Combining

STAMP/STPA

and

Assurance Cases

Stefan Wagner

Institute of Software Technology

STAMP Workshop 2012
Cambridge, MA
19 April 2012
Automobiles evolve...
to electronic systems

Size of software

1970

1980

1990

2000

Degree of interconnection
ISO 26262: Road vehicles – Functional safety

Management of functional safety

Product development: system level

Product development: hardware level

Product development: software level

Production and operation

Supporting processes

ASIL-oriented and safety-oriented analyses

Requires safety case
Safety requirements & objectives

Safety argument

Safety evidence

Safety case
G1  System is safe

G2  All relevant faults considered

C2  Fault Hypothesis

C4  Model of system

C5  Model of environment

C3  Hazard Identification

S1  Fail-safety ensured for fault hypothesis

G3  Faults never lead to failures

G4  Hazards always caused by failures

S2  Hazard-safety ensured for fail-safety

S1  Fail Safety Characterization

S2  Hazard-safety ensured for fail-safety

Sn1  Use of fault pattern libraries

Sn2  Testing using fault injection

Sn3  Formal verification

Sn4  Simulation of model of environment

Sn5  All relevant hazards considered

Sn6  Solution

Sn7  Strategy

Sn8  Context

G  (Sub-)Goal
Cruise control case study with MAN
Abbildung 3: Mittelkonsole mit elektrifiziertem Gangwahlschalter

• Die Erkennung der Tippposition führt zur Abfrage der Motorkupplung.
• Bei erfolgreicher Ermittlung der Tippposition und des Signals Kupplung offen erfolgt die Ansteuerung des Getriebeaktuators.
• Nach der Rückmeldung Gang eingelegt des Positionssensors am Getriebeaktuator wird die Sperrwirkung im GWS durch Stromversorgung aufgehoben (unabhängig von Kupplungssignal).
• Anschließend rastet der Knauf in der Schaltposition ein, die Funktionsbeleuchtung die Ganganzeige im Kombi werden angesteuert.
• Die Sperrwirkung des GWS wird nach der Erkennung der Rastposition des Hebels durch Wegnahme der Stromversorgung wieder hergestellt, damit kein Dauerverbraucher entsteht.
• Der aktuelle Gang wird in Knauf und Kombi angezeigt.
• Die Zeit zwischen der Schaltwunsch-Übermittlung (Knauf in Tippposition bewegt) und der Freigabe der Sperrwirkung soll kleiner als 50ms sein.

Gang Auslegen:
• Knaufposition <> N.
• Die Sperrwirkung wird durch Stromversorgung aufgehoben, sobald die Kupplung getreten ist.
• Bei Verlassen der Rastposition oder Erkennen der Knaufposition = N erfolgt die Übertragung des Fahrerwunsches an den Aktor.

Shift by wire case study with BMW
Safety cases are good for a structured argumentation.

STAMP/STPA are good for a systematic analysis.
Example: The part of the safety case in Figure 10 applies the pattern to the goal that demands to avoid a situation in which the constraint is violated to keep acceleration larger or equal than a lower bound. We simplified this by transforming it to the goal to avoid an acceleration which is smaller than the lower bound.

Avoid situation: violation of \((\text{acceleration} \geq \text{lowerBound})\)

**G1**

Pattern: logical transformation

\[ \text{Resolve } \langle \text{violation of} \rangle \]

Avoid situation: \(\text{acceleration} < \text{lowerBound}\)

**G1.1**

Fig. 10. An Example Usage of the Pattern: Logical Transformation

Usage: We used it another three times in the cruise control safety case for resolving complex goals for combined situations.

5.4.6 Pattern: Formal Elicitation on Model

Context: There is a (formal) behaviour model of the (sub-) system and we have goals for the complete system or sub-system.

Problem: How can we break down the goal to more tangible goals?

Forces: This requires detailed and complete behaviour models, which might not be available for all considered sub-systems. Moreover, it needs a complicated model inspection or sophisticated technical support to perform the elicitation correctly.

Solution: This pattern is an extension of the split by architecture pattern. It solves the same problem but in the context that a formal behaviour model is available. While in split by architecture only the structure of the architecture is employed to decompose the goal, we use the whole information in the model here to understand what influences the goal. We can do this either by a model inspection or automatic techniques, if they are available for the analysed model. In the safety case, we reference the used models in a model node next to the justification node.

Example: We elicit the two possible behaviours so that the car avoids the situation that the current speed is higher than the target speed from the environment, driver and car concept models in Figure 11.

Usage: We often used to elicit the causes that influence the requirement.

5.4.7 Pattern: Identification

Context: The main problem in arguing safety is to show that one has identified all relevant hazards and that none of these can occur. This pattern formalises this approach.

Problem: How can I show that no hazards occur?

Forces: Despite using this pattern, there is no guarantee that you actually identify all hazards.

Solution: We split the goal that no hazards occur into the two sub-goals Identify all hazards and No identified hazards can occur. For these, we then have to argue further.

Example: The example in Figure 12 is from the shift-by-wire case, where we used it on the top level to structure the argumentation.

No hazards occur

**G1**

Identify all hazards

**G2**

Identify all hazards && no identified hazards can occur

**S1**

Identify all hazards && no identified hazards can occur

**J1**

Identification

**G3**

No identified hazards can occur

Usage: We applied the pattern twice in the cruise control case in similar contexts.

5.4.8 Pattern: Failed Expectations

Context: Arguing safety always includes identifying hazards.

Problem: How can I elicit relevant hazards?

STPA

STAMP
Example hazard identification

H-1: Gear for wrong direction
H-2: Shift to unsuitable gear for speed

S1
Use STPA to identify hazards
## Example hazard analysis

<table>
<thead>
<tr>
<th>Control Action</th>
<th>Not Given or not Followed</th>
<th>Given Incorrectly</th>
<th>Wrong Timing or Order</th>
<th>Stopped Too Soon</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shift to gear</td>
<td>Controller does not shift gear to change direction</td>
<td>Controller shifts despite no lever change</td>
<td>Shift too late so that driver opens clutch</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Shift despite no clutch</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Shift despite unsuitable speed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Display gear</td>
<td>Controller does not send new direction to display</td>
<td>Sends wrong gear to display</td>
<td>Not hazardous</td>
<td>–</td>
</tr>
</tbody>
</table>

S2

Use STPA to identify causes for hazards
Example hazard avoidance

Shift controller (software)

Process: Shift/Gear

Actuators

Sensors

S3 Check speed from ICM

Speed

Lever change
Clutch activation

Clutch open
Clutch closed

Shift to gear x
Display gear y

Shift position x
In gear y

Gear position
Gear display

Display gear y
A final step in STPA is to consider how the designed controls could degrade over time and to build in protection against it.

–Leveson (2011)
Example degradation protection

Shift controller (software)

Shift to gear x
Display gear y

Actuators

Speed

Display no gear

S3
Show gear also by lever position

Sensors

Shift position x
In gear y
Clutch open
Clutch closed

Process: Shift/Gear

Speed

Lever change
Clutch activation

Gear position
Gear display
Safety case modules

- Environment and user
  - Concept level
  - Integration level
  - Function level
- Process
  - Process safety argument

Level of detail
Development domain
STAMP hierarchical structure

Safety case modules
Example structure mapping

Engineering

Development

Driver

Automated Controller

Driving

Operations
Process models

Sn1
Inspection in model

Sn2
Formal verification
1. Hazard identification and avoidance
2. Degradation protection
3. Structure
4. Models
A perfect combination of analysis and argumentation