Hazard-based Selection of Test Cases

Functional Safety of Mechatronic Systems

Mario Gleirscher

Software & Systems Engineering
Institut für Informatik
Technische Universität München

May 24, 2011

Safety Case[1]: Assurance of an Airbag Control

Machine I: An airbag system . . .

[1] Cf. Safety case management [Kel98]
[2] Cf. [Wik11]

Safety Case[1]: Assurance of an Airbag Control

Machine I: An airbag system . . .

Safety Case G: Does the airbag release iff it's intended?

[1] Cf. Safety case management [Kel98]
[2] Cf. [Wik11]

Safety Case[1]: Assurance of an Airbag Control

Machine I: An airbag system ...

Context E: ... in a car operated out in a street by a human driver.

"... functional safety methods have to extend to non-E/E/PS parts of the system ..."[2]

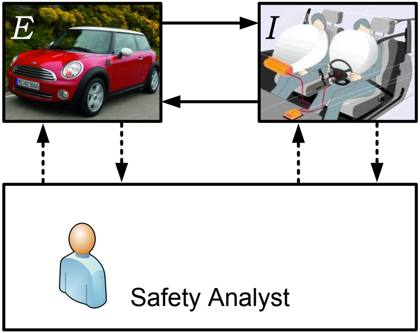
"... functional safety can[not] be determined without considering the environment ..."[2]

[1] Cf. Safety case management [Kel98]
[2] Cf. [Wik11]

1. Functional Safety
   - System Modelling
   - Property Analysis and Specification
2. Hazards
   - Property Analysis and Specification
   - Test Case Selection
3. Conclusion

A System Model M_W of the Airbag World W

Functional[1] model M_W:

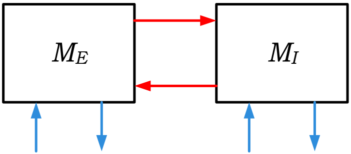


M_I describing the mechatronic system I and
M_E describing its operational environment E.

[1] Cf. [Bro10].

A System Model M_W of the Airbag World W

A system boundary allows interaction across shared phenomena[1]:



$M_E \blacktriangleright M_I \triangleq$ repaired(Airbag), refilled(Gas), signal(activate,Airbag), on(crashSensor), ...

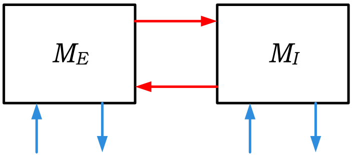
$M_E \blacktriangleleft M_I \triangleq$ released(Airbag), ...

where $A \blacktriangleright B = ctrVar(A) \cap monVar(B)$.

[1] Cf. [Jac01, PM95]

A System Model M_W of the Airbag World W

Supportive phenomena for safety modelling and measurement:

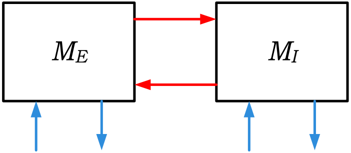


$M_E \setminus M_I \triangleq$ crashed(Car), shocked(Car), deformed(Car), protected(Person), driving(Car), irritated(Passenger), ...

$M_I \setminus M_E \triangleq$ empty(Airbag), ...

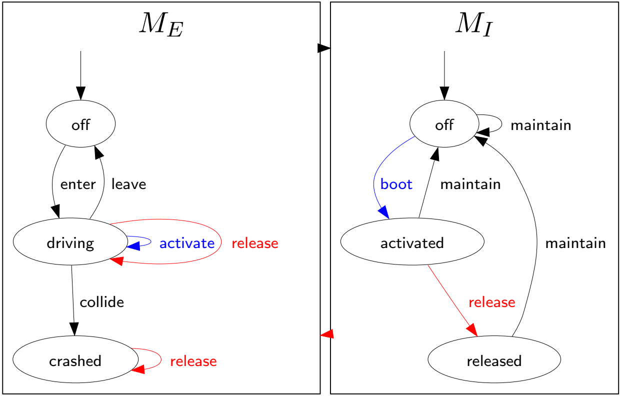
A System Model M_W of the Airbag World W

Interface behaviour $\triangleq$ histories of shared phenomena states:



Intervals	...	n	...	...	$n+j$	...	m	$\longrightarrow$
shocked(Car)	F	T	T	F	F	F	F	...
deformed(Car)	0	2	10	10	10	10	10	...
crashed(Car)	F	F	F	T	T	T	T	...
signal(crash)	F	F	F	T	T	T	T	...
released(Airbag)	F	F	F	F	T	T	T	...
...	...	...	...	...	...	...	...	...

M_W as a Test Model



Independent control states, transitions with action preconditions and effects[1].

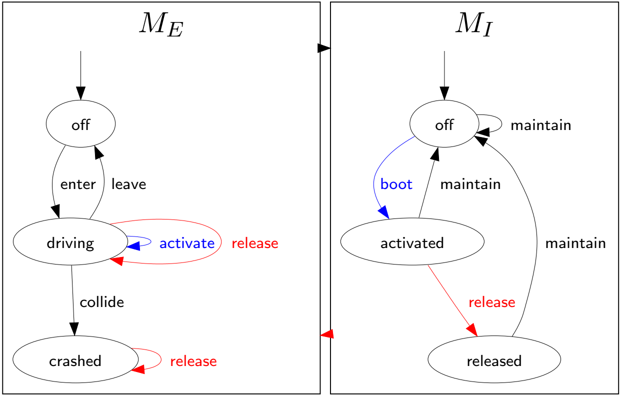
Where to get the information?

System use cases $\rightarrow M_I, M_E$

Domain and context analysis $\rightarrow M_E$

[1] Details in GOLOG script, cf. [Rei01].

M_W as a Test Model

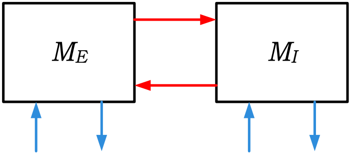


Independent control states, transitions with action preconditions and effects[1].

Problem: Which of M_W's possible or mutated transitions may obstruct safety in M_E?

[1] Details in GOLOG script, cf. [Rei01].

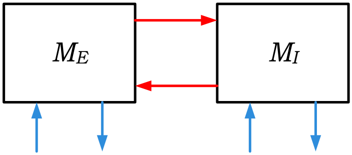
Functional Safety in M_W



Functional safety goal[3]	Behavioral property to globally maintain (or avoid) in E, formally: $\Box\phi$
$G \triangleq$	$\Box protected(Body)$
$G' \triangleq$	$\Box[crashed(Car) \rightarrow \Diamond_{<400ms} absorbed(Body) \wedge \blacksquare\neg crashed(Car) \rightarrow \Box\neg released(Airbag)]$

[3] Cf. [MP95].

Functional Safety in M_W



A/G safety specification: G split into Assumptions for E and Guarantees for I, formally: $\bigvee_i As_i \rightarrow Gr_i \models G$

$As_1 \triangleq$ $\Box[crashed(Car) \leftrightarrow \bullet signal(crash)]$
... "reliable crash sensing expected from E"

$Gr_1 \triangleq$ $\Box[signal(crash) \leftrightarrow \Diamond_{<200ms} released(Airbag)]$
... "reliable bag disengaging required from I"

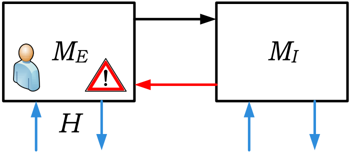
1. Functional Safety
   - System Modelling
   - Property Analysis and Specification

2. Hazards
   - Property Analysis and Specification
   - Test Case Selection

3. Conclusion

Obstacles[2] to Functional Safety in M_W

What obstructs a functional safety goal G in W?



Hazard H Risk of human or environmental harm in E

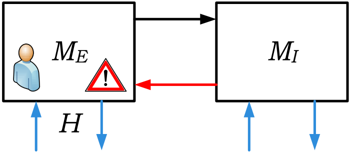
$H_1 \nrightarrow G'$ $\triangleq \Diamond[crashed(Car) \;\wedge\; \bullet harmed(Person)]$

$H_2 \nrightarrow G'$ $\triangleq \Diamond[\blacksquare\neg crashed(Car) \;\wedge\; \bullet harmed(Person)]$

[2] Automated inference possible, e.g. [Let01].

Obstacles[2] to Functional Safety in M_W

How can such obstructions happen in W?

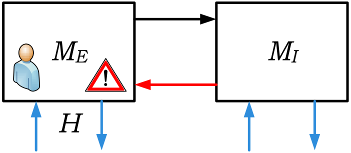


Hazardous state σ	State of M_E (or $M_E \cap M_I$) leading to H
$\sigma_{H_1 G}$	$\triangleq signal(crash) \rightarrow \neg released(Airbag)$
$\sigma_{H_2 G}$	$\triangleq \neg signal(crash) \rightarrow released(Airbag)$
$\sigma_{H_3 A}$	$\triangleq crashed(Car) \rightarrow \neg signal(crash)$
$\sigma_{H_4 A}$	$\triangleq \neg crashed(Car) \rightarrow signal(crash)$

[2] Automated inference possible, e.g. [Let01].

Obstacles[2] to Functional Safety in M_W

How can such obstructions be generated from M_W?

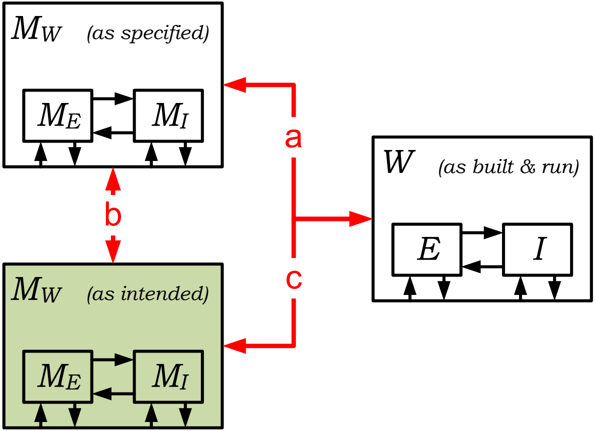


Hazardous state σ	State of M_E (or $M_E \cap M_I$) leading to H
$\sigma_{H_1 G}$	$\triangleq signal(crash) \rightarrow \neg released(Airbag)$
$\sigma_{H_2 G}$	$\triangleq \neg signal(crash) \rightarrow released(Airbag)$
$\sigma_{H_3 A}$	$\triangleq crashed(Car) \rightarrow \neg signal(crash)$
$\sigma_{H_4 A}$	$\triangleq \neg crashed(Car) \rightarrow signal(crash)$

[2] Automated inference possible, e.g. [Let01].

Defects concerning Functional Safety

Causes of (hazardous) system failures:



(Im)mature Specs

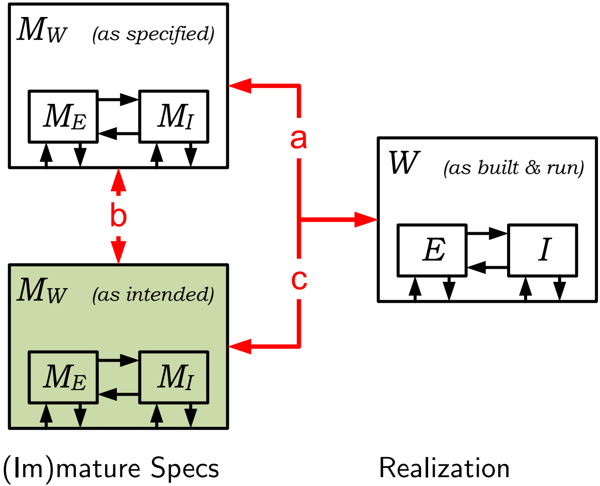
Realization

a) Potential bug or runtime error.

Assurance by system testing too weak and incomplete.

Defects concerning Functional Safety

Causes of (hazardous) system failures:

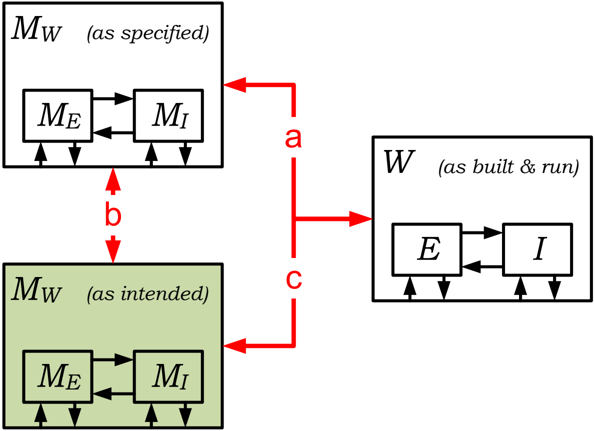


b) Requirements error, e.g. wrong assumption or guarantee; wrong, incomplete or missing transition.

Assurance by requirements validation.

Defects concerning Functional Safety

Causes of (hazardous) system failures:



(Im)mature Specs

Realization

c) **Bug or runtime error**.

Assurance by automated system testing strengthened by validation.

Assure Functional Safety G of a Machine I in a Context E

Constructive Safety Assurance (Requirements Engineer)

1. Safety risks: Does the airbag's behaviour cause hazards?
2. Hazardous exceptions: Is it completely specified?
3. Automation: How to systematically explore such situations?
4. How can they be avoided or kept at minimum risk?

Analytic Safety Assurance (Test Engineer)

1. Selection: How to test beyond the airbag's specification?
2. Coverage: Have all relevant situations be explored, i.e. does an airbag's realization exhibit hazardous behaviour?
3. How to mutate M_W to get interesting test cases?
4. How to automatically generate and execute them?

Assure Functional Safety G of a Machine I in a Context E

Constructive Safety Assurance (Requirements Engineer)

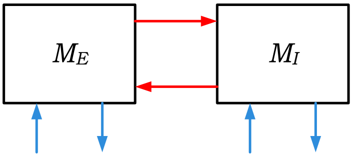
1. Safety risks: Does the airbag's behaviour cause hazards?
2. Hazardous exceptions: Is it completely specified?
3. Automation: How to systematically explore such situations?
4. How can they be avoided or kept at minimum risk?

Analytic Safety Assurance (Test Engineer)

1. Selection: How to test beyond the airbag's specification?
2. Coverage: Have all relevant situations be explored, i.e. does an airbag's realization exhibit hazardous behaviour?
3. How to mutate M_W to get interesting test cases?
4. How to automatically generate and execute them?

Hazard-based Test Case Specifications as Test Goals

Notions relevant for testing-based safety assurance:

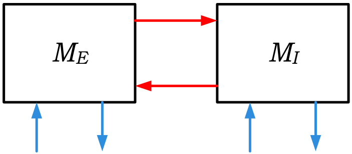


Test case t	$\triangleq$ action sequence possible in M_W
Test suite $\mathcal{T}$	$\triangleq$ set of test cases, e.g.: $\langle collide, release \rangle$, $\langle release, looseControl \rangle$, ...
Test case specification[3] τ	$\triangleq$ state expression over phenomena capturing a test goal

[3] Cf. [Bri10, Pre03].

Hazard-based Test Case Specifications as Test Goals

Specifying negative test cases t based on a hazardous state σ:



Informal: Are there test sequences based on M_W that exhibit unwanted airbag behaviour?

Formal:
$$\tau_1 \triangleq (\exists t).\sigma_{H_2G} \models H$$
$$\triangleq (\exists t).\neg signal(crash, t) \rightarrow released(Airbag, t)$$

Validate M_W and Generate Test Cases

Generate test cases of length 7 in GOLOG:

```
propOfInterest(T) :- not(signal(crash,T)),
  released(Airbag,T).
do(testcontrol(7),s0,T), propOfInterest(T).
```

The selection results in a suite $\mathcal{T}_{\tau_1}$ leading to σ_{H_2G}, e.g.:
$\cong \langle activate, boot, collide, activate, release\rangle$

```
T = do(step, do(release(airbag1),
      do(step, do(activate(airbag2),
        do(step, do(collide(_G110, _G111),
          do(step, do(boot(airbag1),
            do(step, do(activate(airbag1), s0))))))))))
```

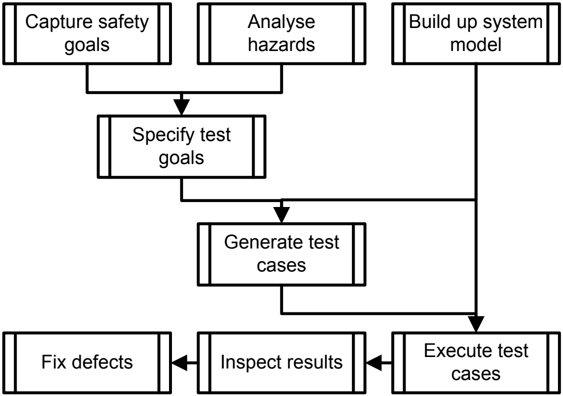
Local coverage yields all paths in M_W to σ_{H_2G}.

1. Functional Safety
   - System Modelling
   - Property Analysis and Specification

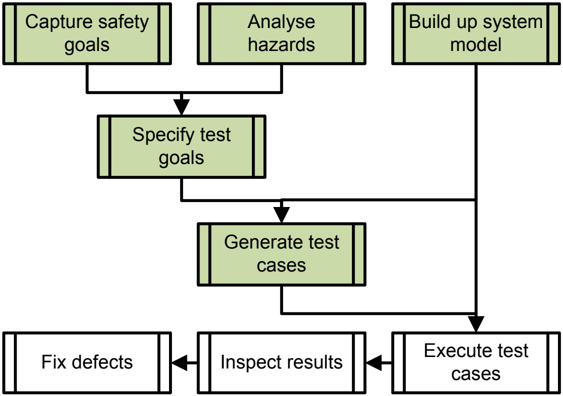
2. Hazards
   - Property Analysis and Specification
   - Test Case Selection

3. Conclusion

A Strategy to select Safety-critical Test Cases



A Strategy to select Safety-critical Test Cases



Further Work

1. Treatment of sets of safety goals or A/G safety specifications,
2. Isolated assurance of a feature,
3. Exploration of hazard mitigation patterns for defect removal, cf. [Gle11].

Contribution to Solving AST Model Problems[3] . . .

. . . REQ 1&2:	How to cover safety requirements by tests?
. . . INT 8:	How to observe architecture to test for functional safety defects?
. . . INT 10:	How to test for hazards?

[3] Cf. Architecture Support for Testing (AST) Model Problems at http://labsewiki.isti.cnr.it/projects/ast/ast2011pisa/main.

References I

[Bri10] Ed Brinksma.
Model-based testing.
volume 31 of *NATO Science for Peace and Security Programme*, Marktoberdorf, 2010.

[Bro10] Manfred Broy.
A logical basis for component-oriented software and systems engineering.
The Computer Journal, 53(10):1758–82, 2010.

[Gle11] Mario Gleirscher.
Hazard-based Selection of Test Cases.
In *Proc. 6th ICSE Workshop on Automation of Software Test (AST'11)*, 2011.

[Jac01] Michael Jackson.
Problem Frames: Analysing & Structuring Software Development Problems.
Addison-Wesley Professional, 2001.

[Kel98] Timothy Patrick Kelly.
Arguing Safety – A Systematic Approach to Safety Case Management.
PhD thesis, University of York, Dept. of Computer Science, 1998.

References II

[Let01] E. Letier.
Reasoning about Agents in Goal-oriented Requirements Engineering.
Thèse de Doctorat en Sciences Appliquées, Université Catholique de Louvain, 2001.

[MP95] Zohar Manna and Amir Pnueli.
Temporal Verification of Reactive Systems: Safety.
Springer, 1st edition, 8 1995.

[PM95] David Parnas and J. Madey.
Functional Documentation for Computer Systems.
Science of Computer Programming, 25:41–61, Octobre 1995.

[Pre03] Walter Alexander Pretschner.
Zum modellbasierten, funktionalen Test reaktiver Systeme.
Dissertation, Technische Universität München, Faculty of Informatics, 2003.

[Rei01] Raymond Reiter.
Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems.
MIT Press, 2001.

References III

[Wik11] Wikipedia.
Functional safety — wikipedia, the free encyclopedia, 2011.
[Online; accessed 15-May-2011].