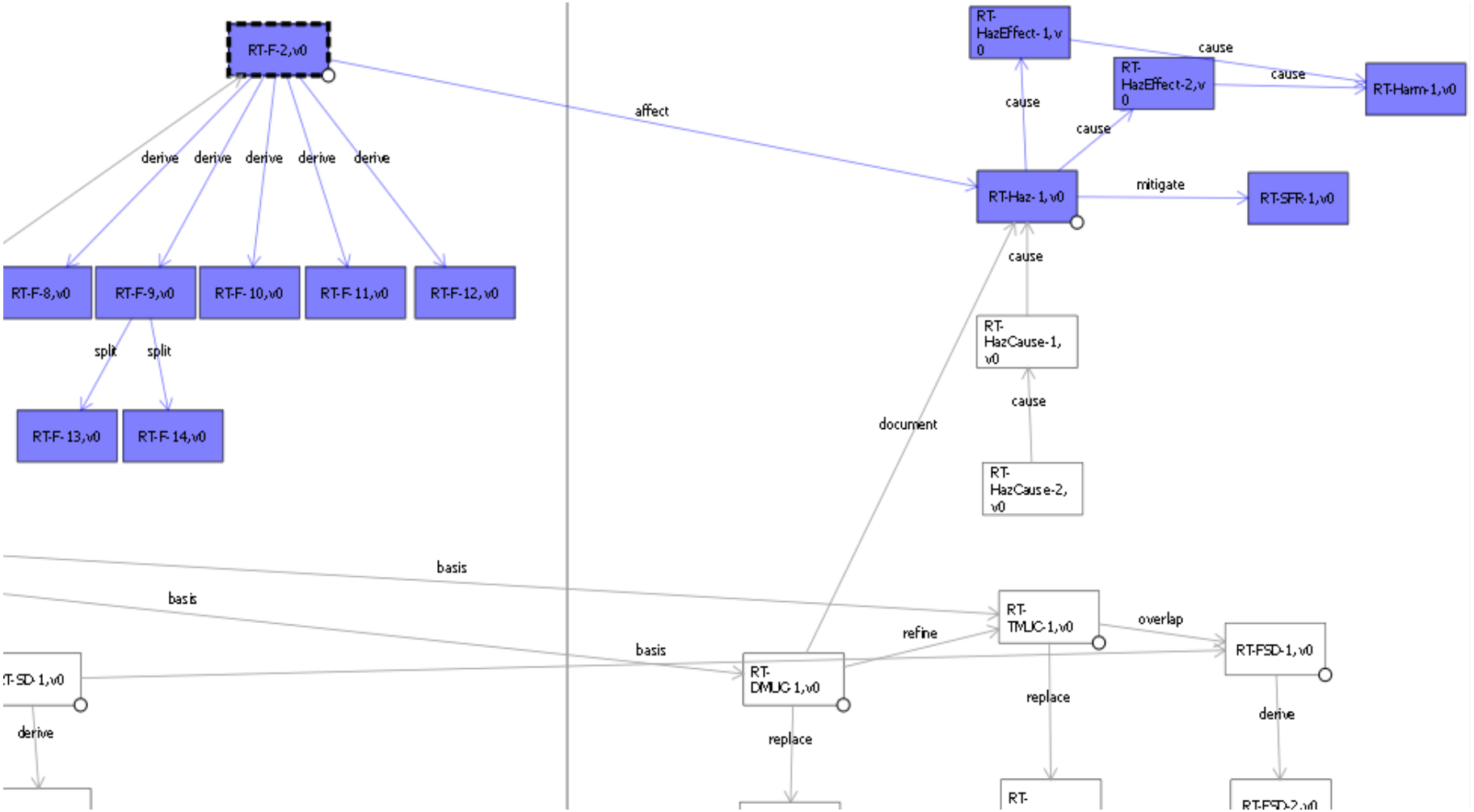


Snapshot from ToSS tool

Traceability (within-level) analysis



Traceability (within-level) analysis



Safety case and argumentation

- Safety case presents
  - structured arguments by relating evidences - generally produced during system development and safety analysis activities
  - in order to argue that a set of claims on the safety of a system have been met.
- For e.g., to provide evidence to the claim that all the hazards identified to a system have been prevented or mitigated, we need to document,
  - the safety requirements have been identified,
  - specified through a systematic safety analysis,
  - and further implemented in the system to deal with the hazards
- Collecting and structuring evidences is mostly manual and resource intensive

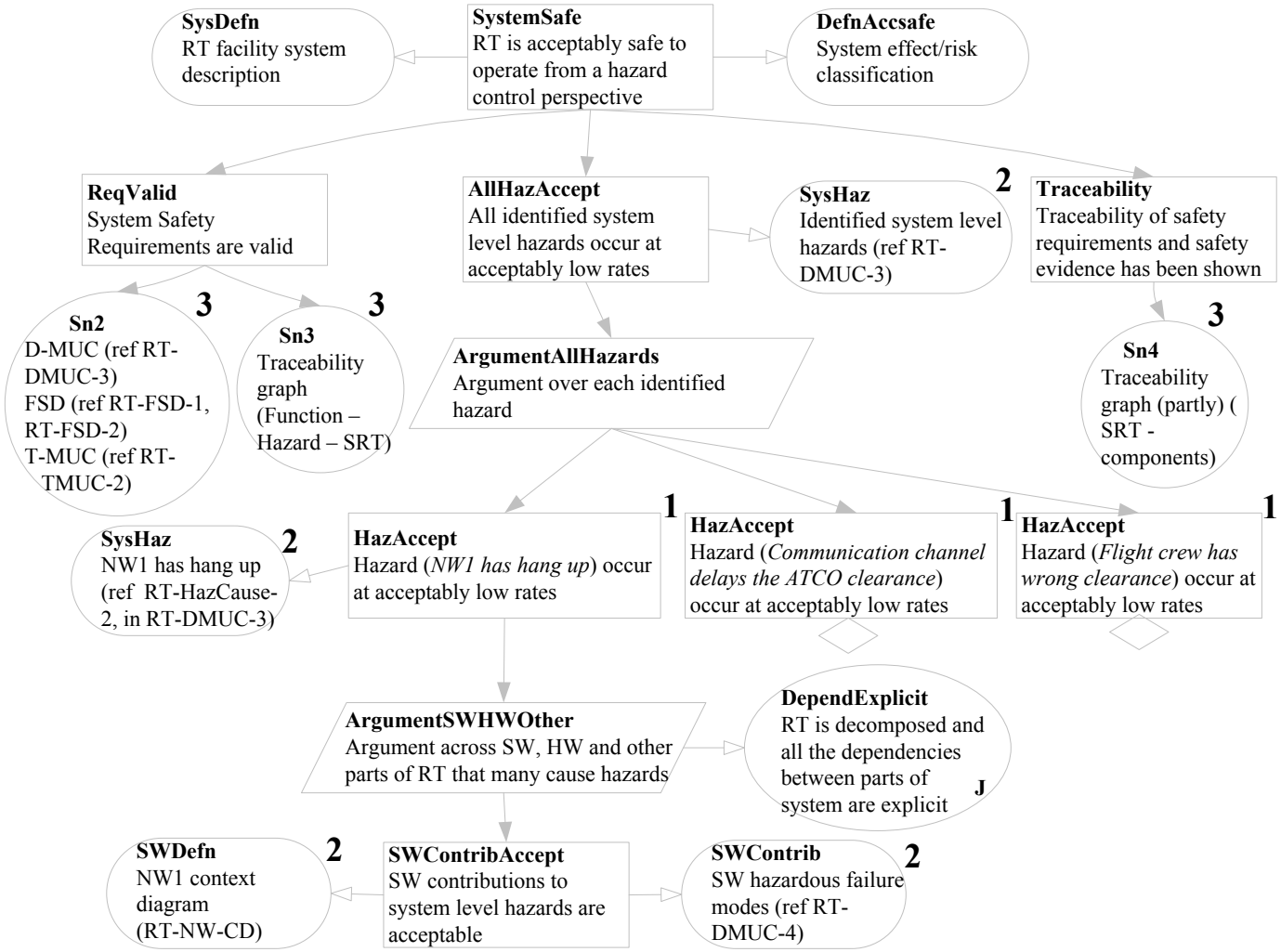
Traceability support for argumentation

- Traceability plays a vital role to identify valid evidence and also to assess whether all the evidence are considered
- For e.g., to demonstrate that the safety requirements reflect the results of the safety analysis, traceability facilitates this by
  - providing evidence in the form of traces between the results - e.g. hazards and failures- from safety analysis and the identified safety requirements
  - safety requirements have been allocated and thereafter implemented by the components of the system

Assurance case (safety)

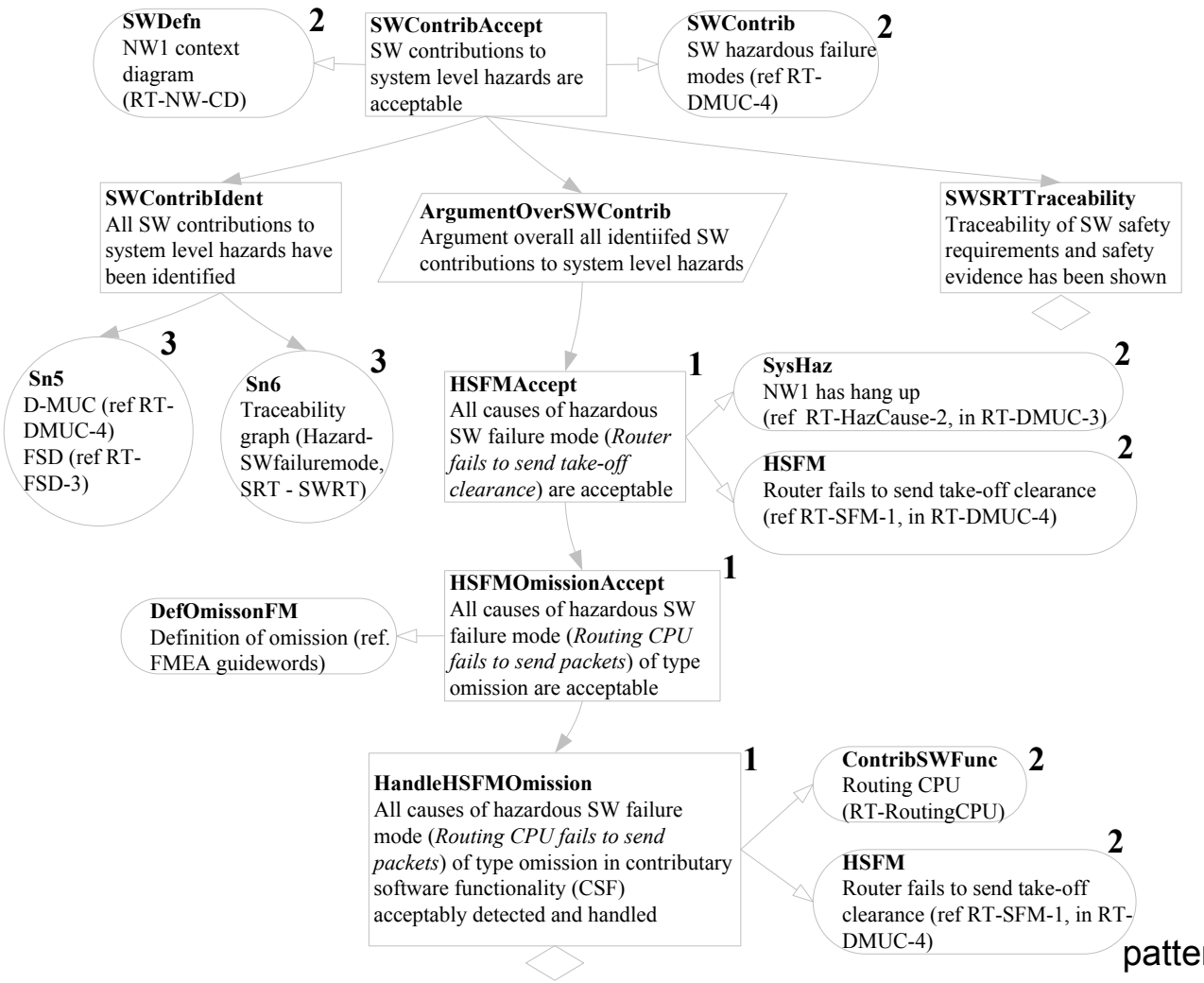
- Using SaTrAp (traceability information) to generate parts of safety case
  - Identify claims
  - Elaborate strategy – decompose claims
  - Identify context
  - Identify evidence
  - Managing safety case

Assurance case (safety)



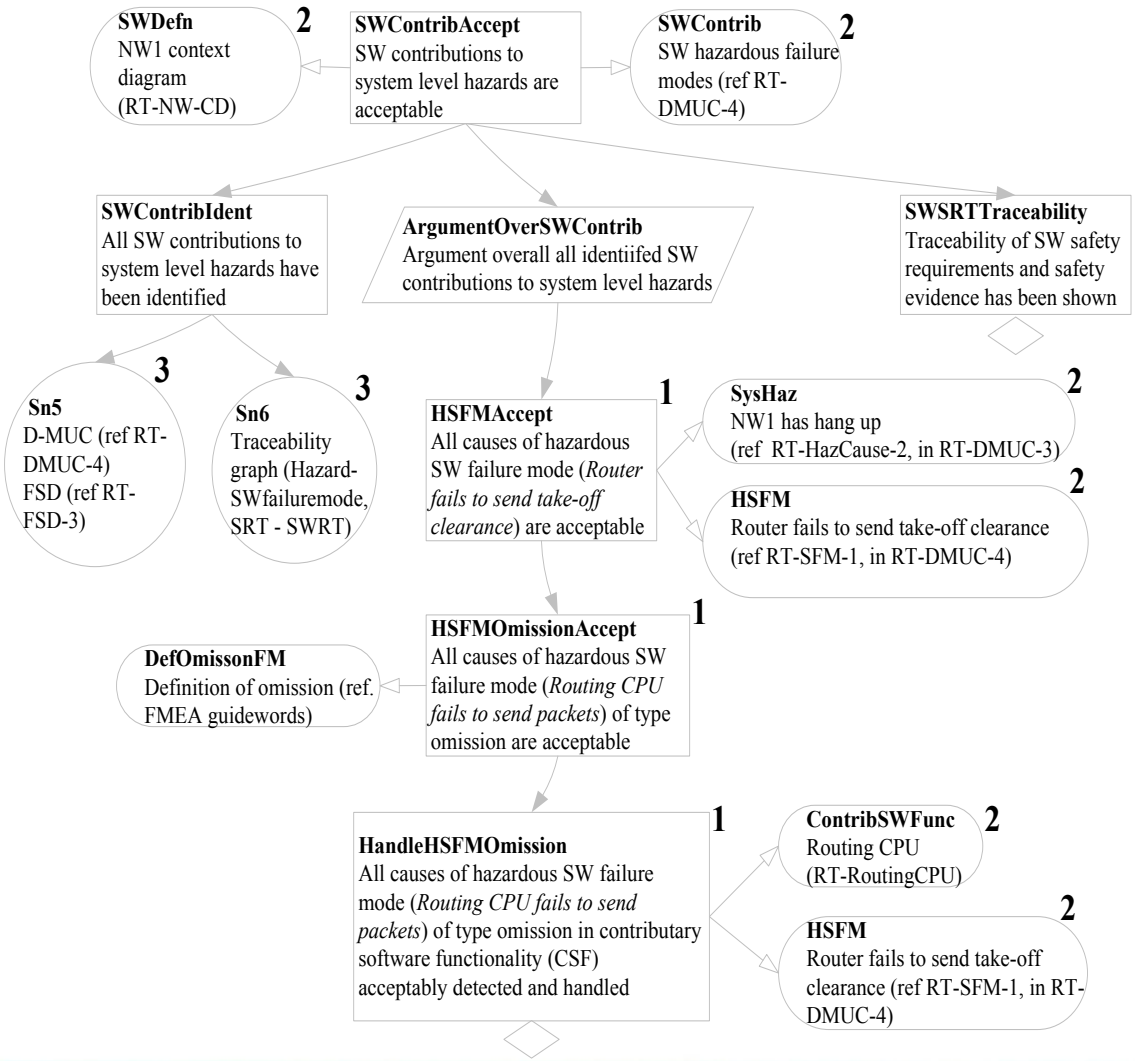
pattern based on [Weaver' 03]

Assurance case (safety)

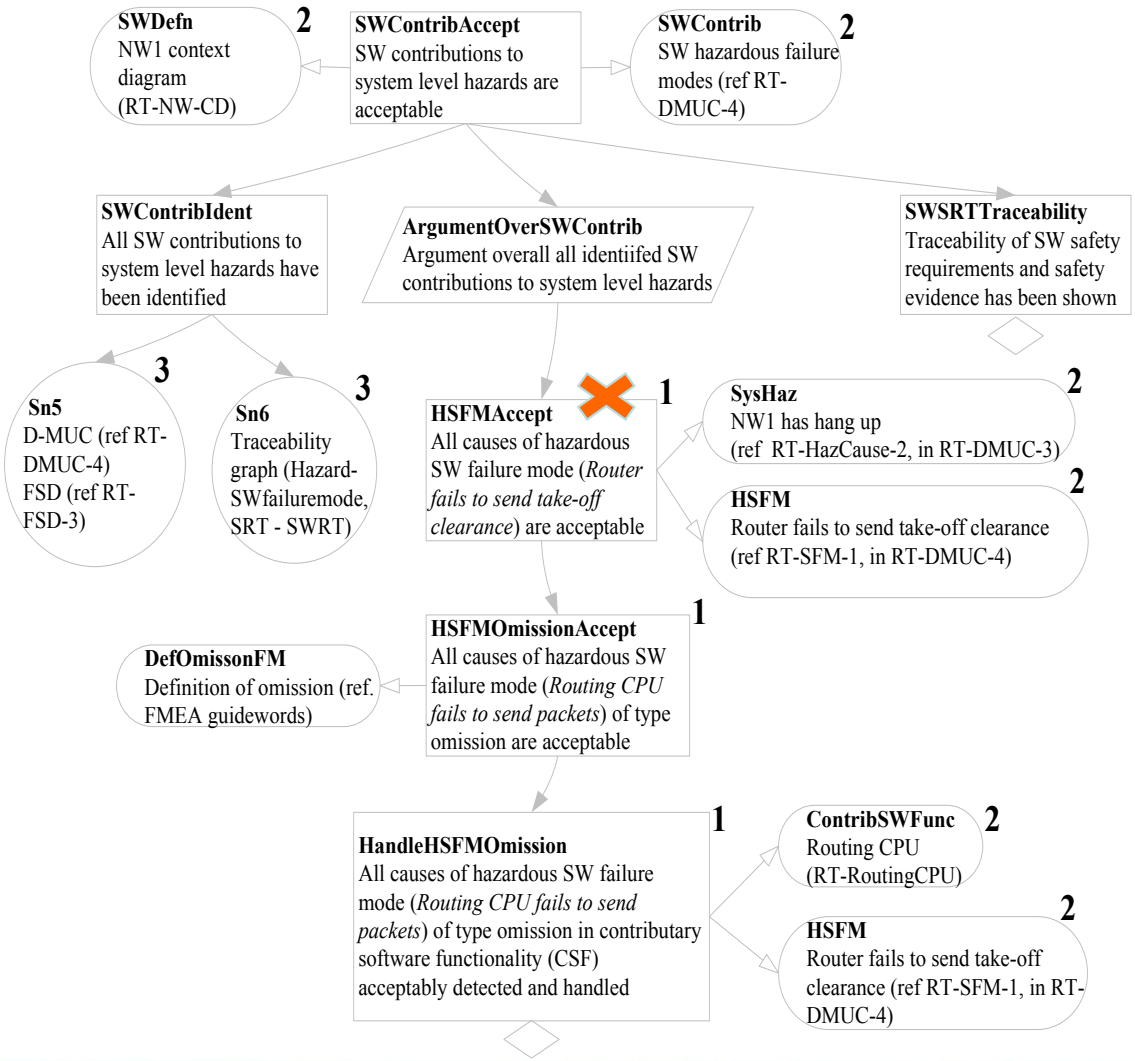


pattern based on [Weaver' 03

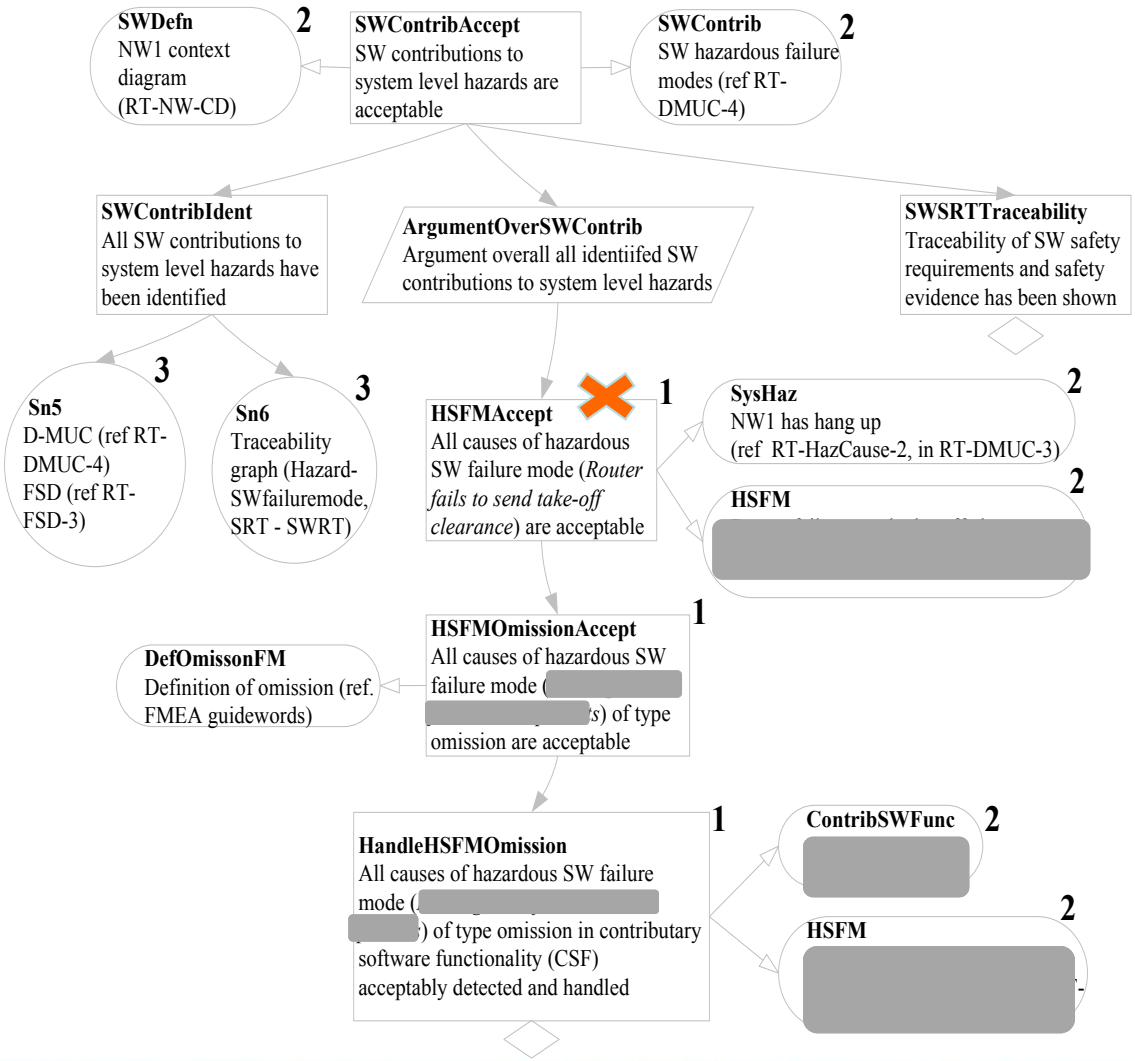
Assurance case (safety)



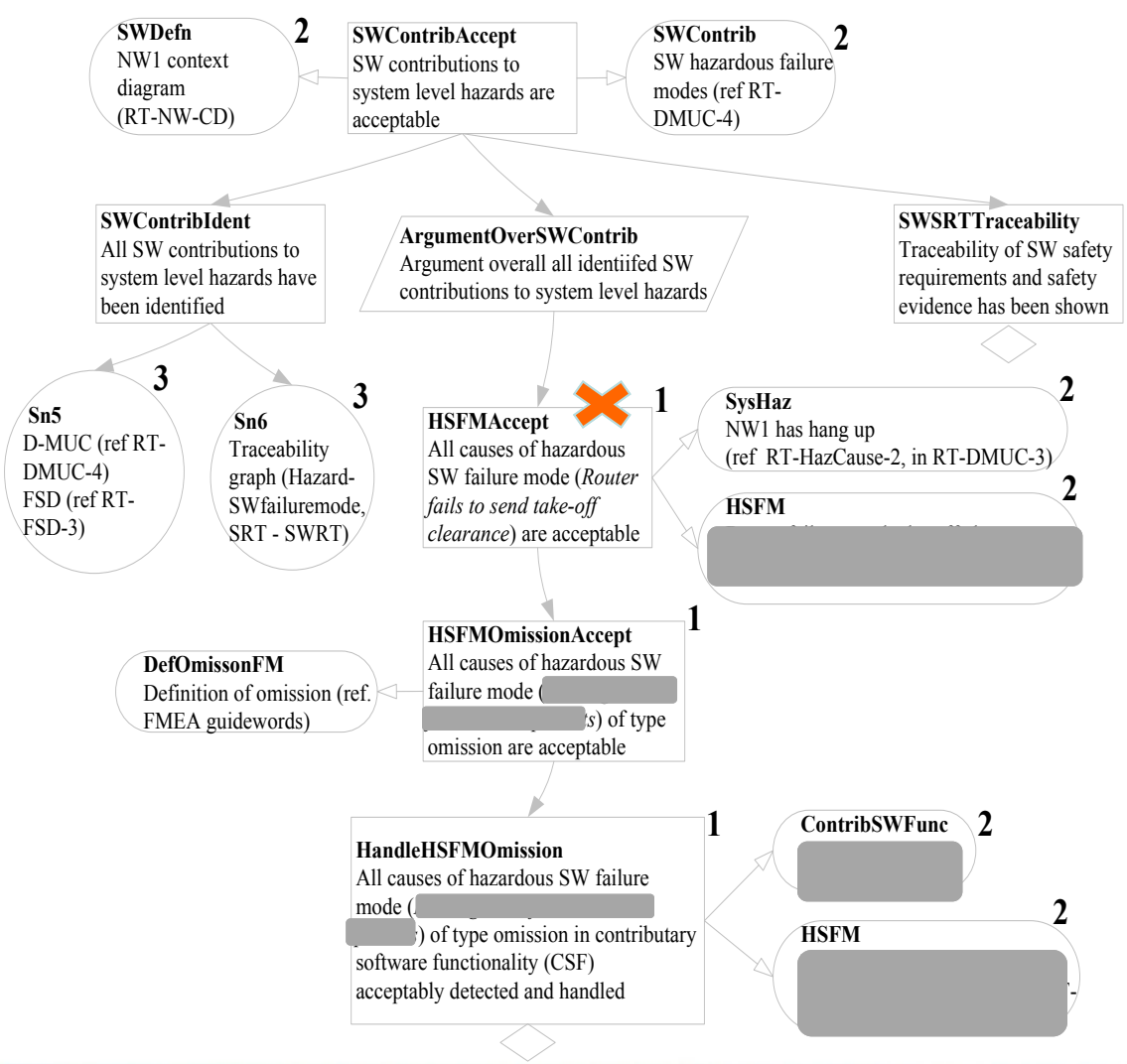
Assurance case (safety)

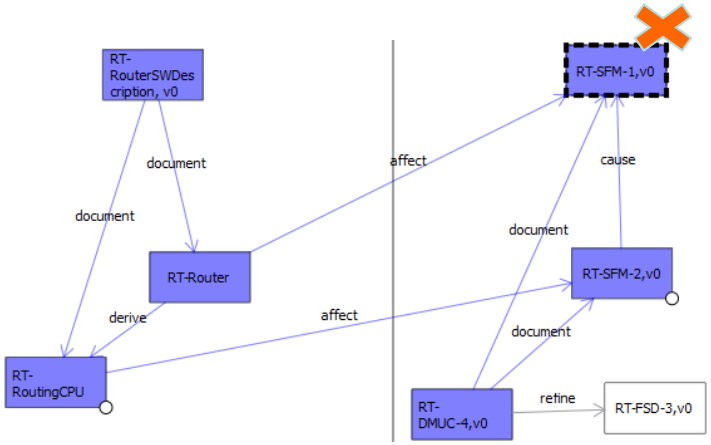


Assurance case (safety)

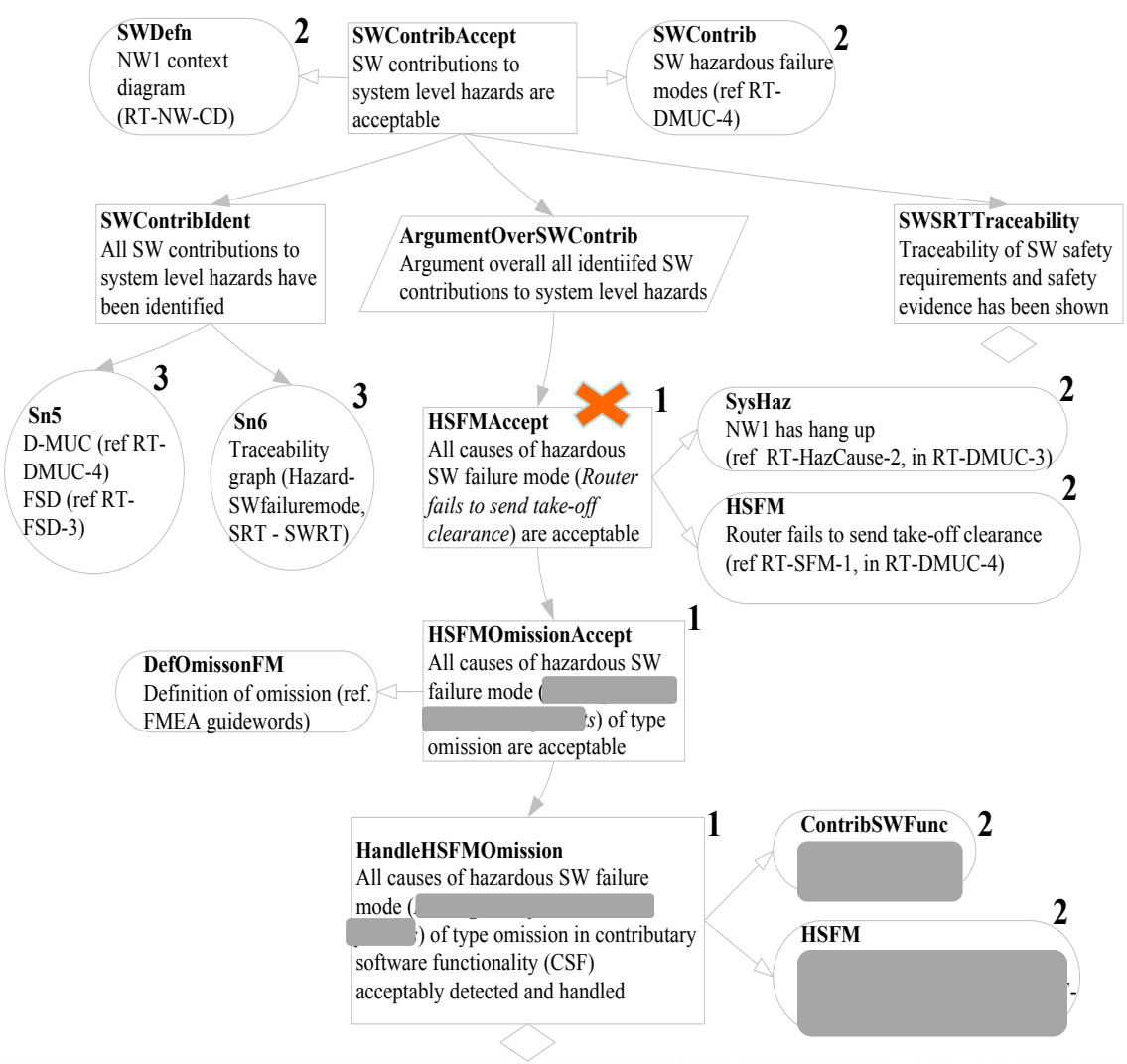


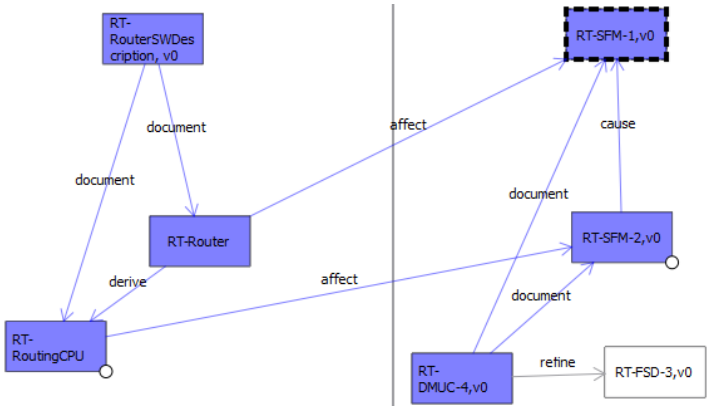
Assurance case (safety)



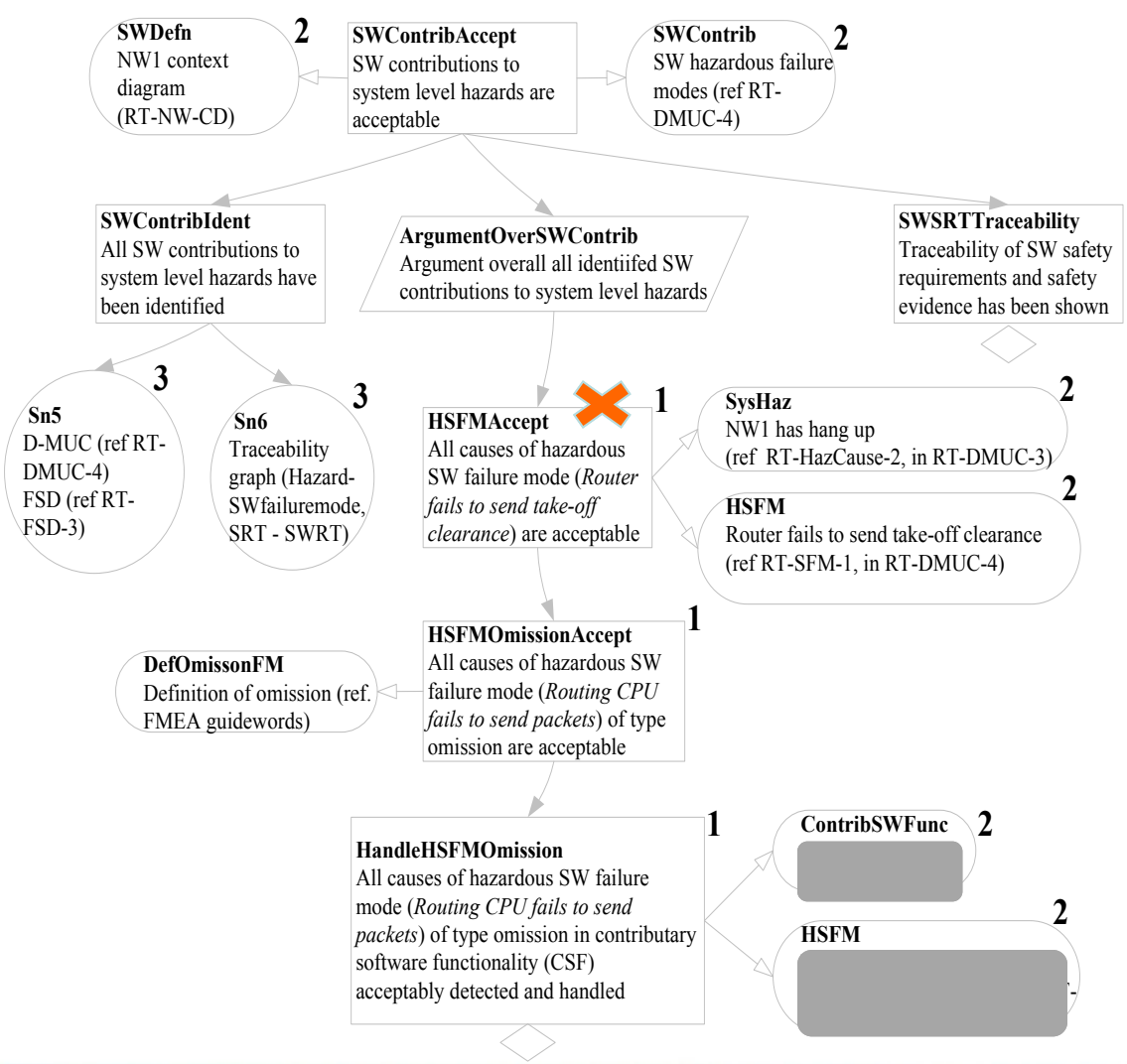


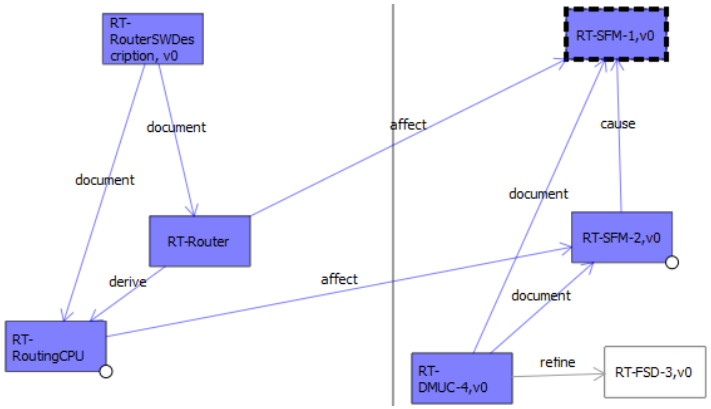
Assurance case (safety)



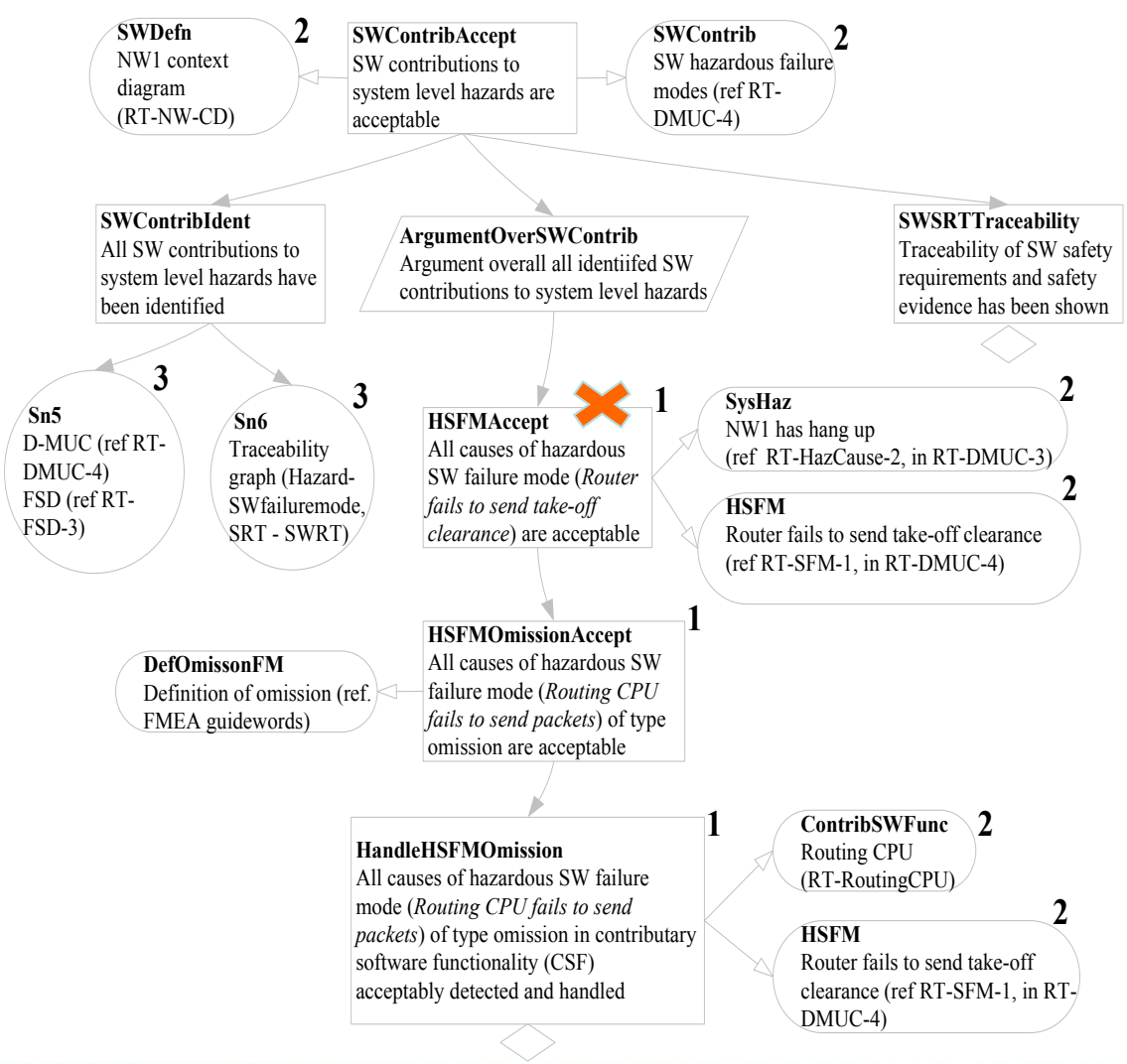


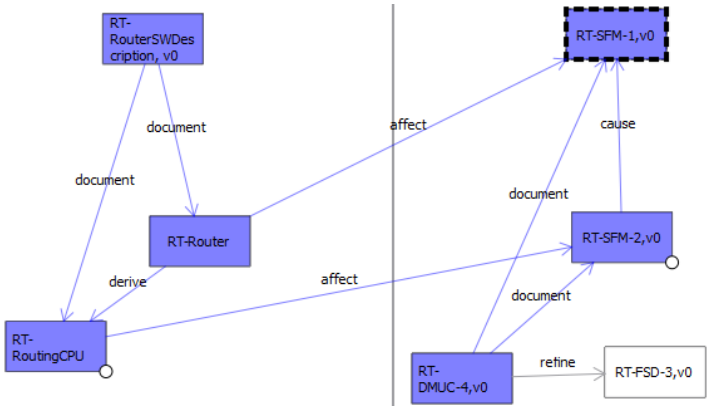
Assurance case (safety)





Assurance case (safety)





Observations – RT example

- Specifying traces was time consuming
  - post development (modelling)
- Complex graph
  - show important information
- Traceability analysis identifies valid traces (impact)
- Able to generate parts of safety case
- Small example

Traceability gaps and challenges

- With multiple organisations involved in the development, assessment and deployment activities
  - lack of common understanding of what needs to be traced, and how it should be traced
- No guidance on traceability
  - main reason for varying degrees of practises among organisations?
- Without a common RM database/tool
  - laborious to identify the relation between artefacts in different documents (that were produced by different organisations)
- As independent actors who were not involved in the project
  - time consuming task to get an overview of the system and its functionality

Improving traceability through SaTrAp

- Traceability process model as guidance
  - with the traceability process model that has been adapted to ATM domain, it was easier to know which artefacts should be traced
- Identifying the missing traces
  - the process model was used as a checklist to check whether the required traces were described in the project documentation
- Automated traceability analysis
  - the approach and tool considerably reduced the effort needed to perform impact analysis with the help of the different traceability analysis

Thank you