STPA in the Product Line Development of Medical Cyberphysical Systems

Michaela Huhn

Joint work with Sara Bessling

Clausthal University of Technology, Department of Informatics
Introduction

- Michaela Huhn
  - PhD in Computer Science: Formal design methods and verification

- Areas of interest
  - Formal verification, also light-weight
  - Model-based design methods
  - Software-intensive embedded systems
  - Model quality
  - Model-based safety analysis
  - Assessment of software safety
  - Software Certification
Motivation

- Medical devices: software-controlled and dependable
- Patient or disease-specific needs → Many variants

- Feature-oriented product line development promises a productivity gain in software development
- Observation: Safety requirements often refer to the specific features of a medical product variant.

- Research question:
  - How to integrate advanced safety analysis and formal verification in product line development?

Case study details from the Pacemaker Grand Verification Challenge:
  - Based on a pacemaker specification by Boston Scientific
Case Study: Cardiac Pacemaker Product Line

- A pacemaker senses the natural pulses in the atrium and/or the ventricle and - under specified conditions - it generates an artificial pace.

- The NASPE/BPEG Code characterizes the pacemaker variants:
  - 1st letter: Chamber(s) paced: A(trium), V(entricle), D(ual)
  - 2nd letter: Chamber(s) sensed: 0 (none), A(trium), V(entricle), D(ual)
  - 3rd letter: Response mode: 0 (none), I(nhibited), T(riggered), D(ual)
  - 4th and 5th letter: Additional features: e.g. R(ate Modulation)

- E.g. DDD or VVI
Feature-Oriented Product Line Development

- Key paradigm: modeling of variability, i.e. „commonalities and differences in terms of requirements, architecture, components, and test artifacts.“

- Variability is represented by features, i.e.
  - a hierarchical decomposition of functionality
  - Operators:
    - AND
    - OR, group cardinalities, XOR, Optional
  - Attributes
  - Constraints

- A product is specified by selecting its features
Feature-Oriented Product Line Development

- **Base Model:** here SCADE System
- **Variability Model at Design Time:** described with CVL or VIATRA
- **Resolution Model:** generated using CVL or VIATRA
- **Generation of DSL Model:** here SCADE

Feature model at requirements level
STPA Step 1: System-level Hazards and Safety Constraints

Hazards
- H1: Bradycardia: missing stimulation (transient or permanent)
- H2: Cardiac arrhythmia: stimulation in a vulnerable phase or over-stimulation
- H3: Cicatrization of cardiac tissue: conduction failure
- H4: Shortening of battery life time: High energy consumption
- H5: Pacemaker mediated tachycardia (PMT): unwanted interference heart ↔ pacemaker

Safety Constraints
- S1: In case the natural pace is missing, an artificial pace is generated (each BI), lower rate limit
- S2: Refractory and blanking periods: ARP, VRP, PVARP, PAVB, AB,VB, PVAB, upper rate limit
- S3: Artificial paces only when the natural pace is missing, AV Hysteresis
- S4: ECU sleep modi (design constraint)
- S5: Upper rate limit, Anti-PMT-heuristics
Basic Control Structure

- Control structure, hazards and safety constraints are product-specific:
  - 0-2 sensing leads
  - Simplified S1 and not S3 for non-sensing variants, less timing constraints in S2 for single chamber variants, H5 only for Dxx variants
STPA Step 2: Potential for Inadequate Control

- Step 2.1: Feature-wise analysis
- Step 2.2: Product-wise analysis (feature interaction)
- Example
  - A00, V00:
    - Single chamber pacing in a patient-specific rate, no sensing

<table>
<thead>
<tr>
<th>Control Action</th>
<th>Not Providing Causes Hazard</th>
<th>Providing Causes Hazard</th>
<th>Wrong Timing /Order Hazard</th>
<th>Stopped Too Soon /Applied Too Long</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pace</td>
<td>No pace within BI</td>
<td>Normal behavior</td>
<td>Pace within refractory</td>
<td>Generated pulse applied too long</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>period</td>
<td>/too short</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BI inappropriate</td>
<td></td>
</tr>
<tr>
<td>No pace</td>
<td>Pacing with low battery</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td></td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>
STPA Step 2: Potential for Inadequate Control

- Step 2.1: Feature-wise analysis
  - e.g. inappropriate refractory period (timing)
- Step 2.2: Product-wise analysis (feature interaction)
  - E.g. inappropriate AVI hysteresis (timing)
- DDD:
  - Dual chamber pacing, dual sensing, patient-specific rates & intervals

<table>
<thead>
<tr>
<th>Control Action</th>
<th>Not Providing Causes Hazard</th>
<th>Providing Causes Hazard</th>
<th>Wrong Timing /Order Hazard</th>
<th>Stopped Too Soon /Applied Too Long</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pace</td>
<td>Neither a natural nor artificial atr./ventr. pace within BI</td>
<td>Pace generated in the presence of natural pace</td>
<td>Pace within refractory periods</td>
<td></td>
</tr>
</tbody>
</table>
<pre><code>                    |                              |                          | Inappropriate timing             |
                    |                              |                          | Generated pulse applied too long/too short |
</code></pre>
<p>| No pace        | Natural and artificial pace | Neither a natural nor artificial atr./ventr. pace within BI | Inappropriate timing |
| Pacing with low battery     |                          | --                                |</p>
STPA Step 3.1: Causal Scenarios for Unsafe Actions

- Part 1: Logical architecture
  - 3.1.1: Feature-wise analysis
    - May a feature-specific design component cause an unsafe control action?
  - 3.1.2: Product-wise analysis (feature interaction)
    - May a control action be missing because a feature is missing or suppressed by another one dominating the control loop?
    - Does the timing or the resource allocation depend on the feature selection?
    - Are control actions doubled/ suppressed because different features address the same safety constraint?
    - Do parameters of a control action depend on an added/missing feature?

Analysis of positive and negative (logical) feature interactions

Pacemaker variants: VVI, AAI
STPA Step 3.2: Causal Scenarios for Unsafe Actions

- **Part 2: Technical architecture**
  Pacemaker: VVI, AAI

  - **3.2.1: Feature-wise analysis**
    - Fault models for feature-specific hardware, fault injection.

  - **3.2.2: Product-wise analysis** (feature interaction)
    - Does the timing / resource allocation rely on the deployment of feature-specific design components?
    - Does a control action rely on a (physical) action principle that interferes with the action principles of another feature (even in non-safety-related control)
      - E.g. activity sensor for rate modulation may measure the respiratory rate, atrial rate, QT or AV interval, or pressure,...

- **Analysis of**
  - **Deployment**
  - **Faults and their consequences**
  - **Interference of physical action principles**
  - **Fault propagation**
  - **Common cause analysis**
Next Design Step: Feature Selection and Product Generation

Observer nodes specifying safety constraints enable formal verification using Design Verifier
### Safety Analysis: Layers considered so far

<table>
<thead>
<tr>
<th></th>
<th>Functional Perspective</th>
<th>Logical Perspective</th>
<th>Technical Perspective</th>
<th>Other relevant perspectives</th>
</tr>
</thead>
<tbody>
<tr>
<td>System of Systems</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
<td></td>
</tr>
<tr>
<td>Systems</td>
<td><img src="image4.png" alt="Image" /></td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
<td></td>
</tr>
<tr>
<td>Subsystems</td>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
<td><img src="image9.png" alt="Image" /></td>
<td></td>
</tr>
<tr>
<td>Building Blocks</td>
<td><img src="image10.png" alt="Image" /></td>
<td><img src="image11.png" alt="Image" /></td>
<td><img src="image12.png" alt="Image" /></td>
<td></td>
</tr>
</tbody>
</table>
## Functional Correctness - SCADE Verification Results

<table>
<thead>
<tr>
<th></th>
<th>PA</th>
<th>PV</th>
<th>PSyn</th>
<th>SRA</th>
<th>SRV</th>
<th>ROA</th>
<th>ROV</th>
<th>RIA</th>
<th>RIV</th>
<th>RSynIV</th>
<th>RVES</th>
<th>RM</th>
</tr>
</thead>
<tbody>
<tr>
<td>A00</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>V00</td>
<td>-</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>D00</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>AAI</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>✓!</td>
<td>-</td>
<td>-</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>VVI</td>
<td>-</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>✓!</td>
<td>-</td>
<td>-</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DDD</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓*</td>
<td>✓*</td>
<td>-</td>
<td>-</td>
<td>✓*</td>
<td>✓*</td>
<td>✓*</td>
<td>✓</td>
<td>-</td>
</tr>
<tr>
<td>DDRD</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓*</td>
<td>✓*</td>
<td>-</td>
<td>-</td>
<td>✓*</td>
<td>✓*</td>
<td>✓*</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

- ✓ proven within seconds
- * only proven with time constants divided by 10 due to complexity problems
- ! Needs more than an hour
- - property doesn’t apply
Observations

- The feature-oriented design with automated product resolution
  - enforced a uniform handling of development artifacts and a uniform interaction architecture
  - many safety constraints could be assigned to single features
    - resulting safety constraints tend to be more fine-grained
  - increased reuse
  - more efficient verification

- STPA allows for a more systematic investigation of the potentially hazardous behavior and derivation of safety constraints
  - Feature-wise analysis has to be complemented with explicit product-wise analysis (feature interaction): (only a few savings)
  - Investigating the technical architecture level really adds issues

- SCADE allows for automated code generation and formal verification.
  - But: slight modifications (design or safety constraint) might induce significant differences in verification times → Further experiments and classification needed
Conclusion

- STPA applied on a product line of software-controlled medical devices
  - Features considered as first class architectural concept
  - Systematic investigation of hazardous behavior by feature- and product-wise STPA ensures that feature interaction is analyzed explicitly
    - the technical architecture view
    - (extending pure software safety considerations)

- Next steps are
  - to complete and extend the case study
    - mode switches, anti-PMT algorithms, ...
  - to complete formal verification (in the presence of faults)
  - to validate the findings in another case study (infusion pump)
  - comparison to other safety analysis approaches, e.g. wrt. coverage of derived safety constraints
Thank you for your attention!