Model Based Safety Analysis

Werner Damm & Thomas Peikenkamp

R&D Division of
Safety Critical Systems

OFFIS

Fachbereich Informatik
Abt. Sicherheitskritische
Eingebettete Systeme

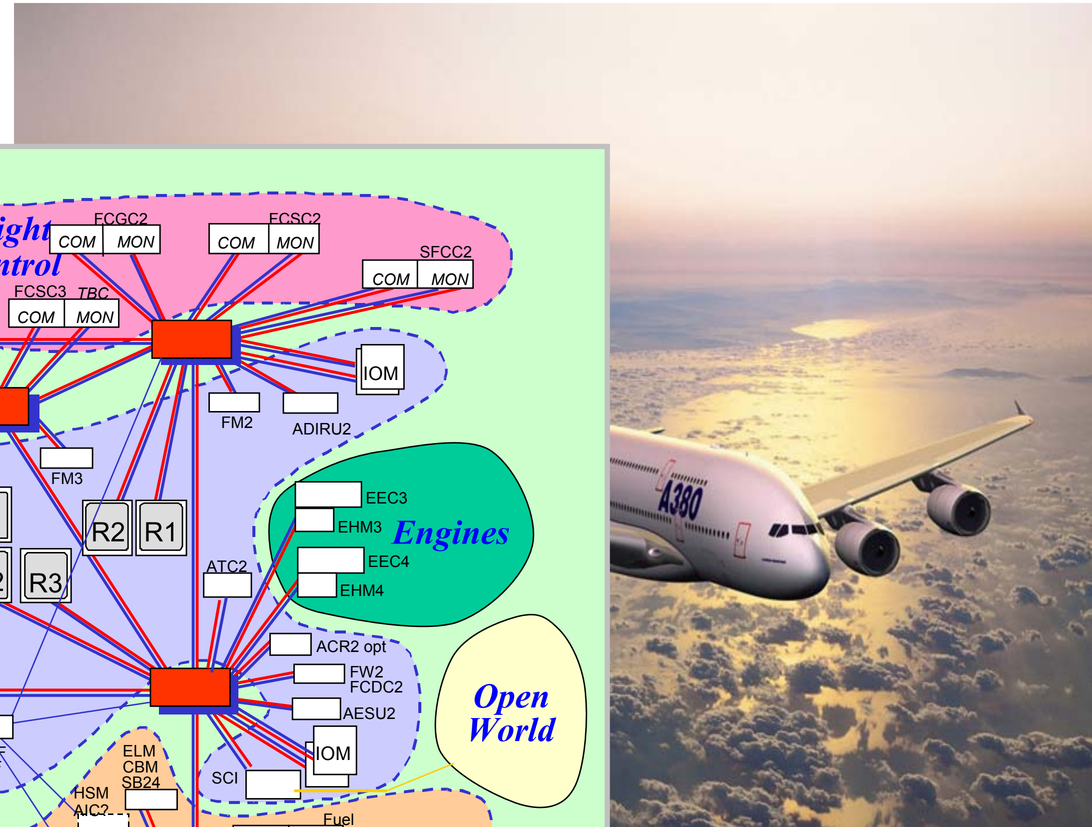
Carl von Ossietzky Universität Oldenburg

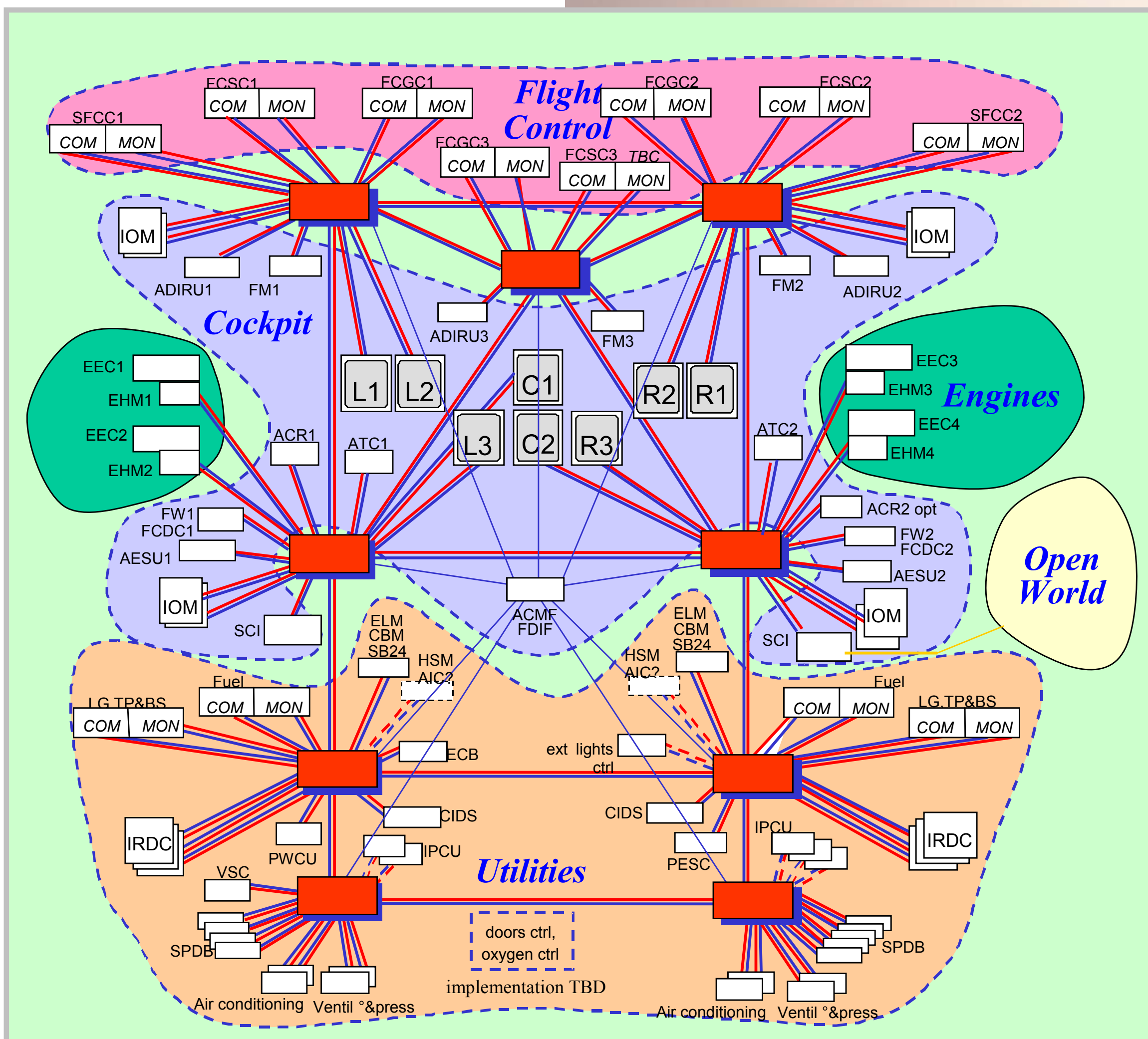
Structure of Presentation

- Introduction
- Model Based Development
- Safety Analysis Process
- Model Based Safety Analysis
- Conclusion

Introduction

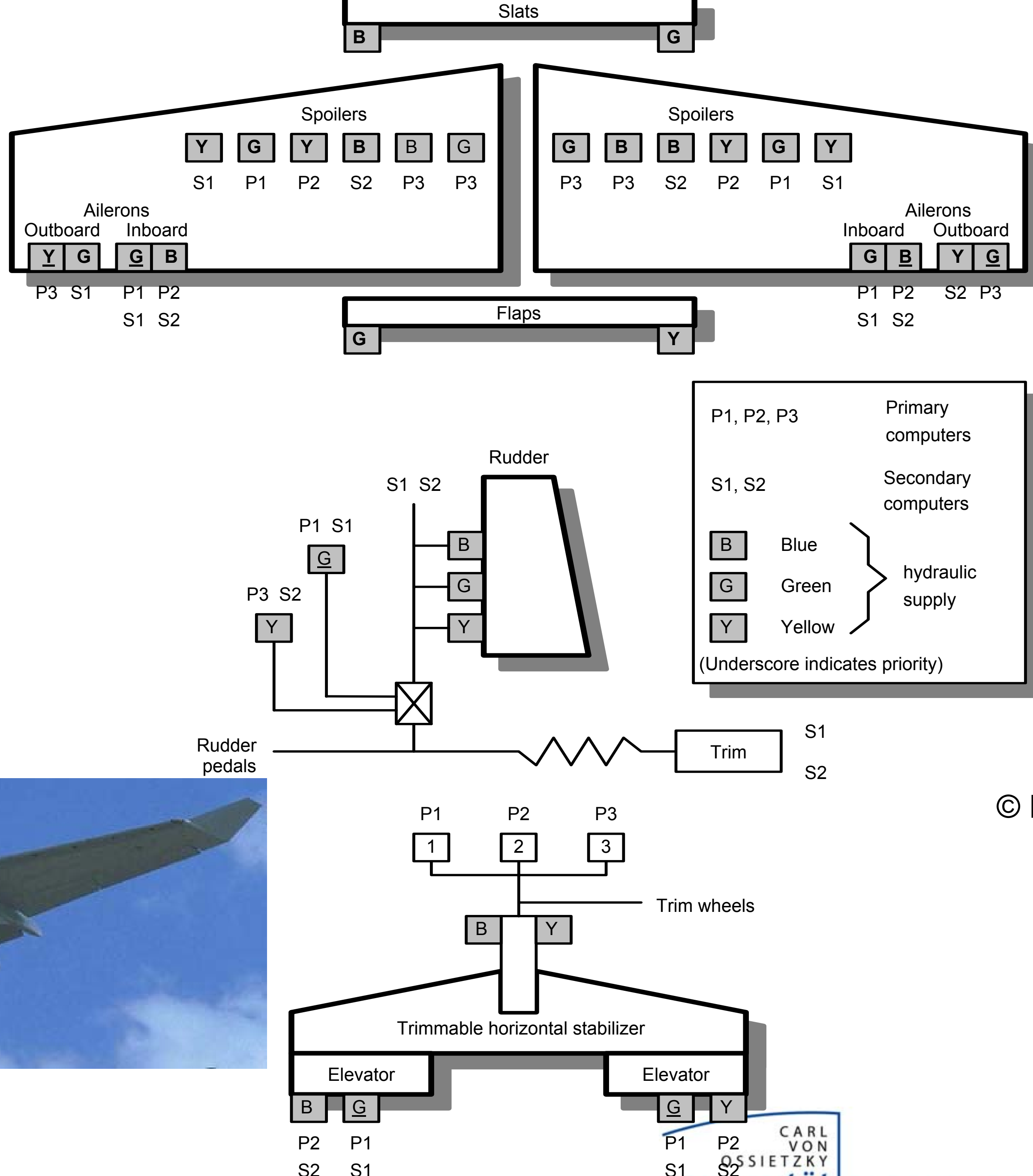
A380



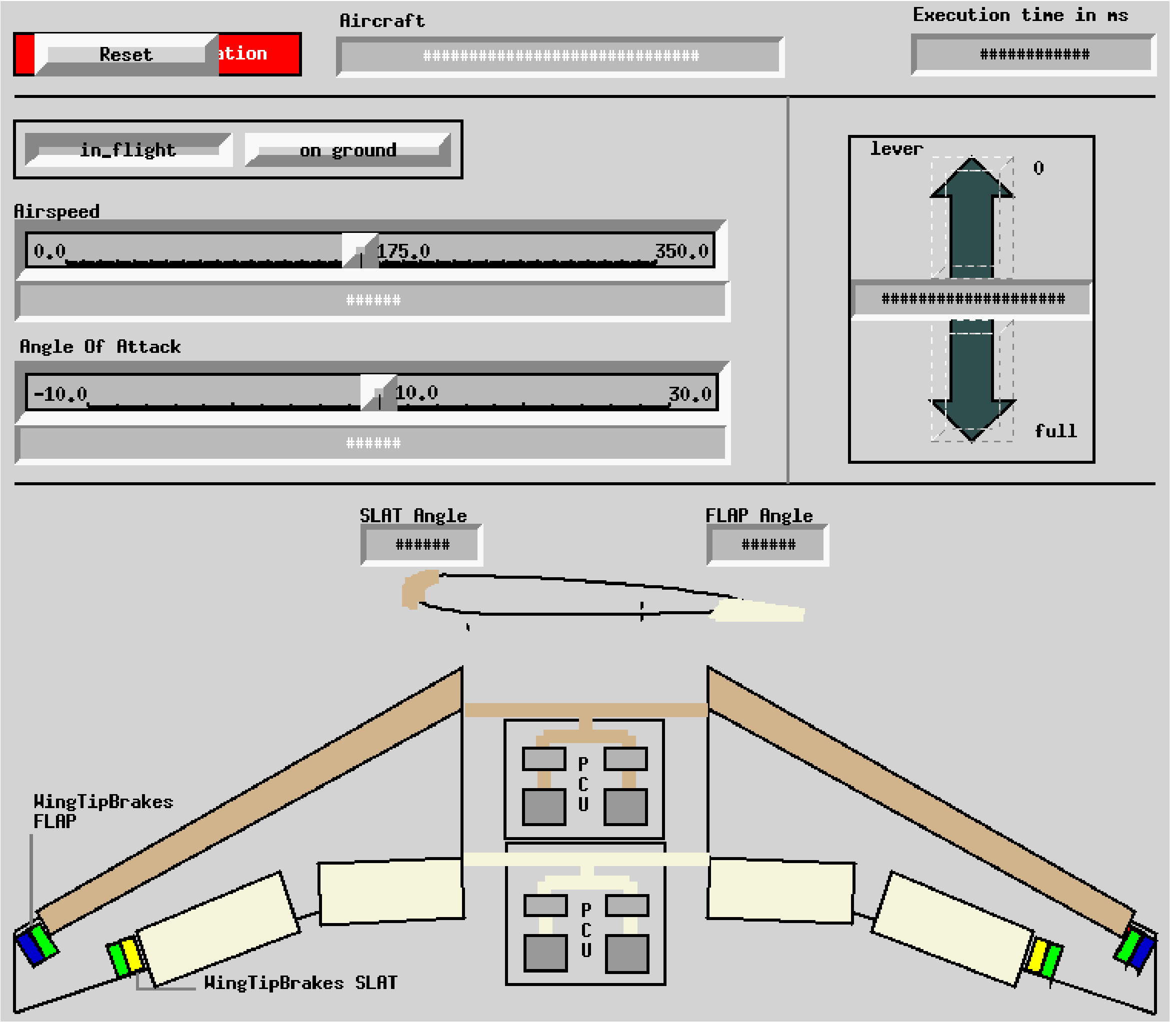


A380: about 100 functions realized in SW, total code size ~65 MB

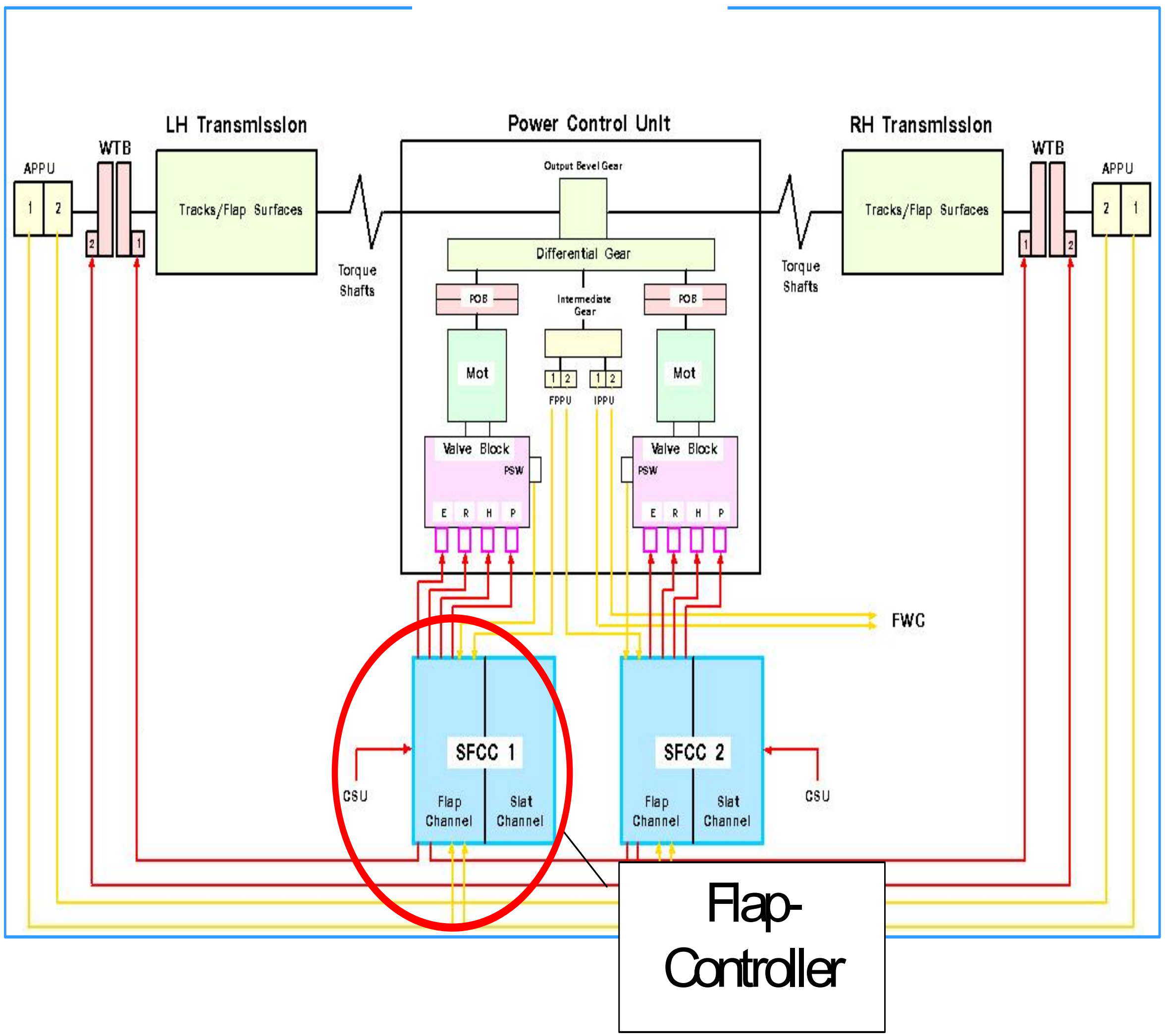
The high-lift system



Sample application: Slat and Flap system



System Architecture



Auto-Flap function

- Automatically adjusts flap position according to load

Critical Events

- Asymmetric Flap Position
- Powered runaway
- Inadvertent Flap Retract due to Auto-Flap Function

Extensive Monitoring

- To detect critical events

Model Based Development of Avionics Applications

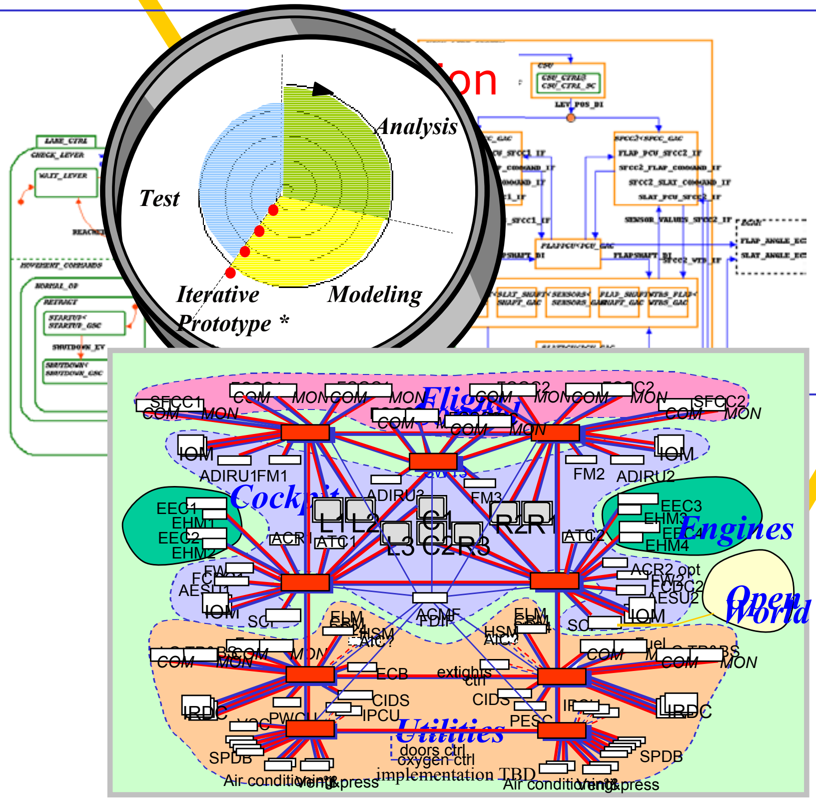
Model based Development Process

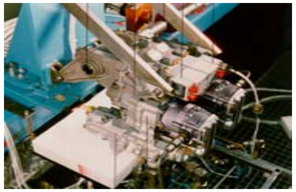
Requirement

"For the current flaps setting, CAS shall not exceed VF."

Aircraft level

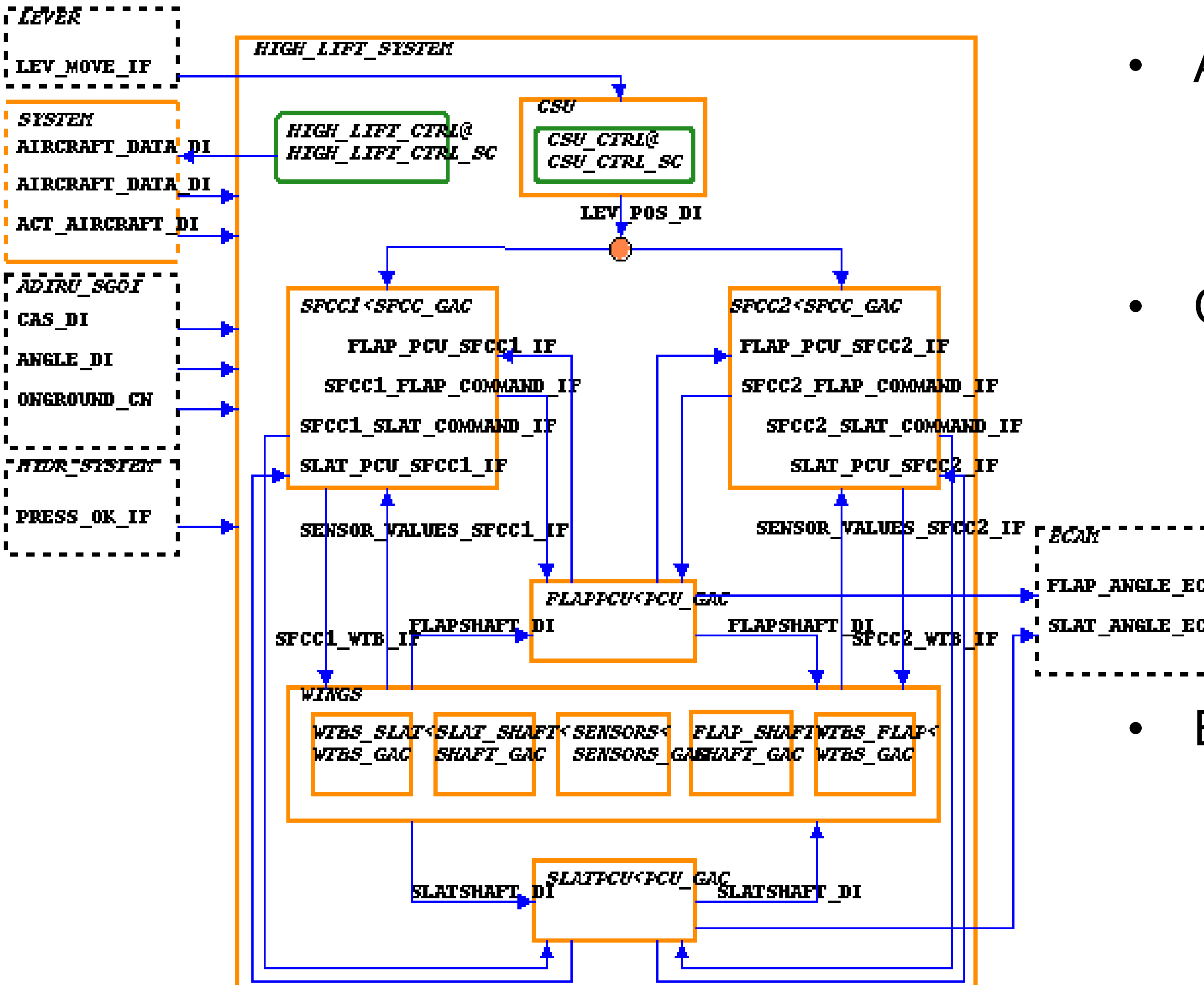
System level





Equipment level

The Statemate Model



- Auto-Flap function
  - Automatically adjusts flap position according to load
- Critical Events
  - Asymmetric Flap Position
  - Powered runaway
  - Inadvertent Flap Retract due to Auto-Flap Function
- Extensive Monitoring
  - To detect critical events

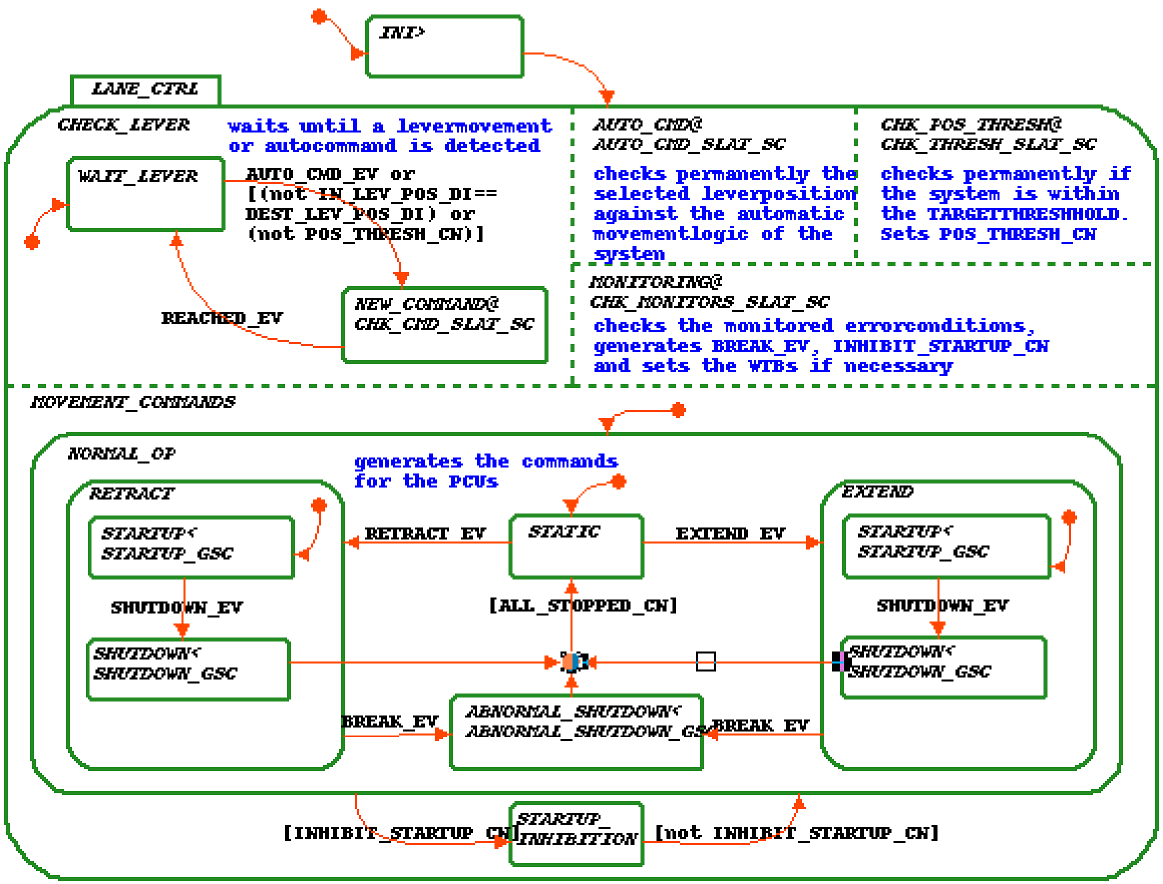
STATEMATE

- Industry standard case tool marketed by I-Logix Inc
- Activity Charts
  - System Architecture
  - Information Flow
  - Environment
- State Charts
  - visual real-time programming language
  - hierarchy
  - orthogonal states
  - algorithms

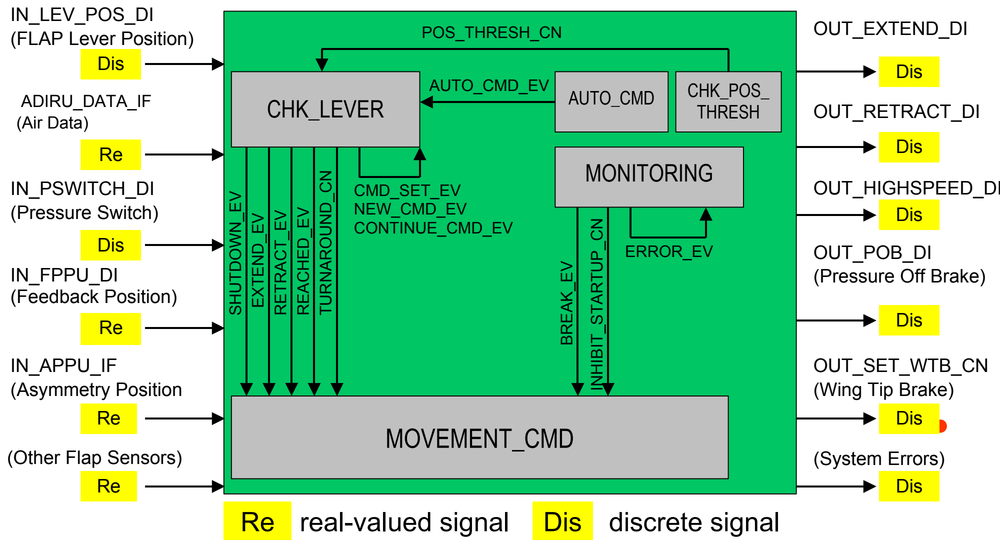
- Animation
- Simulation
- RP code generation
- documentation

Formal published semantics
Damm, Pnueli, … 98

A sample StateChart



Example: Model characteristics



- Static measures
  - 30 charts, most instances of generic charts
  - 164 data-items (mostly floats)
  - 38 conditions, 12 events
  - Arrays, records, user defined types
  - 7 timers
- Explicit representation as flat finite state machine would require 35 000 states
- Exhaustive testing would require to cover 2^{75} possible input values in each step

Verification of Safety Requirements

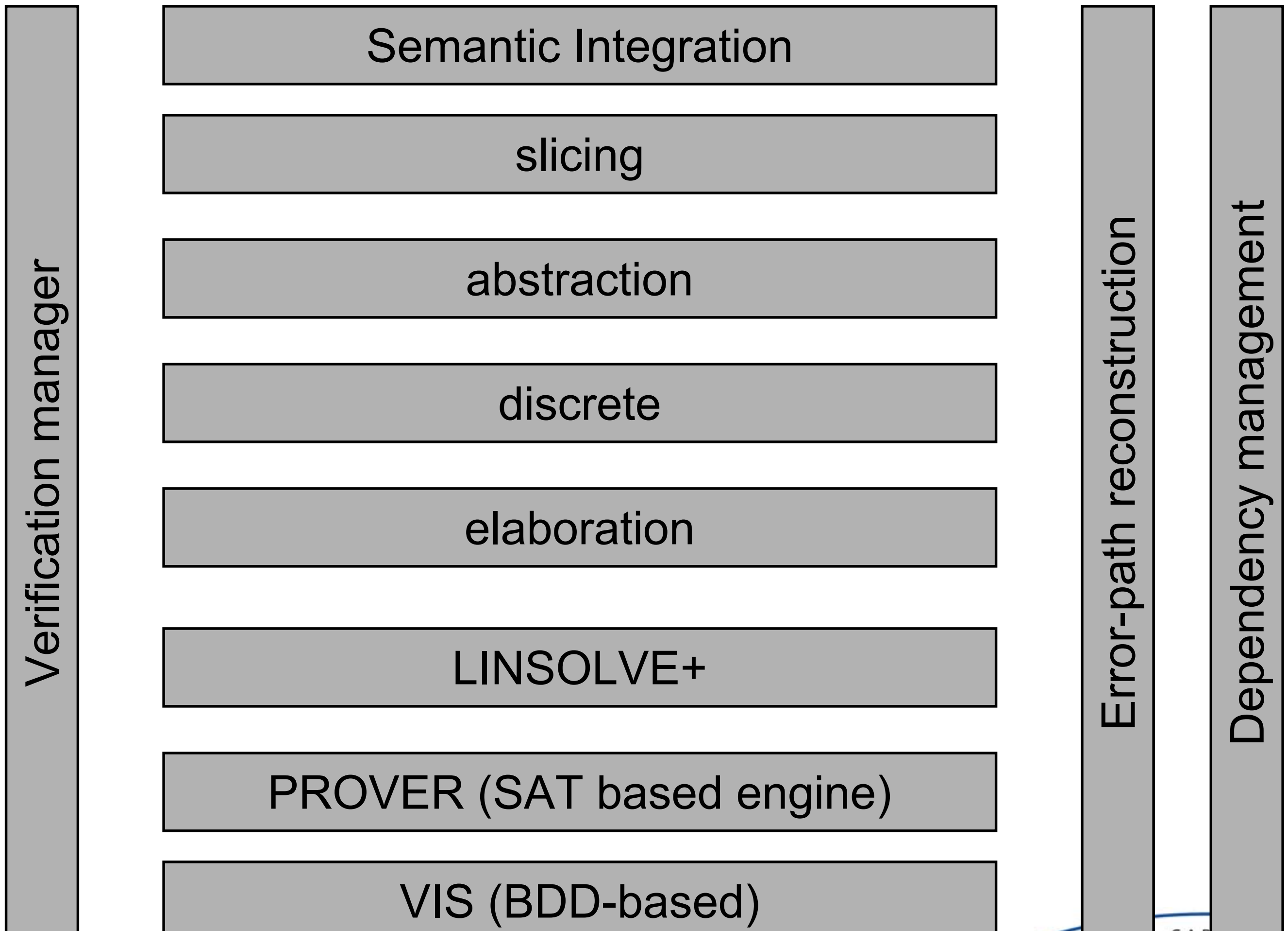
- A typical aircraft level safety requirement related to the High-Lift System:

  "For the current flaps setting, CAS shall not exceed VF."

  - CAS : Calibrated Air Speed
  - VF : maximum allowed speed for a given flaps position + 7 knots.

Verification Environment

ASCET – Matlab/Simulink-Stateflow – Scade -Statemate - UML

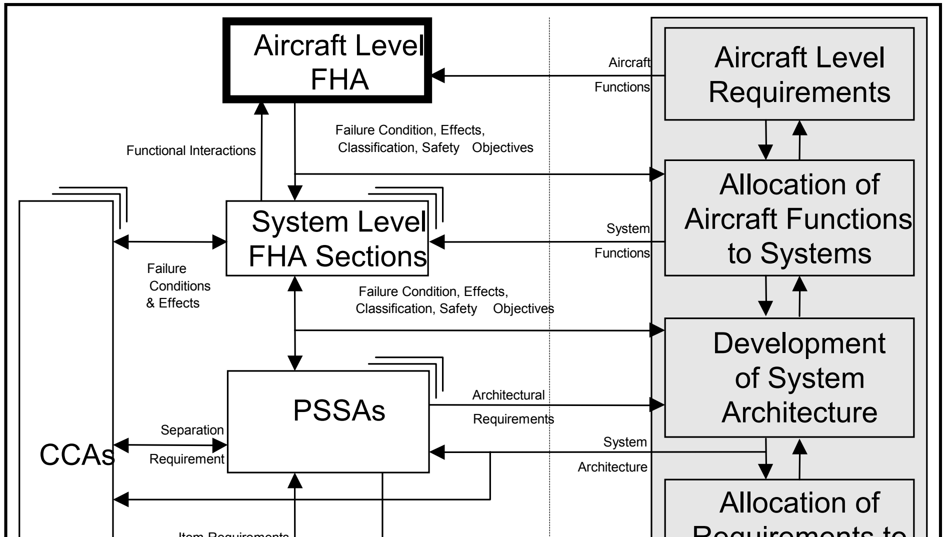


Results

- Verification of full scale ECU models
  - Dealing with complex types (reals, arrays, ...)
  - Dealing with real-time (counters, watchdogs, ..)
  - Dealing with extremely large designs (e.g. a full autopilot)
  - Dealing with the full range of modeling constructs of COTS tools used in industrial practice
- Advances in verification technology
  - Tight integration of BDD, SAT, constraint solving, LP based engines
  - Range of automatic abstraction techniques, including predicate abstraction
  - Infinite state verification for unbounded object creation and real-valued models
- Advances in Formal Requirement Capture
  - Optimized Requirement representations through pattern libraries
  - Live Sequence Charts

See www.ses.informatik.uni-oldenburg.de

The safety analysis process



ARP 4754 and 4761

- Aircraft Recommended Practices
- De facto standard on involved processes

AIRCRAFT FUNCTIONAL HAZARD ASSESSMENT (FHA)

Aircraft function list:

Ex: control aircraft on ground

Aircraft Functional Failure assessment.

For each aircraft function analysis of effects in case of function single failure and in case of failure combination

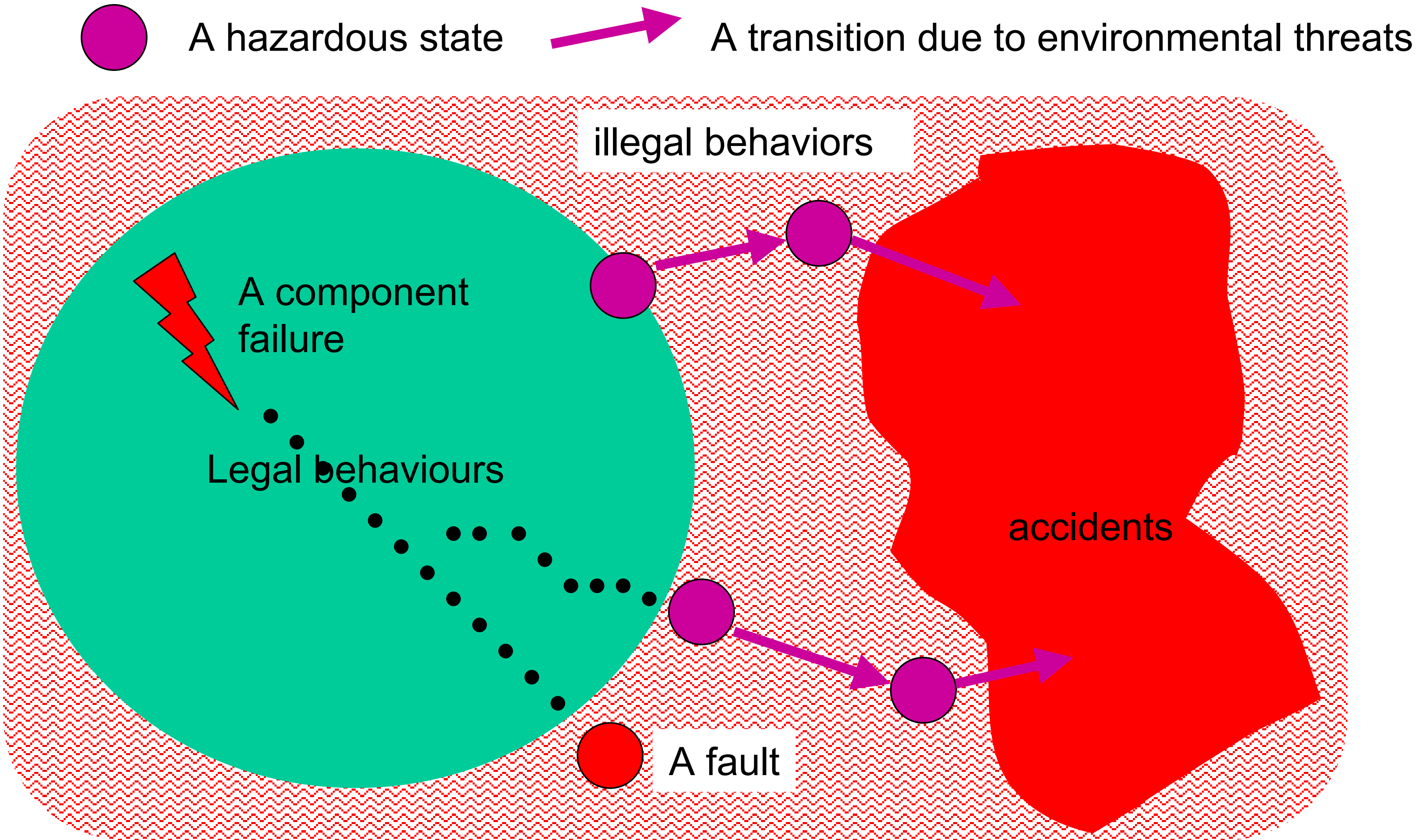
- Functional failure effects
- Detection
- Crew actions
- Effects classification
- Associated significant Failure Condition
- Justification materials
- Qual. And quant. Objectives and requirements

14.09.1993 -
Aircraft thought it was still airborne, because only two tons weight lasted on the wheels due to a strong side wind and the landing maneuver. The computer did not allow braking.
The plane ran over the runway into a rampart.

Causes - official report

Causes of the accident were incorrect decisions and actions of the flight crew taken in situation when the information about windshear at the approach to the runway was received. Windshear was produced by the front just passing the aerodrome; the front was accompanied by intensive variation of wind parameters as well as by heavy rain on the aerodrome itself. Actions of the flight crew were also affected by design features of the aircraft which limited the feasibility of applying available braking systems as well as by insufficient information in the aircraft operations manual (AOM) relating to the increase of the landing distance.

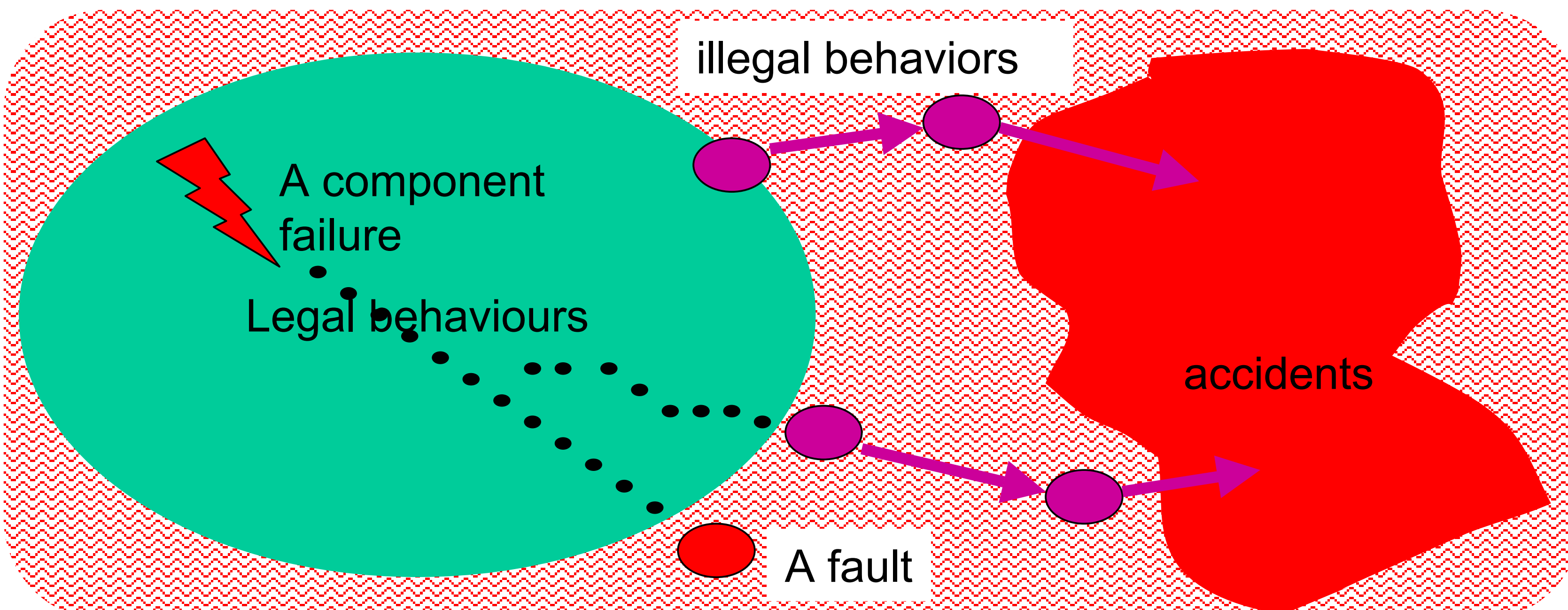
Faults, hazards, and accidents



Failure

- attribute of behavior of physical system/ component of system
- fails to perform under its intended function at a given period of time in spite of operating under specified constraints

- Distinction between
  - systemic failures
    - due to design errors
  - physical failures
    - due to e.g. Fabrication faults, EMC, wear-out, broken interconnect, stuck relays, ...
- Characterization of operating constraints crucial



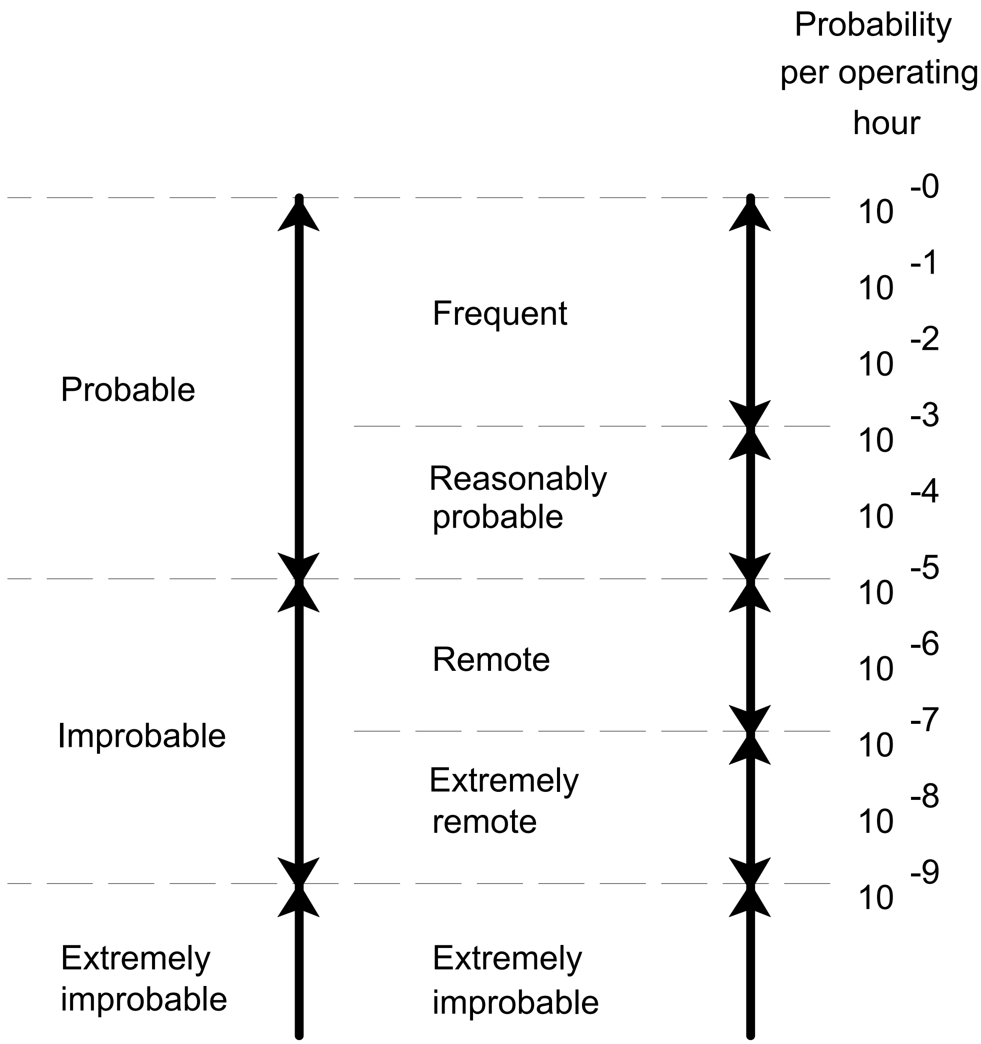
Typical Physical Failures

- Stuck-at
  - Value remains at constant level
- Ramp-down
  - Value gradually decreases to given constant level
- Random
  - Value stays at some randomly chosen value
- Noise
  - Value is randomly changed within given range around nominal value
- Delay
  - Value is transmitted with given delay
- Transient / Persistent
- Attached to design entities
  - Wires, links
  - Sensors
  - Actuators
  - Processors
  - …

Hazard Severity Categories for civil aircraft

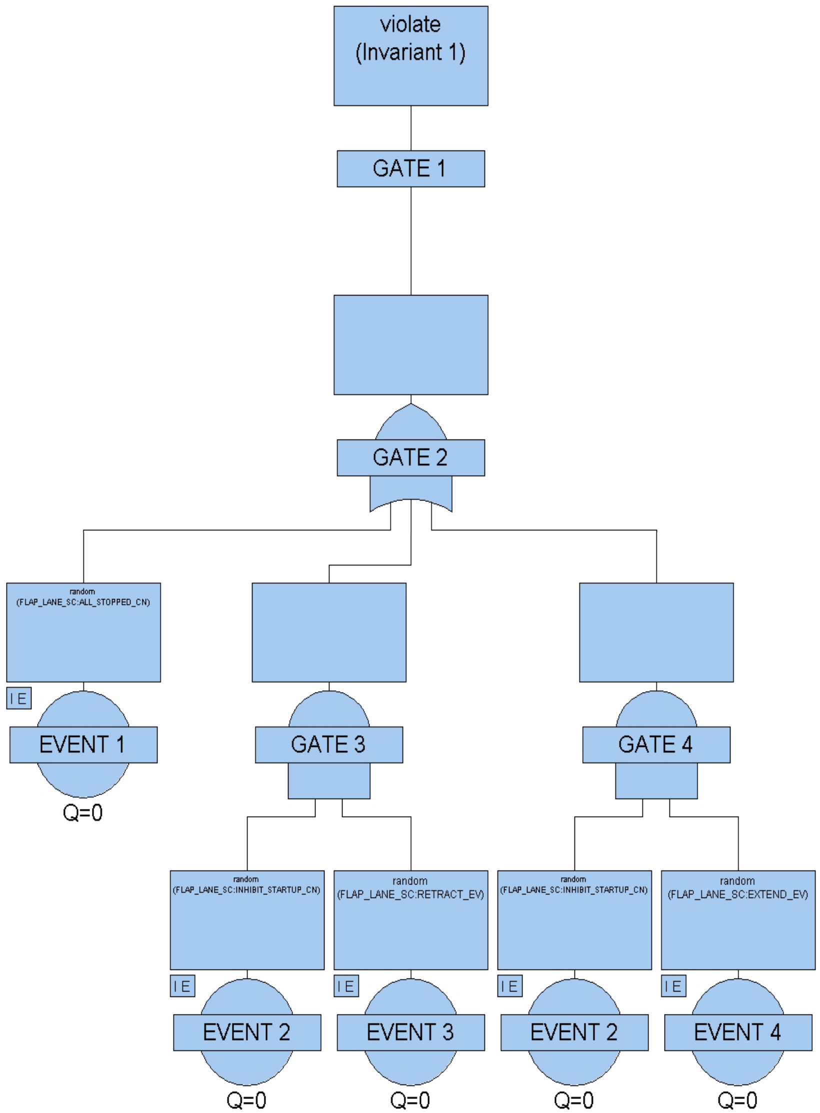
Category	Definition
Catastrophic	would prevent continued safe flight and landing
Hazardous	would reduce the capability of the aircraft or the ability of the crew to cope with adverse operating conditions to the extent that there would be ▪ a large reduction in safety margins or functional capabilities ▪ physical distress or higher workload such that the flight crew could not be relied upon to perform their task accurately or completely ▪ adverse effects on occupants, including serious or potentially fatal injuries to a small number of those occupants
Major	as above, but items viewed disjunctively
Minor	not major and e.g. slight reduction in safety margin, or slight increase in crew workload, such as routine flight plan changes, or some inconveniences to occupants
No effect	on operational capability of aircraft nor incease of crew workload

Hazard probability classes for aircraft systems



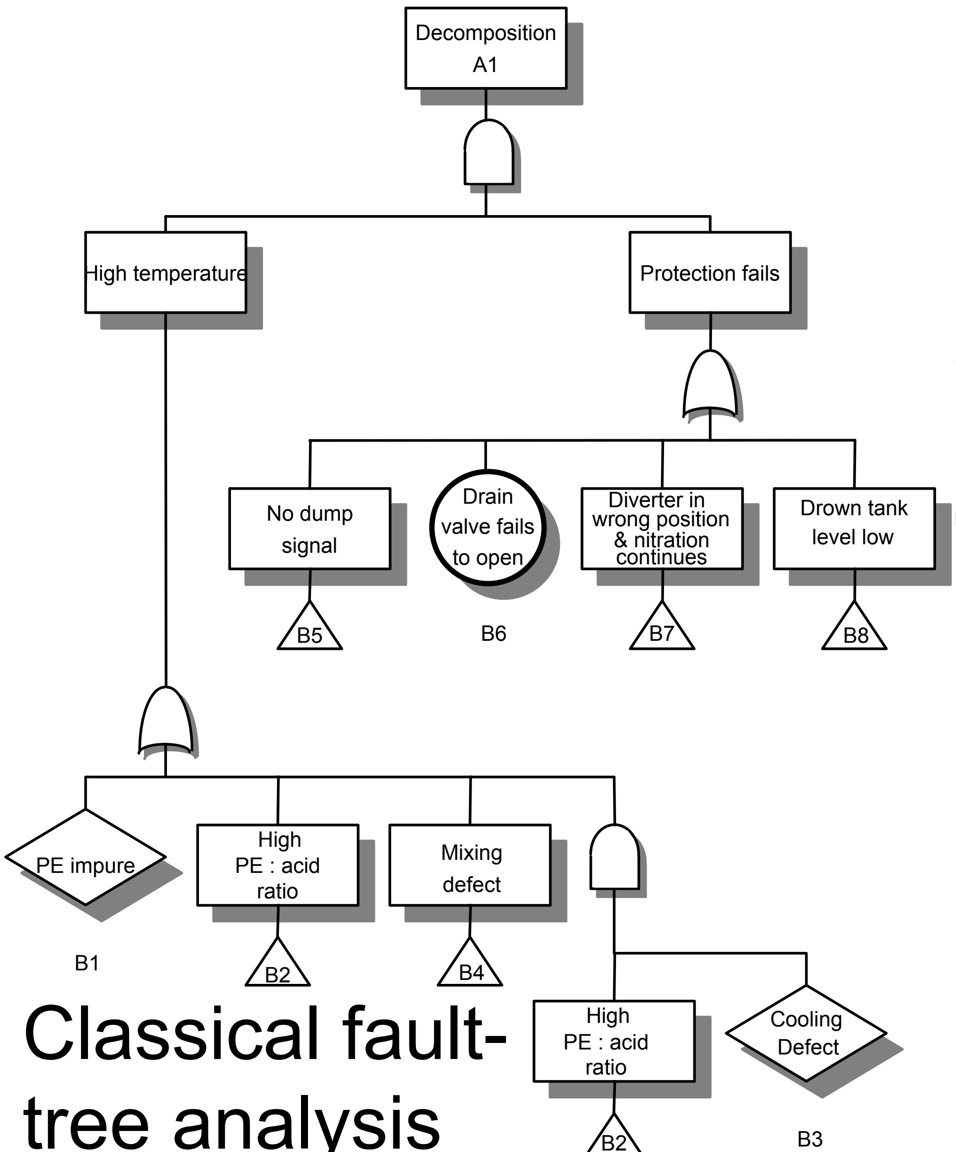
© N.S.

Fault Trees

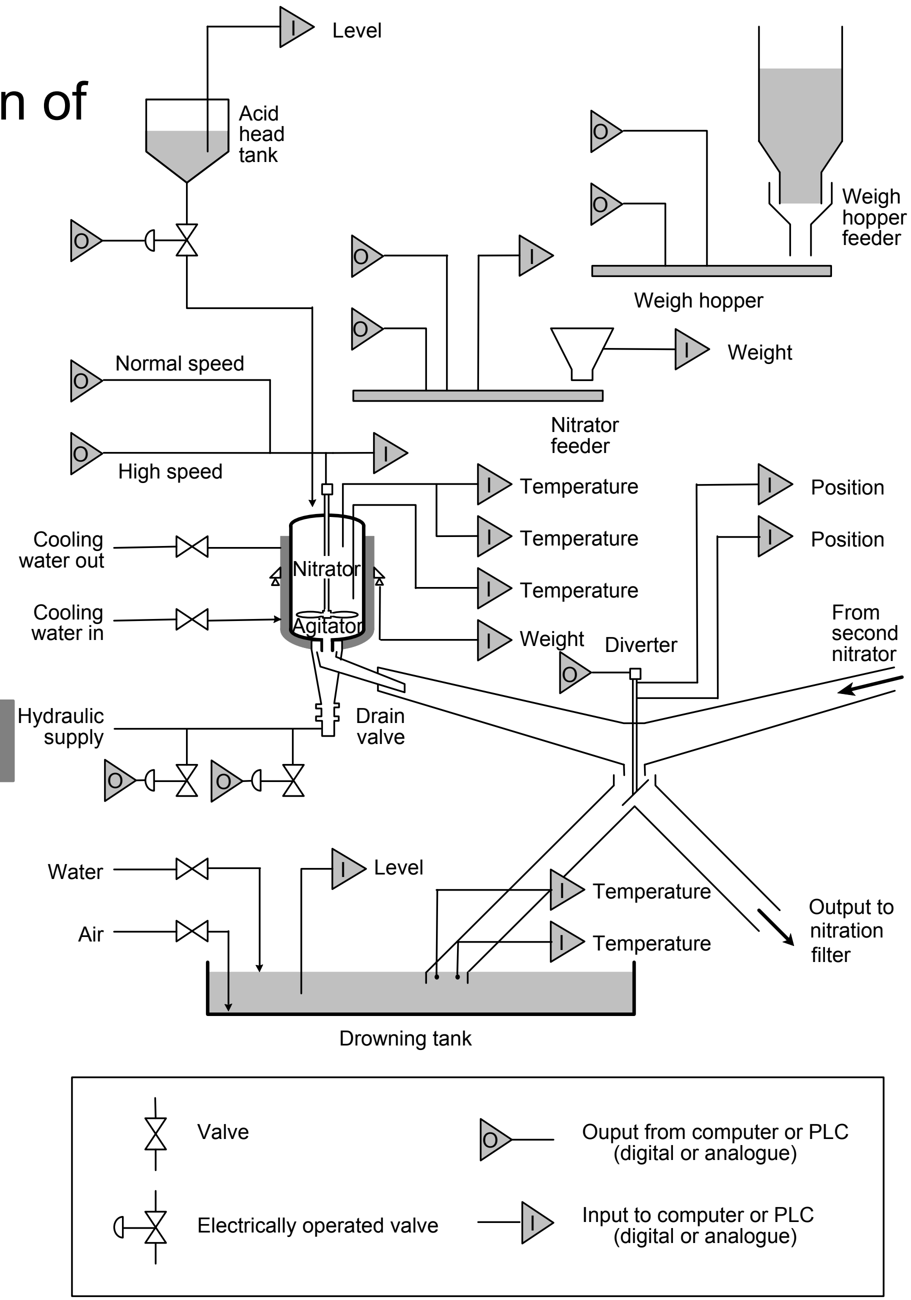


- Start from “Top Level Event”
  - The hazardous situation to be avoided
- Reduces this to failure events
  - Leafs of fault tree
- Explicate causal reasoning
  - Using non-standard semantics of boolean connectives
  - AND: subtrees must have both become true at some point in time
  - OR: one of subtrees must become true at some point in time
- Cut set: a set of events whose joint occurrences causes the TLE
- Minimal cut set: a cut set, where each conjunct is necessary for causing the TLE

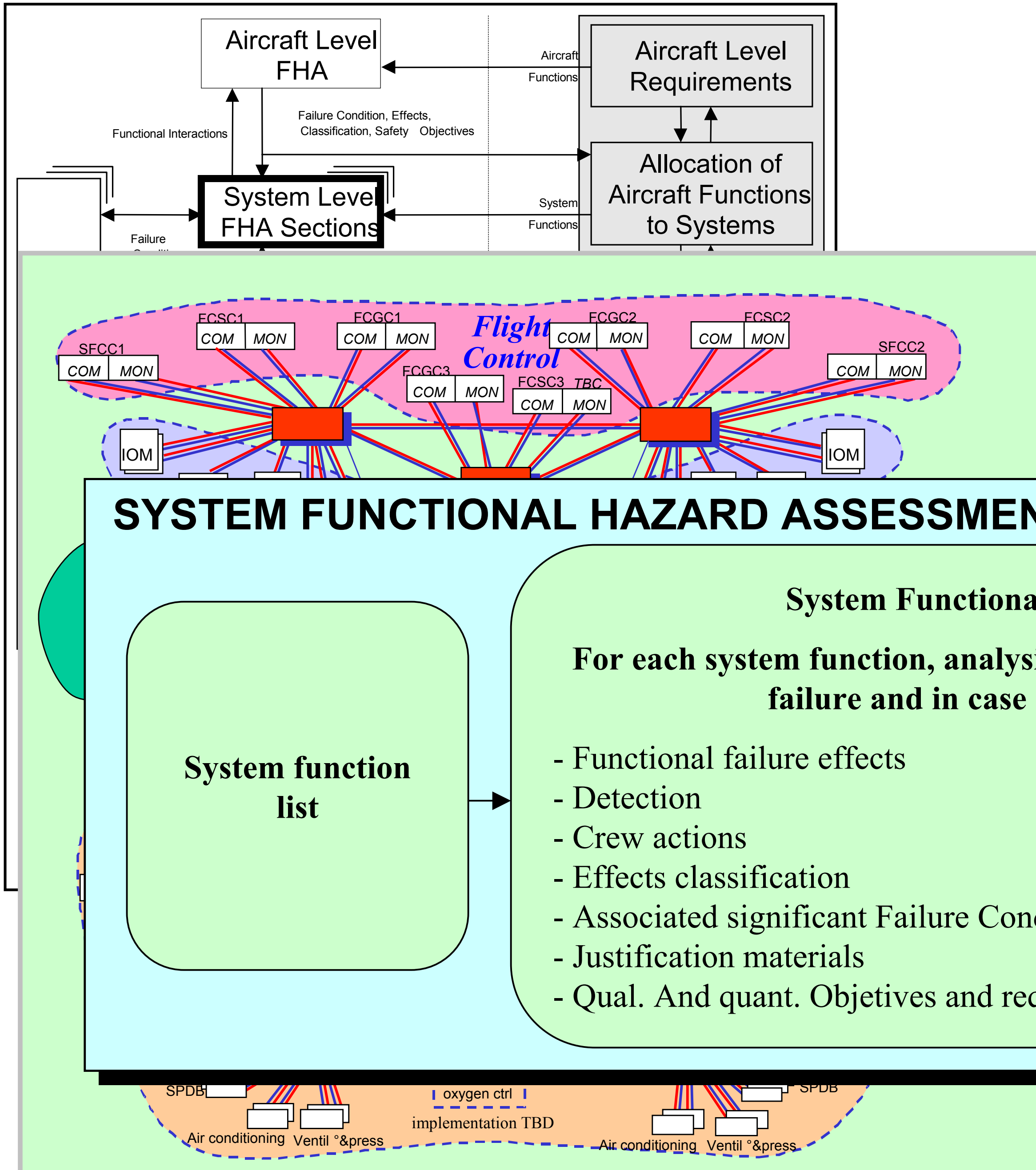
- Uses (informal!) knowledge of safety engineer and structural representation of system



Classical fault-tree analysis



© N.S.



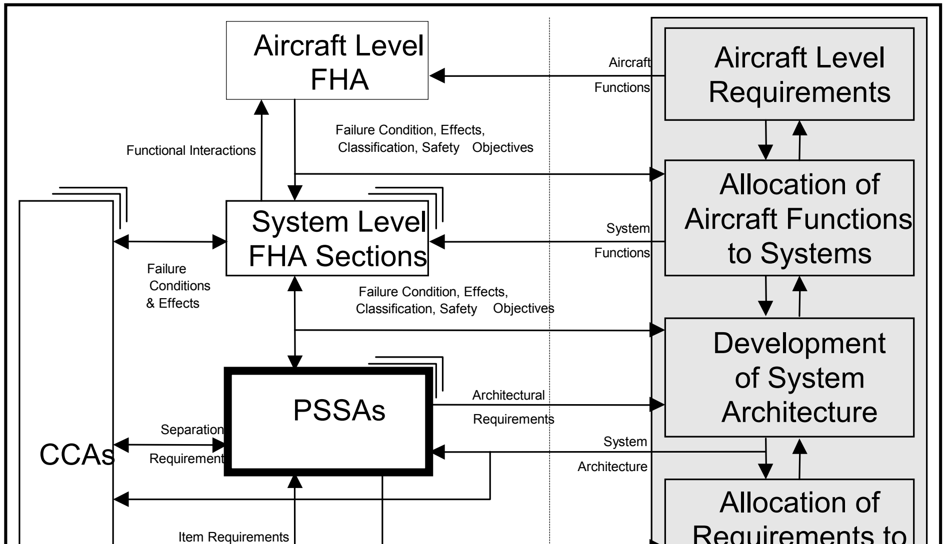
SYSTEM FUNCTIONAL HAZARD ASSESSMENT (FHA)

System function list →

System Functional Failure assessment.

For each system function, analysis of effects in case of function single failure and in case of failure combination

- Functional failure effects
- Detection
- Crew actions
- Effects classification
- Associated significant Failure Condition
- Justification materials
- Qual. And quant. Objetives and requirements



PRELIMINARY SYSTEM SAFETY/ RELIABILITY ASSESSMENT (PSSA)

Failure Condition list from system FHA

Failure Condition supporting materials

For each Failure Condition identified in the FHA, assessment that the Requirement/ Objectives are met:

- Dependence diagram or Fault Tree
- Failure modes and failure apportionnement
- Probability evaluation
- Dormant failures maintenance task periodicities
- Justification material
- Equipment and software criticality and DAL

DEMANDS FOR:

- Common cause studies
- Crew error analysis
- Maintenance error analysis
- Ground and flight tests
- Segregation in installation

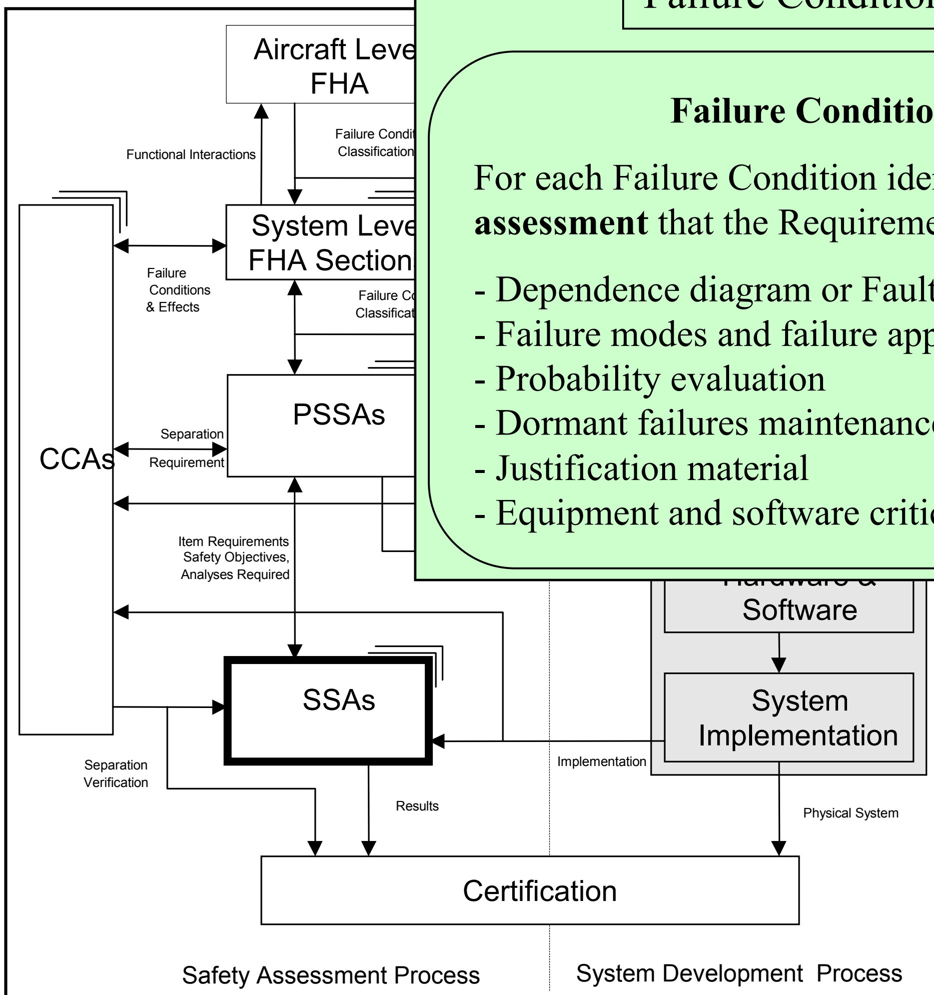
SYSTEM SAFETY/RELIABILITY ASSESSMENT (SSA)

Failure Condition list from system FHA

Failure Condition supporting materials

For each Failure Condition identified in the FHA, **updating of the assessment** that the Requirement/Objectives are met:

- Dependence diagram or Fault Tree
- Failure modes and failure apportionnement
- Probability evaluation
- Dormant failures maintenance task periodicities
- Justification material
- Equipment and software criticality and DAL



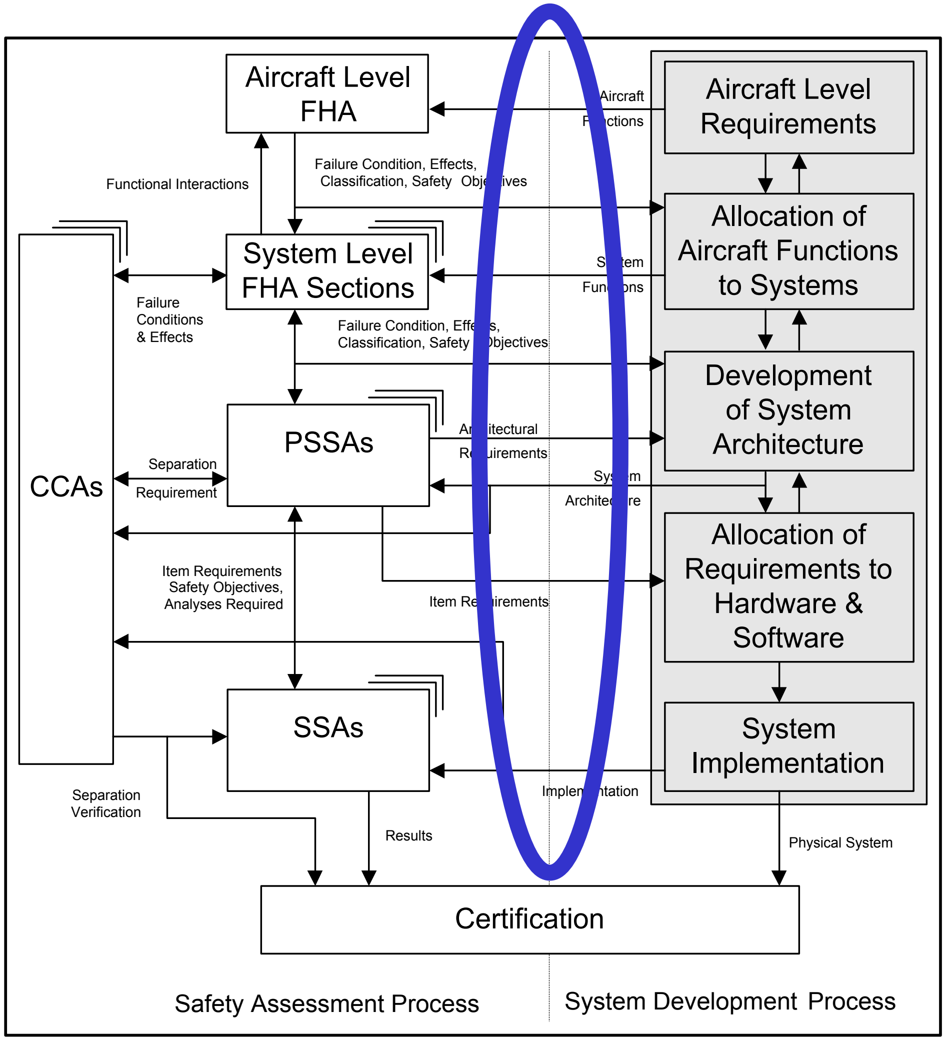
Model Based Safety Analysis

… using Formal Verification Technology

Issues with classical fault tree analysis

- The coherency issue
  - How do models used for safety analysis relate to the actual design?
  - How can safety engineers keep track with ongoing evolvements and changes in design models?
- The plausibility issue
  - How can a system designer relate a cut set to „her“ model?
  - How can she understand, how the cut-set can arise?
- The accuracy issue
  - How can mission phases,
  - How can numerical threshholds
  - .... be assessed without gross overapproximation?
- The completeness issue
  - How can a safety designer assert, that all minimal cut sets have been identified?

The ESACS Approach towards ARP 4754 and 4761



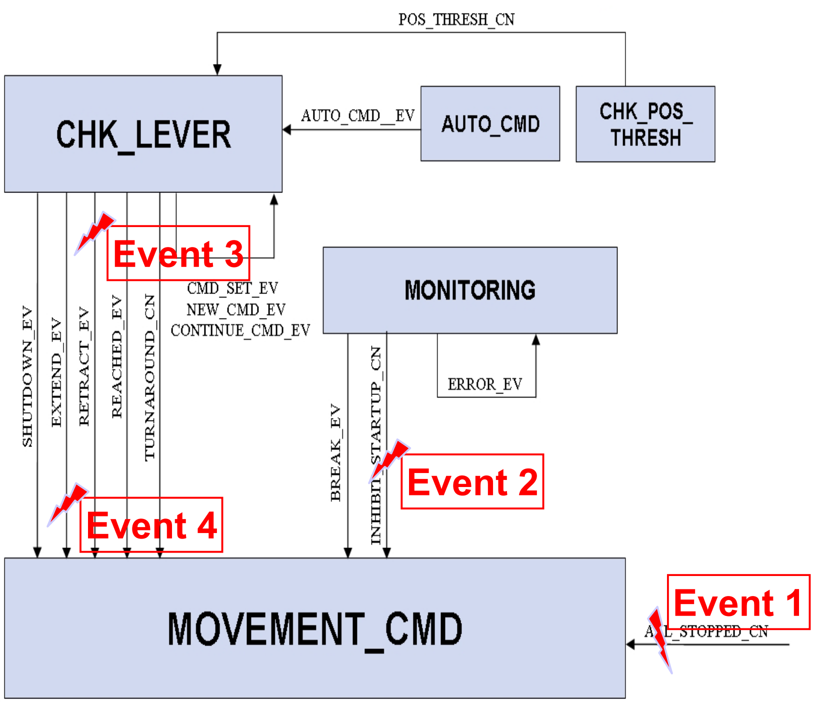
- Model Based Approach
  - Conceptual models for early Analysis
  - Model Based System Development
- Reduce Level of misconception between System-Designers and Safety Engineers

Supported by GROWTH
http: //www.esacs.org

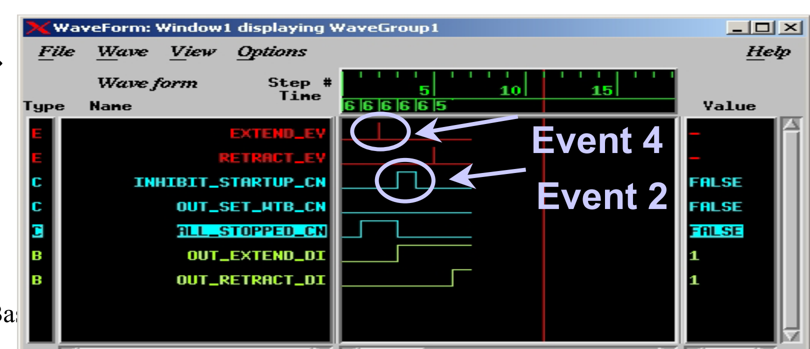
Embedding failures into System Models

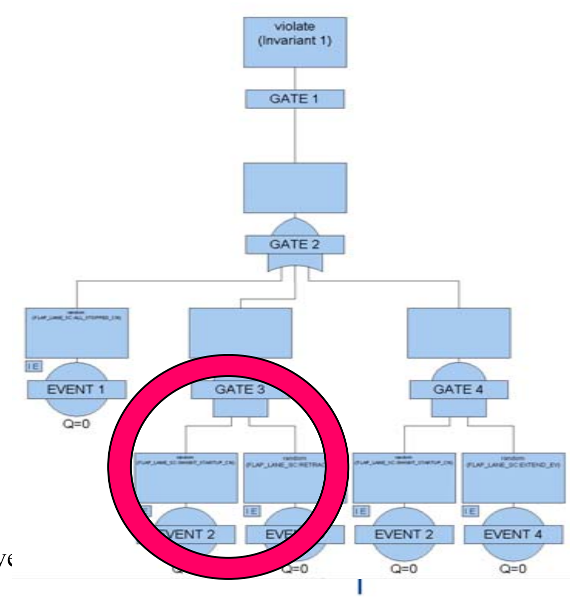
- User specifies fault configuration
  - Associates with design units failure modes
- Fault configurations guide “patching” of semantic representation of Statemate model
  - Each failure is represented by
    - Boolean input: failure occurs when set
    - Boolean local variable: set once failure has been observed
    - Failure model: automata based semantic representation of effect of failure
  - Glue logic disconnects “nominal semantics” driving design unit upon occurrence of failure input, switches to failure model
- Allows full propagation of failure effect on all design entities

Model Checking Based Safety Analysis



- ModelChecking based FTA tool automatically performs fault-tree analysis on system model taking into account injected failure modes
- Computed fault-tree represents all minimal cut sets leading to given top-level event
- Cut sets can be analysed on extended system model using simulation: how can this cut set arise?
- Fault-trees can be exported to FTA+ for analysis of failure probabilities



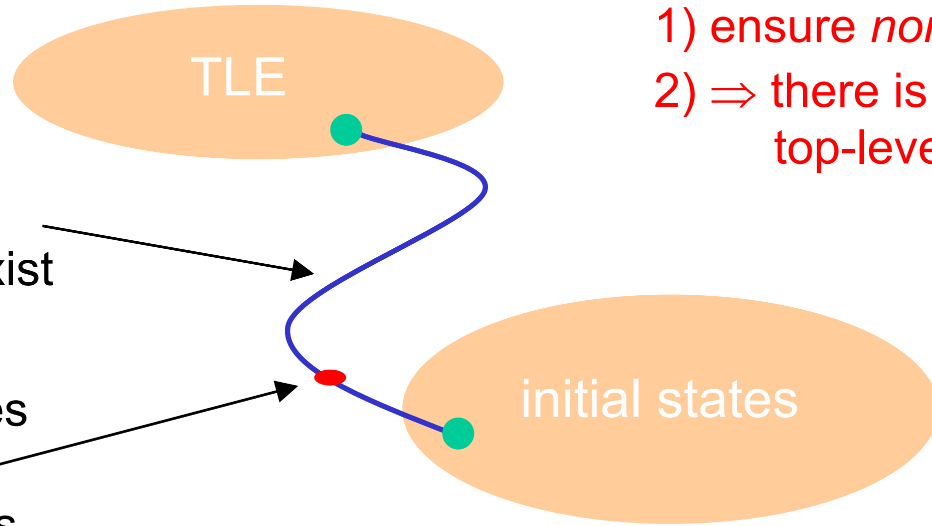


BDD based FT generation

1) ensure *nominal correctness*

2) $\Rightarrow$ there is no path to the top-level event (TLE)

If the design is (nominally) correct this path does not exist



However, failures introduce additional paths

3) Extend the model with failures triggered by *additional inputs*

4) Introduce additional *local variables* recording the occurence of failures (*Failure Variables*)

5) Check *what valuations* of failure variables allow the TLE to be reached ...

BDD-Verfahren

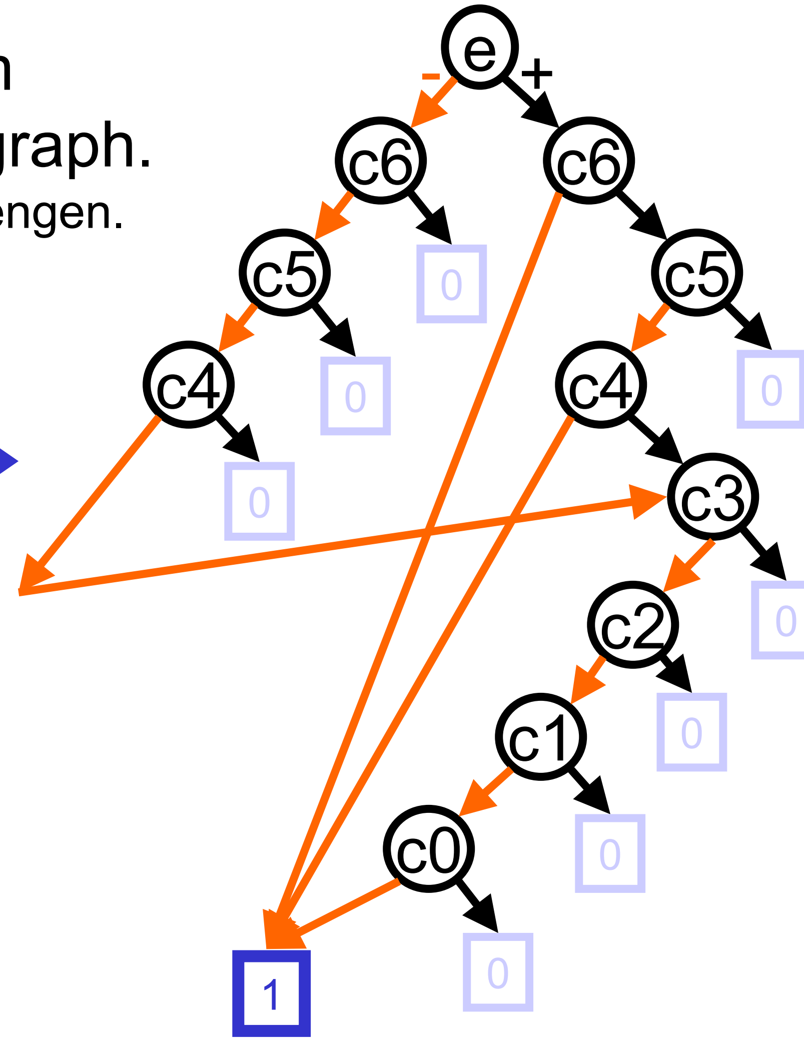
BDD: Binary Decision Diagram
= binärer Entscheidungsgraph.

Dient zur kompakten Darstellung von Mengen.

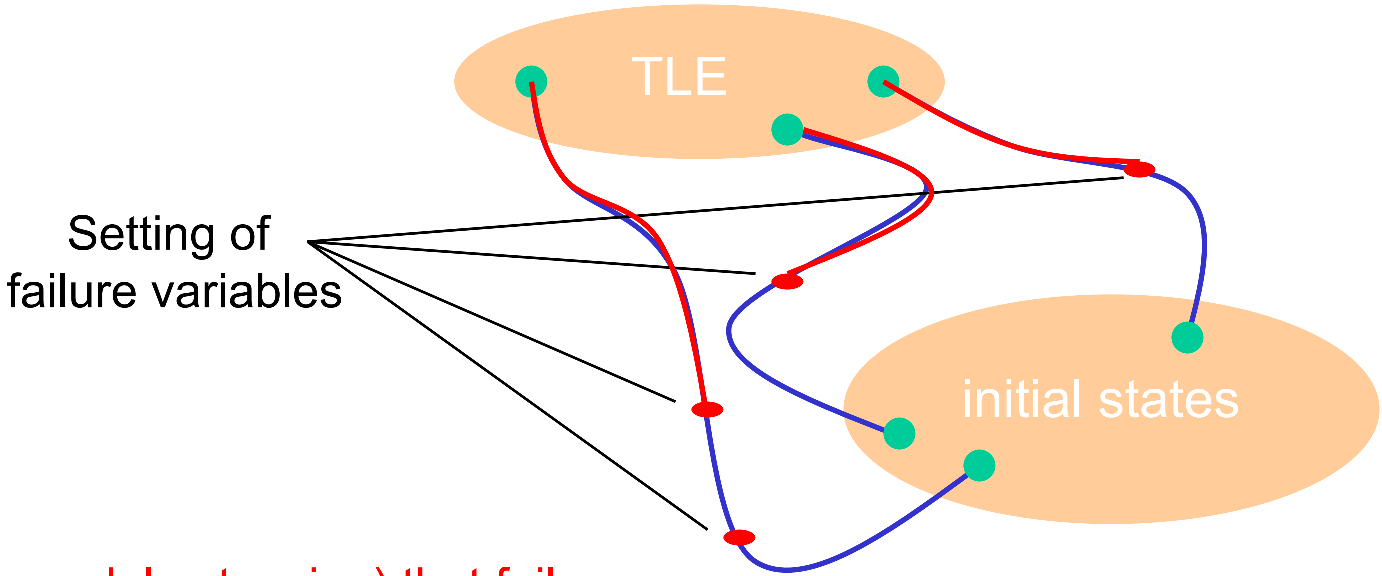
$$\left\{ \begin{array}{l} (e,c=0),\ (e,c=1), \\ (e,c=2), \quad \ldots \\ (e,c=79),\ (e,c=80), \\ (\bar{e},c=0) \end{array} \right\}$$

$\cong$

$$e \wedge c \leq 80$$
$$\vee\ \bar{e} \wedge c=0$$



BDD based FT generation



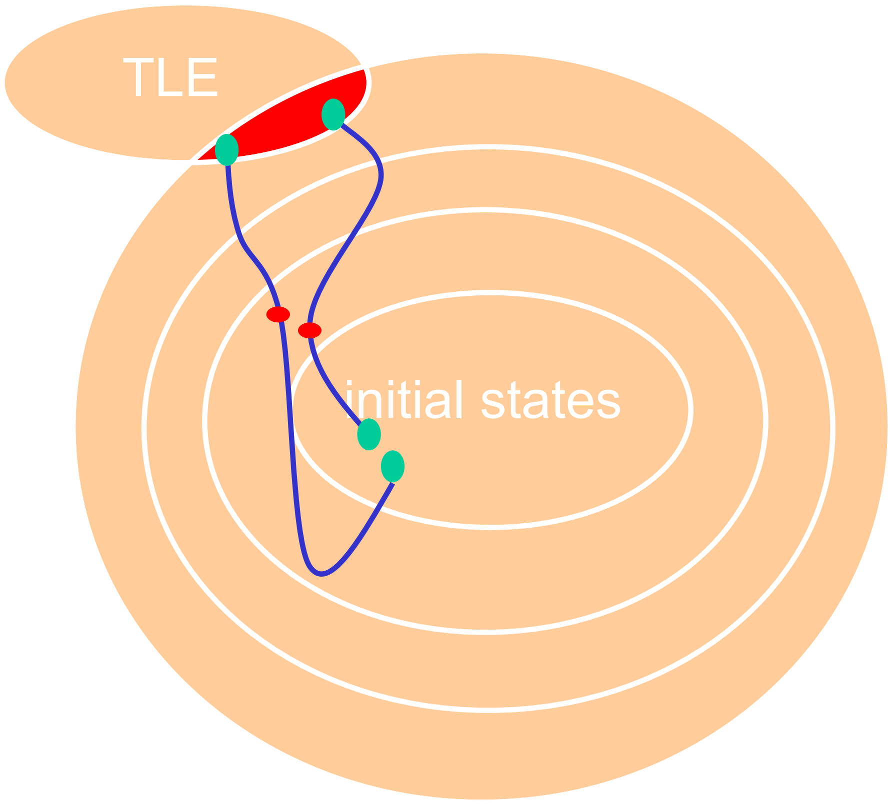
It is guaranteed (by model extension) that failure variables are never reset

⇒ Checking for occurrence of failures can be deferred until TLE has been reached

⇒ Can use classical reachability analysis to check whether failures lead to TLE

Reachability based FT generation

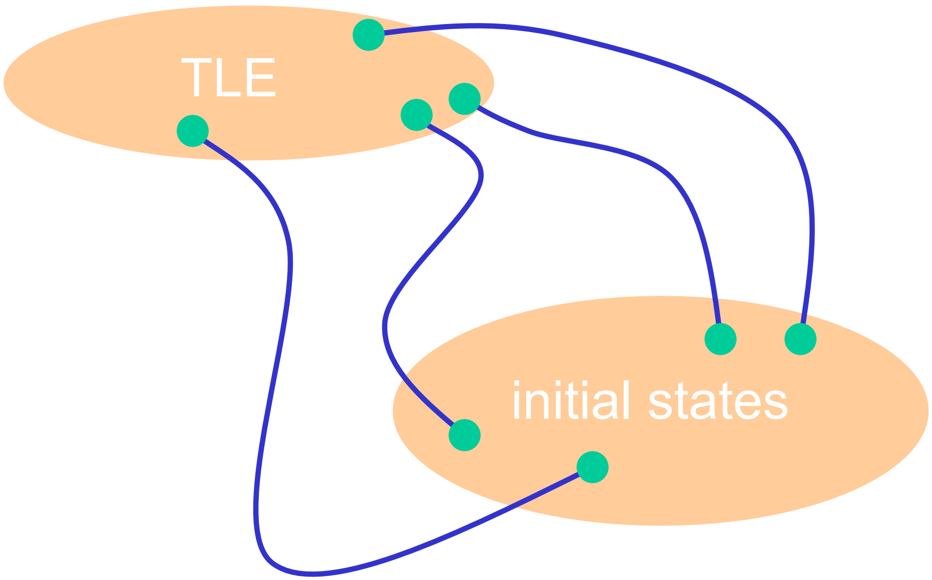
- Compute BDD representing intersection of TLE with set of reachable states
- Project to local variables representing occurrence of failures
- Translate this BDD into disjunctive normal form
- By BDD reduction rules, all conjuncts are minimal cut sets
- Yields flat fault-tree
- (ongoing extension: reflect structure of model)



simple SAT-based methods don´t work

Using extended model T‘:

$Init(s_0) \wedge T'(s_0,fv_1,s_1) \wedge \ldots \wedge T'(s_{n-1},fv_n,s_n)$
$\wedge\ noloop(s_0,\ldots,s_n) \wedge TLE(s_n)$
$\wedge\ fv = fv_1 \vee \ldots \vee fv_n$



Perform „drive-to“ analysis for certain failure combinations with BMC methods

⇒ incomplete (as long as model diameter is not reached)

⇒ (for practical reasons) also incomplete with respect to number of possible failure combinations

Example:

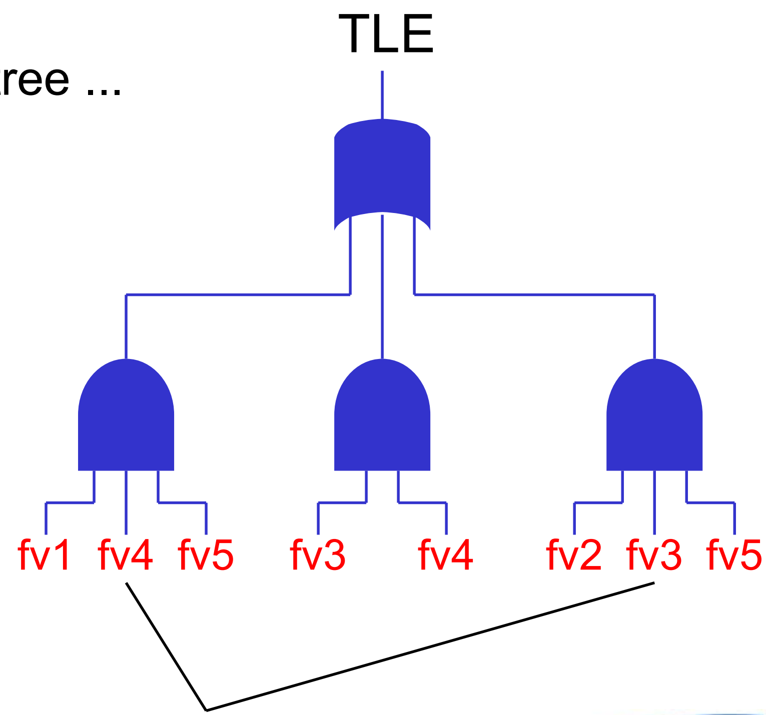
There are 1275 possibilities to have at most two (but at least one) failure activated among 50 possible failures.

Using Abstraction

traditional abstraction techniques are safe also when constructing fault trees (due to percistency of setting of local variables associated with failures)

Resulting fault tree will be too pessimistic:

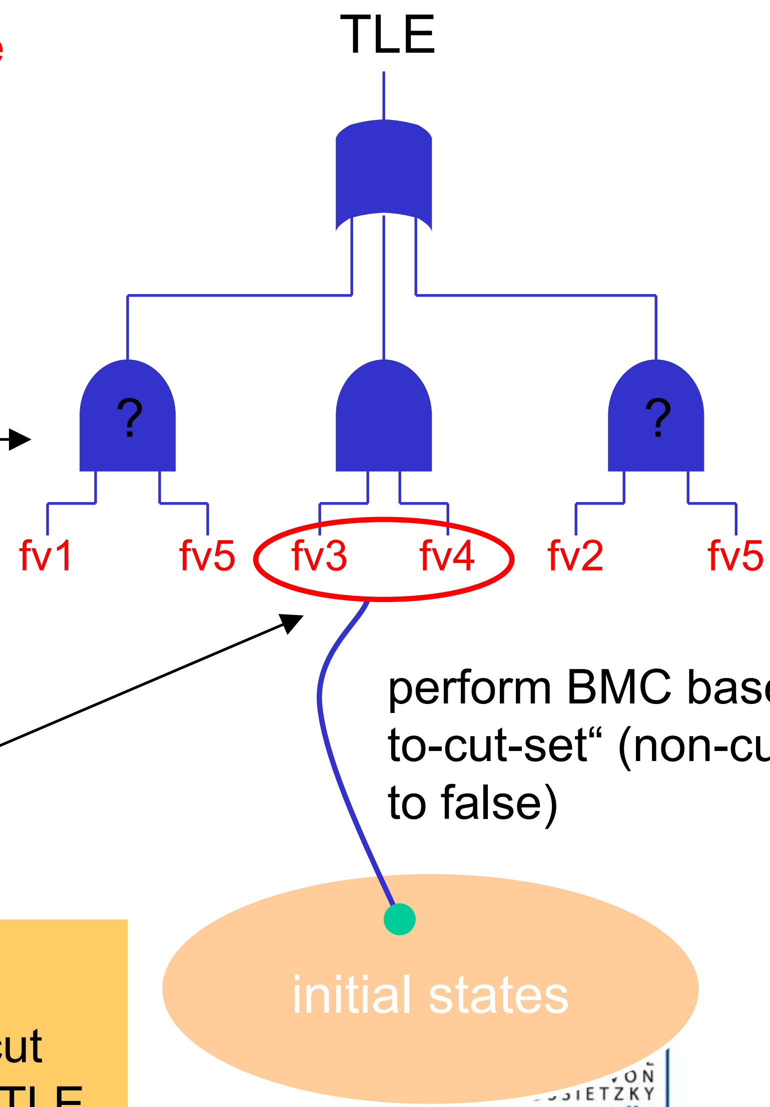
If this is an abstract fault tree ...



... this might be the right one

Concretizing abstract fault-trees

Trying to concretize
abstract cut sets:



Abstract cut set
not reachable
⇒ some failure
variables are missing

Abstract cut set C
reachable
⇒ C is concrete cut set

perform BMC based „drive-to-cut-set" (non-cut-set fv set to false)

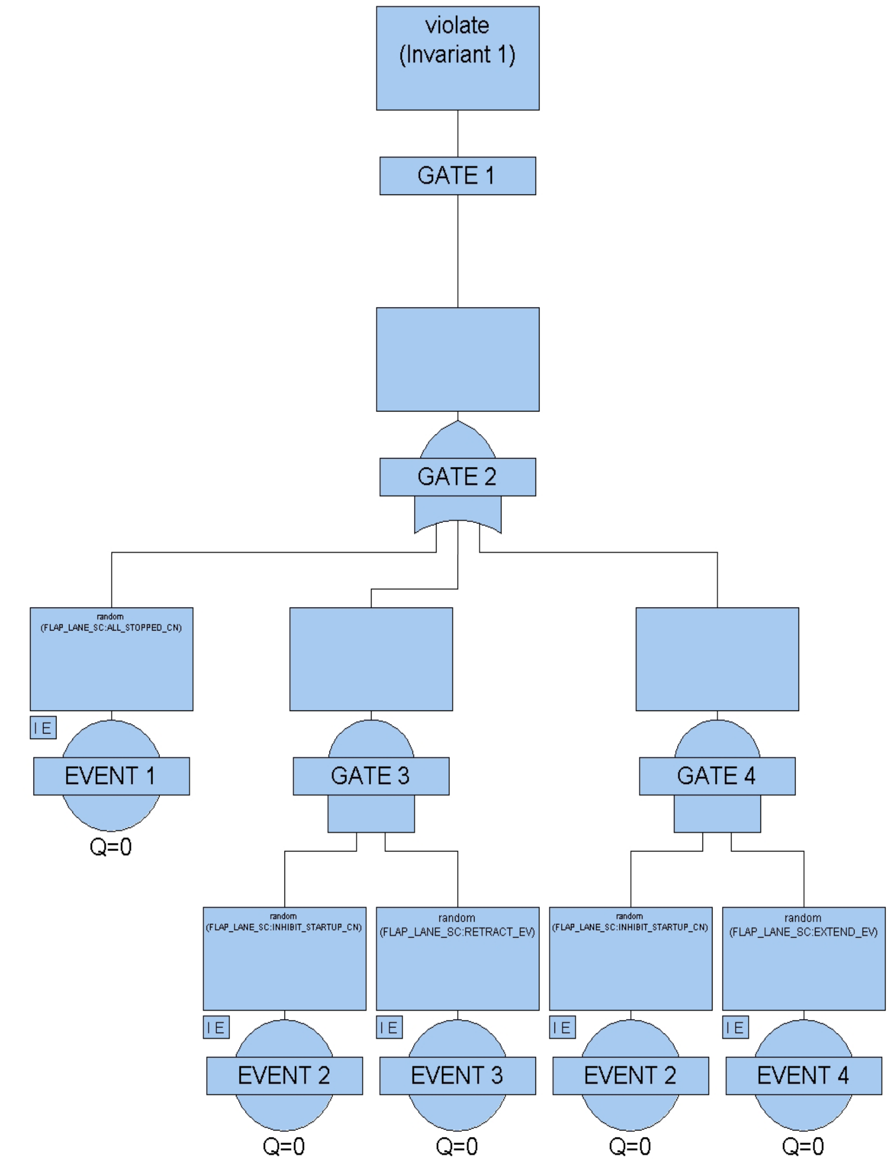
Can use abstraction refinement:

for each (non concretizable) abstract cut set C perform FT computation for C $\wedge$ TLE

Example (cont.)

- **TLE:** The Flap System outputs RETRACT and EXTEND shall never be true at the same time.

- Injected **FMs** (random/persistent):
  - RETRACT_EV (EVENT 3)
  - EXTEND_EV (EVENT 4)
  - SHUTDOWN_EV
  - ALL_STOPPED_CN (EVENT 1)
  - INHIBIT_STARTUP_CN (EVENT 2)
- Cut-sets show, that controller is not protected against failures impacting inhibit-startup
  - Nominal usage: hydraulic pressure too low
  - Uncontrolled occurrences due to failures can cause contradicting actuator settings for flap system



Acknowledgements

- Work performed under Growth projects ESACS and ISAAC
  - Airbus UK, D, F, Alenia, Saab
  - OFFIS team T. Peikenkamp, E. Böde, A. Lüdtke, H. Spenke
- Thanks to Matthias Brettschneider (Airbus DE) and Jean Pierre Heckmann (Airbus F) for many deep discussions
- Figures marked ©N.S. are courtesy to Neil Storey, taken from his book "Safety Critical Computer Systems", Addison Wesley

Published in Proc. Incose 2004, Model-based Safety Analysis of a Flap Control System

Conclusion

- Model Based Safety Analysis is seen as a key objective by avionics companies to further improve the (already high!) quality of the safety analysis process
- Feasibility demonstrated in ESACS, further enhancements and optimization as part of ISAAC project
- Ongoing cooperation with Airbus in Depnet project addresses compositional approaches to safety analysis