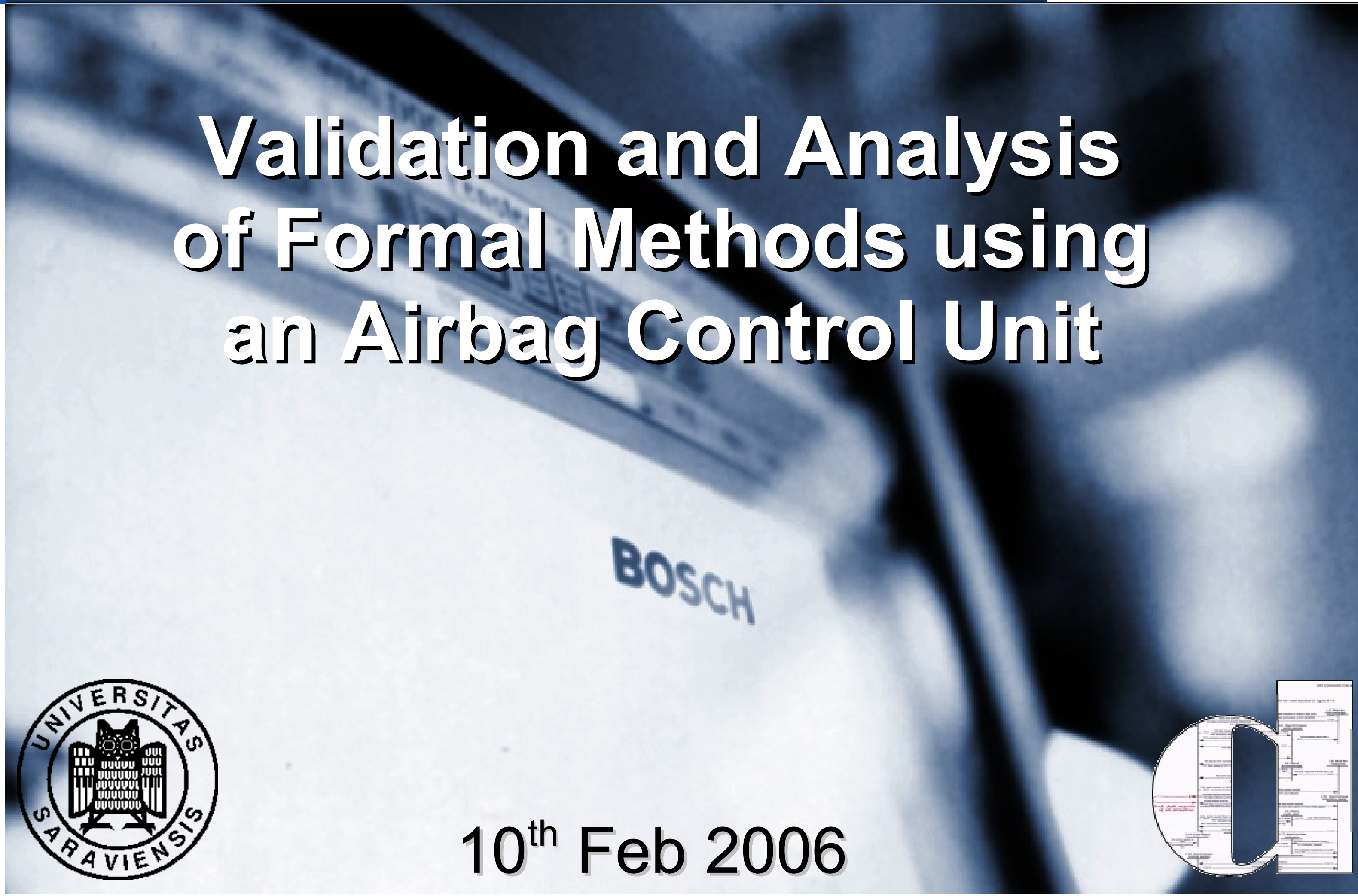


Validation and Analysis of Formal Methods using an Airbag Control Unit

10th Feb 2006

Universität des Saarlandes

Overview

Verification by Simulation

Behavior Model (MoDeST)

Safety Analysis (Markov Chain)

Failure Analysis (Fault Trees)

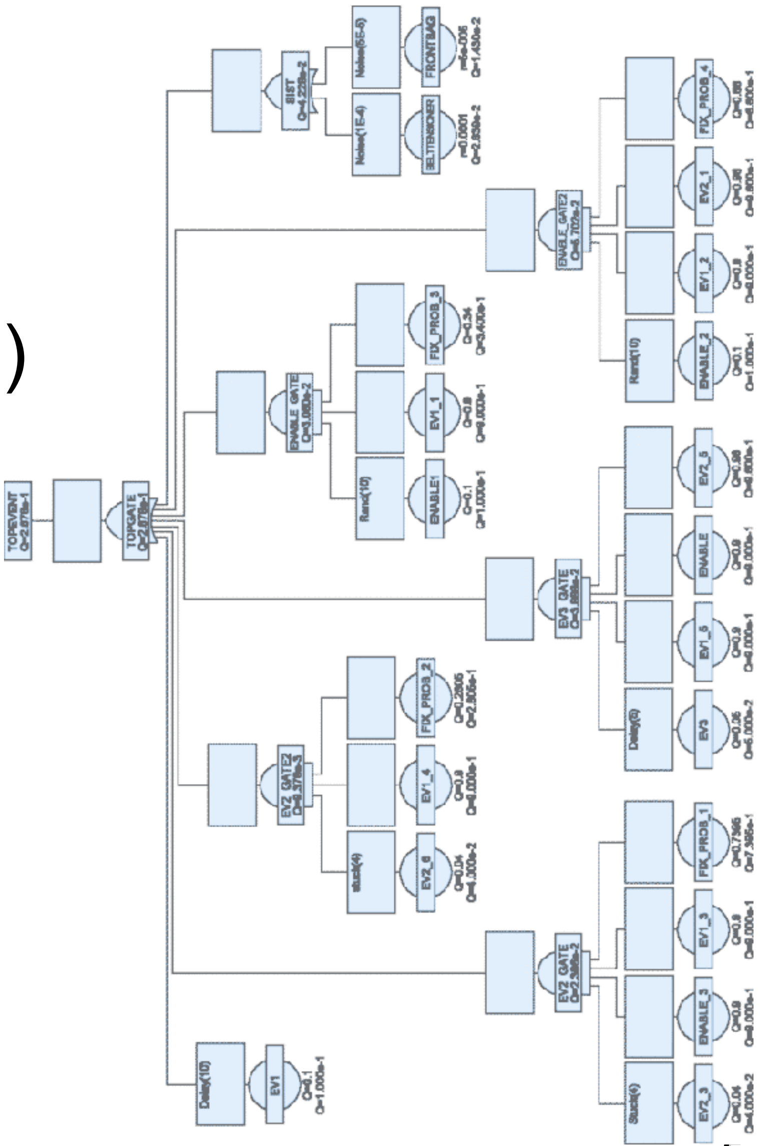
Interesting Point of Investigation

To which extent can simulation be used throughout the analysis using Stochastic Timed Automata

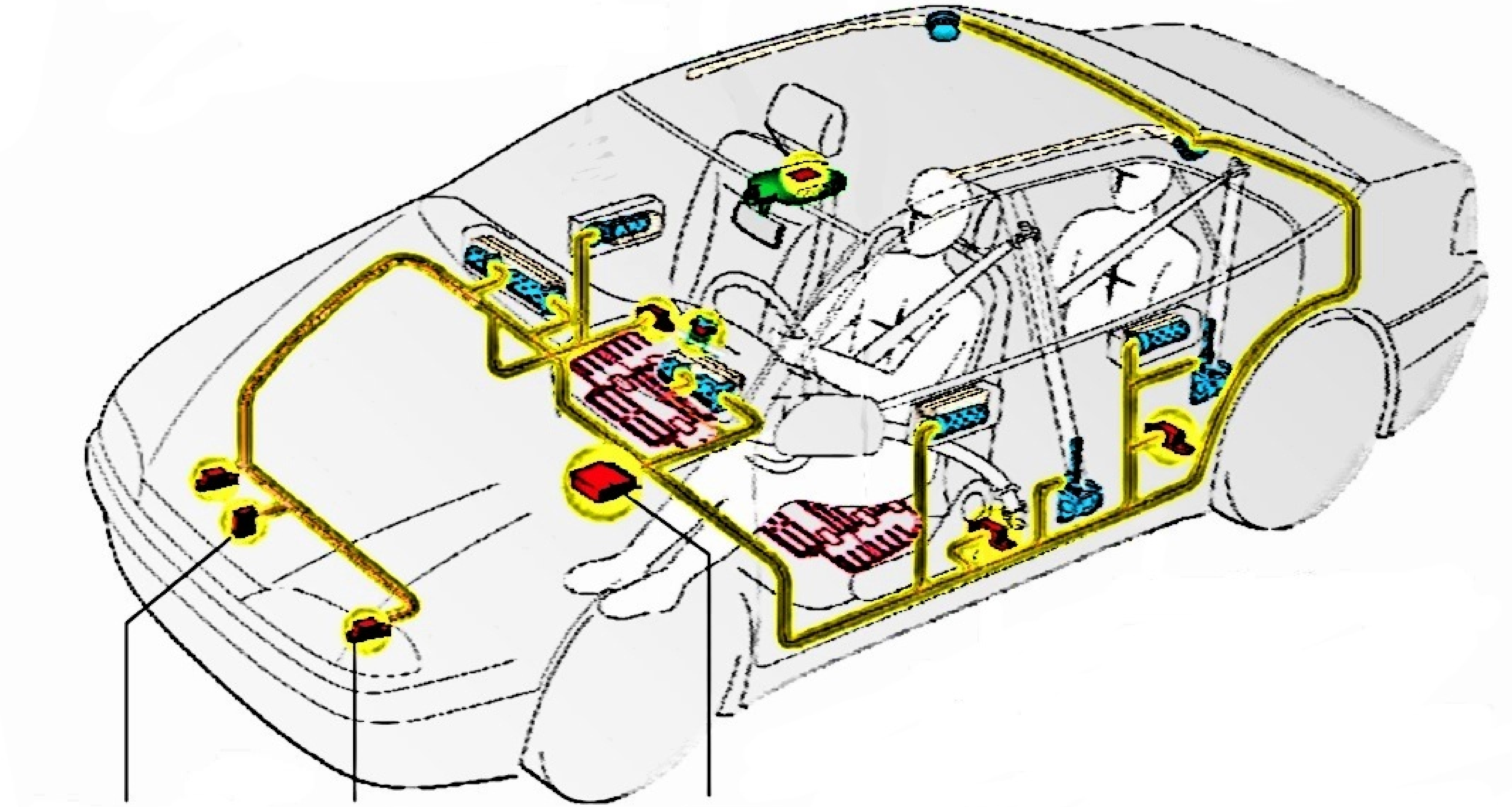
Overview

- **Introduction**
  - The Airbag System
  - MoDeST
  - Uppaal
- **Simulating the ECU**
  (Modeling and Verification)
- **Safety Requirements**
  (Markov Analysis)
  - via Simulation in MoDeST
  - Analytical Approach

- **Fault Tree Generation**
- **Importance Analysis**
  (Fussel-Vesely Importance)
- **Conclusion**



Airbag and Belt Tensioner Deployment depend on configuration determined by

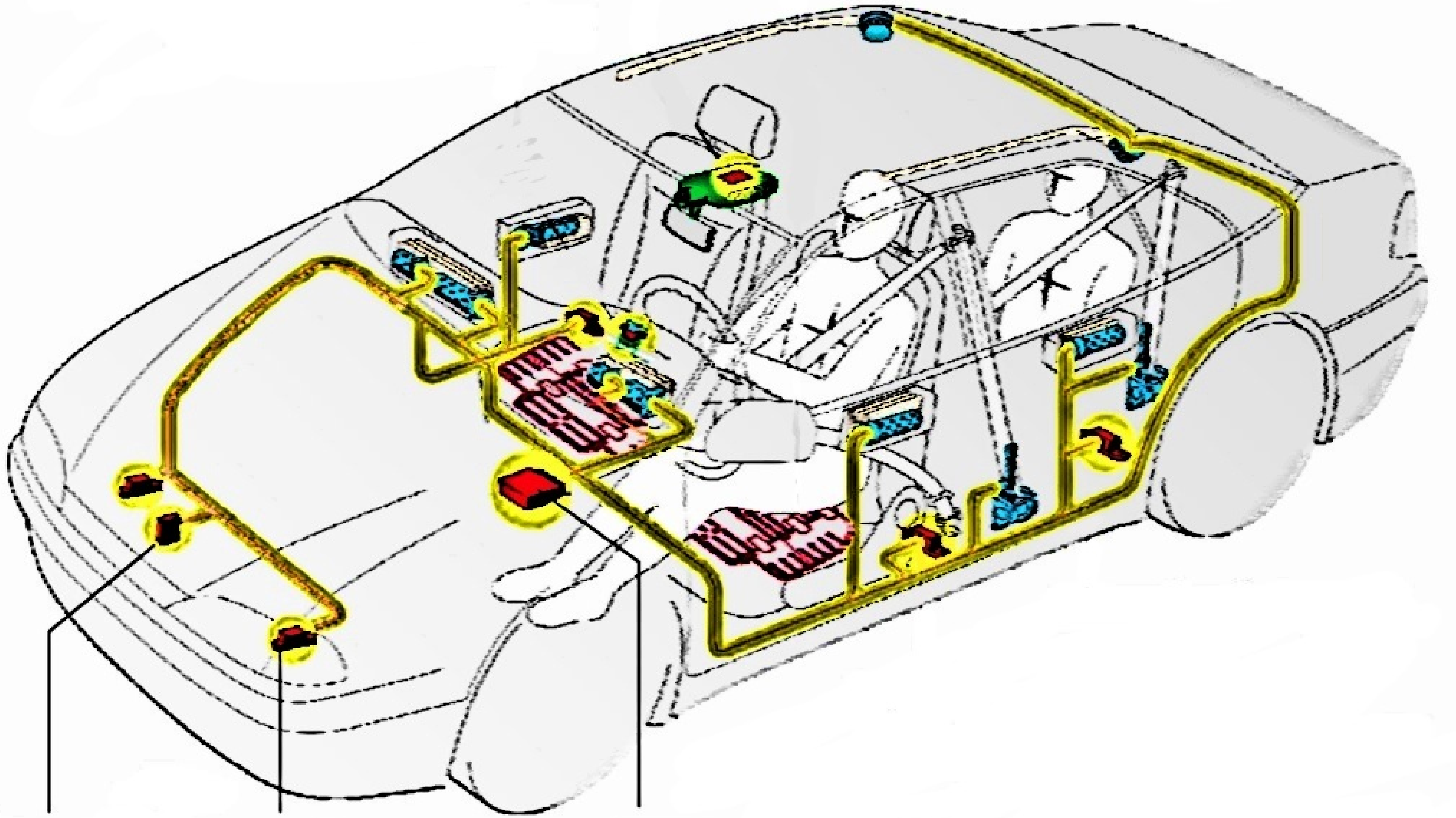


Pre-crash Sensor

Up-Front Sensor

BOSCH

Airbag and Belt Tensioner Deployment depend on configuration determined by



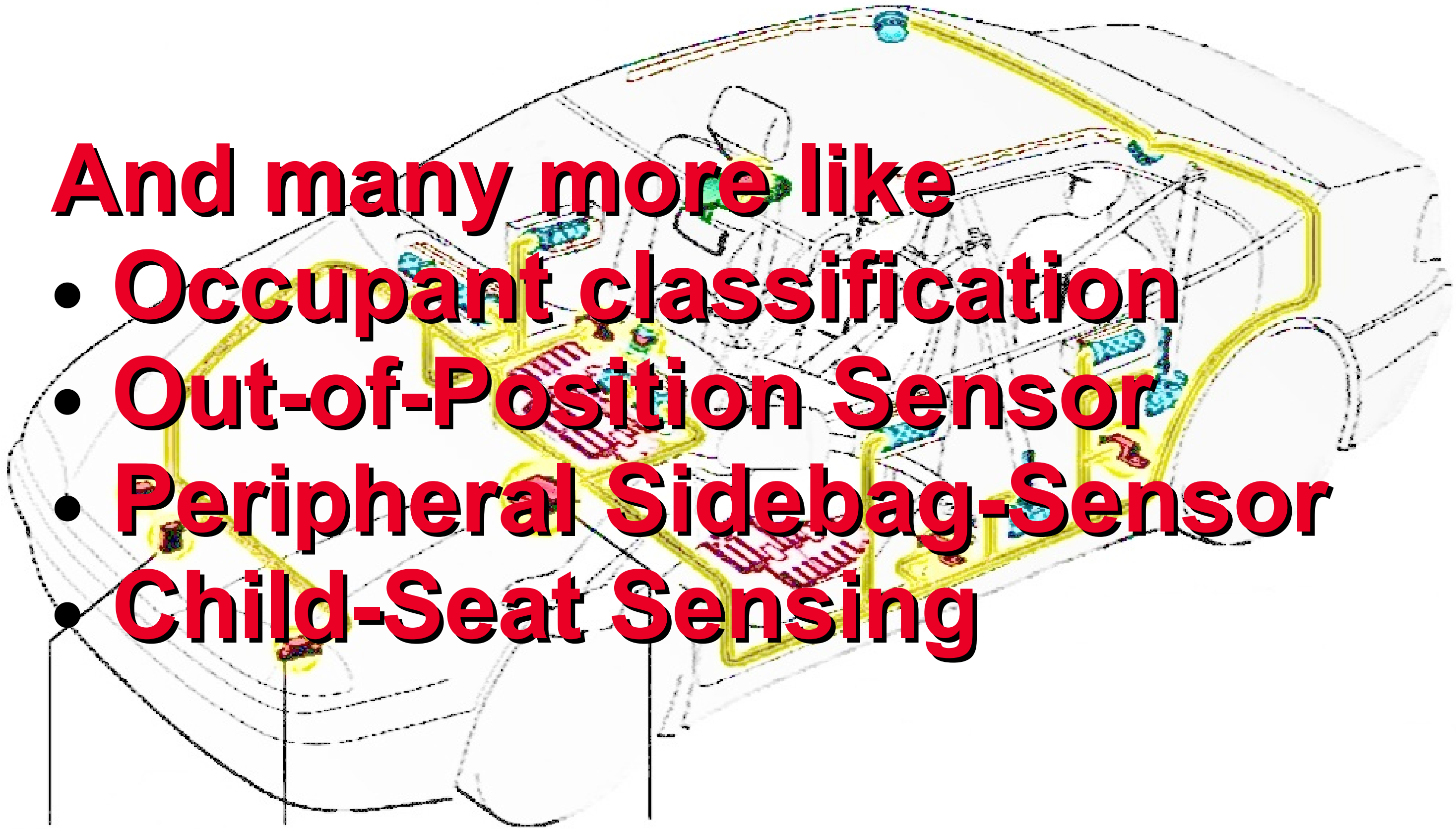
Pre-crash Sensor

Up-Front Sensor

Central Control w/ over-roll sensing

BOSCH

Airbag and Belt Tensioner Deployment depend on configuration determined by



And many more like

- **Occupant classification**
- **Out-of-Position Sensor**
- **Peripheral Sidebag-Sensor**
- **Child-Seat Sensing**

Pre-crash Sensor

Up-Front Sensor

Central Control w/ over-roll sensing

BOSCH

MoDeST (Modeling and Description language for Stochastic Timed systems)

- Definition actions (symmetric), variables, data structures
- Stochastic Distributions
- Probabilistic branching (`palt`)
- Non-deterministic branching (`alt`)
- Local `Clock` variables
- Parallelize Processes (`par`)

MoDeST Example

```
PROCESS sender(){
clock t;
   do{
      ::{= t=0, sample=Exponential(lambda) =};
         when (t==sample) urgent(true) sync
   }
}

PROCESS receiver(){
   do {
      ::sync {= counter+=1 =}
   }
}

PAR{  ::sender()
      ::receiver()
}
```

BOSCH

Uppaal (by Universities in Uppsaala and Aalborg)

- Based on Timed Automata
- Collection of non-deterministic processes
- Asymmetric Communication through complementary pairwise actions (a!,a?), broadcast channels, or shared variables
- Checking invariant- and reachability properties by state-space exploration

Simulation of the ECU

Situation:

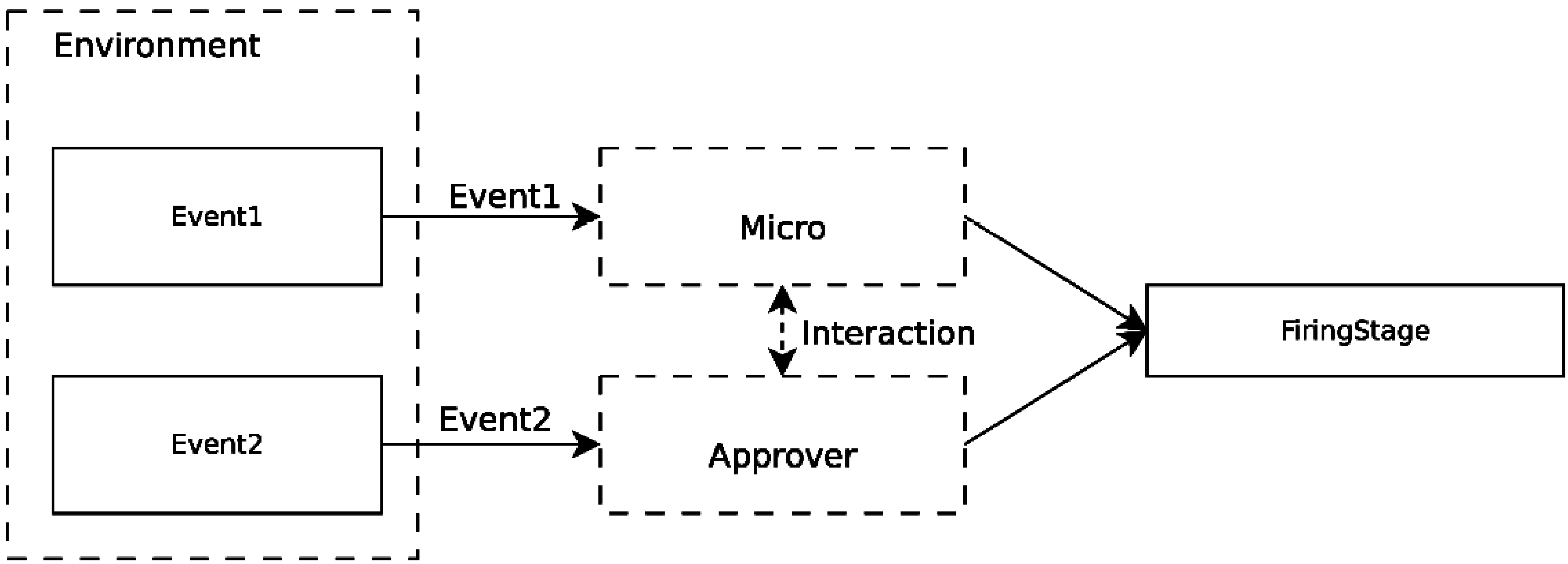
Critical behavior of the airbag control unit that could not be verified in Uppaal.

Goal

- Is it feasible to do model checking via simulation?
- How to adopt for broadcasting?

BOSCH

Scenario Overview



BOSCH

Properties of Interest:

- P0: A[] NOT deadlock
- P1: E<> FiringStage.Fire
- P2: E<> (Micro.Firing AND
  Approver.Enable AND
  NOT FiringStage.Firing)

Properties of Interest:

- P0: A[] NOT deadlock

Simulation

- P1: E<> FiringStage.Fire

Observer

- P2: E<> (Micro.Firing AND
  Approver.Enable AND
  NOT FiringStage.Firing)

Observer

BOSCH

```
Process Observer_Prop1(){
 when(LocationFiringStage==3)
 Property1Satisfied {= Property1+=1 =}
}

Process Observer_Prop2(){
  when(LocationMicroFiring==1 &&
        enable==1 &&
        LocationFiringStage!=2)
   Property2Violated {= Property2+=1 =}
}
```

Simulation Results

- 1 000 batches
- Each for 1 million time units

	Mean Value	Confidence +/-
Property1	1	0
Property2	0	0

What about Property0?

Conclusion

- **Pros**
  - Possible to simulate complex systems where state space exploration is failing
  - Nice way to observe a model during execution
  - Two valuable outputs usable (Trace Path, Reward Variables)
- **Cons**
  - Simulation does never cover the whole state space
  - Danger of purely basing results on simulation
  - Statements about deadlock behavior are risky
  - Awkward to adopt for broadcasting

On-Demand Failure Analysis

- Meet safety integrity requirements
- Does Airbag fails to work at times of an airbag relevant crash
- Determine the company's risk
- Critical point: Unintended or missing airbag deployment

BOSCH

Assumptions

- Overall operation time of one ECU 9.000 hours under normal usage (15 y, 600h/y)
- Total number N_0 of produced ECUs 30.000.000
- An airbag relevant Crash occurs exponentially distributed

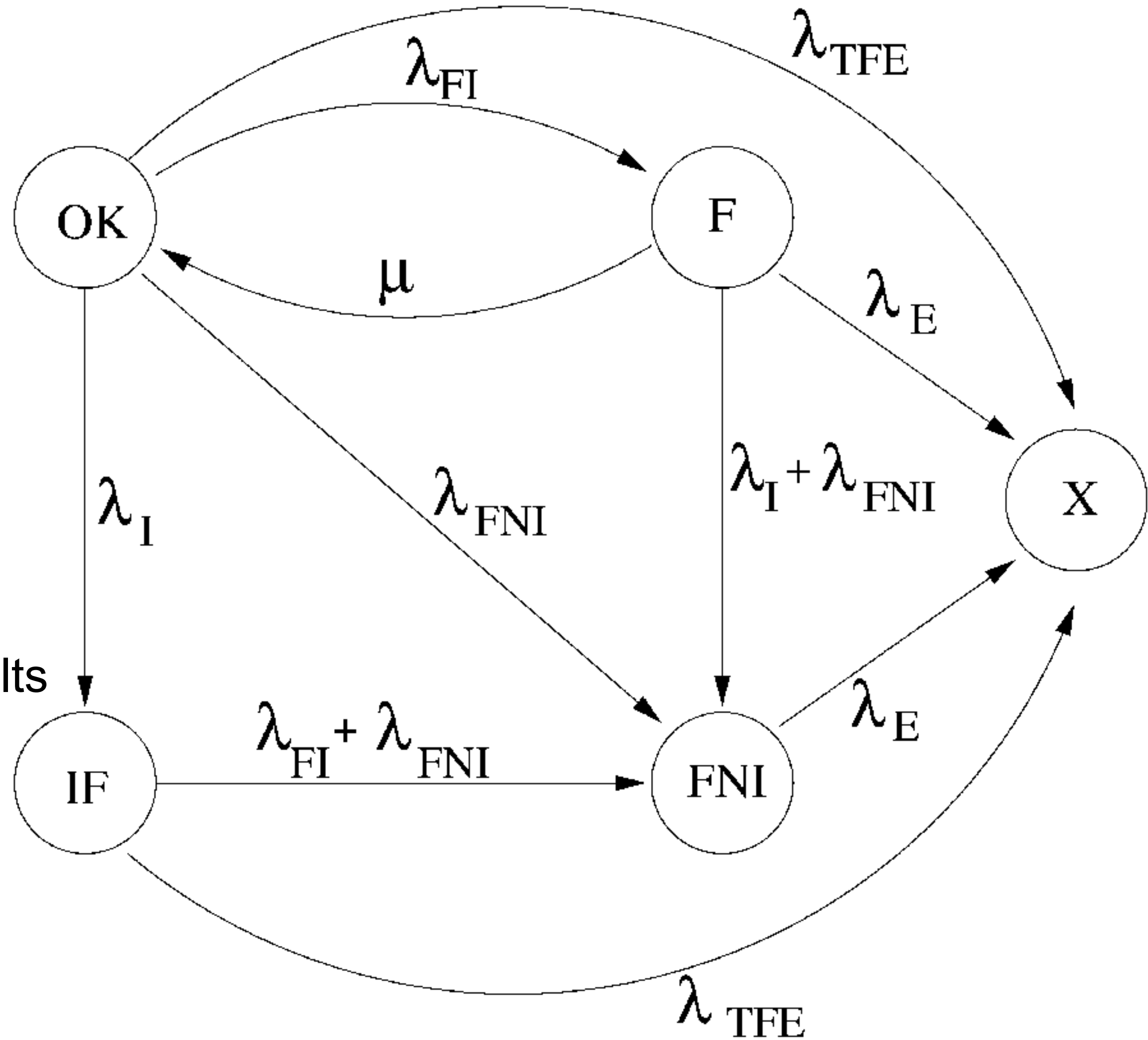
$$Lambda_E = 4.0 \cdot 10^{-5}\, h^{-1}$$

- Indicated Failures get repair after 20 hours on average

$$Mu_{repair} = 5.0 \cdot 10^{-2}\, h^{-1}$$

Markov Chain Analysis

- λ_E
event crash rate
- λ_{FI}
rate of indicated failure
- λ_{FNI}
rate of not indicated failure
- λ_I
indication failure
- μ
repair rate
- P_{TFD}
probability of temporary faults upon processing demand
- λ_{TFE}
rate of temporary failure on demand



Sojourn Times

Time spend in each of the states

Variable	Mean Value	Confidence +/-	Prob
OK	9.9991E06	2.5300E03	9,18%
F	1.9930E-01	7.1528E-04	0,00%
IF	0.9890E8	2.5274E05	90,80%
FNI	2.4996E04	6.3255E00	0,02%

Sojourn Times

Time spend in each of the states

Variable	Mean Value	Confidence +/-	Prob
OK	9.9991E06	2.5300E03	9,18%
F	1.9930E-01	7.1528E-04	0,00%
IF	0.9890E8	2.5274E05	90,80%
FNI	2.4996E04	6.3255E00	0,02%

surprised?

Figures of Interest

- $MTTF$

  Mean Time To Failure
- $P(X)\,|_{t=9.000\,h}$

  one airbag fails after 9000 hours of operation
- $EX_F = P(X)\,|_{t=9.000\,h}\cdot N_0$

  Expected number of failing ECU assuming N_0 are in operation
- $P_{\geqslant 1} = 1-(1-P(X)|_{t=9.000\,h})^{N_0}$

  at least one of N_0 ECU fails during T_0 of operation

BOSCH

Simulating the Model

- Simulation over 10^{15} time units
- 60 million batches

Variable	Mean Value	Confidence +/-
MTTF	1.0108E09	9.9995126E06
P(X)|t=9.000h	1.6666E-08	3.2666666E-08
EX_F	0.5	
P>=1	0.3935	

No Confidence
at 95% reached

BOSCH

Markov Chain Representation

$$
\mathbf{Q} = \begin{bmatrix} \mathbf{T} & T^0 \\ \mathbf{0} & 0 \end{bmatrix}
$$

$$
= \begin{pmatrix} -\sum & \lambda_{FI} & \lambda_I & \lambda_{FNI} & \lambda_{TFE} \\ \mu & -\sum & 0 & \lambda_I + \lambda_{FNI} & \lambda_E \\ 0 & 0 & -\sum & \lambda_{FI} + \lambda_{FNI} & \lambda_{TFE} \\ 0 & 0 & 0 & -\sum & \lambda_E \\ 0 & 0 & 0 & 0 & -\sum \end{pmatrix}
$$

(X)

Interest in transient probabilities of being in state (X) at any time.

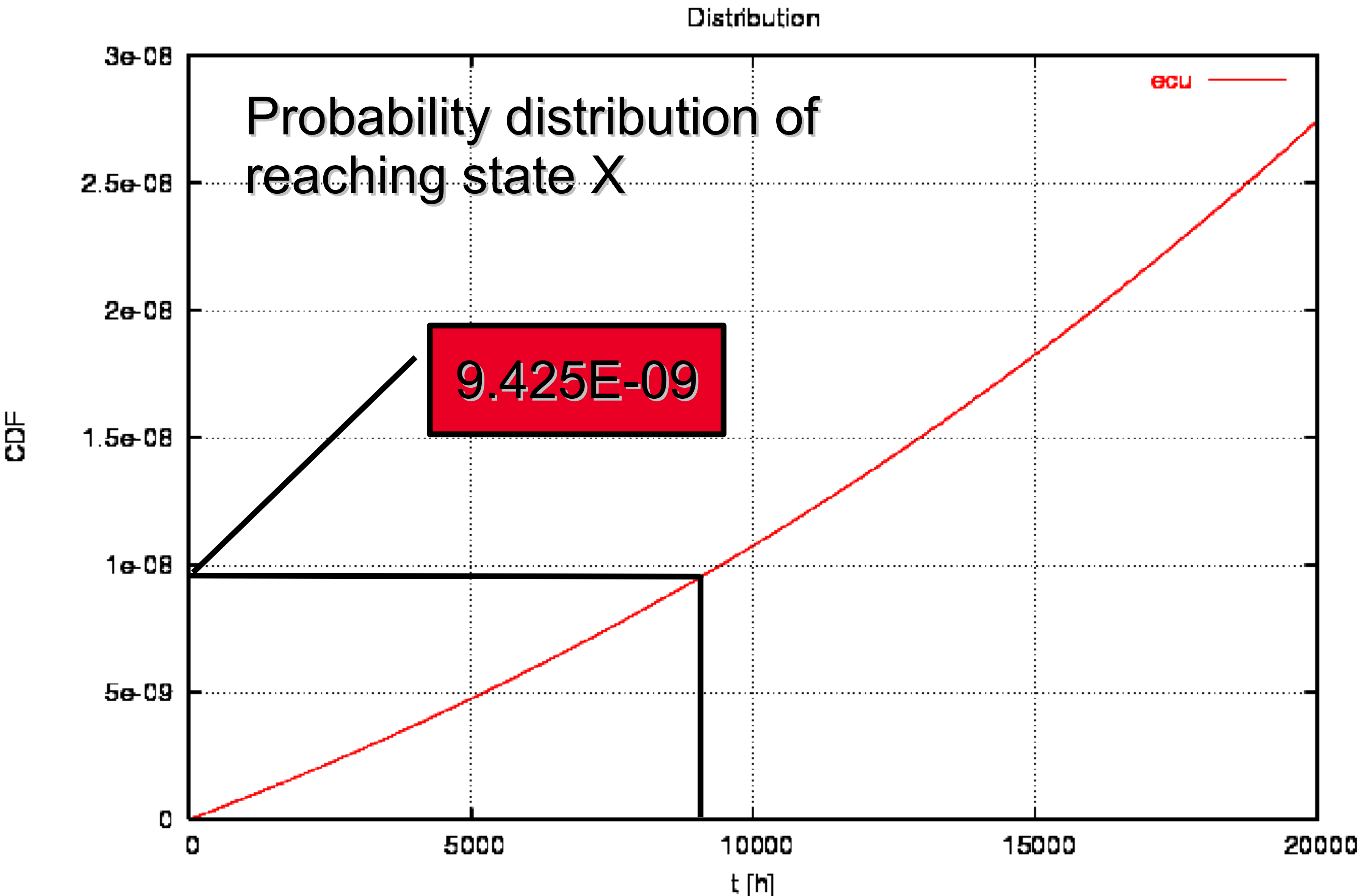
Phase Type distribution with representation (α, T) where

$$\boldsymbol{\alpha} = \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}$$

and

$$\mathbf{T} = \begin{pmatrix} -\sum & \lambda_{FI} & \lambda_I & \lambda_{FNI} \\ \mu & -\sum & 0 & \lambda_I + \lambda_{FNI} \\ 0 & 0 & -\sum & \lambda_{FI} + \lambda_{FNI} \\ 0 & 0 & 0 & -\sum \end{pmatrix}$$

Markov Chain Analysis



BOSCH

Comparison of Simulation and Analytical Results

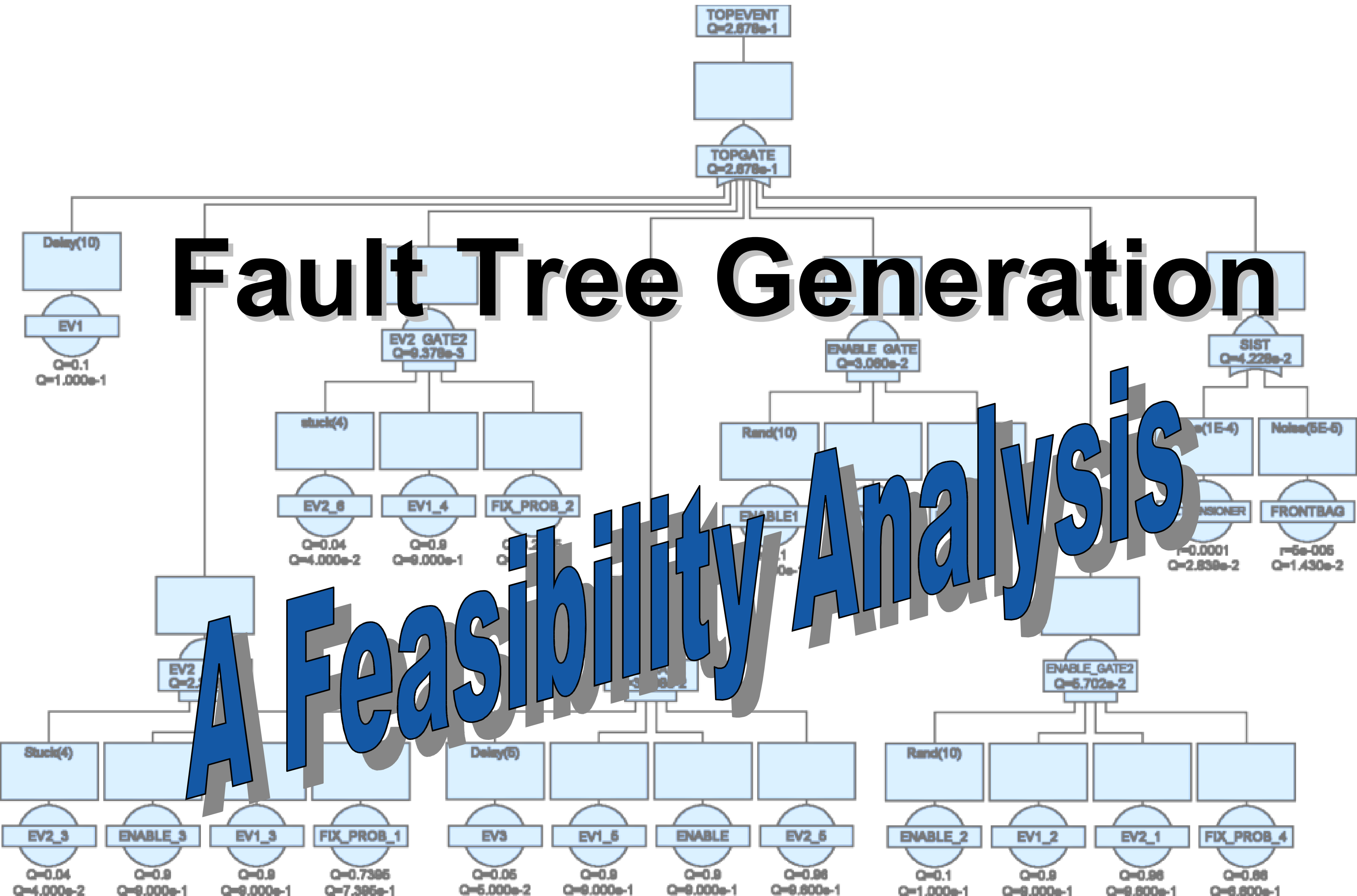
Variable	Simulation	Analytical
$P(X)\|_{t=9.000h}$	1,667E-08	9,425E-06
EX_F	0,5000	0,2828
$P_{>=1}$	0,3935	0,2463

Conclusion

- Way to simulate Markov Chains
- Drawback of simulation especially when dealing with small rates (many samples needed to reach confidence level)
- A long time spend in state IF (indication failure)

Fault Tree Generation

A Feasibility Analysis

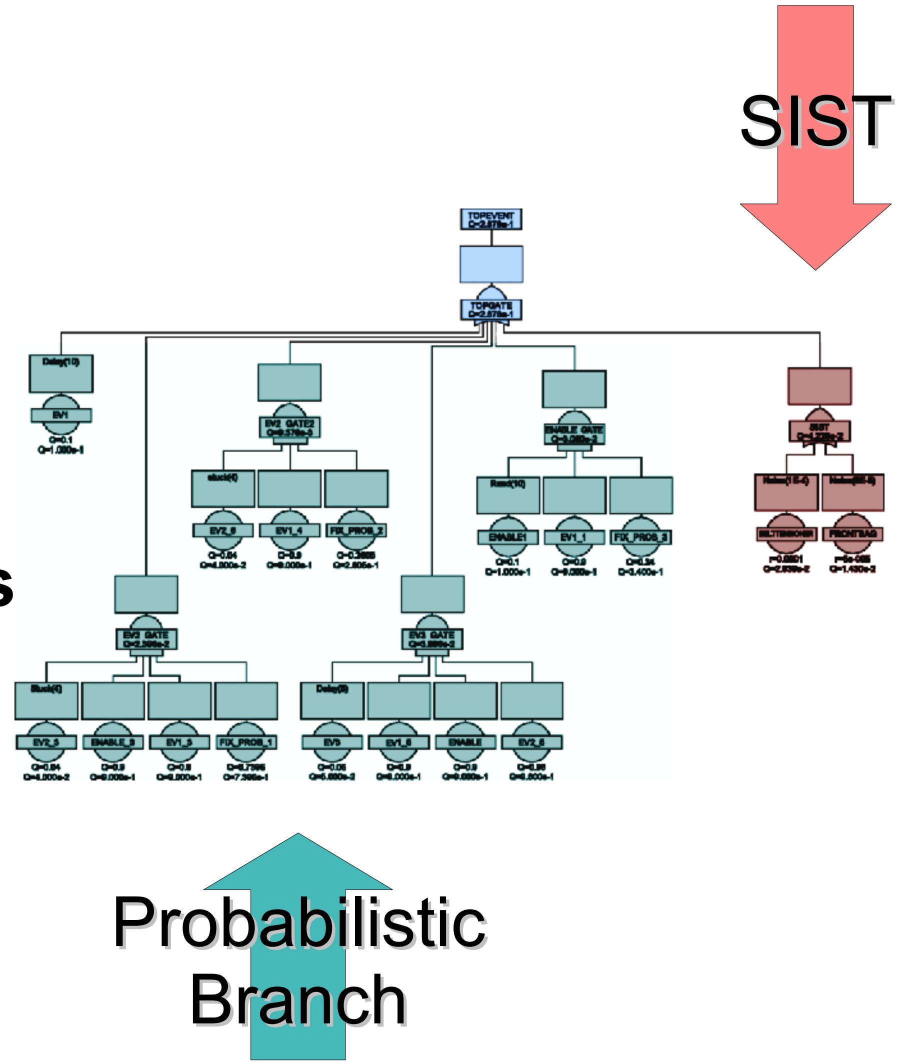


Fault Tree Generation

- Automate failure analysis of complex systems
- Generation of Fault Trees (FTs) by Simulation in MoDeST
- Using Separated Stochastic Independent Subtree (SIST) for stochastic components
- Implement preprocessor translations for errors using **gema**

Fault Tree Generation

- **Modularize Event Groups**
- **Simulate Probabilistic Part for one time unit.**
- **Add exponential events to SIST**



BOSCH

Insertion of Errors into the Simulation Model

Probabilistic Errors for Actions

- `delay(a,p,t,clk)`
- `stuck(a,p)`

Probabilistic Errors for Variables

- `noise(x, p, m)`
- `rand(x, p, m)`

Preprocessor Translation:

```
delay(a,p,t,clk) ::=
  float a_timer;
  exception a_error;
  {= a_timer=t+clk =};
  palt{
      :p:when(clk==a_timer)
        urgent(true) throw a_error
      :1-p: tau
      }; a
```

BOSCH

Errors injected into the Model

Probabilistic Events

	Error Type	Pr.	Values	Delay
Enable	RAND	10	[0,1]	
Event1	DELAY	10		5
Event2	STUCK	4		
Event3	DELAY	5		19

SIST

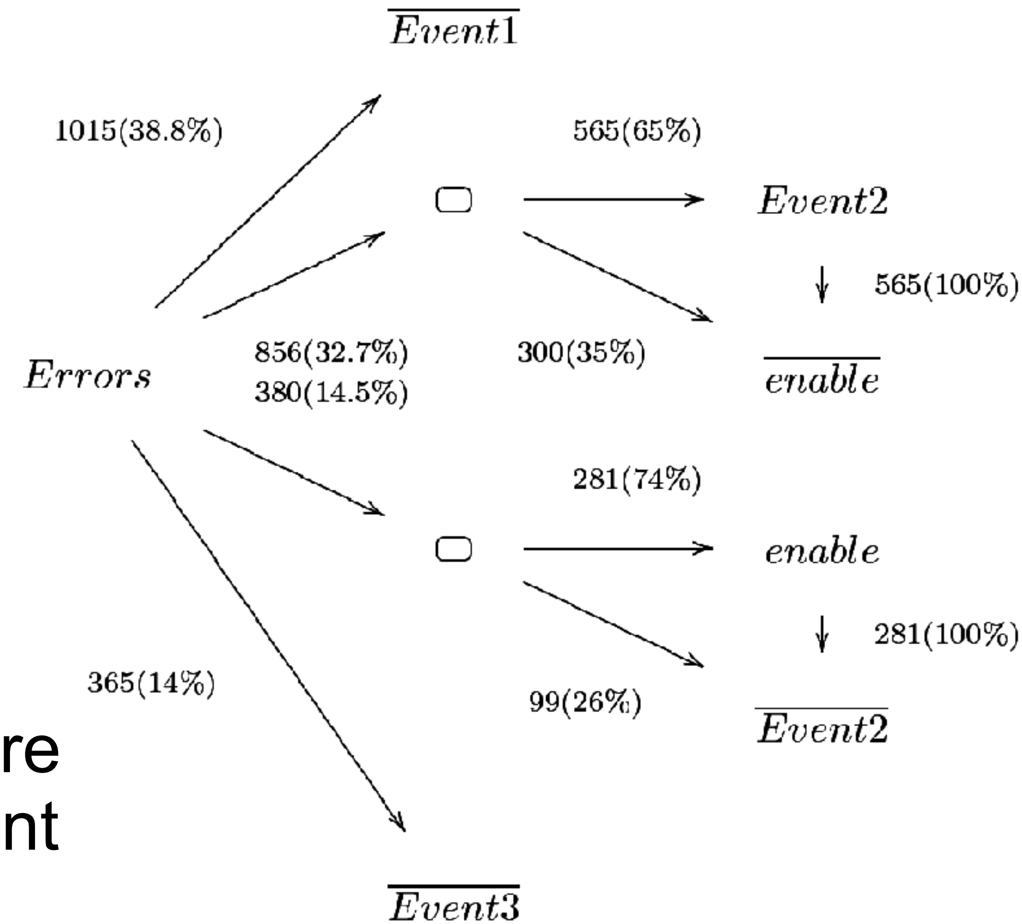
	Lambda	Delay	Magnitude
BT Delay	1E-04	60	30
FB Delay	5E-05	200	100

Simulation of the Behavior Model

- 10.000 batches
- Simulating each batch for 1 time unit (cf. SIST)

Use Trace to identity Error Events w.r.t.

- their Frequency
- their Order

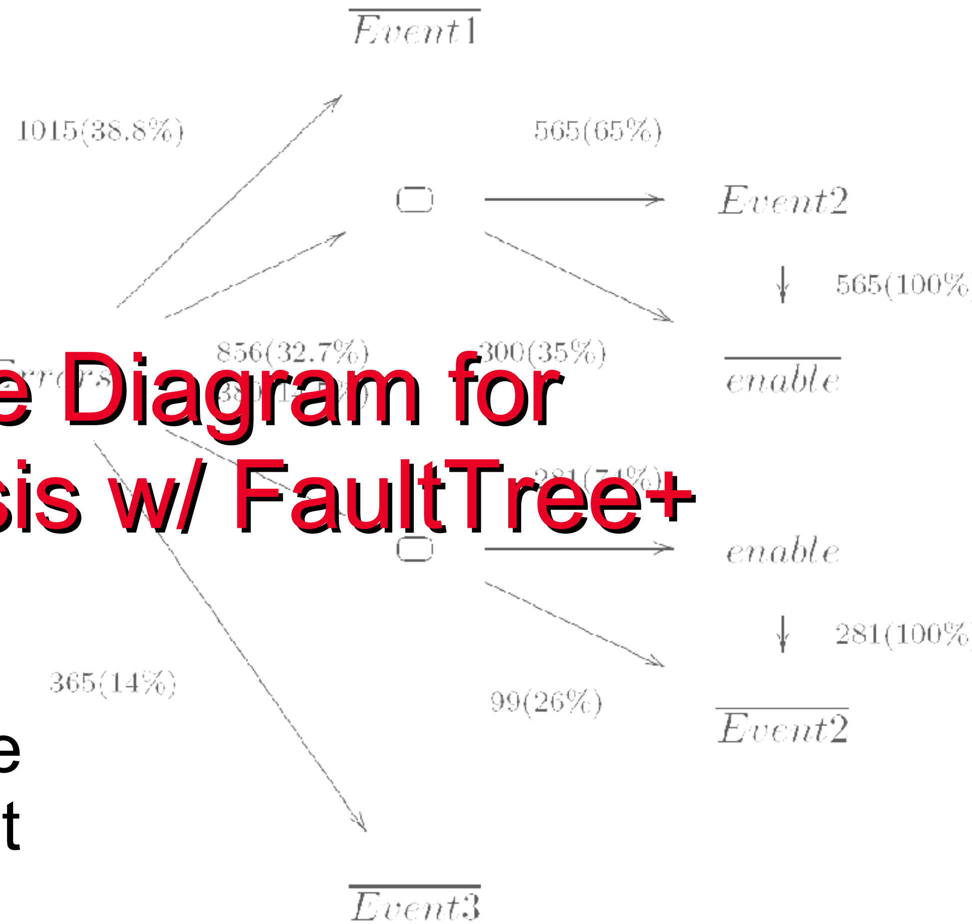


$\overline{Event}$ denotes failure of action/variable Event

BOSCH

Use Trace to identity Error Events w.r.t.

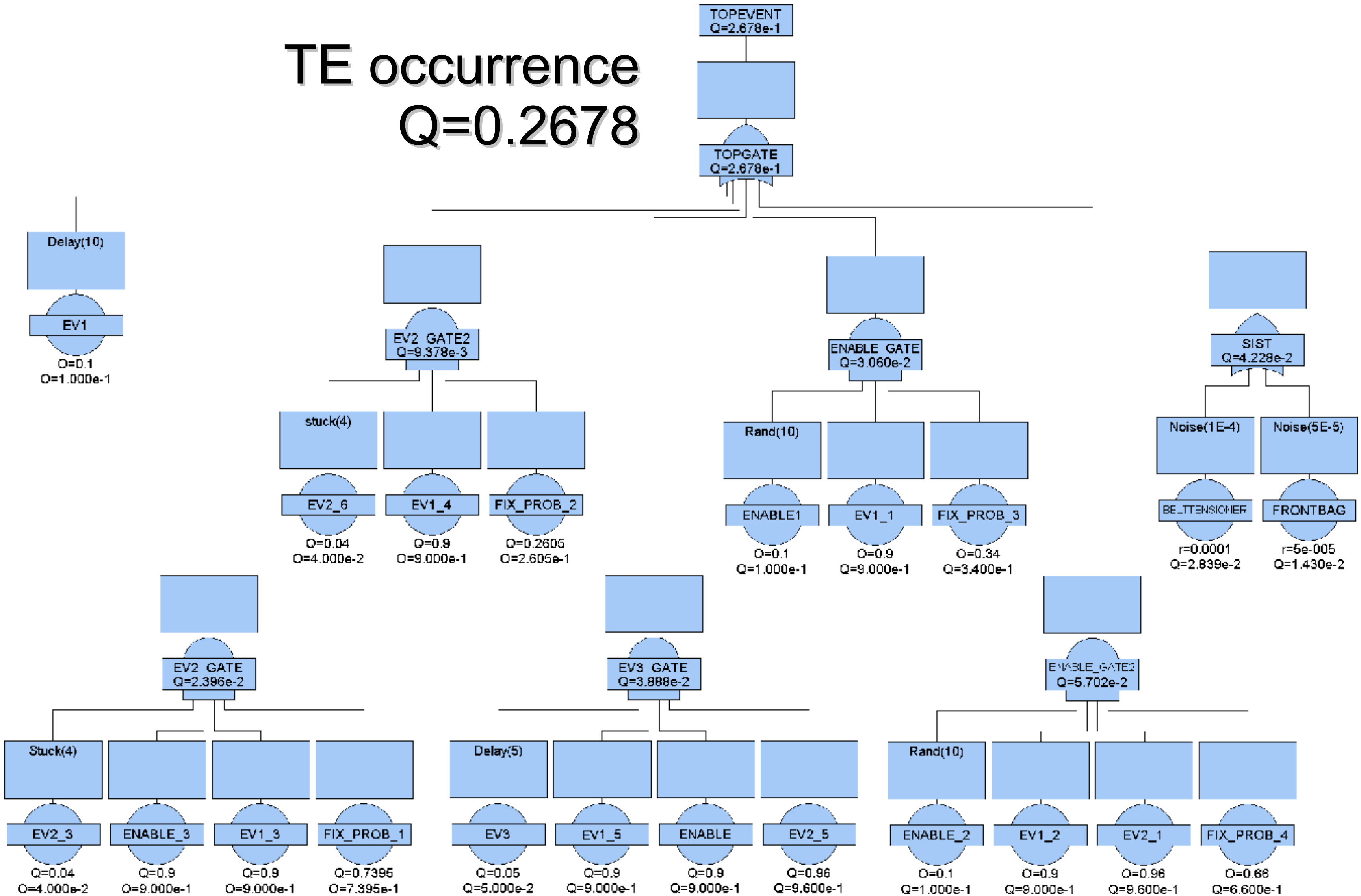
- their Frequency
- their Order



Use Failure Diagram for Failure Analysis w/ FaultTree+

$\overline{Event}$ denotes failure of action/variable Event

TE occurrence
Q=0.2678



BOSCH

Advantages

- Feasibility of automating Fault Tree Generation
- Discrete Event simulation over 1 time unit is fast
- Traces not see during simulation hardly contribute to the final TE probability
- Reflect realistic model behavior including recovery

Drawbacks

- Analysis of the simulation trace was done by hand
  → need for appropriate parsing mechanism

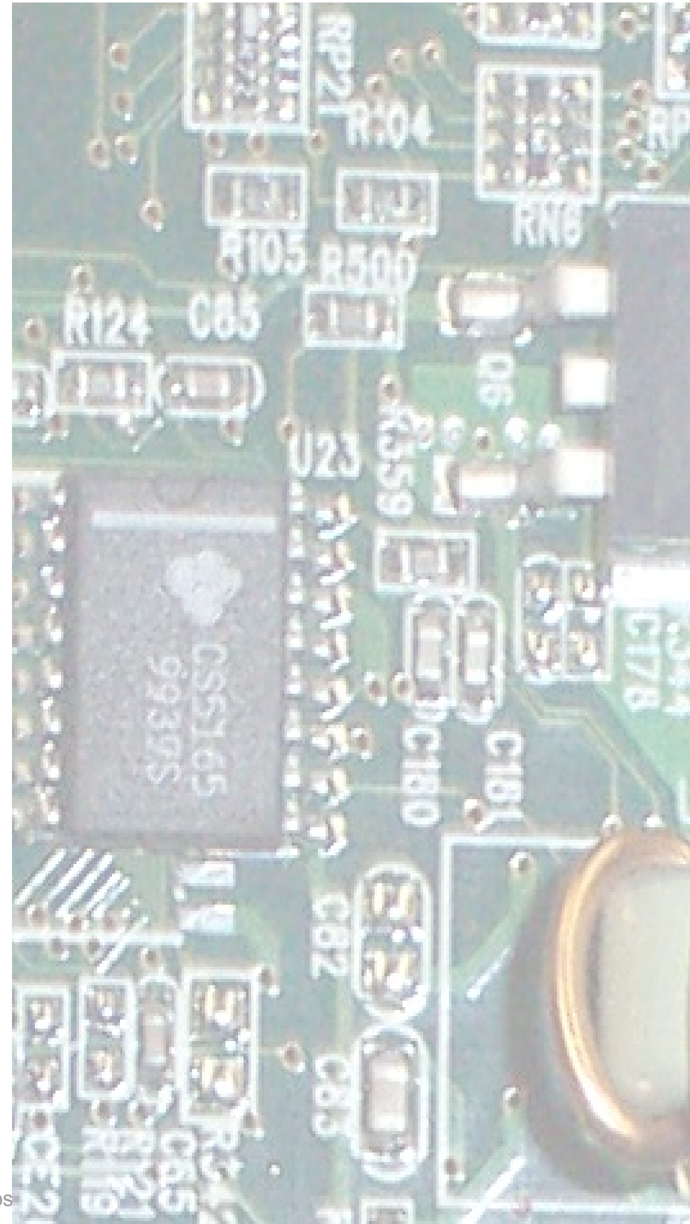
Importance Analysis

Diagnostic Importance Measure

BOSCH

Safe and Failure Critical Systems

- Identity circuits in systems which have the greatest impact on the proper function of the system
- Identify gates within a circuit using Fussel-Vesely



BOSCH

Indicate an event's contribution to the system unavailability

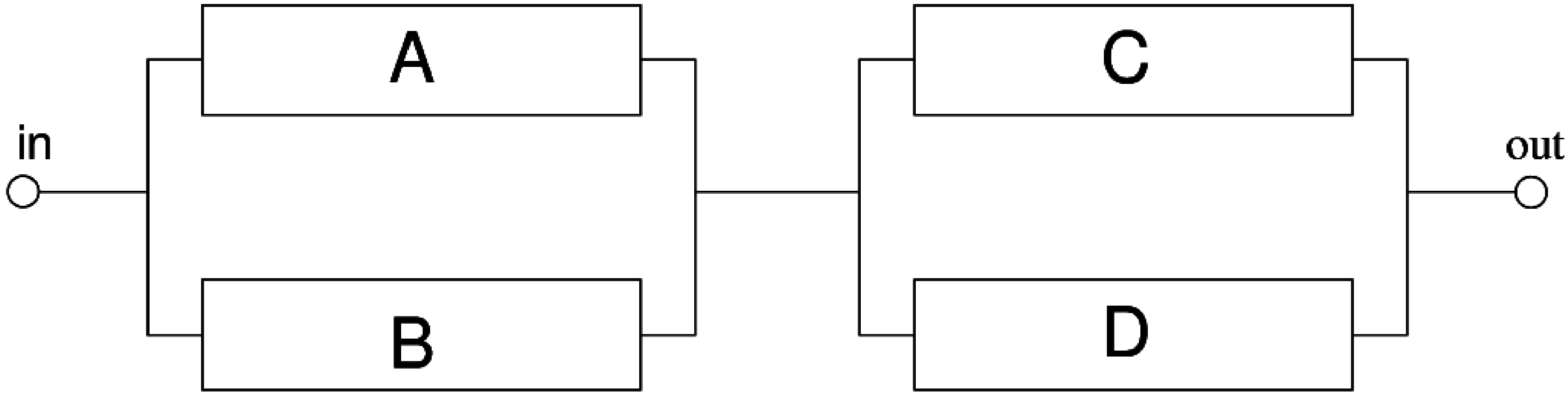
$$I_D(i) = \frac{Q_{system} - Q_{system}(q_i = 1)}{Q_{system}}$$

Q_{system} Probability of the System failing

$Q_{system}(q_i = 1)$ Probability of gate i failing

BOSCH

Example



Obtaining the Diagnostic Importance via simulating 1 million batches

	Prob(i)	Sim DI(i)	FT+ DI(i)
A	1E-01	0,8306	0,8830
B	2E-02	0,1524	0,1670
C	3E-03	0,0031	0,0060
D	5E-01	0,9940	0,9940

Conclusion

- A way to determine the Diagnostic Importance of gates within a circuit using simulation
- When using small rates many simulation runs are required

Nuts and Bolts of Simulation

- Versatile method for expressing complex systems like STA (Verification, Markov Chains, Failure Analysis, Importance Analysis)
- Even more powerful having a parser at hand
- Alternative approach to analyze system where state exploration fails
- When having small rates (<1.0E-08) many batches are needed for confidence
- No exhaustive state exploration

References

[Aue05] Marko Auerswald. SRS ECU Behavior Modeling, IST Project AMETIST. Technical report, Bosch Industrial Case Study, March 2005.
[BHB+04] Matthias Bretschneider, Hans-Jürgen Holberg, Eckard Böde, Ingo Brückner, Thomas Peikenkamp, and Karriet Spenke. Model-based Safety Analysis of a Flap Control System. Technical report, ICOSE 2004
[BHKK03] Henrik Bohnenkamp, Holger Hermanns, Jost-Pieter Katoen, and Ric Klaren. The MoDeST Modeling Tool and Its Implementation.
[Bra01] Kraftfahrzeugtechnik, volume 2. Vieweg Handbuch, 2001.
[Buc00] Kerstin Buchacker. Definition und Auswertung erweiterter Fehlerbäume für die Zuverlässigkeitsanalyse technischer Systeme. Dissertation, Universität Erlangen, Juli 2000. Band33, Nummer 3
[DHKK01] P.R. D'Argenio, H. Hermanns, J.P. Katoen, and R. Klaren. MoDeST - a modelling and description language for stochastic timed systems.
[Fau05] FaultTree+. Handbook and Demo-Licence, Version 11.0

References

[Fau05] FaultTree+. Handbook and Demo-Licence at www.isograph -software .com, Version 11.0

[Gra03] David. N. Gray. Gema - a general purpose macro processor.

[Hub03] Thomas Huber. Zuverlässigkeit und Ausfallsicherheit elektronischer Systeme im Automobil. Technical report, Robert Bosch GmbH, IIR-Konferenz, 2003. Safety First - For Intelligent Restraint System Technologies.

[LPY97] Kim G. Larsen, Paul Pettersson, and Wang Yi. Uppaal in a Nutshell. Int. Journal on Software Tools for Technology Transfer, October 1997.

[MMT00] J. Mauss, V. May, and M. Tatar. Towards Model-based Engineering: Failure Analysis with MDS. Technical report, ECAI-2000 Workshop on Knowledge-Based Systems for Model-Based Engineering, August 2000.

[Moc05] Ralf Mock. Risiko und Sicherheit von Netzwerken. Technical report, Laboratorium für Sicherheitsanalytik, Juni 2005. lecture notes.

[Neu81] Marcel F. Neuts. Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach. The John Hopkins University Press, 1981.

End

Thanks for paying attention

Q&A