Getting Rid of Clone-And-Own: Moving to a Software Product Line for Temperature Monitoring

Elias Kuiter, Jacob Krüger, Sebastian Krieter, Thomas Leich, Gunter Saake

University of Magdeburg, Harz University of Applied Sciences, METOP GmbH

SPLC 2018

September 10 – 14 | Gothenburg, Sweden

Using Clone-And-Own

Using Clone-And-Own

+ Easy to implement
+ Fast to deploy

Using Clone-And-Own

+ Easy to implement
+ Fast to deploy

– Hard to maintain
– Hard to extend

Using Clone-And-Own

+ Easy to implement
+ Fast to deploy

– Hard to maintain
– Hard to extend

$\Rightarrow$ **Migrating to a software product line**

Domain

- **Heat Monitoring**
  1. Technische Alternative
     - Heating control systems

Domain

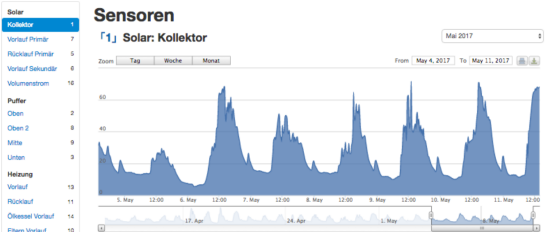
- **Heat Monitoring**
  1. Technische Alternative
     - Heating control systems
  2. HCP-Technology
     - Coolers and temperature control solutions

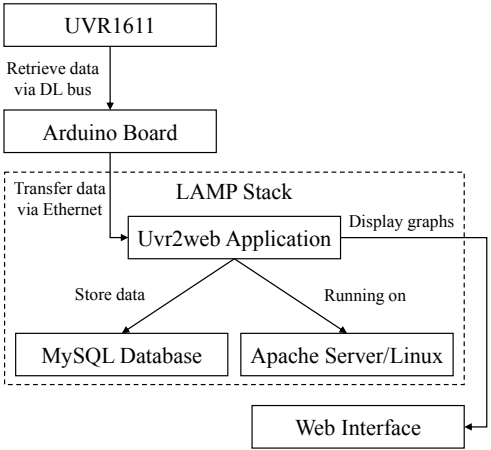
Domain

- **Heat Monitoring**
  1. Technische Alternative
     - Heating control systems
  2. HCP-Technology
     - Coolers and temperature control solutions
  3. Dometic
     - Temperature control solutions

Cloned Systems

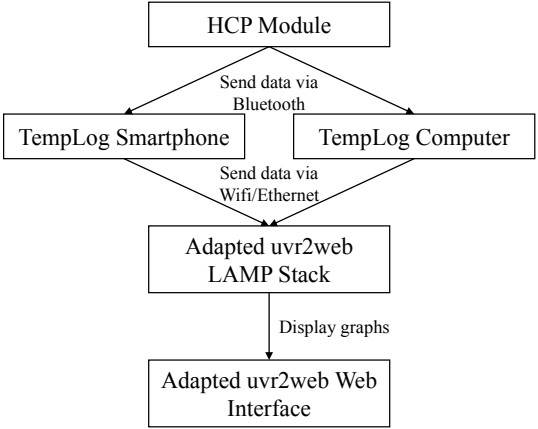
- **First system**: *uvr2web*
  - Reads temperature
  - Displays live feed and statistics





Cloned Systems

- **Second system**: *TempLog*
- **Third system**: *TempLog Dometic*
- **More systems:**
  - Different interfaces
  - New hardware
  - ...



Clone-and-Own: First Problems

- **Arising problems with developing our systems**
  1. Introducing new features
     - *Which variants must be updated?*

Clone-and-Own: First Problems

- **Arising problems with developing our systems**
  1. Introducing new features
     - *Which variants must be updated?*
  2. Fixing bugs
     - Which variants must be repaired?

Clone-and-Own: First Problems

- **Arising problems with developing our systems**
  1. Introducing new features
     - *Which variants must be updated?*
  2. Fixing bugs
     - Which variants must be repaired?
  3. Managing variability among different variants
     - How is variability implemented?

On The Way to an SPL

- Challenges during migration:

On The Way to an SPL

- **Challenges during migration**:
  - Deciding which behavior comprises a feature
  - Mismatch between the intended and actual variability

} Feature Modeling

On The Way to an SPL

- **Challenges during migration**:
  - Deciding which behavior comprises a feature
  - Mismatch between the intended and actual variability

  } Feature Modeling

  - Extracting variable feature code
  - Managing the source code policies

  } Domain Implementation

On The Way to an SPL

- **Challenges during migration**:
  - Deciding which behavior comprises a feature
  - Mismatch between the intended and actual variability

  } Feature Modeling

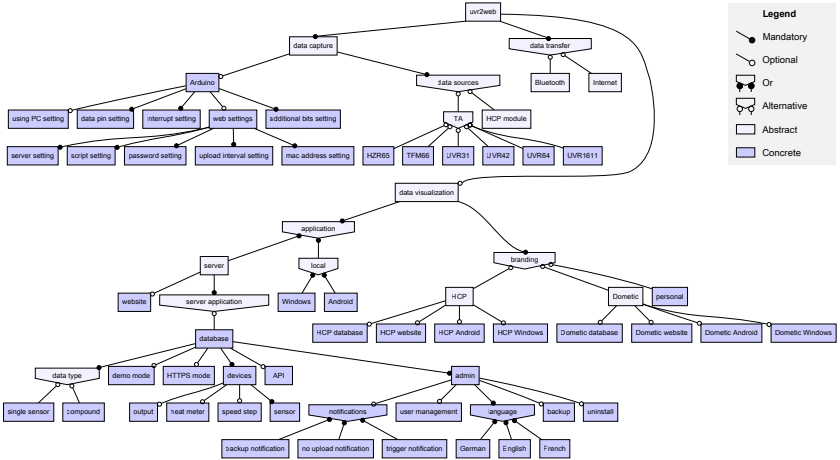
  - Extracting variable feature code
  - Managing the source code policies

  } Domain Implementation

  - Initializing configuration management and product derivation

  } Application Engineering

Feature Modeling

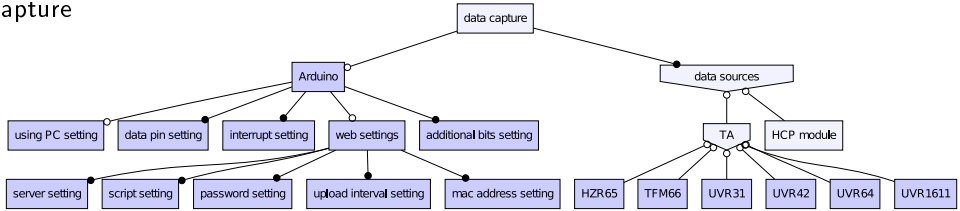


Feature Modeling

- **Main problem:** *tyranny of the dominant decomposition*

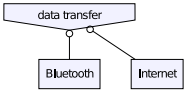
Feature Modeling

- **Main problem:** *tyranny of the dominant decomposition*
- Sub-trees:
  - Data capture



Feature Modeling

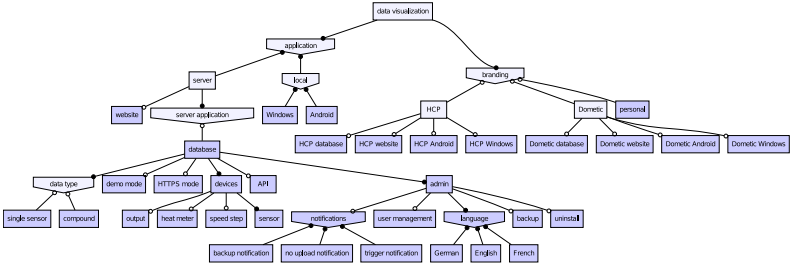
- **Main problem:** *tyranny of the dominant decomposition*
- Sub-trees:
  - Data capture
  - Data transfer



Feature Modeling

- **Main problem:** *tyranny of the dominant decomposition*

- Sub-trees:
  - Data capture
  - Data transfer
  - Data visualization



Feature Modeling

- **Main problem:** *tyranny of the dominant decomposition*

- Sub-trees:
  - Data capture
  - Data transfer
  - Data visualization

$$TA \wedge database \Rightarrow (compound \wedge sensor \wedge output)$$
$$single\ sensor \Rightarrow (sensor \wedge \neg output \wedge \neg heat\ meter \wedge \neg speedstep)$$
$$UVR1611 \wedge database \Rightarrow (heat\ meter \wedge speedstep)$$
$$backup\ notification \Rightarrow backup$$
$$HCP\ module \Leftrightarrow (HCP \vee Dometic)$$
$$database \Leftrightarrow (personal \vee HCP\ database \vee Dometic\ database)$$
$$Android \Leftrightarrow (HCP\ Android \vee Dometic\ Android)$$
$$website \Leftrightarrow (HCP\ website \vee Dometic\ website)$$
$$website \Rightarrow (HCP\ module \wedge database \wedge Android \wedge Windows)$$
$$\ldots$$

- Cross-tree constraints:
  - Handling dependencies of legacy systems

Domain Implementation

Before feature extraction:

Variant	SLOC
uvr2web	5,148
TempLog	6,837
Total	**11,985**

Domain Implementation

Before feature extraction:

Variant	SLOC
uvr2web	5,148
TempLog	6,837
Total	**11,985**

⇒ Introducing variability:

- Preprocessor
- Runtime variability
- Build system

Domain Implementation

Before feature extraction:

Variant	SLOC
uvr2web	5,148
TempLog	6,837
Total	**11,985**

After feature extraction:

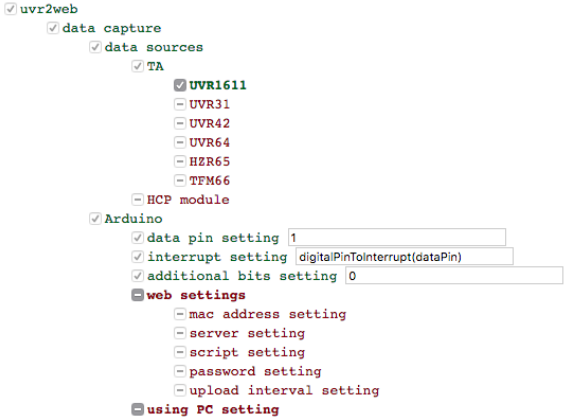
Feature	SLOC
Windows	1,533
Android	1,478
website	1,013
database	890
devices	659
Arduino	623
Other features	4,388
Total	**10,584**

Application Engineering

- **Own web-based tooling:**
  - *feature-configurator*
    - Configuration
    - Uses decision propagation

Application Engineering

- **Own web-based tooling:**
  - *feature-configurator*
    - Configuration
    - Uses decision propagation
  - *feature-php*
    - Product derivation
    - Enables multiple variability mechanisms (preprocessor, runtime variability, ...)



Case Study

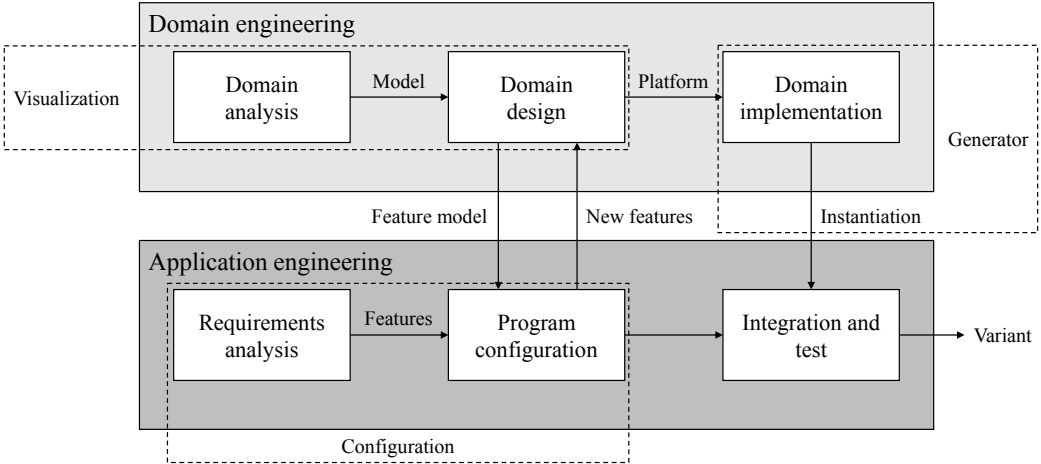
- **Resulting product line**
  - Mainly C++ and PHP
  - Using preprocessors, runtime variability, and build system
  - 10,584 lines of code
  - 69 features
  - 17 cross-tree constraints

Artifacts Evaluated acm Reusable

GitHub

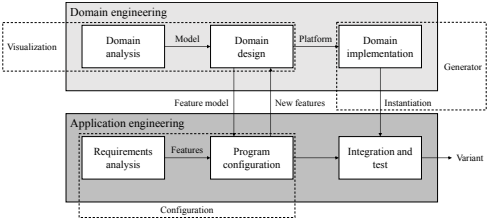
https://github.com/ekuiter/uvr2web-spl/

Developed Variability Tools



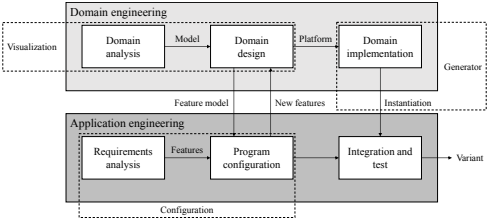
Developed Variability Tools

- **Why implemented new tools?**
  - Closed source
  - Too heavy weight / too many dependencies



Developed Variability Tools

- **Why implemented new tools?**
  - Closed source
  - Too heavy weight / too many dependencies



⇒ **Own modularized tool framework:**

- *feature-model-viz*: domain analysis / visualization
- *feature-php*: domain implementation / product derivation
- *feature-configurator*: configuration

Developed Variability Tools

- **Integration of all tools:**
  - *feature-web*
  - Web-based:
    - Feature model visualization
    - Configuration
    - Automatic product derivation & download
  - LGPLv3 license

Artifacts Evaluated acm Reusable

GitHub

https://github.com/ekuiter/feature-web/

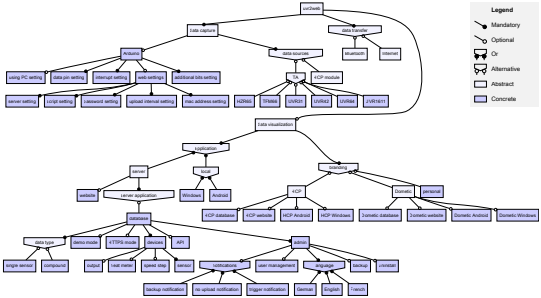
Developed Variability Tools

feature-web in action:

https://uvr2web.de/?p=download

Conclusion

SPL Case Study



Tools

GitHub

feature-php

feature-model-viz

feature-configurator