Managing complexity –
Keep system development on track using SysML

MedConf 2012 – Munich, 26 Sep 2012

Mechatronic AG
Thomas Jetter, Thorsten Nieß
Mechatronic AG

- Founded in 1987
- 70 Employees
- Industry: Medical devices
- Service portfolio for devices and components:
  - Development
    - Software
    - Electronics
    - Mechanics
  - Manufacturing

Höhn (Subsidiary)

Darmstadt (Head office)
Motivation
History

- 2007: First discussions
- 2008: Evaluation of SysML
- 2009: First use in pilot project
- 2010: First presentation on MedConf
- 2012: Standard for new projects
Example:

Sting healer
Approach/Agenda

- Generate system context
- Define use cases
- Develop system architecture
- Risk analysis on basis of system architecture
- Derive sub system requirements
- Plan tests
- Generate documents
- Best practices
### Capture product requirements (import)

<table>
<thead>
<tr>
<th>#</th>
<th>ID</th>
<th>Name</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PRS:1</td>
<td>Heat up plate</td>
<td>The device shall heat up the heat plate to operating temperature.</td>
</tr>
<tr>
<td>2</td>
<td>PRS:2</td>
<td>Automatic production test</td>
<td>During manufacturing the device shall be testable via an Automatic Testing Equipment (ATE).</td>
</tr>
<tr>
<td>3</td>
<td>PRS:3</td>
<td>Keep operating temperature</td>
<td>The device shall keep the temperature of the heat plate at operating temperature for 10 s.</td>
</tr>
<tr>
<td>4</td>
<td>PRS:4</td>
<td>Switch off heating</td>
<td>The device shall switch off the heating after the operating temperature has been kept constant for 10 s.</td>
</tr>
<tr>
<td>5</td>
<td>PRS:5</td>
<td>Inform user about reached operating</td>
<td>The device shall inform the user once the operating temperature has been reached.</td>
</tr>
<tr>
<td>6</td>
<td>PRS:6</td>
<td>Time information for user at operating</td>
<td>The device shall inform the user about the elapsed time once operating temperature has been reached.</td>
</tr>
<tr>
<td>7</td>
<td>PRS:7</td>
<td>Operating temperature</td>
<td>The operating temperature shall be 51°C at a tolerance of +/- 0.5°C.</td>
</tr>
<tr>
<td>8</td>
<td>PRS:8</td>
<td>Exchange batteries</td>
<td>The device shall allow the user to exchange the batteries.</td>
</tr>
<tr>
<td>9</td>
<td>PRS:9</td>
<td>Cleaning</td>
<td>The device shall be easily cleanable by the user.</td>
</tr>
<tr>
<td>10</td>
<td>PRS:10</td>
<td>Contamination</td>
<td>For the heat plate a material shall be used that avoids contamination with blood.</td>
</tr>
</tbody>
</table>
Capture product requirements (structure)
Capture product requirements (derive)

- **User Information**
  - Id = "PRS:11"
  - Text = "The device shall inform the user about its current state of operation."

- **Comment**
  - Unclear wording: What means 'elapsed time'??
Generate system context

- Generate system context
- Define use cases
- Develop system architecture
- Risk analysis on basis of system architecture
- Derive sub system requirements
- Plan tests
- Generate documents
- Best practices
Generate system context

```
<table>
<thead>
<tr>
<th>bdd [Package] Structure [ System Context ]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperature</td>
</tr>
<tr>
<td>Humidity</td>
</tr>
<tr>
<td>Patient</td>
</tr>
<tr>
<td>Operator</td>
</tr>
<tr>
<td>Previous Patient</td>
</tr>
<tr>
<td>Sting Healer</td>
</tr>
<tr>
<td>ATE</td>
</tr>
</tbody>
</table>
```
Define use cases

- Capture product requirements
- Define use cases
- Develop system architecture
- Risk analysis on basis of system architecture
- Derive sub system requirements
- Plan tests
- Generate documents
- Best practices
Identify use cases

package Primary Use Cases

Primary Use Cases

- Exchange Battery
- Clean Device
- Treat Sting
- Test Device

Participants:
- Patient
- Previous Patient
- Operator
- ATE
Compile use case activities

act [Activity] Treat Sting [Treat Sting]

- Switch On
- Signal
- Heat Up
- Signal Ready

«comment»
Unclear wording: What means 'elapsed time'?

User Info: "Progress"
[else]

(time elapsed)
Switch Off

MedConf 2012 - Munich, 26 Sep 2012
Use cases refine requirements

req [Package] System [PRS refined]

- «requirement» Heat up plate
- «requirement» Keep operating temperature
- «requirement» Switch off heating
- «requirement» Time information for user at operating temperature
- «requirement» Inform user about reached operating temperature

Treat Sting
Develop system architecture

- Capture product requirements
- Generate system context
- Develop system architecture
- Risk analysis on basis of system architecture

- Derive sub system requirements
- Plan tests
- Generate documents
- Best practices
Identify system blocks

1: press → 2: wakeup
5: Signal Heat Up → 4: Signal Heat Up
7: Signal Ready → 6: Signal Ready
Develop system architecture (composition)

```
bbd [Package] Sting Healer [ Sting Healer ]
```

- «block» HeatPlate
- «block» Heating
- «block» Button
- «block» Micro Controller
- «block» Battery
- «block» Speaker
- «block» Temperature Sensor
- «block» Housing
- «block» Comms Interface
Develop system architecture (interconnection)
Specify system behaviour
Risk analysis on basis of system architecture
Risk analysis on basis of system architecture
Risk analysis on basis of system architecture

Over Temperature Protection

Id = "MDD_DES:0001"
Text = "The device provides an over temperature protection that operates independently from the closed loop temperature control. In case the temperature of the heat plate exceeds 55°C the temperature protection disconnects the heating element from its power supply. In this situation the device shall signal an error to the user."
Risk analysis on basis of system architecture
Risk analysis on basis of system architecture

Diagram: State machine showing operating modes with mitigation.
System blocks satisfy requirements

![Diagram showing system blocks and their satisfaction of requirements.]

- **Keep operating temperature** satisfied by **Micro Controller**
- **Operating temperature** satisfied by **Temperature Sensor**
- **Over Temperature Protection** satisfied by **Heating**
- **Temp Fuse**
Derive sub system requirements

- Capture product requirements
- Generate system context
- Define use cases
- Develop system architecture
- Derive sub system requirements
- Plan tests
- Generate documents
- Best practices
Derive sub system requirements

**req [Package] Software [ SRS With Mitigation ]**

- **Operating temperature**
  - «requirement»
  - «deriveReqt»

- **Keep operating temperature**
  - «requirement»
  - «deriveReqt»

- **Over Temperature Protection**
  - «Requirement Mitigation Textual»
  - «deriveReqt»

- **Temperature Control: Cycle Time**
  - «requirement»
  - Id = "SRS:3"
  - Text = "For temperature control the software shall implement a cycle time of 100 ms max."

- **Temperature Control: Characteristics**
  - «requirement»
  - Id = "SRS:1"
  - Text = "For temperature control the software shall implement a P-control loop."

- **Alarm signal at over temperature**
  - «requirement»
  - Id = "SRS:4"
  - Text = "If the over temperature protection has attacked the software shall generate an alarm signal via the speaker for a duration of 3 seconds."
Plan tests

- Capture product requirements
- Generate system context
- Define use cases
- Develop system architecture
- Risk analysis on basis of system architecture
- Plan tests
- Generate documents
- Best practices
Plan tests

```
req [Package] Software [ Software Verification Plan ]

<requirement>
Alarm signal at over temperature

id = "SRS:4"
Text = "If the over temperature protection has
attacked the software shall generate an alarm
signal via the speaker for a duration of 3 seconds."

<verify>

<block>
Test Setup 1

<satisfy>

<TestCase Textual>
Alarm signal at over temperature

AutomationType = Full
Id = "TC_SRS:1"
Purpose = ""

- Verify that software generates the over temperature alarm correctly.

TestDescription = ""

- Switch the device on via Testing Interface (simulate button press)
- Activate TempFuse attack signal via Testing Interface (First run during state
  "Heat Up", 2. run during state "Hold Temperature")
- Supervise the Speaker activation via Testing Interface
- Check if Speaker is activated for the specified time.

TestStrategy = "Functional Test"
```
Account for testability
Specify test setups

![Diagram showing test setups with TestAvenue PC, USB, Digital I/O GPIB, Sting Healer, and connections between them.]
Generate documents

- Capture product requirements
- Generate system context
- Define use cases
- Develop system architecture
- Risk analysis on basis of system architecture
- Derive sub system requirements
- Generate documents
- Best practices
### Generate documents, Requirements

<table>
<thead>
<tr>
<th>ReqID</th>
<th>Req Name</th>
<th>Requirement</th>
<th>Trace</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRS:1</td>
<td>Temperature Control: Characteristics</td>
<td>For temperature control the software shall implement a P-control loop.</td>
<td>Derived from [PRS:1] Heat up plate [PRS:3]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Keep operating temperature</td>
<td>Rational:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MATLAB Simulation of P-control loop shows that residual deviation is &lt; 0.2K.</td>
<td>[PRS:7] Operating temperature</td>
</tr>
<tr>
<td>SRS:2</td>
<td>Sound signal at seconds interval</td>
<td>Once the device has reached its operating temperature the software shall generate an audible signal via the speaker with each elapsed second.</td>
<td>Derived from [PRS:6] Time information for user at operating temperature</td>
</tr>
</tbody>
</table>
# Generate documents, Test plans

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Purpose/Automation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TC_SRS:1</strong></td>
<td>Purpose:</td>
</tr>
<tr>
<td><strong>Name:</strong></td>
<td>• Verify that software generates the over temperature alarm correctly.</td>
</tr>
<tr>
<td>Alarm signal at over temperature</td>
<td>Automation:</td>
</tr>
<tr>
<td></td>
<td>• Full</td>
</tr>
<tr>
<td></td>
<td>• Test strategy:</td>
</tr>
<tr>
<td></td>
<td>• Functional Test</td>
</tr>
</tbody>
</table>

## Requirements verified by this test case

**[SRS:4]**

Alarm signal at over temperature

If the over temperature protection has attacked the software shall generate an alarm signal via the speaker for a duration of 3 seconds.

### Description

- Switch the device on via Testing Interface (simulate button press)
- Activate TempFuse attack signal via Testing Interface (First run during state "Heat Up", 2 run during state "Hold Temperature")
- Supervise the Speaker activation via Testing Interface
- Check if Speaker is activated for the specified time.

### Expected Result

The SW generates an audible signal for 3 s +/- 300 ms in case the over temperature fuse attacks.
Generate documents, Test plans

act [Alarm signal at over temperature]

Switch on device

Activate Temp Fuse attack signal

Supervise speaker activation signal for 5 s

else

Speaker activation = 3 s +/- 300 ms

else

Check speaker activation signal

else

Speaker off

Fail

Pass
Generate documents, Trace reports

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Trace</th>
</tr>
</thead>
</table>
| MDD_DES:0001| Over Temperature Protection | **Derived from**  
N/A: Directly sourced from System FMEA  
**Satisfied by**  
Name: Temp Fuse  
Scope: Sting Healer  
**Verification via derived**  
[ERS:2] → Is unverified  
[SRS:4] → verified by [TC_SRS:1]  
**Derived**  
[ERS:2]  
Cut off heating in case of over temperature  
[SRS:4]  
Alarm signal at over temperature |
Best Practices

- Capture product requirements
- Generate system context
- Define use cases
- Develop system architecture
- Risk analysis on basis of system architecture
- Derive sub system requirements
- Plan tests
- Best practices
Best Practices

- Avoid over modelling \(\dagger\) Use text!
- Don’t model documents \(\dagger\) Document the model!
- Orphan and childless analysis: Early and often
- Limit used SysML language set, especially during introduction phase
Many thanks!