AMASS
ECSEL Joint Undertaking

http://www.amass-ecsel.eu/

Barbara Gallina, MDH

Stefano Puri, Intecs

Bernhard Kaiser, B&M

…

Safe Cooperating Cyber-Physical Systems
using Wireless Communication | SafeCOP

http://www.safecop.eu/

Hans Hansson, MDH

Henrik Thane, MDH

Sasikumar Punnekkat, MDH

…

Facing design and assurance challenges of security-informed safety-critical vehicle platoons via FLAR2SAF

Irfan Sljivo

Irfan.sljivo@mdh.se

MÄLARDALEN UNIVERSITY
SWEDEN

5th Scandinavian Conference on System and Software Safety,

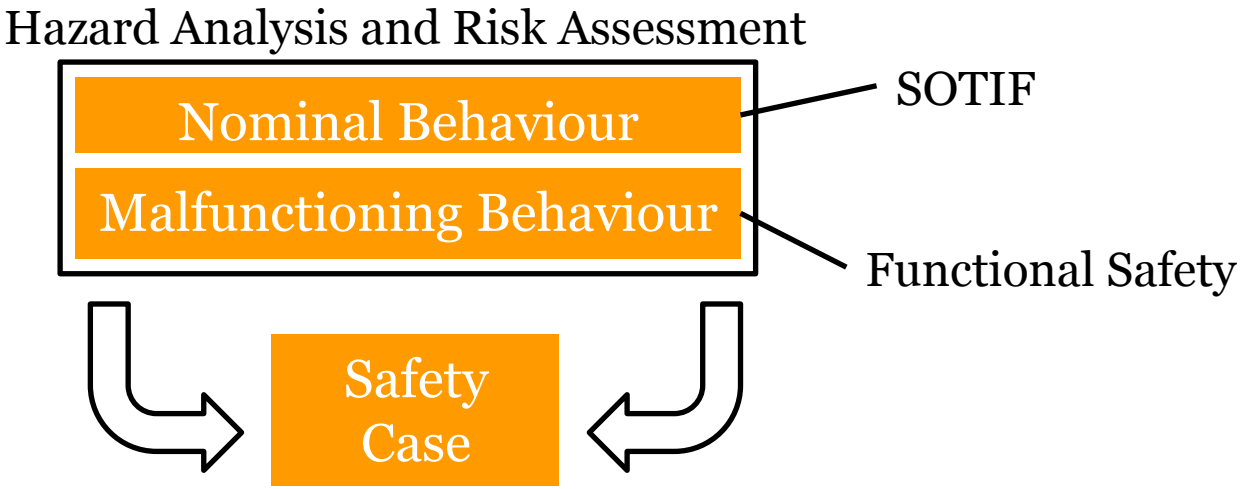
Stockholm, May 2017

Agenda

- Safety Cases
- Certification and Reuse
- Cooperative Functions
- FLAR2SAF for Cooperative Functions
- Vehicle Platooning Example

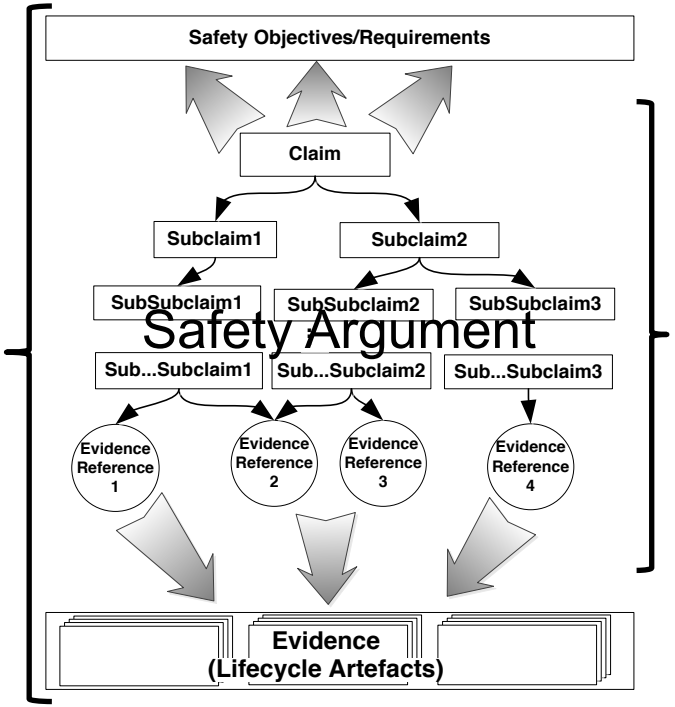
Safety-Critical Systems

- Safety-critical systems
  - Malfunctioning can result in harm or loss of human life, or damage to property or the environment
  - But not only malfunctioning is safety-relevant, sometimes the harm can be done even in absence of failures



Safety Case

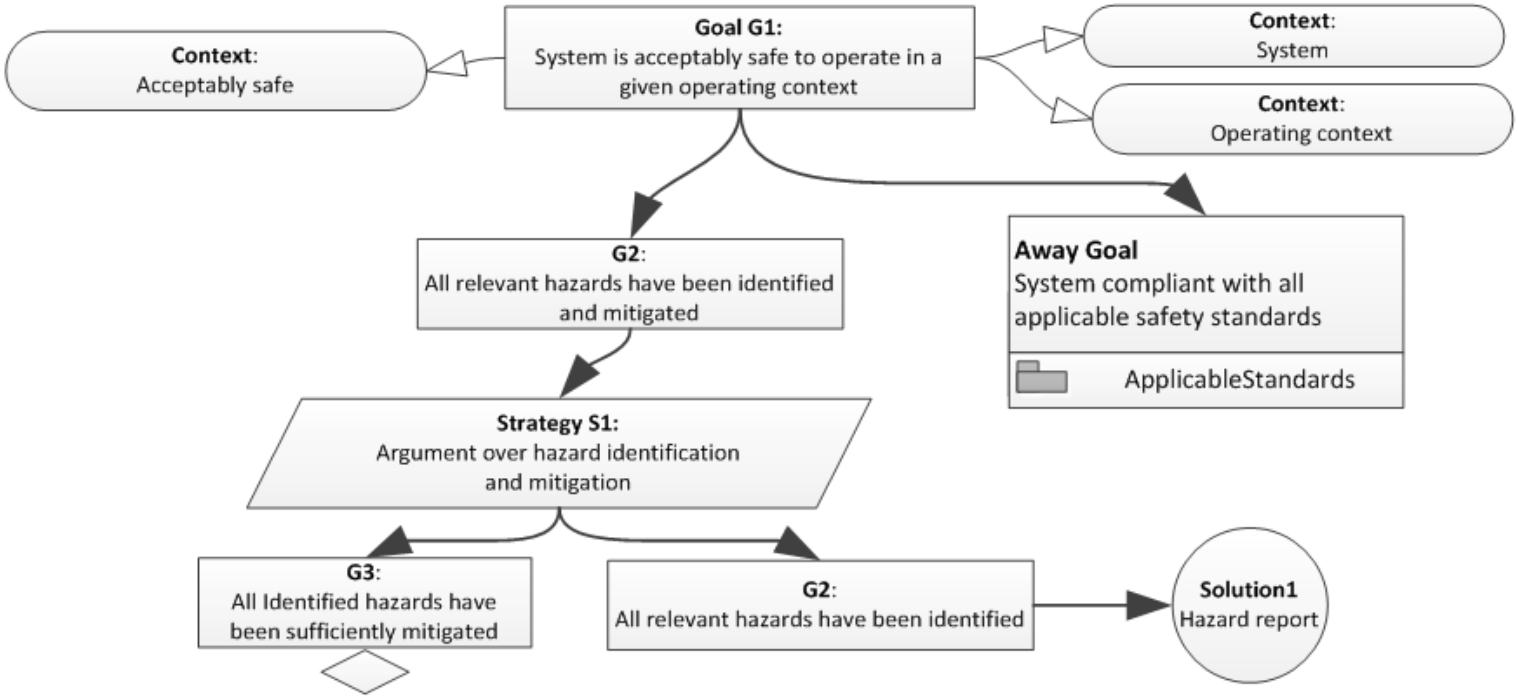
- A *safety case* is documented in form of a structured argument to clearly communicate that the system is acceptably safe to operate in a given context [Kelly, 1998]



- *Safety argument* is the "spine" of the safety case showing how safety objectives/ requirements are connected with evidence

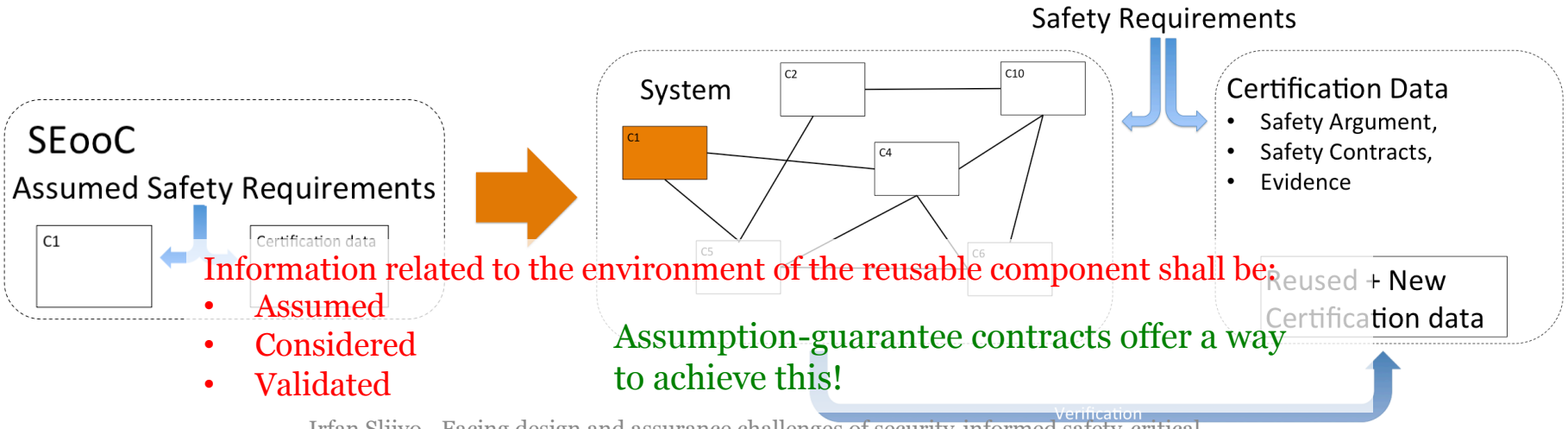
- *Assurance case – safety case generalisation*
- *Goal Structuring Notation* (GSN) - a graphical argumentation notation that can be used to specify elements of any argument [GSN, 2011]

GSN – An Argument Example



Safety-critical Systems: Certification* and Reuse

- *Composable certification – an approach that assumes **reuse of certification data*** as a way to reduce the cost and time needed to achieve certification (*development of an assurance case)
- Single-domain reuse within the safety standards:
  - Automotive ISO 26262 - Safety Element out of Context (*SEooC*)
  - Avionics DO-178B/C– Reusable Software Component (*RSC*)



Component Contracts

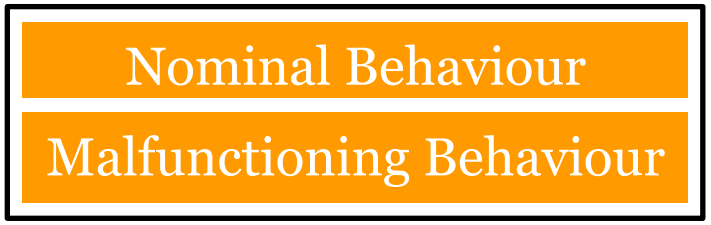
- "Design by Contract" – Bertrand Meyer (1992)
- Assumption/guarantee contracts C=(A, G)
  - A component offers the guarantees G if assumptions A on its environment are satisfied
- Contract viewpoints
  - Functional, timing, safety…
    - Safety contract – a contract that addresses safety requirements
    - Security contract – a contract that addresses security requirements…
    - Note that one contract may be addressing multiple concerns!

Safety Contract Derivation

- Just as hazard analysis is the basis for safety engineering at the system level, derivation of contracts plays the similar role on the component level

Hazard Analysis and Risk Assessment

- Failure Mode Effects Analysis (FMEA)
- Fault Tree Analysis (FTA)
- Fault Propagation and Transformation Calculus (FPTC)



- Safety contracts capture safety-relevant behaviours of components deemed relevant from the perspective of hazard analysis

FLAR2SAF

Failure Logic Analysis Results (FLAR)

A/G Contracts

Evidence

Common Assurance and Certification Metamodel – CACM

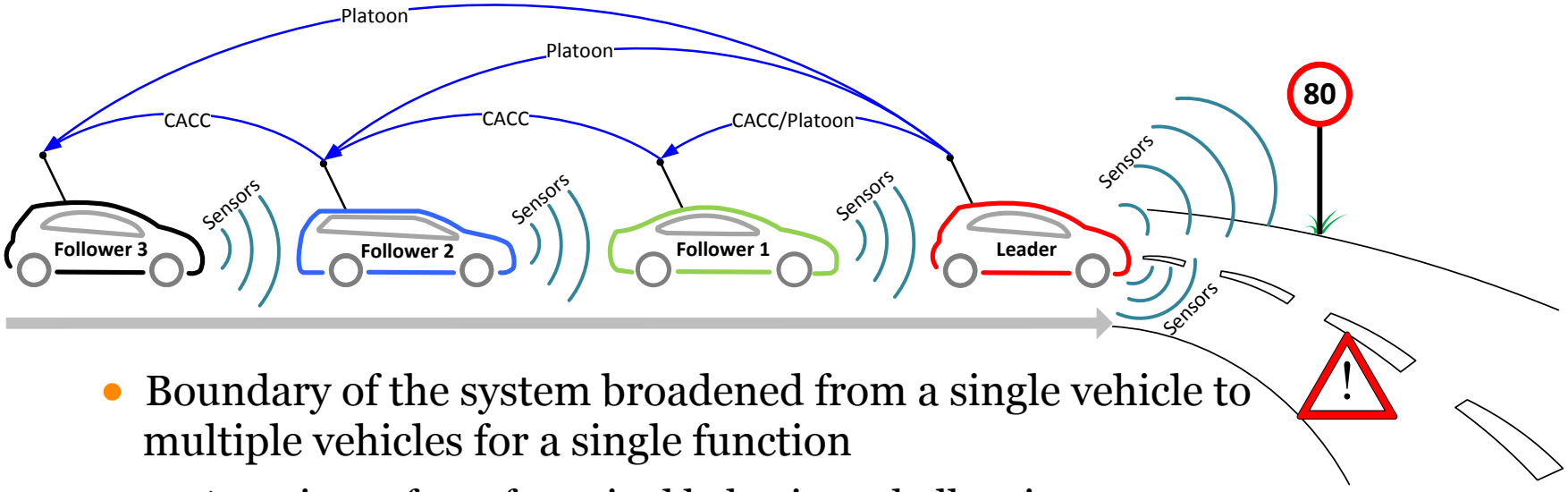
Safety Argument-Fragment (SAF) generation

Evidence reuse

FLAR2SAF for Cooperative Functions

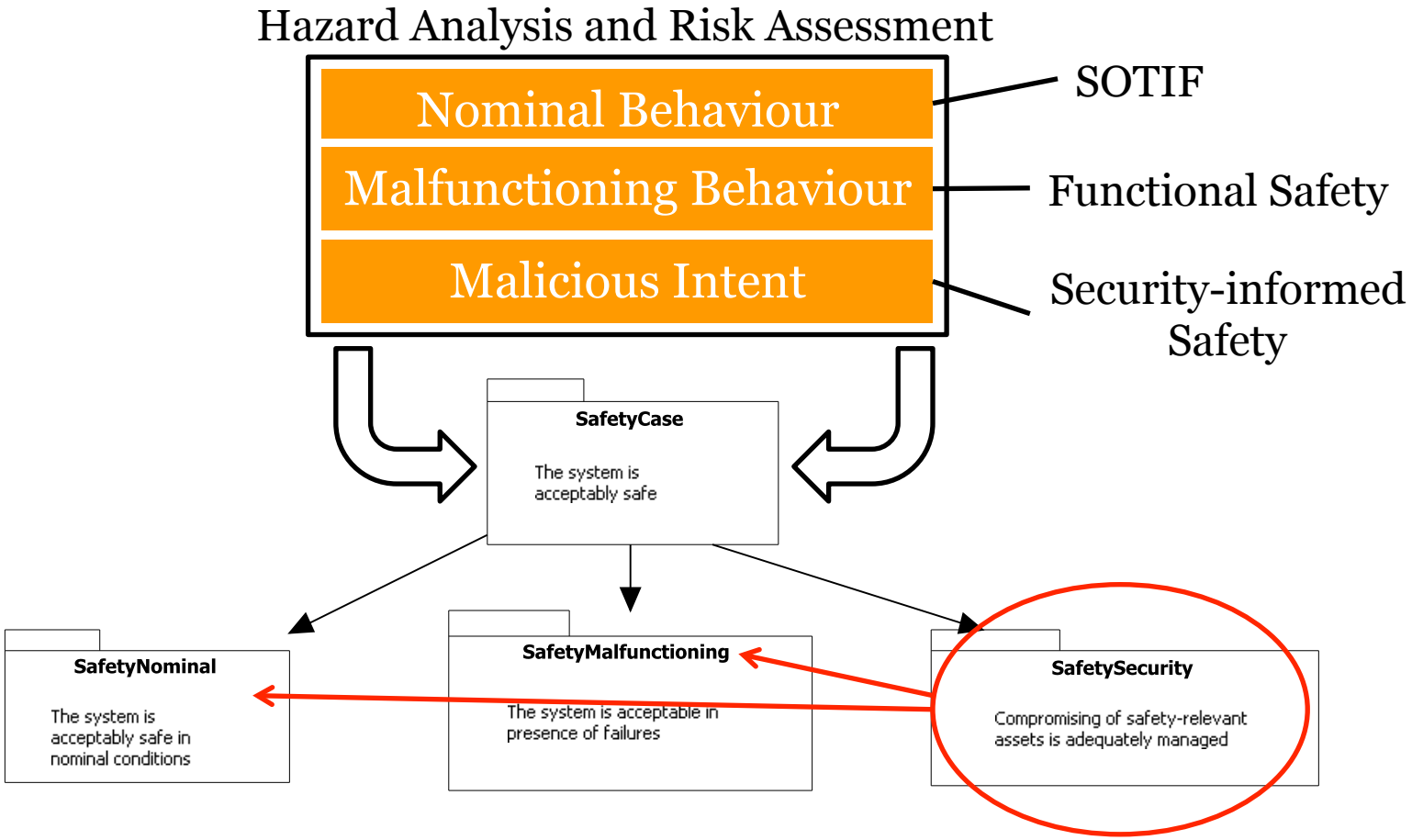
- Deriving contracts for cooperative functions
  - What should the contracts capture?
    - Is failure behaviour sufficient to cover?
- How can such contracts support design and assurance challenges of cooperative functions?

Cooperative Functions: Vehicle Platooning



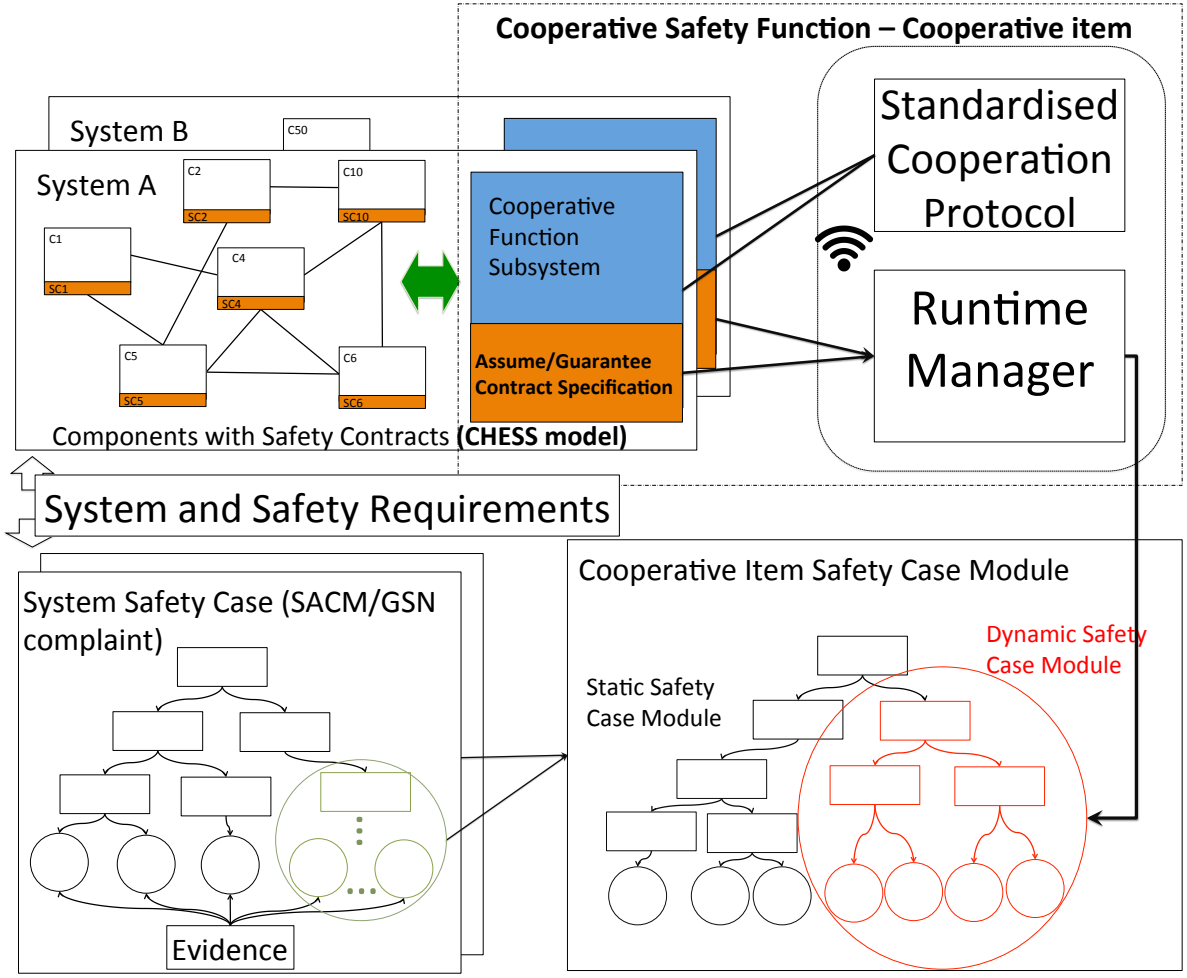
- Boundary of the system broadened from a single vehicle to multiple vehicles for a single function
  - Assuring safety of nominal behaviour challenging
  - The failures of interest for malfunctioning behaviour:
    - Technical failures in the local car
    - Technical failures in another participating car
    - Failures in the communication
  - Trust in the other participants in the cooperation under question

Security-informed Safety Case



Safety needs to become more like security, it needs to be more dynamic!

SafeCOP Safety Assurance Concept



- Standardised cooperation protocol must be the basis for cooperative safety function (e.g., platooning standard to which each vehicle must comply to participate in the cooperative function)
- Runtime manager
  - Checks contracts during runtime
    - Which are active, and
    - Which are violated

Runtime Manager

- Dual role
  - State manager
    - Degrading performance to improve dependability
  - Runtime assurance
    - Supporting the dynamic part of the safety case
      - Support for multiconcern assurance

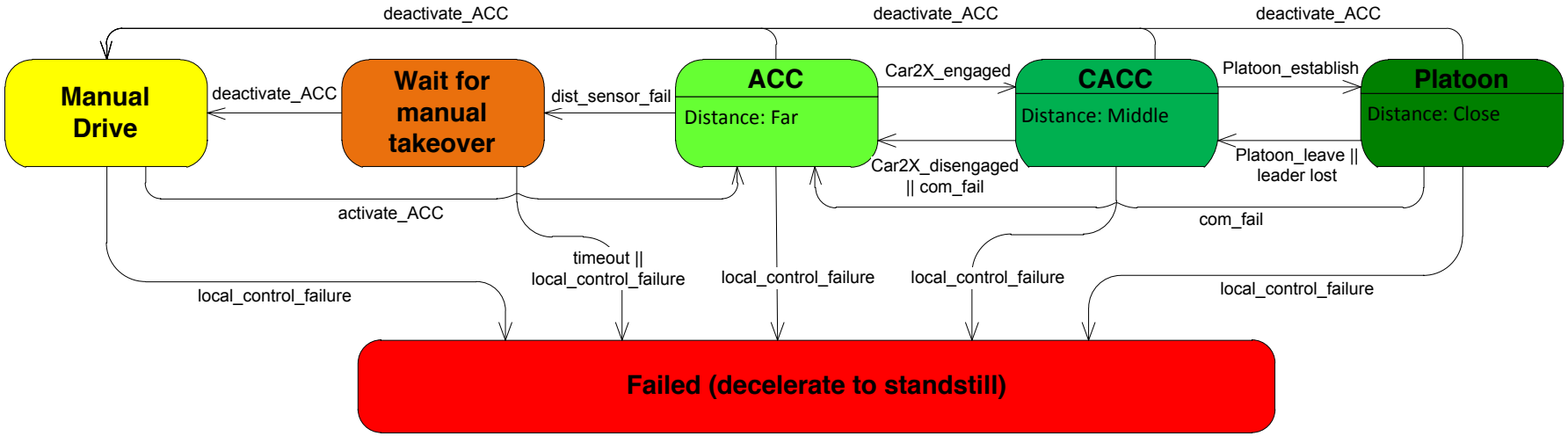
Runtime Assurance

- We built the initial confidence that the system is acceptably safe based on the contract checking and the fact that the contracts are sufficiently complete

- How do the reported runtime contract violations influence our initial confidence in the contracts?
  - With security in the loop, its time to start using the counter example elements in the safety arguments

- Answering the question:
  - Is the system still acceptably safe?

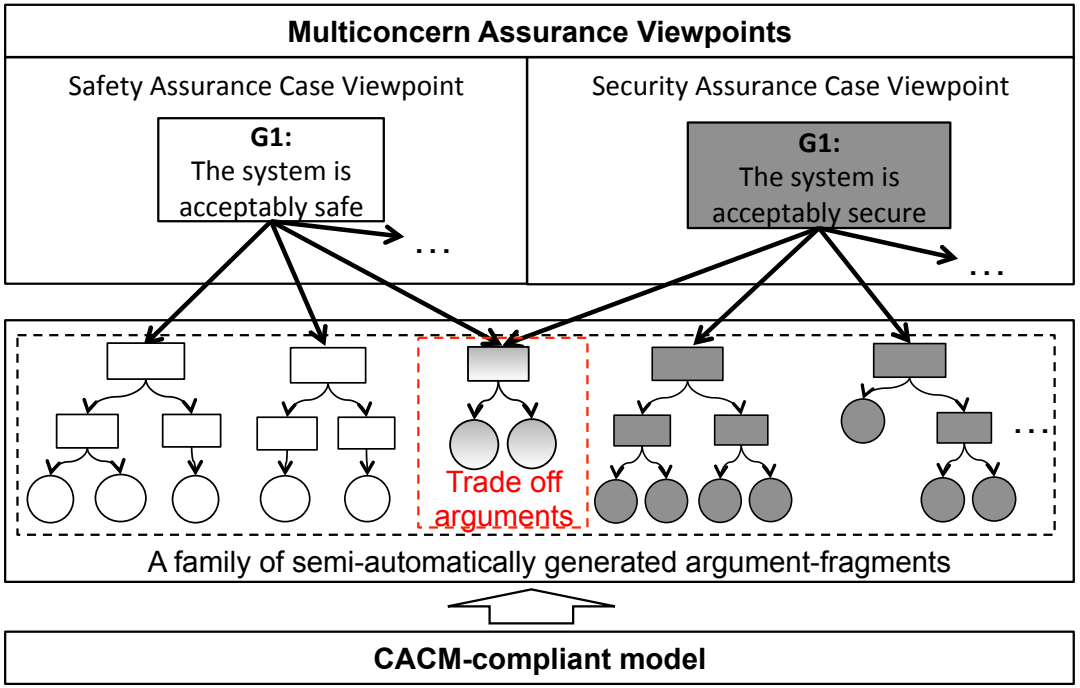
RM as State Manager



- RM degradation mode contracts capture failure detection conditions leading from one mode to another
- An informal contract example for the platoon mode:
  - **A:** No failures (due to malfunctioning or intrusion) and braking of the predecessor vehicle is recognised within 30ms
  - **G:** The distance to the predecessor vehicle is always greater than 20m AND a sudden braking manoeuvre of the preceding vehicle does not lead to a resulting distance of less than 2m

Multiconcern Assurance

- We extend CACM with concern-specific tags to both system and assurance elements
  - Requirements and contracts in the system domain
  - Argument elements and evidence in the assurance domain



Conclusions

- Safety case needs to adapt for inclusion of security
  - It needs to be more dynamic
- Digitalising certification assets and structuring according to a metamodel offers a way to battle the increasing dynamicity
- We need to constantly evaluate our confidence in the contracts
- Safety and security analyses need to come closer
  - Whether we have joint or separate analyses, it does not matter as long as we can bring them together by specifying both safety and security contracts

Thank you!

Questions and comments?

Related Papers

- Irfan Sljivo and Barbara Gallina and Jan Carlson and Hans Hansson. Generation of Safety Case Argument-Fragments from Safety Contracts. The 33rd International Conference on Computer Safety, Reliability and Security. 978-3-319-10505-5, 170-185, Sep. 2014
- Irfan Sljivo and Omar Jaradat and Iain Bate and Patrick Graydon. Deriving Safety Contracts to Support Architecture Design of Safety Critical Systems. 16th IEEE International Symposium on High Assurance Systems Engineering. 978-1-4799-8111-3, 126-133. IEEE, Jan. 2015
- Irfan Sljivo and Barbara Gallina and Jan Carlson and Hans Hansson. Using Safety Contracts to Guide the Integration of Reusable Safety Elements within ISO 26262. The 21st IEEE Pacific Rim International Symposium on Dependable Computing, Nov. 2015
- Irfan Sljivo and Barbara Gallina and Jan Carlson and Hans Hansson and Stefano Puri. A Method to Generate Reusable Safety Case Fragments from Compositional Safety Analysis. Journal of Systems and Software: Special Issue on Software Reuse (SR-JSS 2016), Jul. 2016
- Irfan Sljivo, Barbara Gallina. Building Multiple-Viewpoint Assurance Cases Using Assumption/Guarantee Contracts. 1st International workshop on Interplay of Security, Safety and System/Software Architecture (ISSA-2016), Nov. 2016
- Samer Medawar, Irfan Sljivo, Detlef Scholle. Cooperative Safety Critical CPS Platooning in SafeCOP. 5th EUROMICRO/IEEE Workshop on Embedded and Cyber-Physical Systems (ECYPS2017), Jun 2017