Usage Scenarios for Feature Model Synthesis.

Steven She, Krzysztof Czarnecki, Andrzej Wasowski
University of Waterloo
IT University of Copenhagen

September 30, 2012.
Why Synthesize a Feature Model?

Variability is often scattered over multiple artifacts in large software projects.

STACK enables the stack(9) facility … stack(9) will also be compiled in automatically if DDB(4) is compiled into the kernel.

```c
#ifdef DDB
#ifndef KDB
#error KDB must be enabled for DDB to work!
#endif
#endif
```

S. She, K. Czarnecki, A. Wasowski.
Why Synthesize a Feature Model? (cont.)

- A feature model provides an overview of variability in the software system.
- Automated tool support and product configuration.
Feature Models.

Feature models describe the common and variable product characteristics of products in a product line.
Feature Models.

Feature models describe the common and variable product characteristics of products in a product line.
Feature models describe the common and variable product characteristics of products in a product line.
Feature Models.

Feature models describe the common and variable product characteristics of products in a product line.
Feature Models.

Feature models describe the common and variable product characteristics of products in a product line.
Feature models describe the common and variable product characteristics of products in a product line.

- **Phone**
  - Processor
  - NFC
  - Camera
  - 4G

- **ARM** implies **OMAP** implies **Snapdragon**
Feature Models.

Feature models describe the common and variable product characteristics of products in a product line.
Feature Model Synthesis.

Features

Configs or Dependencies

Feature Model Synthesis

Feature Model

\{(Phone, Processor, ARM, Snapdragon, NFC)\}

(4G \land NFC \rightarrow Phone)

\land

(Processor \leftrightarrow Phone)

\land

(ARM \land Snapdragon \rightarrow Processor)

\land

(4G \rightarrow Snapdragon)

\land

(ARM \rightarrow \neg Snapdragon)

S. She, K. Czarnecki, A. Wasowski.

Usage Scenarios for Feature Model Synthesis.
Feature Model Synthesis.

\{\text{Phone, Processor, ARM, Snapdragon, NFC}\}

\((4G \land \text{NFC} \rightarrow \text{Phone})\)
\land \ (\text{Processor} \leftrightarrow \text{Phone})
\land \ (\text{ARM} \land \text{Snapdragon} \rightarrow \text{Processor})
\land \ (4G \rightarrow \text{Snapdragon})
\land \ (\text{ARM} \rightarrow \neg \text{Snapdragon})

S. She, K. Czarnecki, A. Wasowski.

Usage Scenarios for Feature Model Synthesis.
Feature Model Synthesis.

Artifacts with Variability

Analysis

Features

Configs or Dependencies

Feature Model Synthesis

Feature Model

{Phone, Processor, ARM, Snapdragon, NFC}

(4G $\land$ NFC $\rightarrow$ Phone)

(Processor $\leftrightarrow$ Phone)

(ARM $\land$ Snapdragon $\rightarrow$ Processor)

(4G $\rightarrow$ Snapdragon)

(ARM $\rightarrow$ $\neg$Snapdragon)

S. She, K. Czarnecki, A. Wasowski.

Usage Scenarios for Feature Model Synthesis.
Artifacts with Variability.

- README: Documentation
- ddb.c: Source Code
- FM: Feature Model
- DOORS: Requirements
- Makefile: Build System
- program\textsubscript{1}, program\textsubscript{2}, program\textsubscript{3}: Clone-and-Own Code

...and more.
Artifacts with Variability.

Many different kinds of input artifacts, each requiring specialized analysis.

...and more.

S. SHE, K. CZARNECKI, A. WAsOWSKI.

USAGE SCENARIOS FOR FEATURE MODEL SYNTHESIS.
Workflow for Feature Model Synthesis.

- **Analysis** recovers the abstract input needed for feature model synthesis from the input artifacts.
- **Feature model synthesis** builds a FM given the abstract input.
Challenges of Feature Model Synthesis.

Which feature model describes features \{A, B, C\}, and the following constraints:

\begin{align*}
B &\rightarrow A \\
C &\rightarrow A \\
C &\rightarrow B
\end{align*}

Both! Different feature models can describe the same set of configurations (or dependencies).
Challenges of Feature Model Synthesis.

Which feature model describes features \{A, B, C\}, and the following constraints:

- $B \rightarrow A$
- $C \rightarrow A$
- $C \rightarrow B$

Both! Different feature models can describe the same set of configurations (or dependencies).
Workflow for Feature Model Synthesis.

- Supplemental information, user input or heuristics can be used to select a distinct feature model.

Supplemental Information, User Input, Heuristics
Workflow for Feature Model Synthesis.

Artifacts with Variability → Analysis → Feature Model Synthesis → Feature Model

User Input, Heuristics

Dependencies or Configurations

Features

Supplemental Information
Outline.

1 Overview of Feature Model Synthesis
2 Scenario Criteria
3 Scenarios
4 Discussion and Conclusions
Outline.

1. Overview of Feature Model Synthesis
2. Scenario Criteria
3. Scenarios
4. Discussion and Conclusions
Scenario Criteria.

1. Input Artifacts
2. Precision of Configuration Analysis
3. Required Synthesis Precision
4. Size
## Input Artifacts.

<table>
<thead>
<tr>
<th>Variability Abstraction</th>
<th>Abstraction-Realization Interface</th>
<th>Variability Realization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable Artifacts</td>
<td>Feature model</td>
<td>VPs and feature-to-VP mapping</td>
</tr>
<tr>
<td>Instances</td>
<td>Feature configs.</td>
<td>VP configs.</td>
</tr>
</tbody>
</table>
Precision of Configuration Analysis.

Configurations represented by the input artifacts

Analysis

Configurations recovered through analysis
Precision of Configuration Analysis.

Configurations represented by the input artifacts

Analysis

Configurations recovered through analysis

Sound and complete (exact) recovery.
Precision of Configuration Analysis.

Configurations represented by the input artifacts

Configurations recovered through analysis

Sound recovery (under approximation).

S. She, K. Czarnecki, A. Wasowski.

Usage Scenarios for Feature Model Synthesis.
Precision of Configuration Analysis.

Configurations represented by the input artifacts

Configurations recovered through analysis

Complete recovery (over approximation).

S. SHE, K. CZARNECKI, A. WASOWSKI.

USAGE SCENARIOS FOR FEATURE MODEL SYNTHESIS.
Precision of Configuration Analysis.

Configurations represented by the input artifacts

Configurations recovered through analysis

Analysis

Arbitrary recovery.
Required Synthesis Precision.

- Configs represented by the input artifacts
- Configs recovered through analysis
- Configs represented in the feature model

Analysis → FM Synthesis
Required Synthesis Precision.

Configs represented by the input artifacts

Analysis

Configs recovered through analysis

FM Synthesis

Configs represented in the feature model
Required Synthesis Precision.

Configs represented by the input artifacts

Analysis

Configs recovered through analysis

FM Synthesis

Configs represented in the feature model
Required Synthesis Precision.

Configs represented by the input artifacts

Configs recovered through analysis

Configs represented in the feature model

Analysis

FM Synthesis

Sound and complete (exact) recovery.
Required Synthesis Precision.

Configs represented by the input artifacts

Analysis

Configs recovered through analysis

FM Synthesis

Configs represented in the feature model

Sound recovery (under approximation).
Required Synthesis Precision.

Complete recovery (over approximation).

Configs represented by the input artifacts → Analysis → Configs recovered through analysis → FM Synthesis → Configs represented in the feature model.
Required Synthesis Precision.

Configgs represented by the input artifacts

Analysis

Configgs recovered through analysis

FM Synthesis

Configgs represented in the feature model

Arbitrary recovery.
Size.

- Classified size of a scenario by estimating the number of features required for synthesis.
- Based on SPLOT’s model repository¹ and a collection of FMs gathered from the system’s domain.

**Small**  Several hundred features.
**Medium**  Thousand features.
**Large**  Several thousand features.

¹http://www.splot-research.org
Outline.

1. Overview of Feature Model Synthesis
2. Scenario Criteria
3. Scenarios
   - Scenario 1: From a Configurable Platform
   - Scenario 2: From a Set of Variants
   - Scenario 3: Feature Model Operations
   - Scenario 4: Feature Model Merge Workflows
4. Discussion and Conclusions
Workflow for Feature Model Synthesis.
Scenario 1: From a Configurable Platform

<table>
<thead>
<tr>
<th>Variability Abstraction</th>
<th>Abstraction-Realization Interface</th>
<th>Variability Realization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable Artifacts</td>
<td>Feature model</td>
<td>VPs and feature-to-VP mapping</td>
</tr>
<tr>
<td>Instances</td>
<td>Feature configs.</td>
<td>VP configs.</td>
</tr>
</tbody>
</table>
Configurable Platform (Code).

```c
#ifdef DDB
#ifndef KDB
#error KDB must be enabled for DDB to work!
#endif
#endif
```

- Scenario for synthesizing a FM for FreeBSD (She et al. 2011)
Configurable Platform (Code): Precision.

- Configs represented by the input artifacts
- Configs recovered through analysis
- Configs represented in the feature model

Analysis → Complete Recovery

FM Synthesis → Sound Recovery
Configurable Platform (Requirements).

Req. 1. Product A must have **Feature X**, while Product B may optionally implement **Feature X**. Every product must have **Feature Y**.

Req. 2. ...
## Scenario 2: From a Set of Variants

<table>
<thead>
<tr>
<th>Variable Artifacts</th>
<th>Feature model</th>
<th>VPs and feature-to-VP mapping</th>
<th>Configurable platform requirements, models, code, etc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instances</td>
<td>Feature configs.</td>
<td><strong>VP configs.</strong></td>
<td><strong>Variants requirements, models, code, etc.</strong></td>
</tr>
</tbody>
</table>

### Variability Abstraction

- **Abstraction Interface**

### Abstraction-Realization Interface

- **Variability Realization**

S. She, K. Czarnecki, A. Wasowski.
Variants (Models).

- (Ryssel et al. 2012) developed model matching on SimuLink models.
- (Rubin et al. 2012) developed model matching on UML state charts and class diagrams.
Variants (Requirements).

REQS for Product A

Req. 1. Product A has a MP3 player and a break system.

Req. 2. ...

REQS for Product B

Req. 1. Product B has a 6-disc CD player and a break system.

Req. 2. ...

• Experience reported at an automotive company. VPs were manually identified between requirements documents.
Variants (Code).

\[ \text{Code}_A \]

\[
\begin{align*}
\text{class } \text{Calculator} & \{ \\
\text{def } \text{add}(x,y) & \{ \ldots \} \\
\text{def } \text{subtract}(x,y) & \{ \ldots \}
\}
\]

\[ \text{Code}_B \]

\[
\begin{align*}
\text{class } \text{Calculator} & \{ \\
\text{def } \text{add}(x,y) & \{ \ldots \} \\
\text{def } \text{subtract}(x,y) & \{ \ldots \} \\
\text{def } \text{multiply}(x,y) & \{ \ldots \}
\}
\]

- (Jepsen et al. 2007) report building a FM from products built with clone-and-own code variants at Danfoss drives.
Variant Workflows.

(Scenario. 2a, 2b, 2c) Variants (requirements, models, and code)

Additional scenario involving VP configs in the paper.
## Scenario 3: Feature Model Operations

<table>
<thead>
<tr>
<th>Variability Abstraction</th>
<th>Abstraction-Realization Interface</th>
<th>Variability Realization</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Variable Artifacts</strong></td>
<td><strong>Feature model</strong></td>
<td>VPs and feature-to-VP mapping</td>
</tr>
<tr>
<td><strong>Instances</strong></td>
<td><strong>Feature configs.</strong></td>
<td><strong>VP configs.</strong></td>
</tr>
</tbody>
</table>

S. She, K. Czarnecki, A. Wasowski.
Feature Model Operations.
(Scenario. 3) Feature model operations

- (Acher 2011)’s thesis work was based on this scenario.
- (Fahrenberg et al. 2011) also described similar operations on models using FMs as an example.
Feature Model Operations: Precision.

- Configs represented by the input artifacts
- Configs recovered through analysis
- Configs represented in the feature model

Analysis → Sound Recovery → FM Synthesis → Exact Recovery

Usage Scenarios for Feature Model Synthesis.
Scenario 4: Feature Model Merge Workflows

<table>
<thead>
<tr>
<th>Variable Artifacts</th>
<th>Variability Abstraction</th>
<th>Abstraction-Realization Interface</th>
<th>Variability Realization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature model</td>
<td>VPs and feature-to-VP mapping</td>
<td>Configurable platform requirements, models, code, etc.</td>
<td></td>
</tr>
</tbody>
</table>

| Instances | Feature configs. | VP configs. | Variants requirements, models, code, etc. |
Feature Model Merge Workflows.

<table>
<thead>
<tr>
<th>ID</th>
<th>Revision Diffs</th>
<th>Operating System</th>
<th>Storage</th>
<th>Open</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Windows; Linux</td>
<td>Files; Database</td>
<td>No</td>
</tr>
<tr>
<td>FooWiki</td>
<td>Plugin</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(a) A wiki engine description

(b) Corresponding FM

Product descriptions (Acher et al. 2012)
Outline.

1. Overview of Feature Model Synthesis
2. Scenario Criteria
3. Scenarios
4. Discussion and Conclusions
Discussion.

- Analysis of variable artifacts recover dependencies, while variants recover sets of VP or feature configurations.
- Alternatively, FMs can be used as input (e.g. FM operations and merge).
- Heuristics or user input can be used to select a distinct hierarchy for the FM depending on the scenario.
- Additional configurations recovered by a complete analysis could be pruned with a sound synthesis and vice versa.
Conclusions.

• FM synthesis is required in a wide range of scenarios with significantly different input artifacts.

• Different input artifacts require different analysis techniques and synthesis workflows.

• Scenarios can be used for a qualitative evaluation of FM synthesis techniques.

• Usage scenarios provide requirements for synthesis techniques.