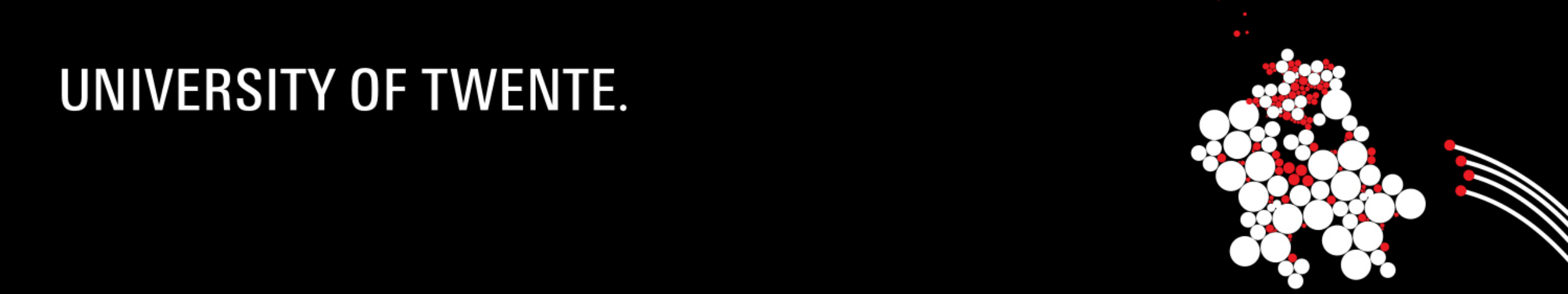
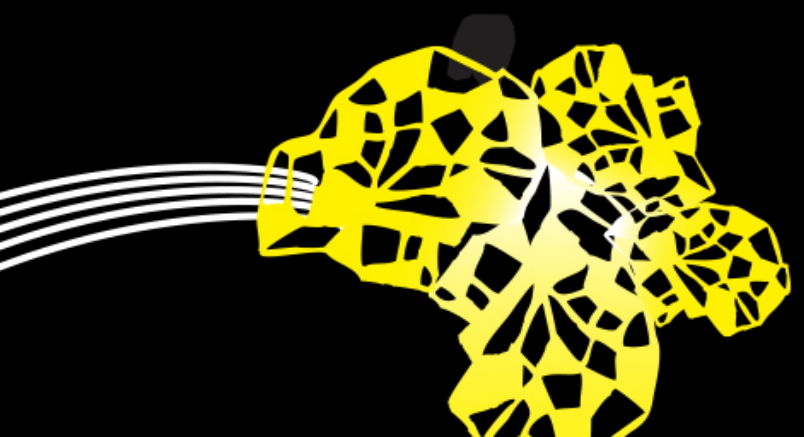
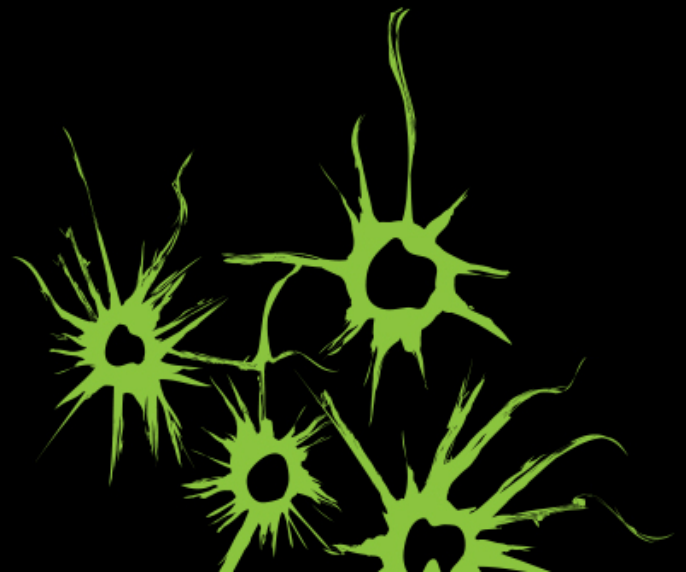
UNIVERSITY OF TWENTE.



UNIFORM ANALYSIS OF FAULT TREES THROUGH MODEL TRANSFORMATIONS



Enno Ruijters,
Stefano Schivo,
Mariëlle Stoelinga,
Arend Rensink,
Rajesh Kumar



TALK OUTLINE

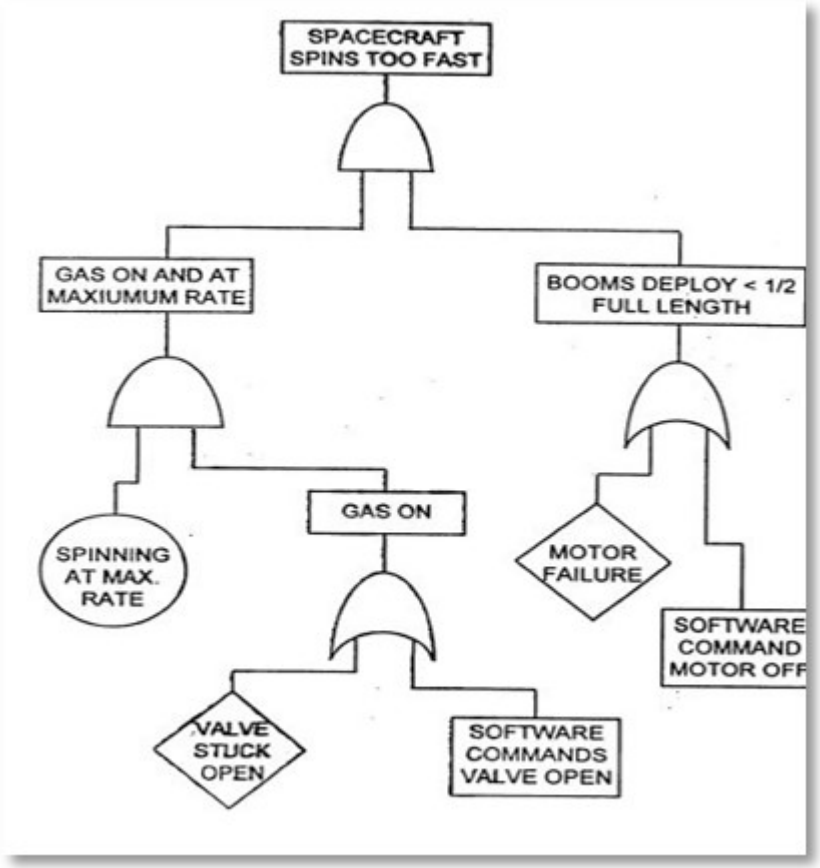
- Introduction & background
  - Fault trees
  - Attack trees
- Methodology
  - Meta-model
  - Transformations
- Case studies
- Conclusions

INTRODUCTION: RAMS

- Dependability of critical systems:
  - Airplanes
  - Power stations
  - Medical devices
- Formal analysis provides important guarantees

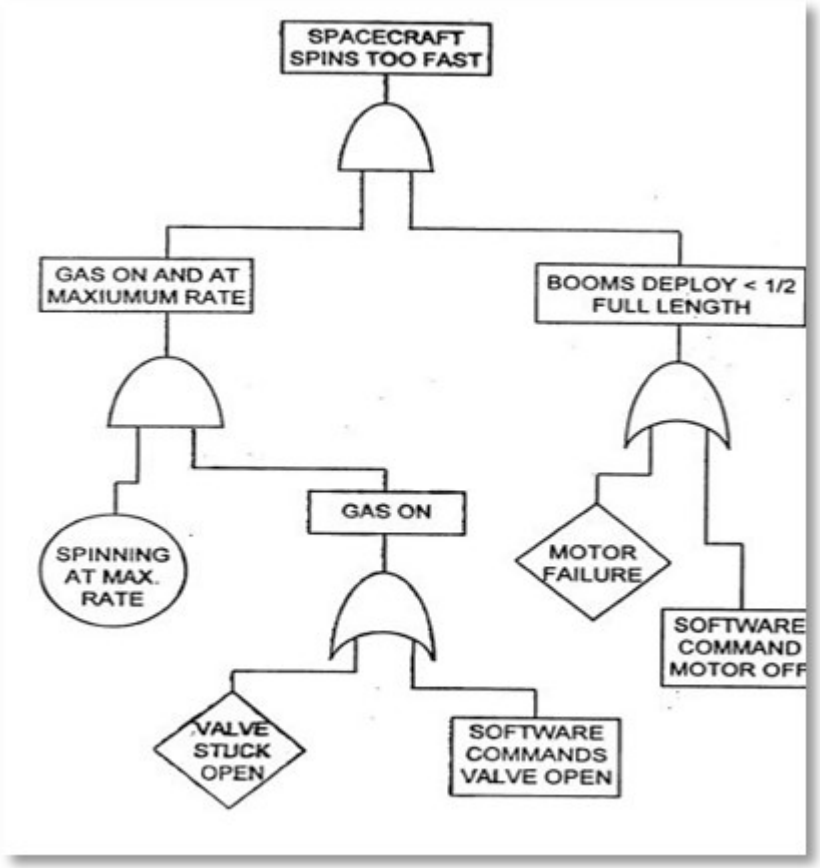
RAMS: METHODS

- FME(C)A / HAZOP
  - Spreadsheet
- Domain-specific modeling
  - AADL, UML, SAVE, etc.
- Reliability block diagrams
- Fault trees

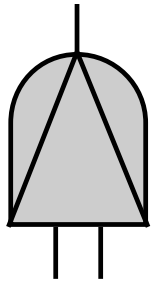


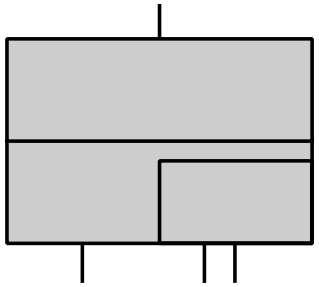
RAMS: FAULT TREES

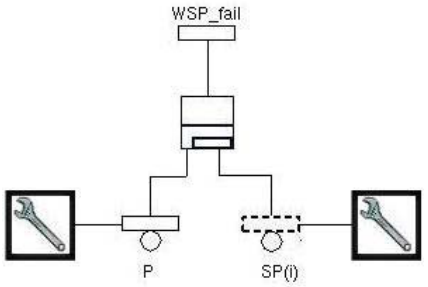
- Industry-standard RAMS tool.
- How do component failures propagate to system failures?
- Used by NASA, ESA, Boeing, etc.
- Top node: Undesired event
- Leaves: basic events (components failures)
- Intermediate nodes: gate combining failures
  - AND, OR, k-out-of-N

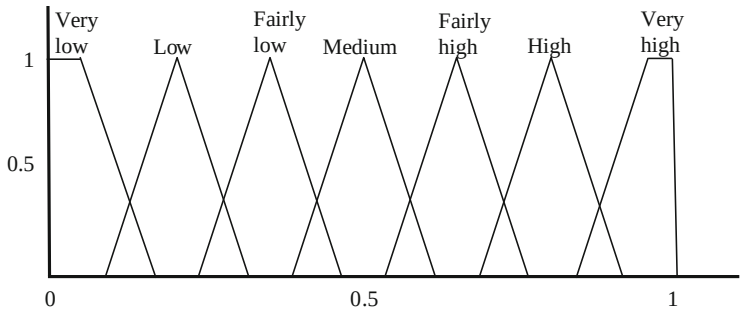


FAULT TREES: EXTENSIONS





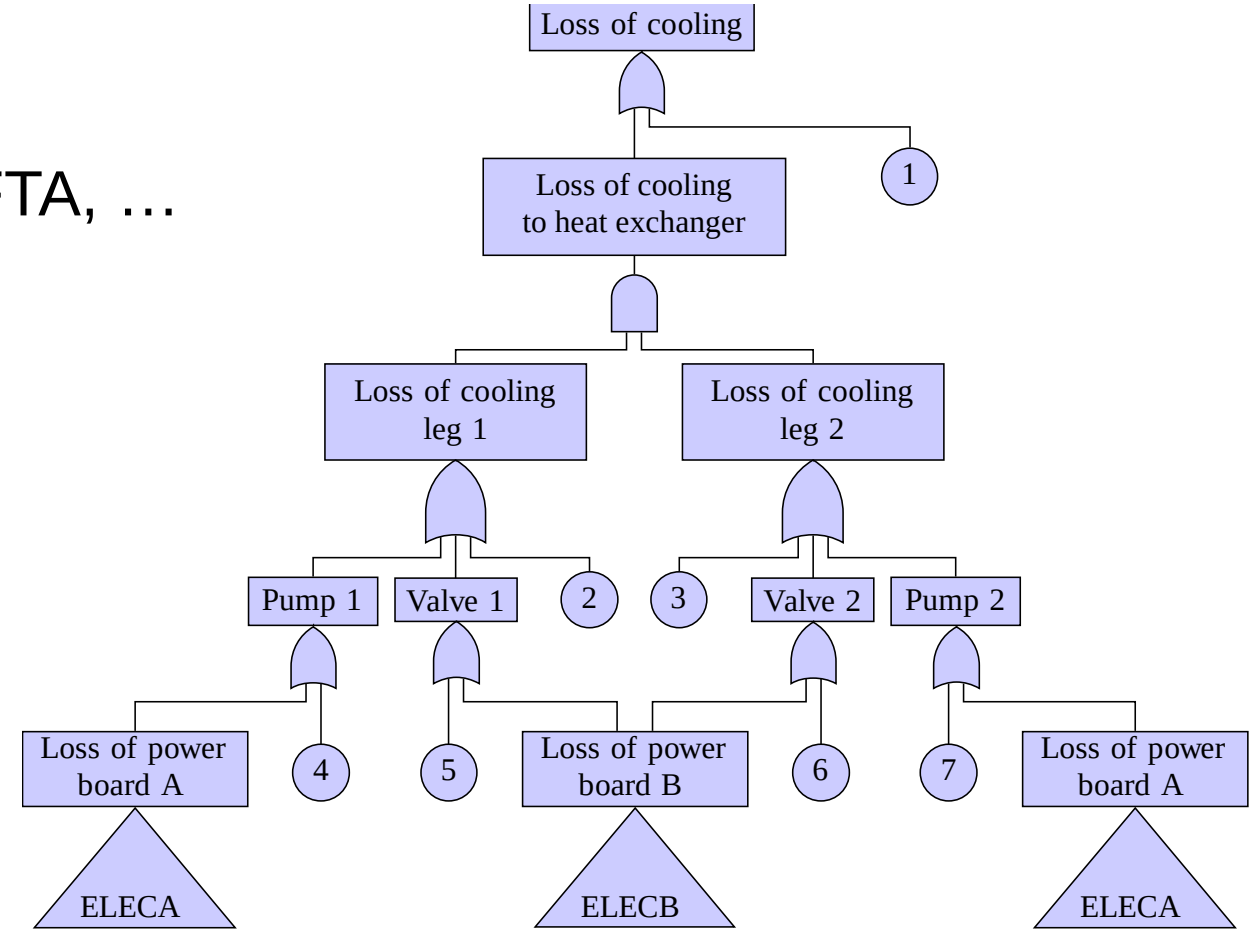




- Order dependent gates
  - Dynamic fault trees (DFT)
- Dependent failure rates
  - Extended fault trees, DFTs
- Repairs/maintenance
  - Repairable fault trees
  - Fault maintenance trees
- Uncertainty
  - Fuzzy fault trees

FAULT TREES: TOOLS

- Commercial
  - IsoGraph FaultTree+, RiskSpectrum FTA, …
- Open-source
  - OpenFTA, DFTCalc, StormDFT, …
- How do they work together?
  - They don't, really.

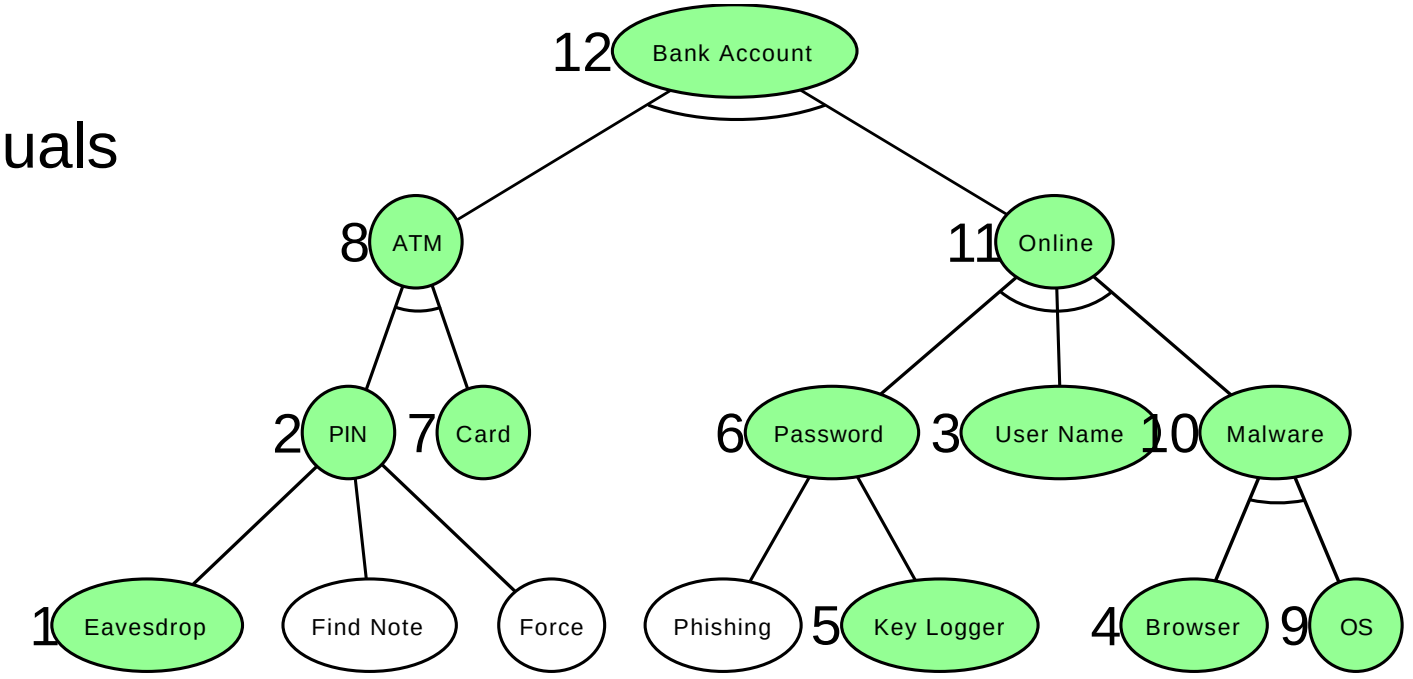


INTRODUCTION: SECURITY

- Safety is not just about the system
- Malicious actors
  - Terrorists
  - Robbers
  - Disgruntled employees
- Part of, and needs similar analysis as, RAMS.
- One method: attack trees

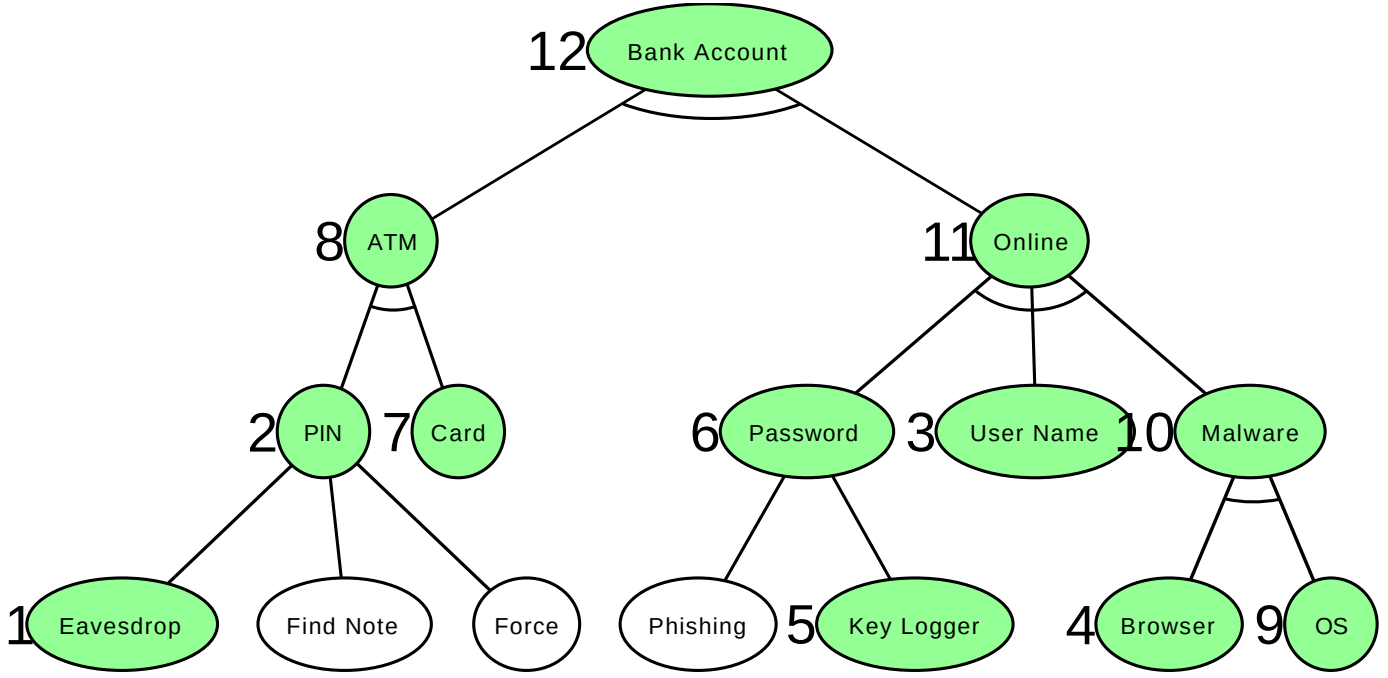
INTRODUCTION: ATTACK TREES

- Tool for analyzing vulnerabilities
- Like fault trees, but for security
- Same concepts, slightly different visuals
  - AND-gates denoted by arc
- Possible new metrics
  - Cost to attack
  - Different attacker models



INTRODUCTION: ATTACK TREES

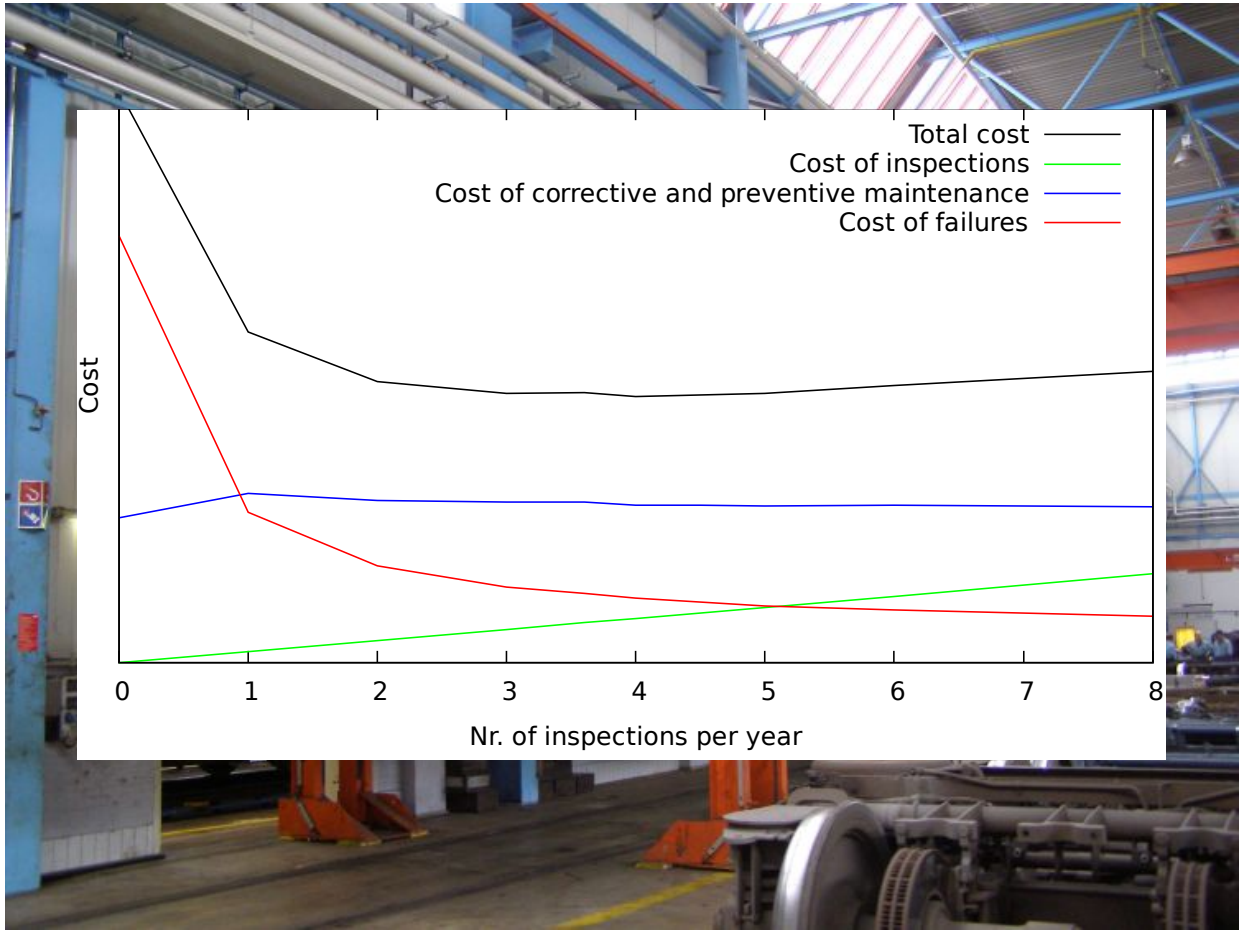
- Extensions
  - Sequential gates
  - Countermeasures
- Tools
  - Commercial, e.g. AttackTree+
  - Open-source, e.g. ADTool
- Again, no real interoperability



INTRODUCTION: ATTACK-FAULT TREES

- Combine attacks and faults in one tree.
- Motivation: interactions between faults and attacks
  - E.g. effects of damaging some components
- Bring together modellers to find more weaknesses.

MAINTENANCE



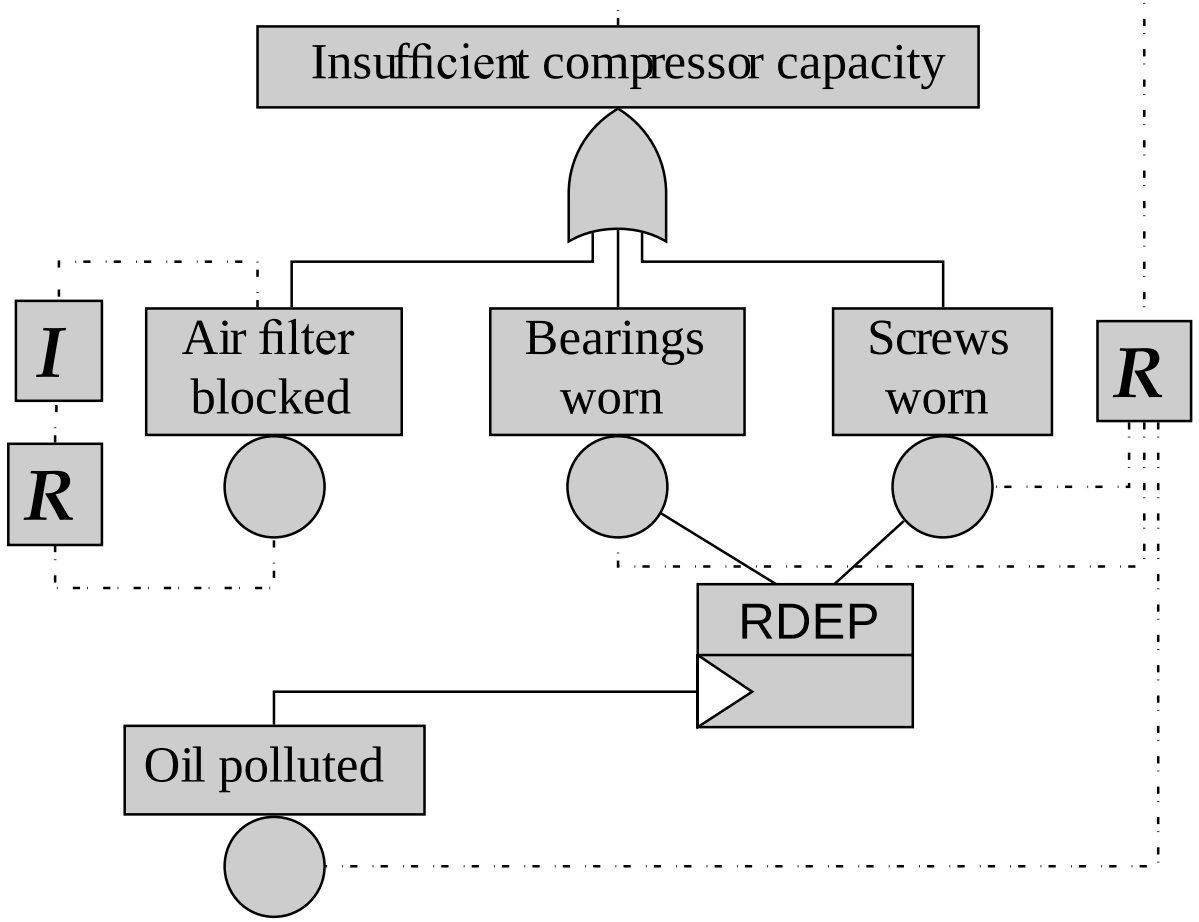
- **Crucial**: Large impact on reliability, life span.
- **Costly:** downtime, labour, equipment, …

- Optimize:
  - Minimal total cost
  - Minimal cost in spec.
  - Maximal reliability in budget

MAINTENANCE

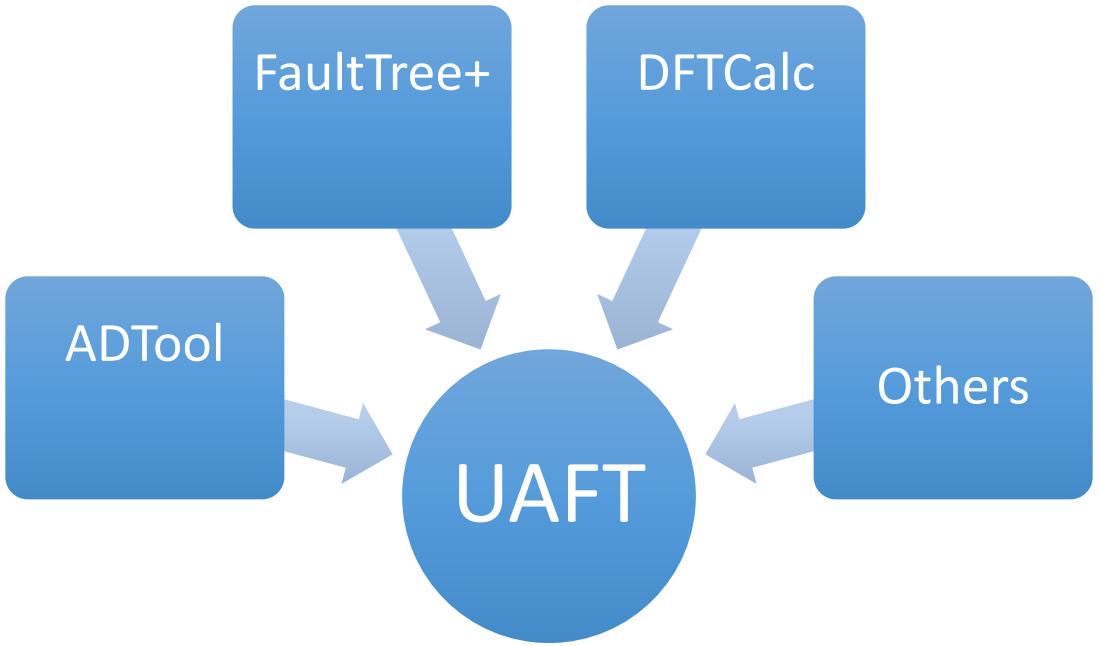
- Types of maintenance:
  - Corrective
  - Preventive
- Timing of maintenance:
  - Age-based
  - Use-based
  - Condition-based

FAULT MAINTENANCE TREES



- Combine fault trees and maintenance
  - Degradation of components
  - Inspections
  - Repairs
  - Dependent failure rates
- Analysis via stochastic timed automata

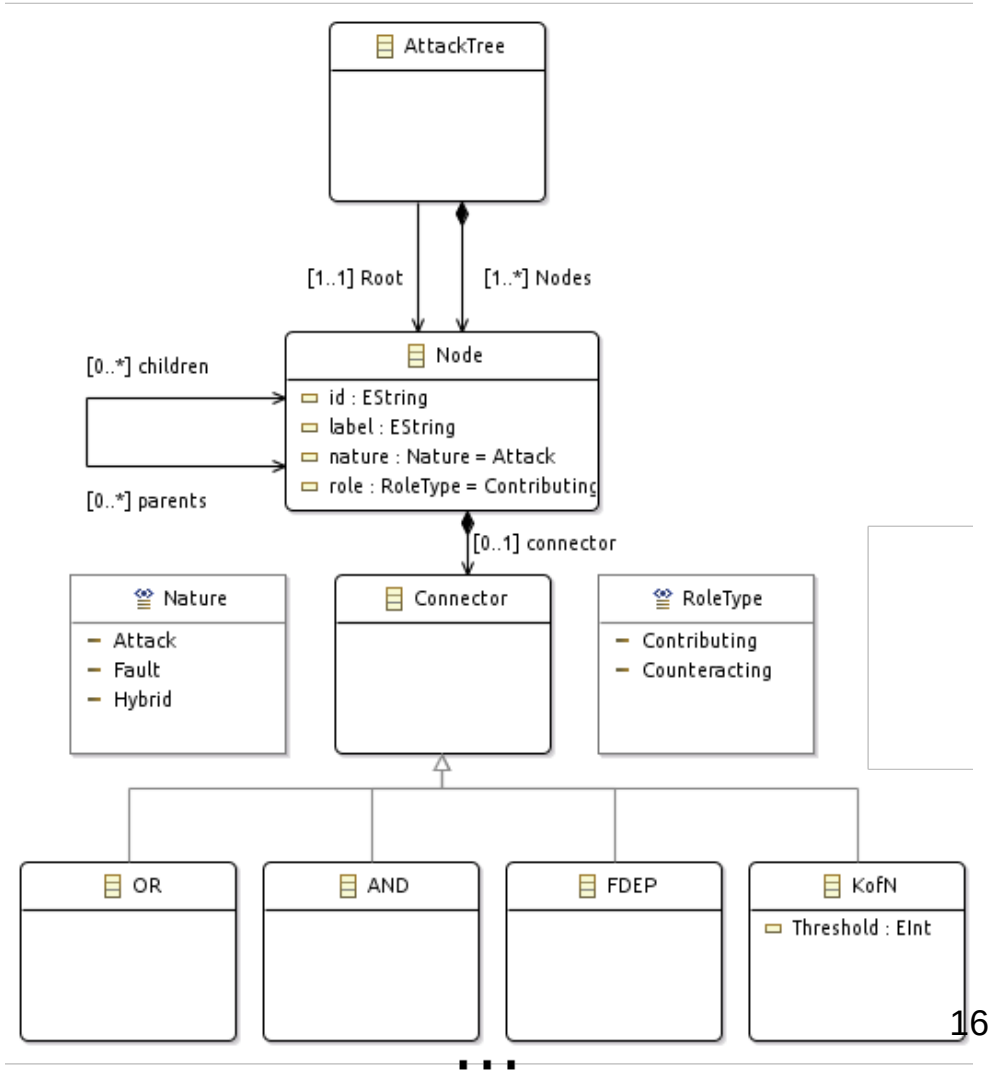
OUR CONTRIBUTION: UNIFIED META-MODEL



- Support for many different formalisms
- Allow combinations (e.g. attack-fault trees)
- Transformations to & from existing tools
- New analysis framework for combined models

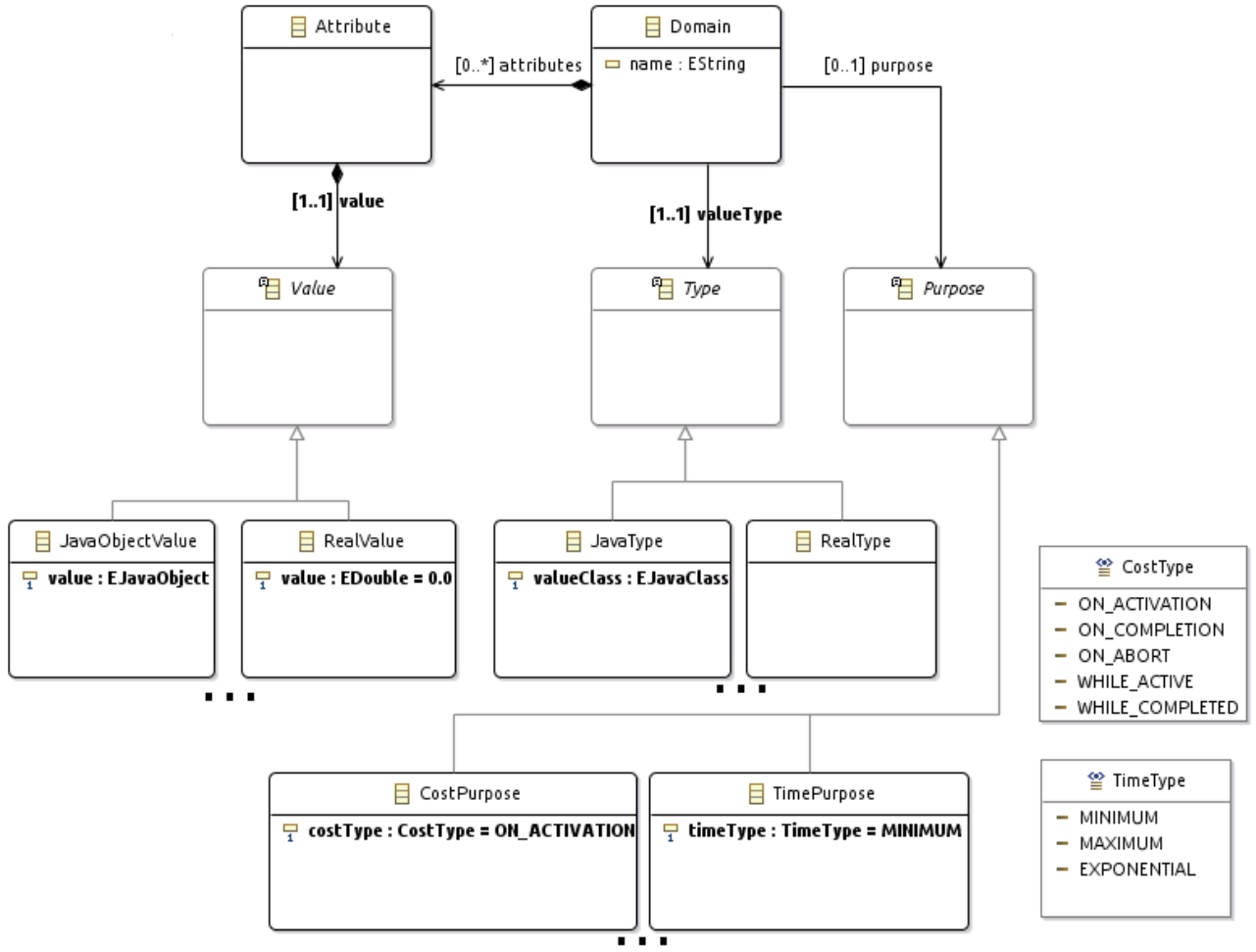
STRUCTURE

- Gates, basic events, and their relations.
- Support for many different gates.
  - AND, OR, SAND, SPARE, etc.
- Supports counteracting nodes.
  - Countermeasures / inhibitors
- Easy to extend with new gates.

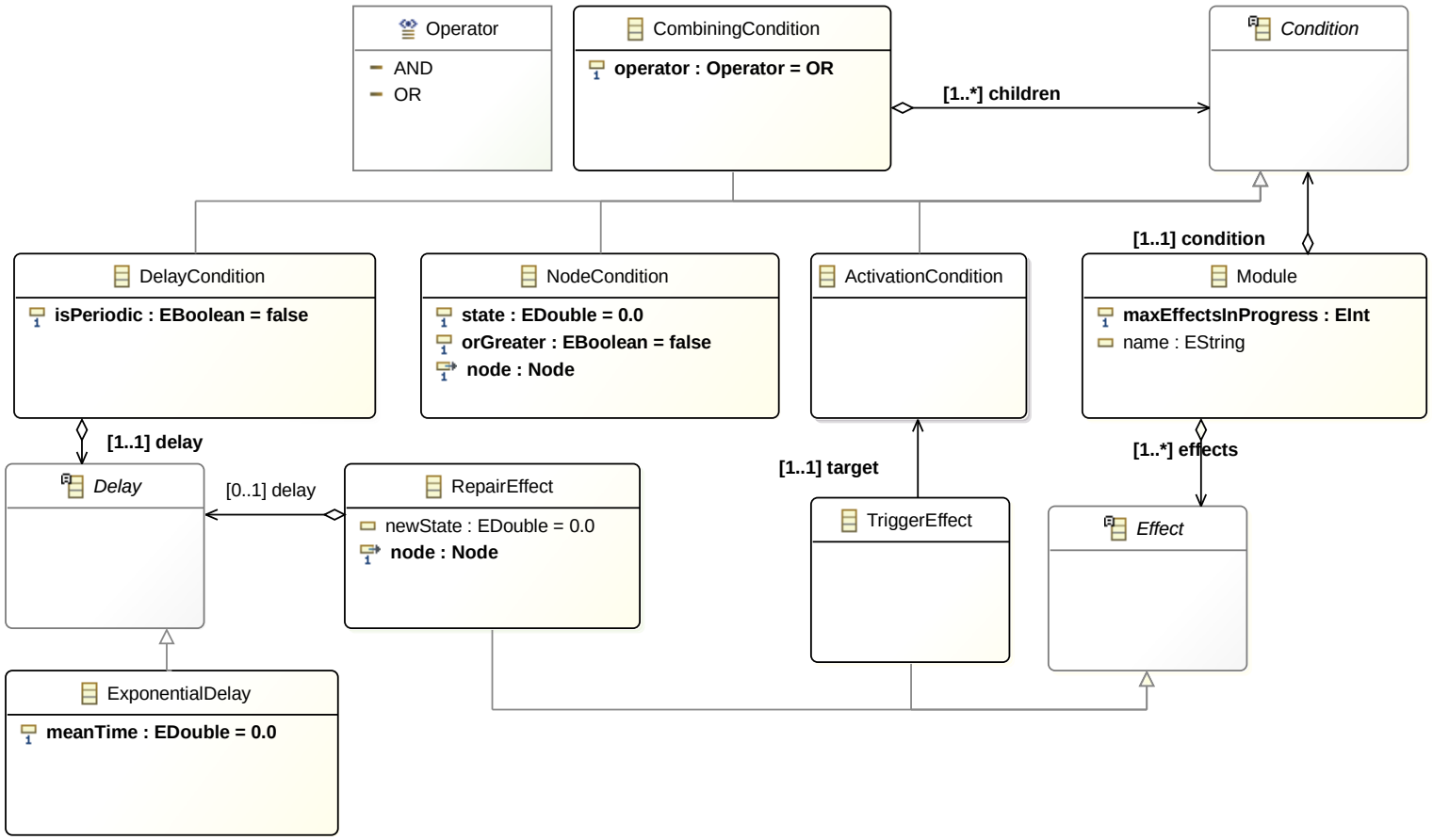


VALUES

- Values associated with nodes
  - E.g. MTTF, cost to attack, etc.
- Includes semantic domain and type information
  - I.e. “This is the time to complete an attack, and must be a real number”.
- Again, easy to extend with new domains and types.



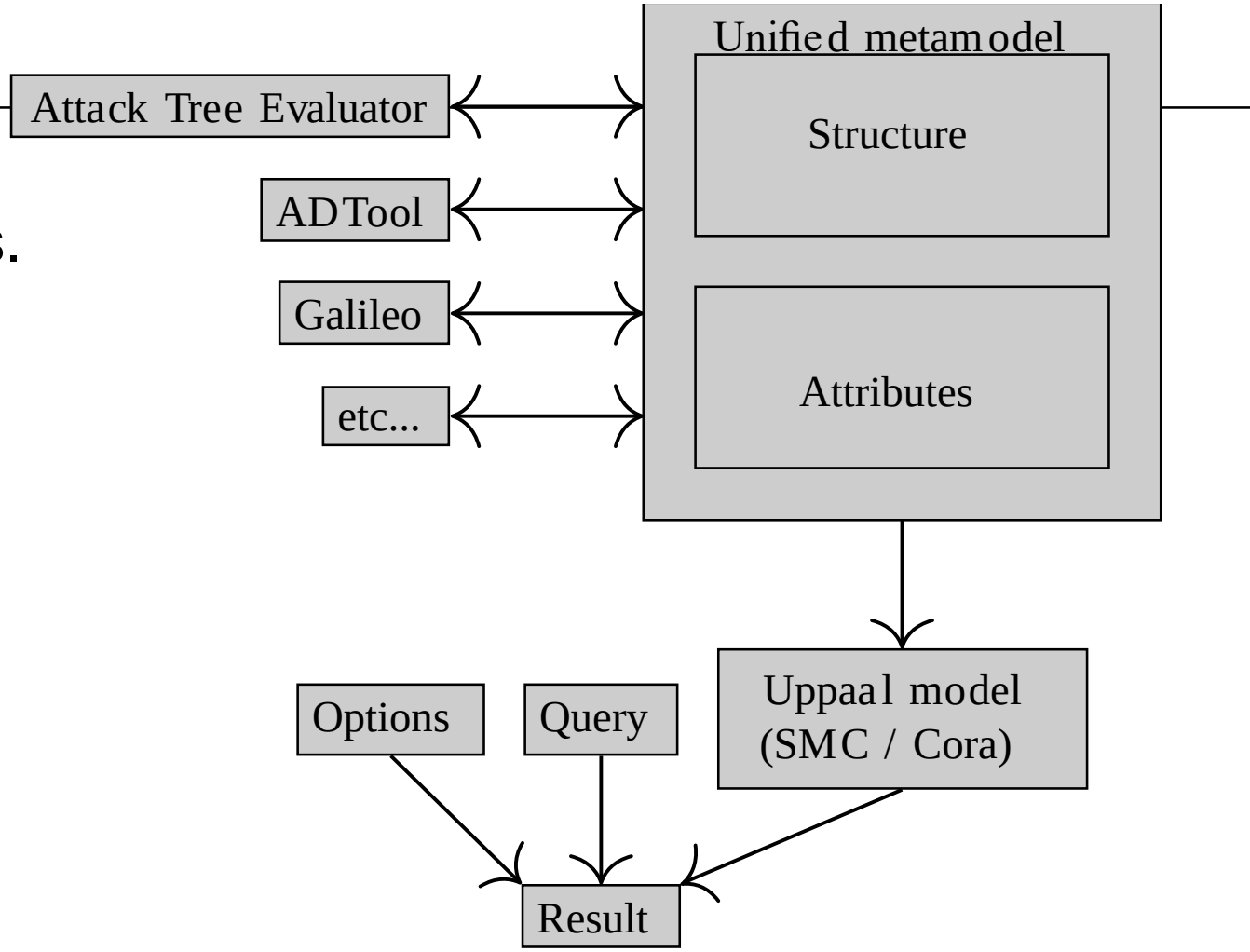
MAINTENANCE



- Models inspections and repairs.
- Set of modules, each with:
  - Conditions: Time, inspections
  - Effects: Repairs, trigger other modules
- Can be developed separate from the AFT.

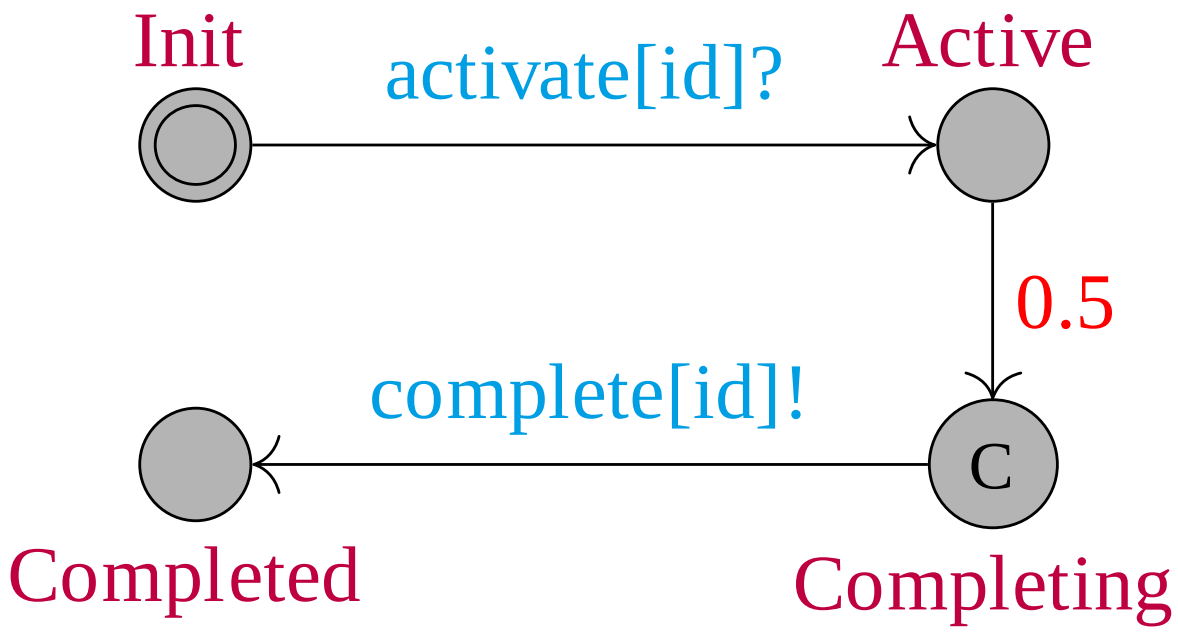
MODEL TRANSFORMATIONS

- Transform to & from existing tools.
- Automated selection of transformations.
- Preserve semantics whenever possible.
- Uses the Epsilon framework.



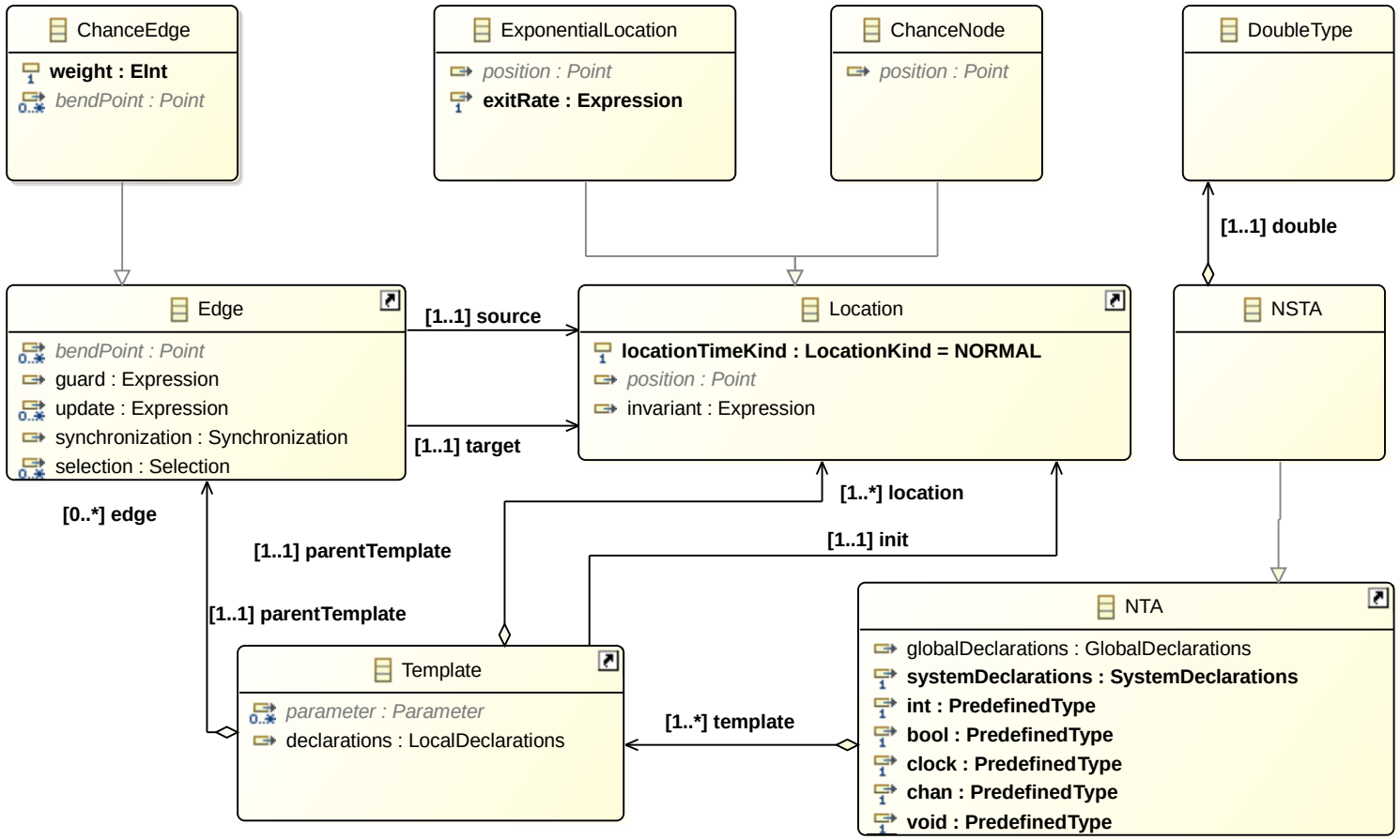
ANALYSIS VIA TIMED AUTOMATA

- Translate to UPPAAL model for analysis.
- Support for models with features from different formalisms.
- Textual queries can be automatically executed.
- Based on metamodel from the University of Paderborn with our own extensions.



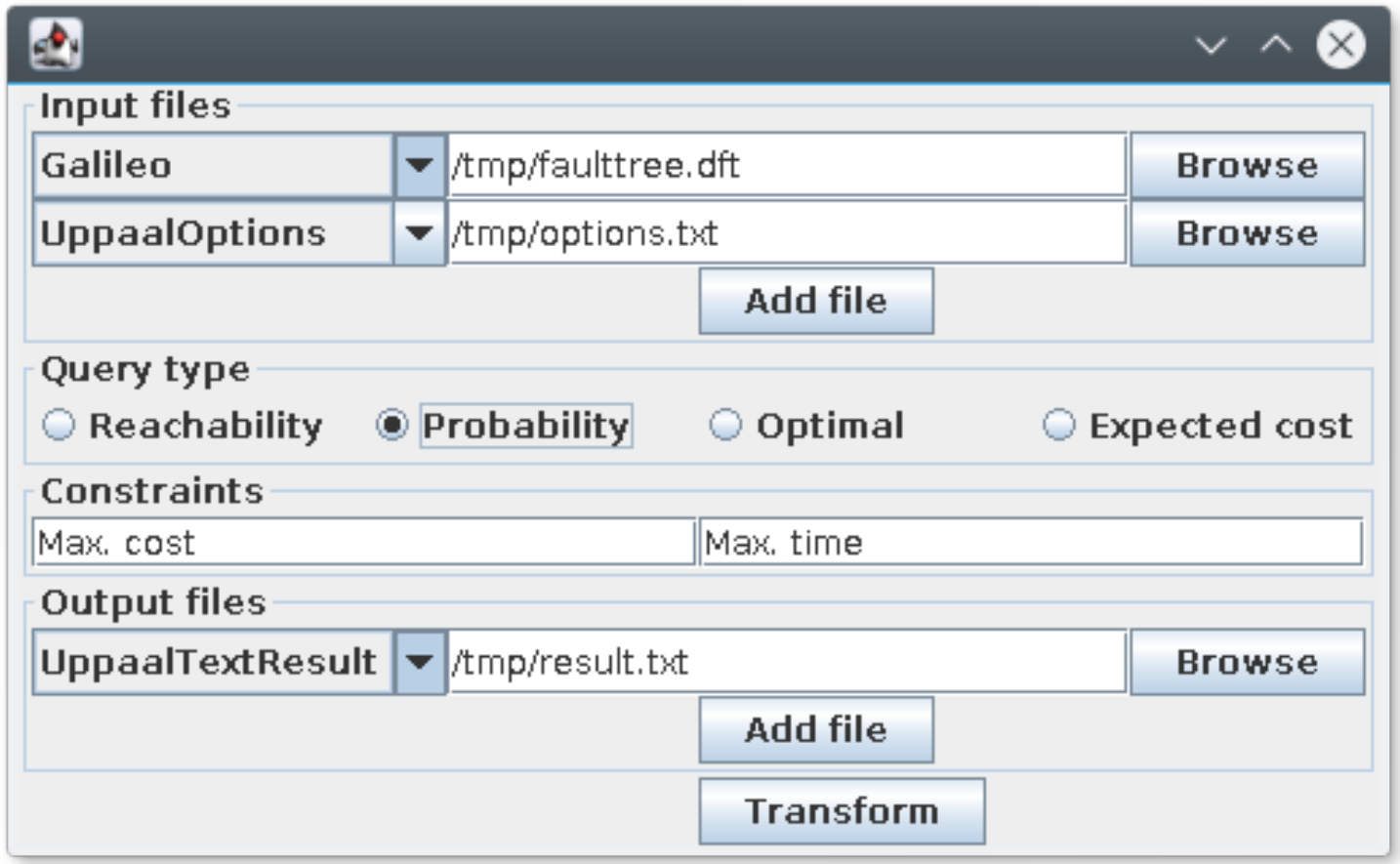
Timed automaton for basic event with exponentially distributed failure time.

UPPAAL METAMODEL



- Adds SMC extensions to Uppaal metamodel:
  - Double type
  - Locations with exponential rates
  - Branch nodes
- Related extensions to XML output.

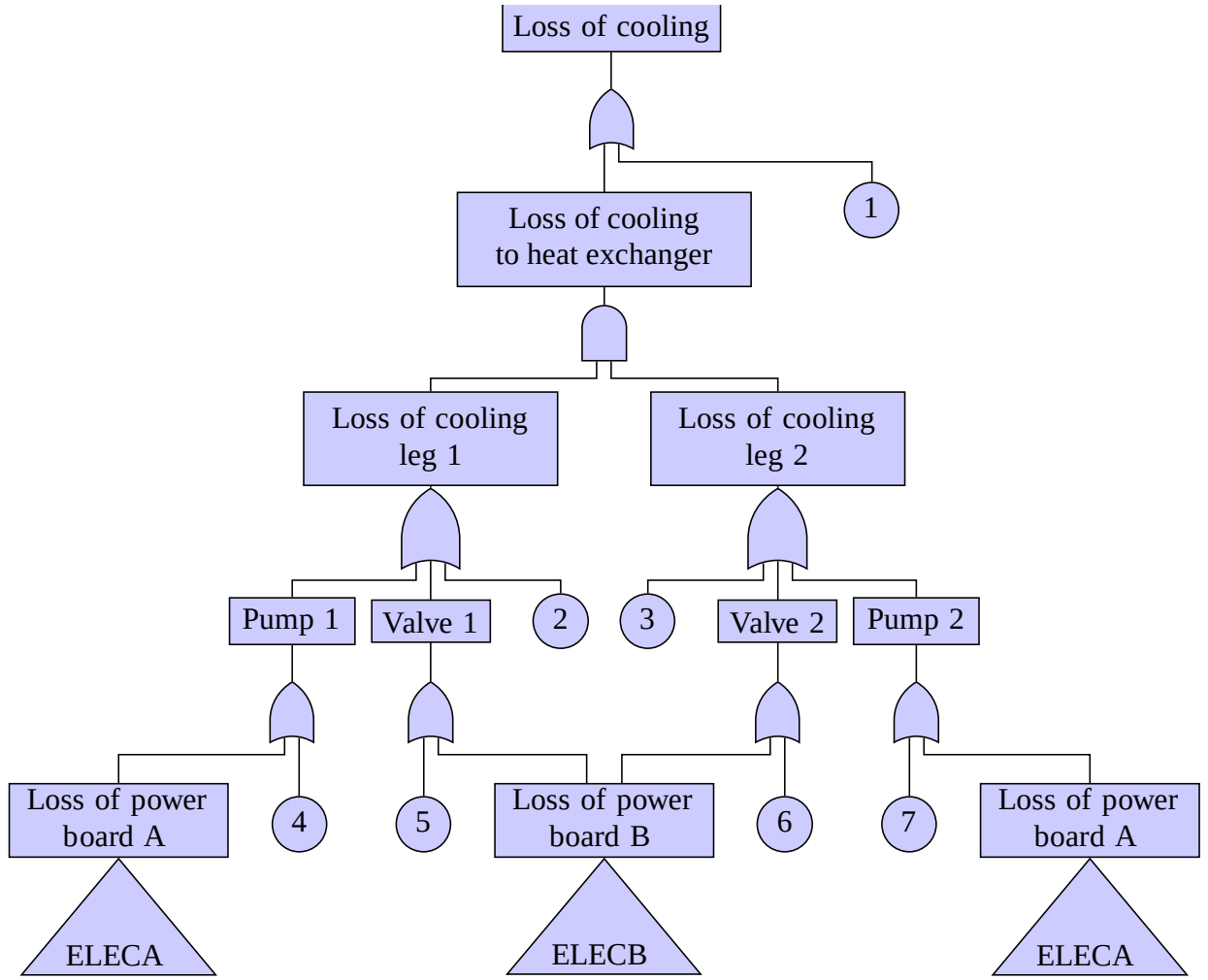
TOOL



- Push-button transformation between tools
- GUI for basic Uppaal queries
  - Support for arbitrary queries in textual form
- CLI for integration in tool chains

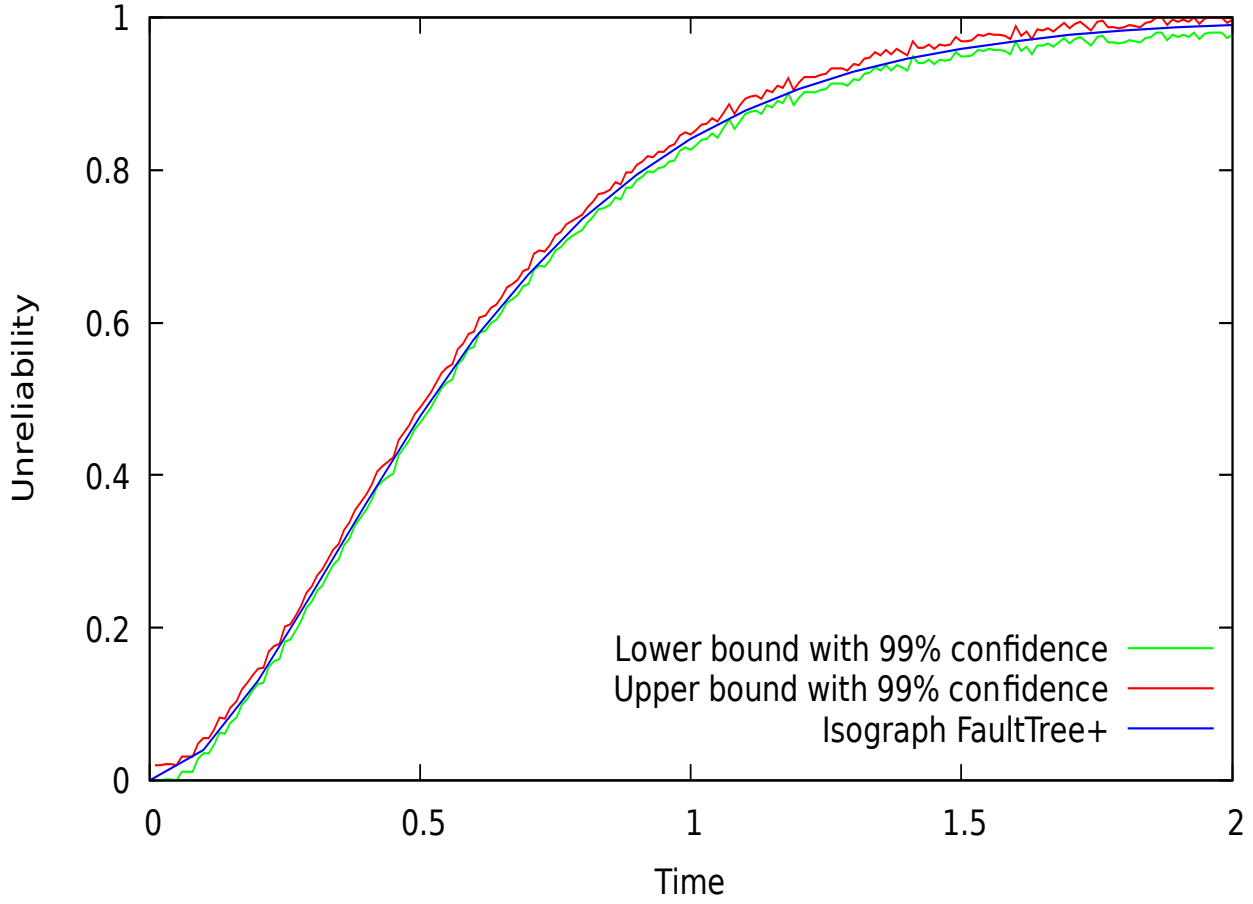
CASE STUDY 1: FAULT TREE

- Example taken from Isograph FaultTree+.
- Models a cooling system with redundant pumps and power supply.
- Unchanged from FaultTree+, except that repairs have been removed.



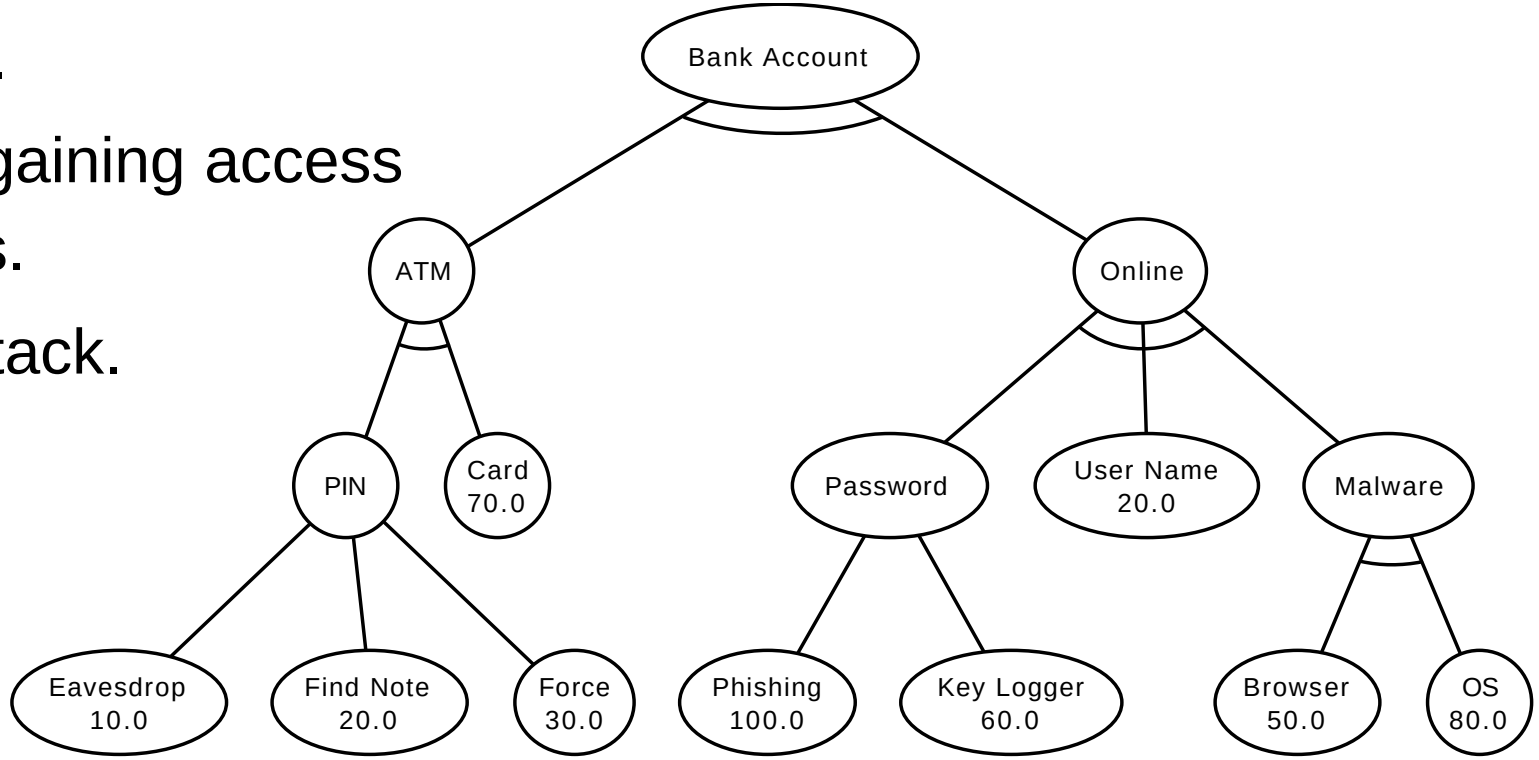
CASE STUDY 1: RESULTS

- Analyzed with FaultTree+, UPPAAL, and DFTCalc.
- Conversion time negligible.
- UPPAAL analysis: 5 minutes.
- 99% confidence interval of width 1%.
- FaultTree+ and DFTCalc produced identical results.



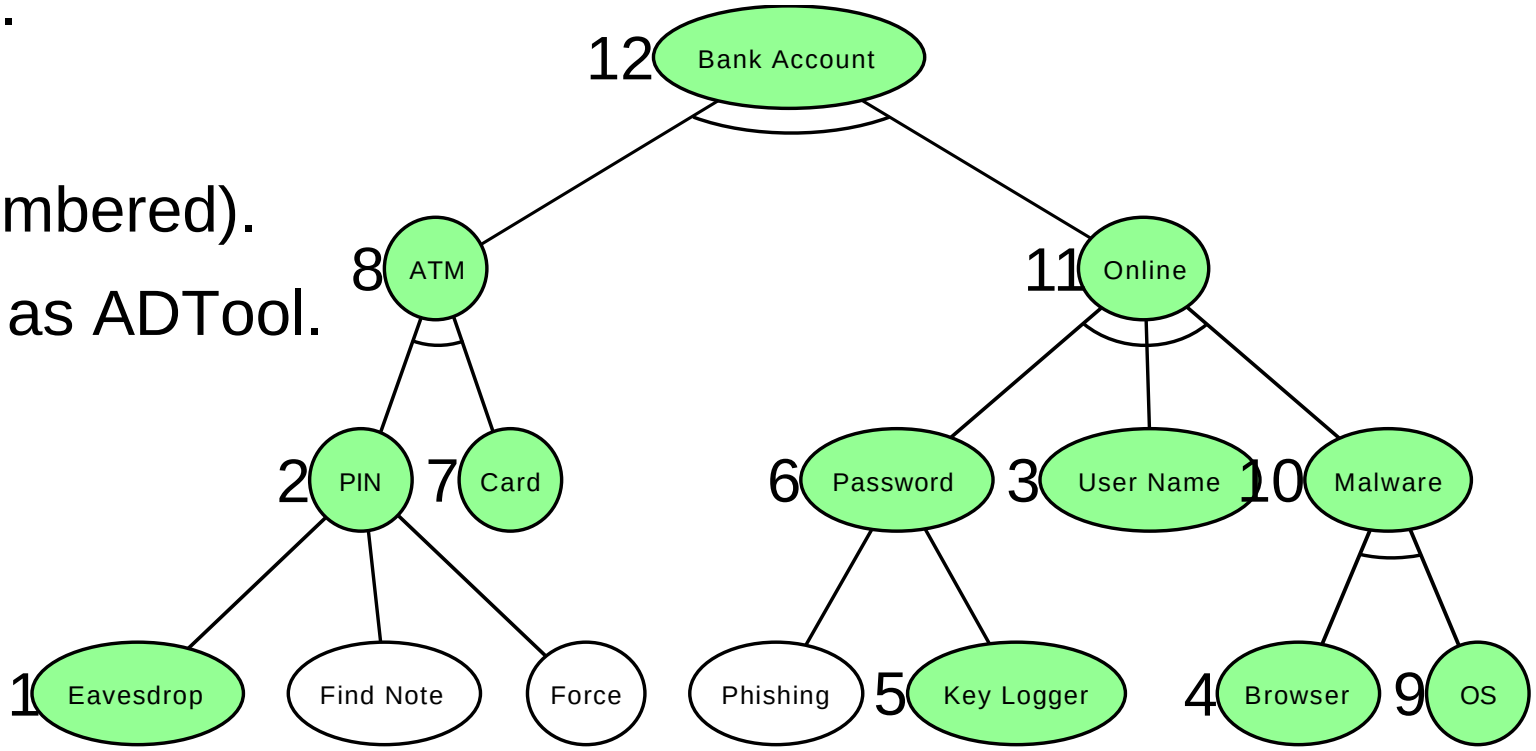
CASE STUDY 2: ATTACK TREE

- Input from ADTool.
- Model of attacker gaining access to a bank accounts.
- Values: Time to attack.



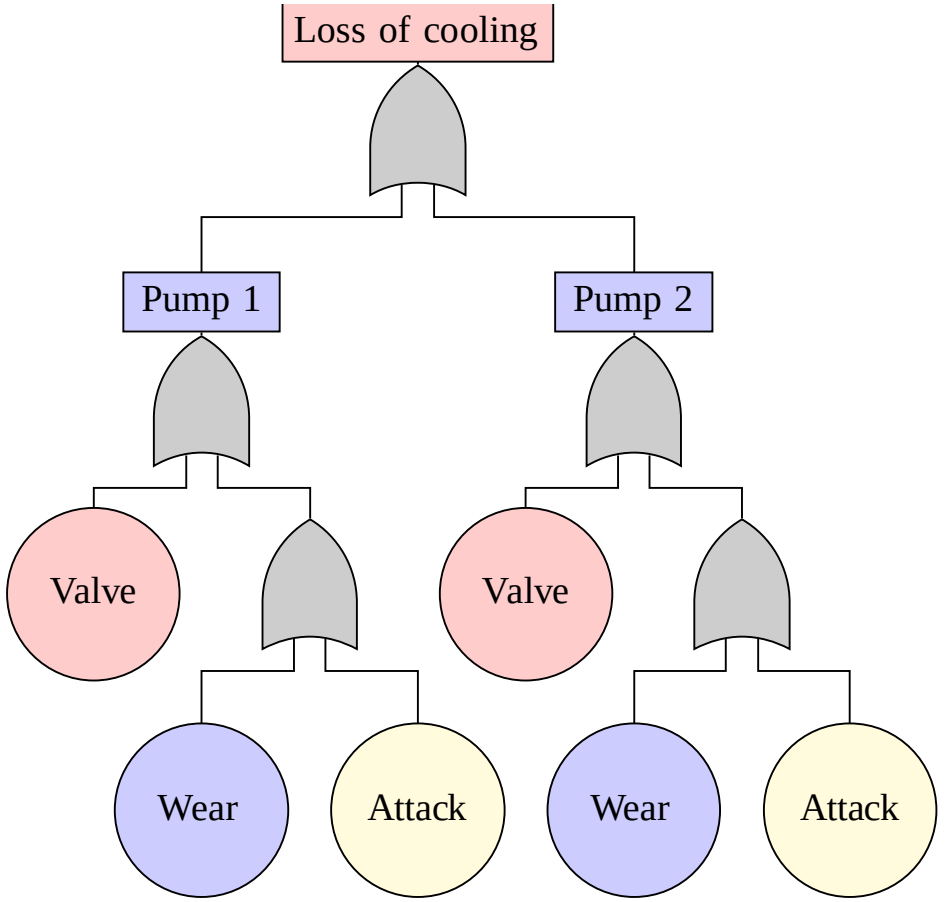
CASE STUDY 2: RESULTS

- Analysis via UPPAAL.
- Result: fastest attack
  - (green steps as numbered).
- Time for attack same as ADTool.



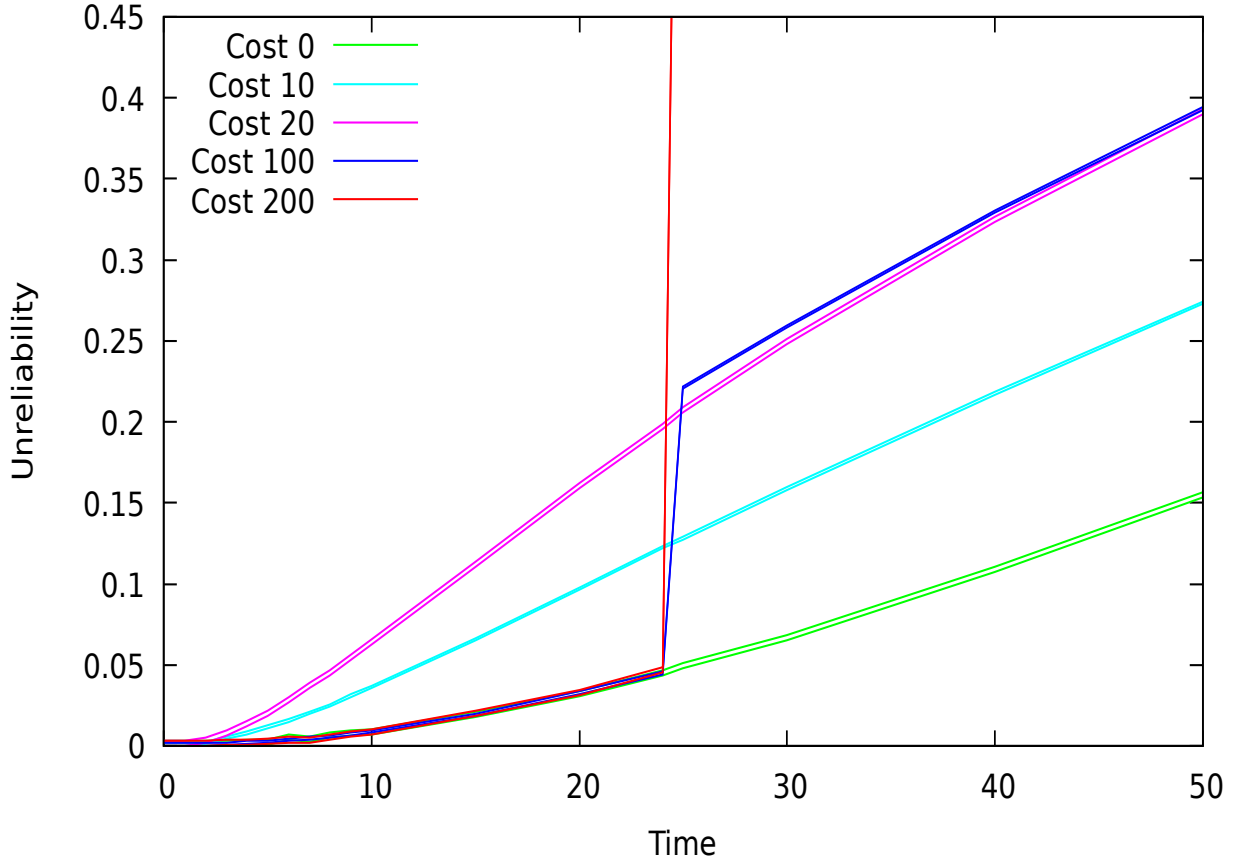
CASE STUDY 3: COMBINED ATTACK-FAULT TREE

- Model: Cooling with two pumps and associated valves.
- Pumps can fail on their own.
- Attacker options:
  - Accelerate pump failure (cost 10)
  - Cause valve failure at time 25 (cost 100)



CASE STUDY 3: COMBINED ATTACK-FAULT TREE

- Analysis of unreliablity in UPPAAL (99% confidence shown)
- Regular failure rate 15% after 50 time units.
- Attacker can force failure at time 25 for cost 200.
- Forcing one valve failure (cost 100) no more effective than accelerating two pumps (cost 20).



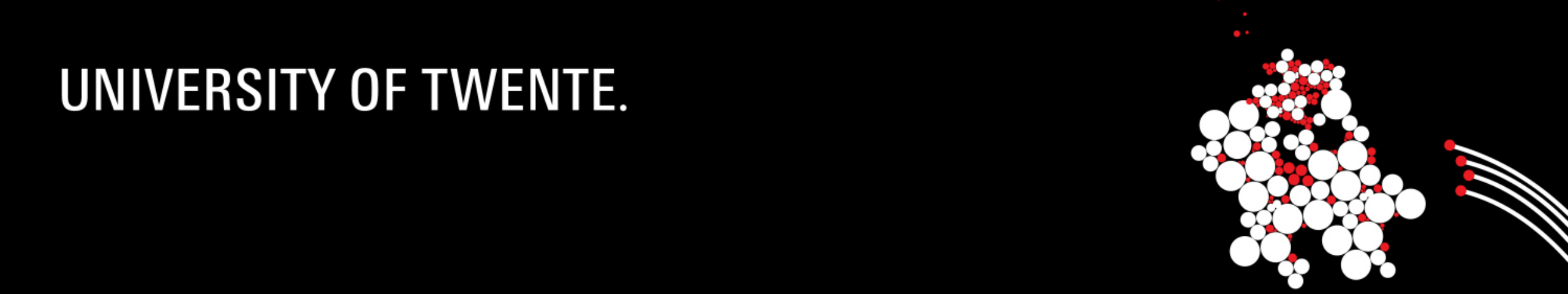
CONCLUSIONS

- Framework unifies and integrates different fault tree and attack tree formalisms.
- Include maintenance in fault/attack trees.
- Extensible to new formalisms.
- We support analysis of combined models.
  - Reliability, cost, MTTF, you name it.
- Tool support for lay users.

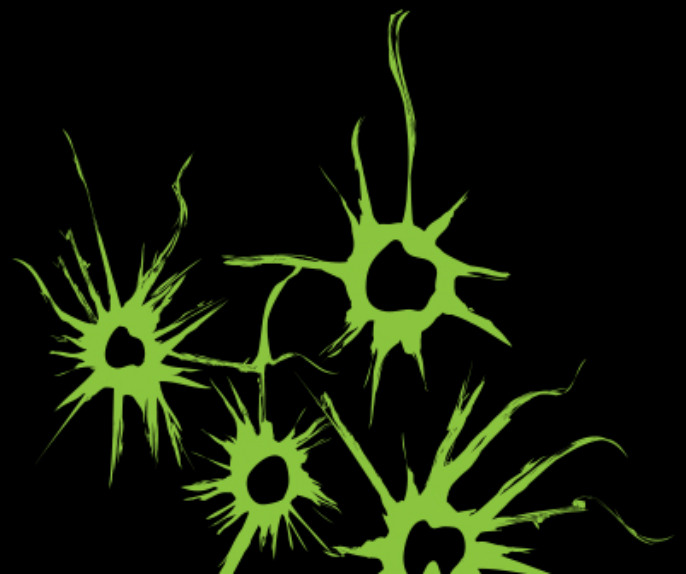
RESEARCH CHALLENGES

- More advanced automatic query generation.
- Support for better outputs (esp. plots, augmented trees)
- Compatibility between features (e.g. maintenance and sequential gates).
- GUI for inputting trees.
- More output for analysis
  - JANI, rare-event simulation, etc.

UNIVERSITY OF TWENTE.



AND NOW: SLIGHTLY DIFFERENT BUT RELATED

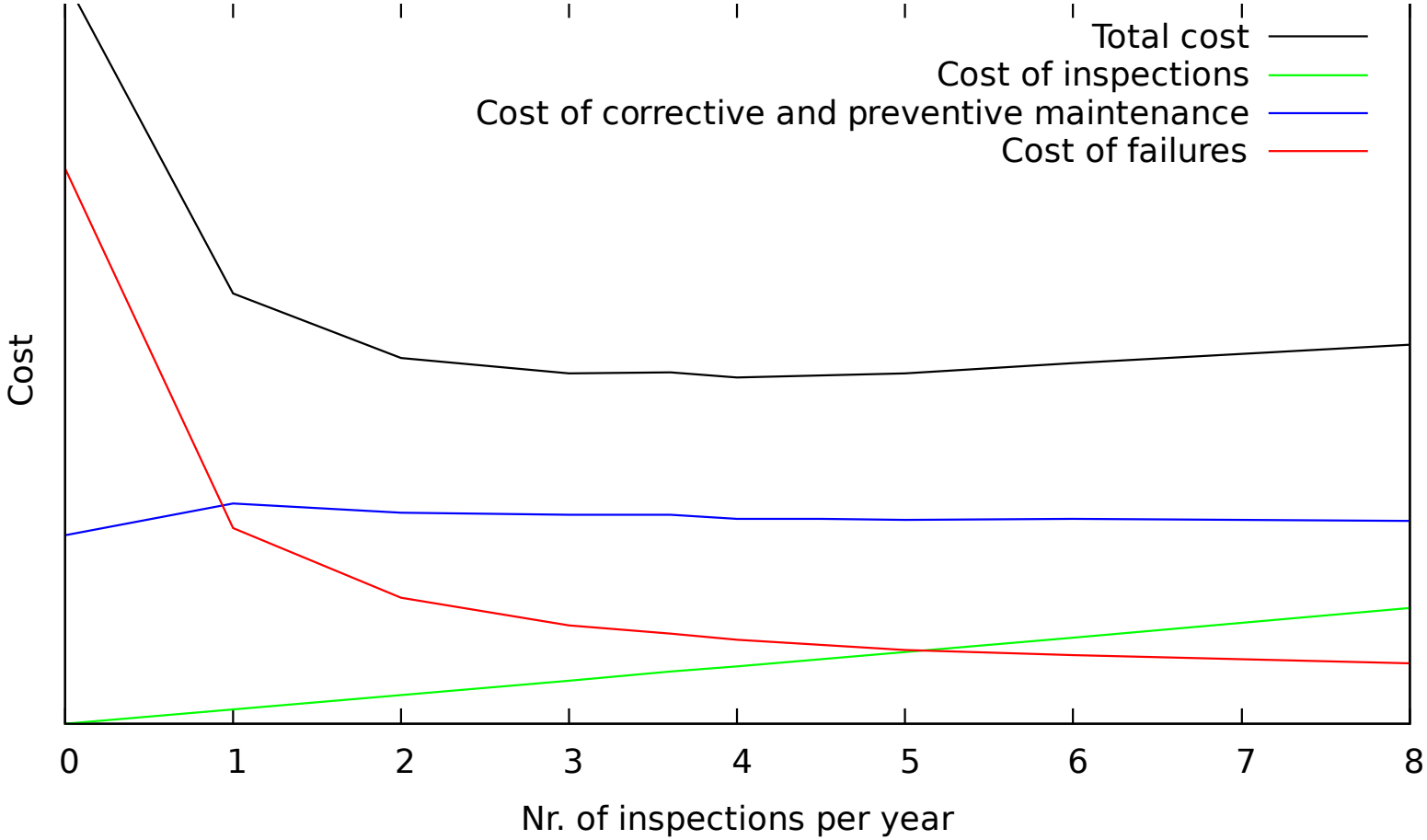


EI-JOINT

ProRail

- Case study for fault maintenance trees.
- Collaboration with ProRail
- 50.000 installed in The Netherlands
- Relatively frequent cause of disruptions

PREVIOUS RESULTS EI-JOINT

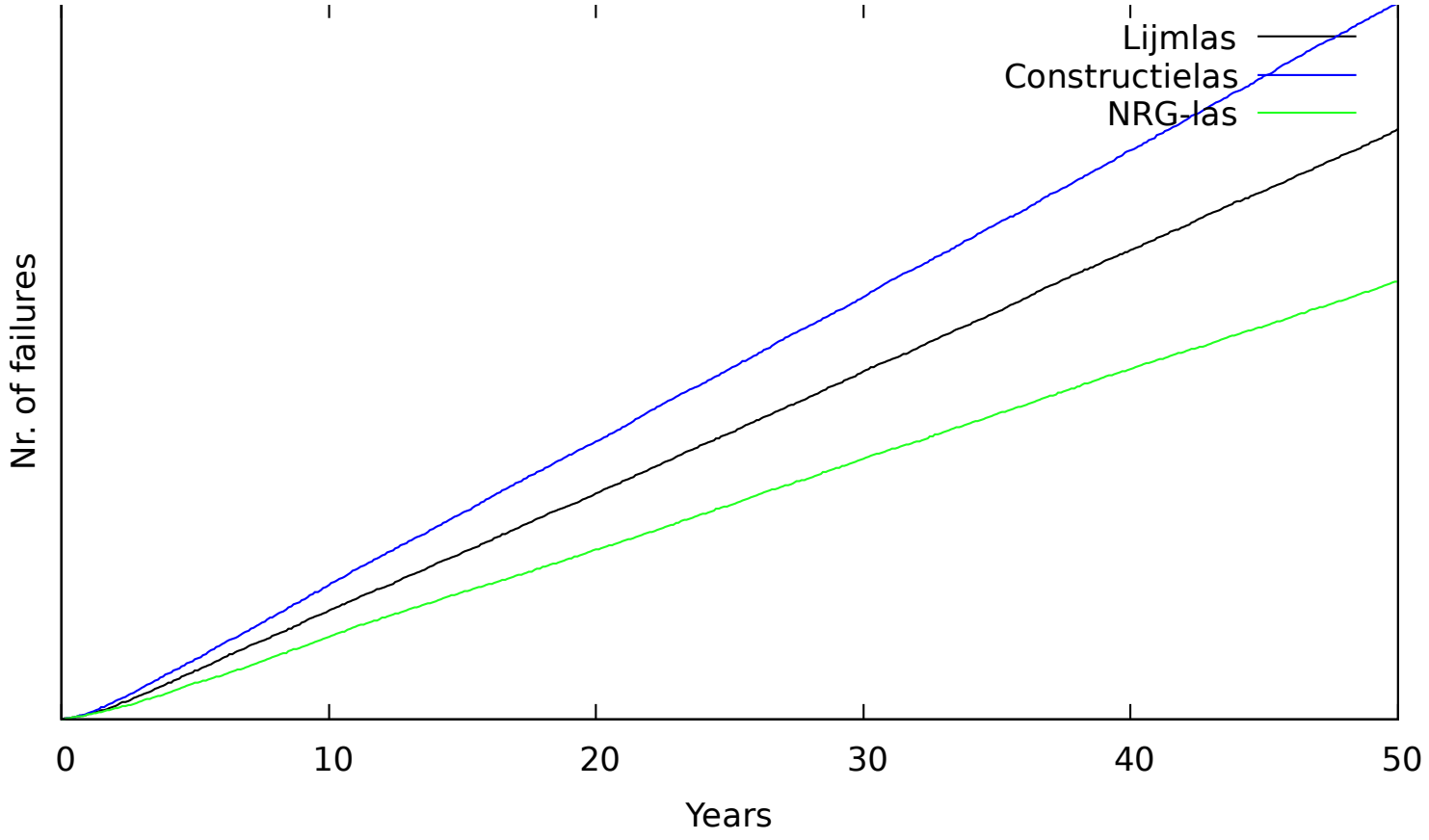


- Failure causes
- Reliability
- Cost breakdown
- Effect of inspections on cost

NRG-JOINT

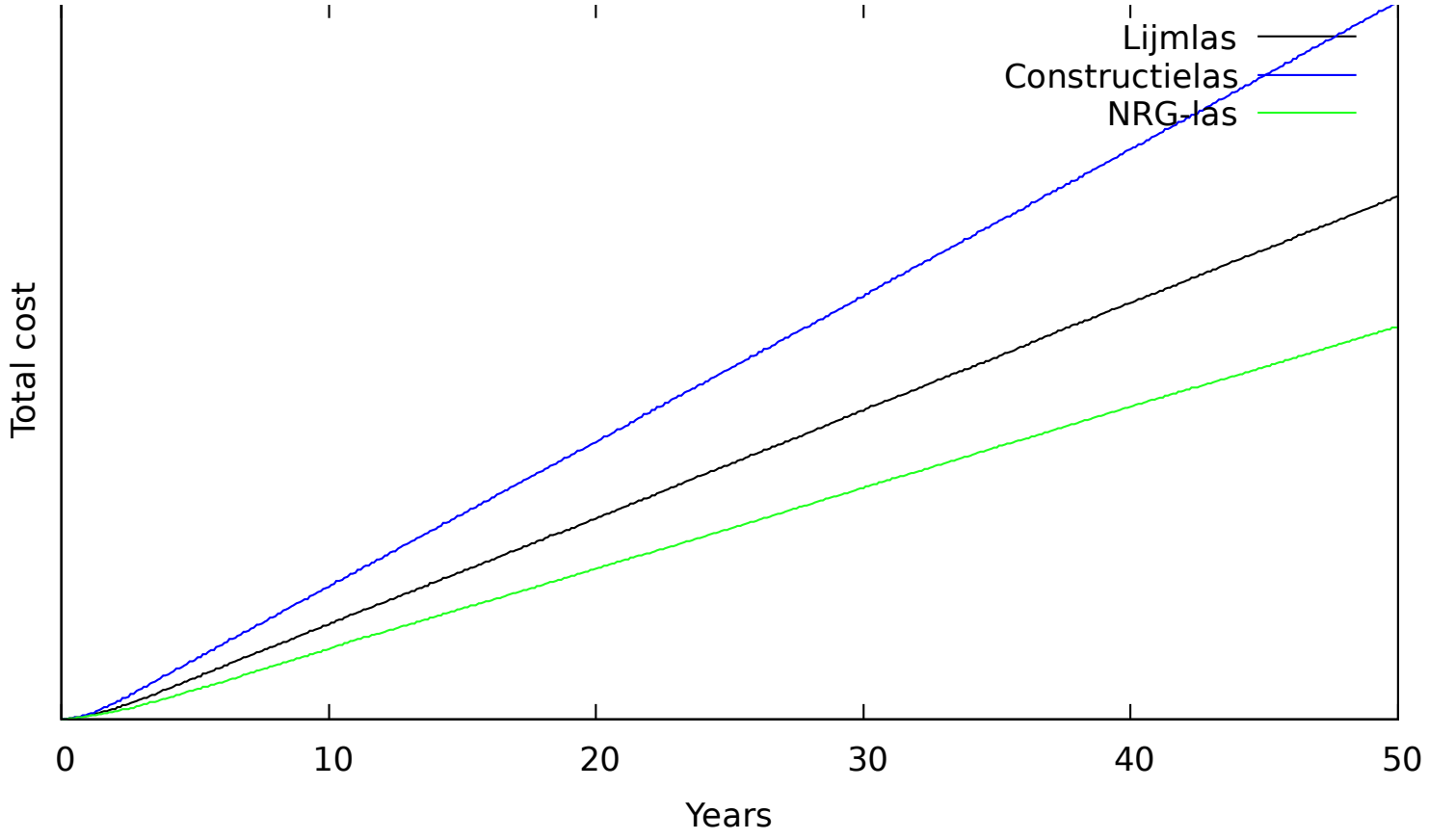
- New and improved joint developed for ProRail.
- Longer plates attaching to track.
- Six bolts instead of four.
- Bolts repositioned to reduce stress.
- Does not need to be installed on top of double sleeper.
- More reliable.
- More expensive.

RESULTS NRG-JOINT

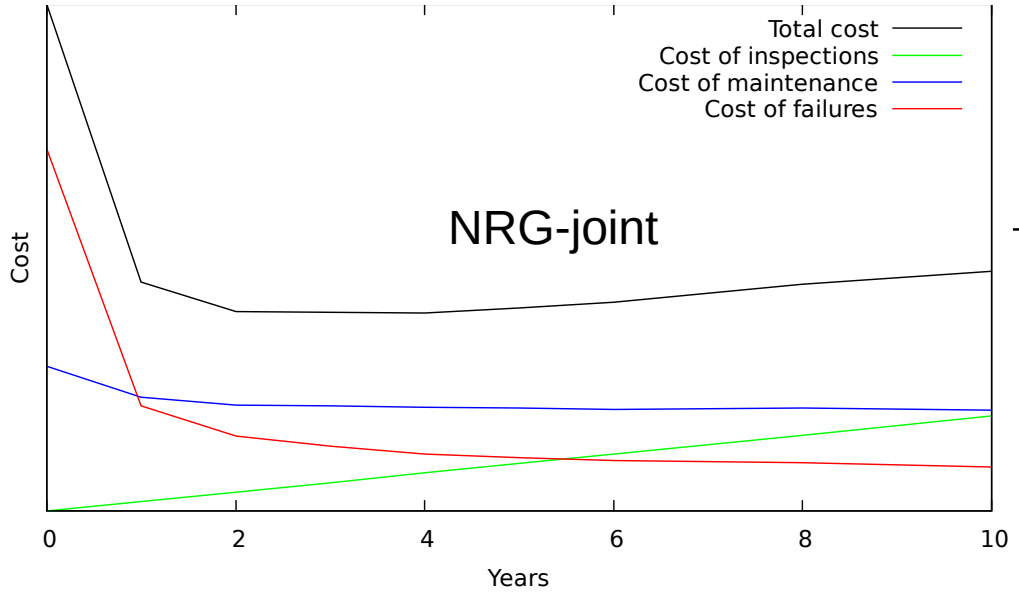


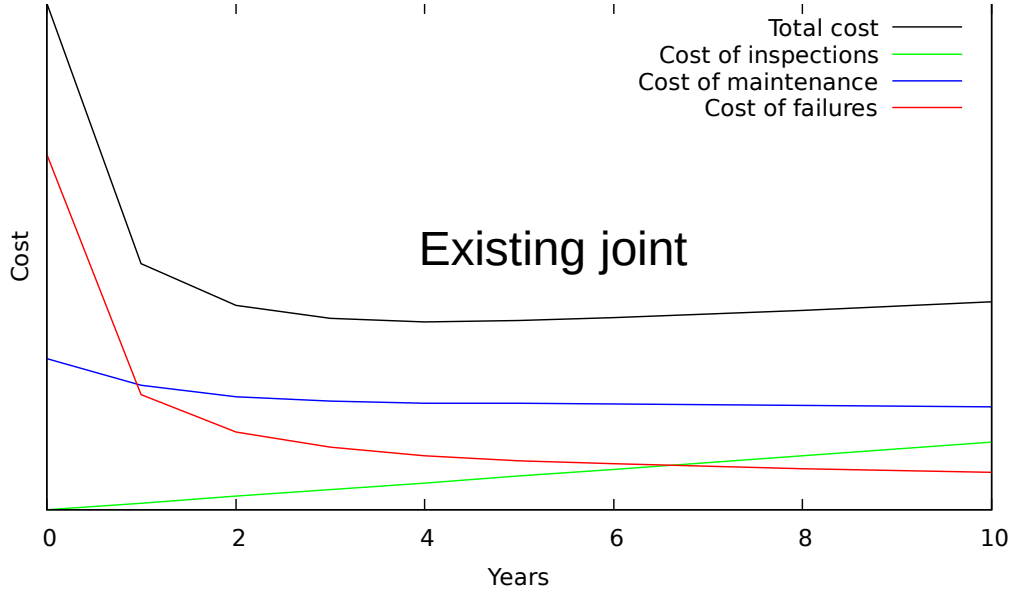
- Nr. of failures over time for three types of joint:
  - Glued (previous case)
  - Constructed in situ
  - NRG (new)
- NRG-joint has significantly fewer failures (at same maintenance policy).

RESULTS NRG-JOINT



- (Yes, it is a diffent image)
- Substantial cost reduction post-installation.
- Analysis used to choose deployment strategy:
  - Immediate replacement
  - Replace worn out joints





RESULTS NRG-JOINT

- Comparing maintenance strategies:
  - Lower cost for existing strategy (as previous slide).
  - More sensitive to variations on maintenance policy.

THANK YOU!