Guaranteeing Configuration Validity in Evolving Software Product Lines

Michael Nieke, Christoph Seidl, Sven Schuster, 2016/01/28
Technische Universität Braunschweig
Institute of Software Engineering and Automotive Informatics
Motivation

- Car Manufacturer (OEM) with Car Software Product Line
- Sells many similar variants of a car to a major customer
- Particular important variant/configuration
- Cars have a common interface to communicate with external devices
- Car SPL is evolving
- Support for the communication interface is dropped (e.g., new standard)
Prevent important configurations from breaking

- Model the evolution of SPLs / Feature Models
- Analyze the impact of Feature Model evolution on existing configurations
  - i.e., are configurations still valid after the evolution?
Evolution of Feature Models

Parking Assistance → Side Distance Sensor
Adaptive Cruise Control → Distance Sensors
Evolution of Feature Models – Temporal Feature Models (TFMs)

General Idea

- To seamlessly integrate evolution in feature models: make evolution a first-class entity

- **Temporal Elements**

- Each *Temporal Element* has a **Temporal Validity**

- **Temporal Validities** define the timespan in which an element is valid
Evolution of Feature Models – Temporal Feature Models (TFMs)

General Idea

Each evolvable element has to be a *Temporal Element*:

- Features
- Groups
- Types of Features / Groups (i.e., Cardinalities)
- Feature ↔ Group relations
- Constraints
Temporal Validities

<table>
<thead>
<tr>
<th>Feature</th>
<th>( \vartheta_{since} )</th>
<th>( \vartheta_{until} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>( t_0 )</td>
<td>( t_1 )</td>
</tr>
<tr>
<td>F2</td>
<td>( t_0 )</td>
<td>( \infty )</td>
</tr>
<tr>
<td>F3</td>
<td>( t_1 )</td>
<td>( \infty )</td>
</tr>
</tbody>
</table>

Temporal Validity \( \vartheta = [\vartheta_{since};\vartheta_{until}] \)
Evolution of Feature Models – Evolution Operations
Semantics of Evolution

• **Temporal Validities** may be used uniformly to **represent evolution**

• Generally independent of **type** of evolution

• Introduce **Evolution Operation** to provide **semantics** of the evolution

• **Evolution Operations** can be **extracted** from *Temporal Validities*

• Temporal Validities are independent of Evolution Operations
Evolution of Feature Models – Evolution Operations

- Add Feature/Group
- Move Feature/Group
- Change Type of Feature/Group
- Delete Feature/Group

Temporal Validities

<table>
<thead>
<tr>
<th></th>
<th>$\vartheta_{\text{since}}$</th>
<th>$\vartheta_{\text{until}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distance Sensors</td>
<td>$t_1$</td>
<td>$\infty$</td>
</tr>
</tbody>
</table>

$t_0 \rightarrow t_1$
Evolution of Feature Models – Evolution Operations

Temporal Validities

<table>
<thead>
<tr>
<th>Feature/Group</th>
<th>(\vartheta_{since})</th>
<th>(\vartheta_{until})</th>
</tr>
</thead>
<tbody>
<tr>
<td>InfotainmentSystemName</td>
<td>(\infty)</td>
<td>(\infty)</td>
</tr>
</tbody>
</table>

- Add Feature/Group
- Move Feature/Group
- Change Type of Feature/Group
- Delete Feature/Group
- Rename Feature
### Evolution of Feature Models – Evolution Operations

- **Add Feature/Group**
- **Move Feature/Group**
- **Change Type of Feature/Group**
- **Delete Feature/Group**
- **Rename Feature**

#### Temporal Validities

<table>
<thead>
<tr>
<th>Feature/Group</th>
<th>$\vartheta_{since}$</th>
<th>$\vartheta_{until}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infotainment System</td>
<td>$\infty$</td>
<td>$t_1$</td>
</tr>
</tbody>
</table>
Evolution of Feature Models – Evolution Operations

Temporal Validities

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>$\vartheta_{since}$</th>
<th>$\vartheta_{until}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>InfotainmentSystemName</td>
<td>$\infty$</td>
<td>$t_1$</td>
</tr>
<tr>
<td>EcoInfotainmentSystemName</td>
<td>$t_1$</td>
<td>$\infty$</td>
</tr>
</tbody>
</table>

- Add Feature/Group
- Move Feature/Group
- Change Type of Feature/Group
- Delete Feature/Group
- Rename Feature
- Add/Delete Constraint
Guaranteeing Valid Configurations
Broken Configuration Categories

- Measure the possibility of evolution to invalidate a certain configuration

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not broken</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Refactoring</td>
<td>No impact on the variability space</td>
<td>Feature is renamed</td>
</tr>
<tr>
<td>Unaffected</td>
<td>Variability space is modified</td>
<td>Type of feature is changed from mandatory to optional</td>
</tr>
<tr>
<td></td>
<td>Not relevant for considered configuration</td>
<td></td>
</tr>
<tr>
<td>Extended</td>
<td>Variability space is extended</td>
<td>A new feature is added</td>
</tr>
<tr>
<td></td>
<td>Developers should be informed</td>
<td></td>
</tr>
<tr>
<td>Outdated</td>
<td>Contained elements are not valid anymore</td>
<td>A selected feature is deleted</td>
</tr>
<tr>
<td>Conflicting</td>
<td>Contained elements are not consistent with the feature model</td>
<td>A new constraint prohibits the selection of two selected features</td>
</tr>
</tbody>
</table>
Guaranteeing Valid Configurations
Change Impact Analysis

Temporal Feature Model

Evolution Operation

Analysis

Input

Categorization

Not broken category

Broken category

Configuration
Guaranteeing Valid Configurations
Change Impact Analysis

Temporal Feature Model

Evolution Operation

Input

Analysis

Locked Configuration

Not broken category

Evolution performed

Categorization

Broken category

Evolution blocked

Evolution blocked

Evolution performed
Example Configurations

Basic
+ Infotainment System
+ Smartphone Integration

Economic
+ Infotainment System
+ Consumption Indicator

Premium
+ Infotainment System
+ Assistance Systems
+ Parking Assistance
+ Adaptive Cruise Control
+ Side Distance Sensor
+ Fast Front Dist. Sensor
Example
Change Impact Analysis

- Car
  - Assistance Systems
  - Infotainment System
  - Smartphone Integration
  - Adaptative Cruise Control
    - Slow Front Distance Sensor
    - Fast Front Distance Sensor
  - Consumption Indicator

Add Feature / Attribute

- Basic
  + Infotainment System
  + Smartphone Integration

- Economic
  + Infotainment System
  + Consumption Indicator

- Premium
  + Infotainment System
  + Assistance Systems
  + Parking Assistance
  + Adaptative Cruise Control
  + Side Distance Sensor
  + Fast Front Distance Sensor

Extended
Example
Change Impact Analysis

- Basic
  - +Infotainment System
  - +Smartphone Integration

- Economic
  - +Infotainment System
  - +Consumption Indicator

- Premium
  - +Infotainment System
  - +Assistance Systems
  - +Parking Assistance
  - +Adaptive Cruise Control
  - +Side Distance Sensor
  - +Fast Front Dist. Sensor

Delete Feature
Outdated

Evolution

Car
- Assistance Systems
- Infotainment System
- Smartphone Integration
- Adaptive Cruise Control
- Consumption Indicator

Parking Assistance
- Side Distance Sensor
- Slow Front Distance Sensor

Fast Front Distance Sensor

Eco Infotainment System
- Distance Sensors
- Smartphone Integration

Side Distance Sensor
- Slow Front Distance Sensor

Fast Front Distance Sensor

Car

Parking Assistance
- Adaptive Cruise Control

Slow Front Distance Sensor

Smartphone Integration

Eco Infotainment System

Side Distance Sensor
- Slow Front Distance Sensor

Fast Front Distance Sensor

Car

Assistance Systems

Distances

Slow Front Distance Sensor

Fast Front Distance Sensor

Parking Assistance

Adaptive Cruise Control

Eco Infotainment System

Distance Sensors

Smartphone Integration

Slow Front Distance Sensor

Fast Front Distance Sensor

Side Distance Sensor
Example Configurations

- Basic
  - +Infotainment System
  - +Smartphone Integration

- Economic
  - +Infotainment System
  - +Consumption Indicator

- Premium
  - +Infotainment System
  - +Assistance Systems
  - +Parking Assistance
  - +Adaptive Cruise Control
  - +Side Distance Sensor
  - +Fast Front Dist. Sensor

- Unaffected
Example
Change Impact Analysis

Basic
+ Infotainment System
+ Smartphone Integration

Economic
+ Infotainment System
+ Consumption Indicator

Premium
+ Infotainment System
+ Assistance Systems
+ Parking Assistance
+ Adaptive Cruise Control
+ Side Distance Sensor
+ Fast Front Dist. Sensor

Move Feature
Conflicting
Example
Change Impact Analysis

Evolution

Car

Assistance Systems

Infotainment System

Smartphone Integration

Parking Assistance

Adaptive Cruise Control

Consumption Indicator

Slow Front Distance Sensor

Fast Front Distance Sensor

Side Distance Sensor

Basic

+ Infotainment System
+ Smartphone Integration

Economic

+ Infotainment System
+ Consumption Indicator

Premium

+ Infotainment System
+ Assistance Systems
+ Parking Assistance
+ Adaptive Cruise Control
+ Side Distance Sensor
+ Fast Front Dist. Sensor

Rename Feature

Refactoring
Example

Locking

- Car
  - Assistance Systems
  - Infotainment System
    - Parking Assistance
    - Adaptive Cruise Control
      - Slow Front Distance Sensor
      - Fast Front Distance Sensor
    - Consumption Indicator

- Evolution

- Basic
  - + Infotainment System
  - + Smartphone Integr.
  - Evolution performed

- Economic
  - + Infotainment System
  - + Consumption Indicator
  - Evolution blocked

- Premium
  - + Infotainment System
  - + Assistance Systems
  - + Parking Assistance
  - + Adaptive Cruise Control
  - + Slow Front Distance Sensor
  - + Fast Front Dist. Sensor
  - Evolution blocked
Summary

Temporal Feature Model

Evolution Operation

Analysis

Input

Locked Configuration

Not broken category

Evolution performed

Categorization

Evolution blocked

Broken category
Future Work
Staged Configurations

- Change Impact Analyses also works for partial configurations

- *Configuration Locking* is „all or nothing“

  - Stage Locking

  Adapted From: Czarnecki et al.: Staged configuration through specialization and multilevel configuration of feature models
Future Work
Semantic Repair

Temporal Feature Model

Evolution Operation

Analysis

Input

Broken category

Locked Configuration

Evolution

Locked Configuration

Semantic Repair

Input

Broken category

Locked Configuration
Guaranteeing Configuration Validity in Evolving Software Product Lines

Temporal Feature Model

Evolution Operation

Analysis

Input

Not broken category
Evolution performed

Categorization

Broken category
Evolution blocked

Locked Configuration
Classic Meta Model

“Classic” Cardinality-Based Feature Model Meta Model:

- FeatureModel
  - rootFeature
  - Feature
    - name : EString
    - [1..*] features
  - Group
    - [0..1] parentOfFeature
    - [0..*] groups
    - [0..1] parentOfGroup

CardinalityBasedElement
- minCardinality : EInt =
- maxCardinality : EInt =
Realization in the Meta Model

Transformation to *Temporal Elements*:
## Affected Elements

<table>
<thead>
<tr>
<th>Evolution Operation</th>
<th>Affected Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Features, Groups, Attributes, Constraint</td>
</tr>
<tr>
<td>delete</td>
<td>Features, Groups, Attributes, Constraint</td>
</tr>
<tr>
<td>change type</td>
<td>Features, Groups</td>
</tr>
<tr>
<td>rename feature</td>
<td>Features</td>
</tr>
<tr>
<td>move feature</td>
<td>Group Compositions</td>
</tr>
<tr>
<td>move group</td>
<td>Feature Children</td>
</tr>
</tbody>
</table>

- Temporal Validities are independent of Evolution Operations
- To support evolution, Evolution Operations are not mandatory
- Extensible → new Evolution Operations can be arbitrarily defined
### Derivation Rules

\[ \vartheta = [t_0; t_1] \quad \vartheta_{since} = t_0, \vartheta_{until} = t_1 \]

<table>
<thead>
<tr>
<th>Evolution Operation</th>
<th>Properties of Temporal Validities between two points in time ( \tau_1, \tau_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>( \tau_1 &lt; \vartheta_{since} \leq \tau_1 &lt; \vartheta_{until} )</td>
</tr>
<tr>
<td>delete</td>
<td>( \vartheta_{since} \leq \tau_1 &lt; \vartheta_{until} \leq \tau_2 )</td>
</tr>
<tr>
<td>change type</td>
<td>( \text{card}_1 \neq \text{card}<em>2, ) ( \vartheta</em>{\text{card}<em>1\text{since}} \leq \tau_1 &lt; \vartheta</em>{\text{card}<em>1\text{until}} \leq \vartheta</em>{\text{card}<em>2\text{since}} \leq \tau_2 &lt; \vartheta</em>{\text{card}_2\text{until}} )</td>
</tr>
<tr>
<td>rename feature</td>
<td>( \text{name}_1 \neq \text{name}<em>2, ) ( \vartheta</em>{\text{name}<em>1\text{since}} \leq \tau_1 &lt; \vartheta</em>{\text{name}<em>1\text{until}} \leq \vartheta</em>{\text{name}<em>2\text{since}} \leq \tau_2 &lt; \vartheta</em>{\text{name}_2\text{until}} )</td>
</tr>
<tr>
<td>move feature</td>
<td>( \text{group}_1 \neq \text{group}<em>2, ) ( \vartheta</em>{\text{compo}<em>1\text{since}} \leq \tau_1 &lt; \vartheta</em>{\text{compo}<em>1\text{until}} \leq \vartheta</em>{\text{compo}<em>2\text{since}} \leq \tau_2 &lt; \vartheta</em>{\text{compo}_2\text{until}} )</td>
</tr>
<tr>
<td>move group</td>
<td>( \text{parentFeature}_1 \neq \text{parentFeature}<em>2, ) ( \vartheta</em>{\text{featureChild}<em>1\text{since}} \leq \tau_1 &lt; \vartheta</em>{\text{featureChild}<em>1\text{until}} \leq \vartheta</em>{\text{featureChild}<em>2\text{since}} \leq \tau_2 &lt; \vartheta</em>{\text{featureChild}_2\text{until}} )</td>
</tr>
</tbody>
</table>
## Categorizations (I)

<table>
<thead>
<tr>
<th>Add and Delete Operation</th>
<th>Add</th>
<th>Delete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature / Attribute</td>
<td>Extended</td>
<td>Outdated (if defined)</td>
</tr>
<tr>
<td>Group without features</td>
<td>Unaffected</td>
<td>Unaffected</td>
</tr>
<tr>
<td>Constraint</td>
<td>(possibly) <strong>Conflicting</strong></td>
<td>Unaffected</td>
</tr>
</tbody>
</table>

### Change Type Operation (From) → (To)

<table>
<thead>
<tr>
<th>Change Type Operation</th>
<th>Broken Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandatory → Optional</td>
<td>Unaffected</td>
</tr>
<tr>
<td>Optional → Mandatory</td>
<td><strong>Conflicting</strong> (if deselected)</td>
</tr>
<tr>
<td>Alternative → Or</td>
<td>Unaffected</td>
</tr>
<tr>
<td>Or → Alternative</td>
<td><strong>Conflicting</strong> (if more than 1 feature selected)</td>
</tr>
<tr>
<td>Unbounded → Or</td>
<td><strong>Conflicting</strong> (if no feature selected)</td>
</tr>
<tr>
<td>Unbounded → Alternative</td>
<td><strong>Conflicting</strong> (if not exactly 1 feature selected)</td>
</tr>
<tr>
<td>Or / Alternative → Unbounded</td>
<td>Unaffected</td>
</tr>
</tbody>
</table>
## Categorizations (II)

<table>
<thead>
<tr>
<th>Move Operation</th>
<th>Broken Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Move Group to other parent feature</td>
<td><strong>Conflicting</strong> (if feature of group selected but new parent not)</td>
</tr>
<tr>
<td>Feature from Alternative Group</td>
<td><strong>Conflicting</strong> (if feature selected)</td>
</tr>
<tr>
<td>Feature to Alternative Group</td>
<td><strong>Conflictig</strong> (if feature selected)</td>
</tr>
<tr>
<td>From Or</td>
<td><strong>Conflictig</strong> (if feature is the only selected)</td>
</tr>
<tr>
<td>From Unbounded</td>
<td>Unaffected</td>
</tr>
<tr>
<td>To Unbounded / Or</td>
<td>Unaffected</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rename Feature</th>
<th>Broken Category</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Refactoring</strong></td>
</tr>
</tbody>
</table>
Example

Change Impact Analysis

(Parking Assistance v Adaptive Cruise Control) → Front Distance Sensor

Evolution

Basic
+ Infotainment System
- Assistance Systems

Economic
+ Infotainment System
+ Consumption Indicator

Premium
+ Infotainment System
+ Assistance Systems
+ Parking Assistance
+ Adaptive Cruise Control
+ Side Distance Sensor
+ Front Distance Sensor

Unaffected
Example
Change Impact Analysis

(Parking Assistance v Adaptive Cruise Control) → Front Distance Sensor

Evolution

Basic
+ Infotainment System
- Assistance Systems

Economic
+ Infotainment System
+ Consumption Indicator

Premium
+ Infotainment System
+ Assistance Systems
+ Parking Assistance
+ Adaptive Cruise Control
+ Side Distance Sensor
+ Front Distance Sensor

Add Constraint

Conflicting