AGANDA

Software Product Line

Feature model

Three types of Feature modelling notation
  • Basic feature models
  • Cardinality-based feature models
  • Extended feature models

Example of feature model
SOFTWARE PRODUCT LINE

• Software product family
• a set of software-intensive systems sharing a common managed set of features that satisfy the specific needs of a particular market segment or mission and are developed from common set core assets in a prescribed way
• Feature Model is the core of SPL
SOFTWARE PRODUCT LINE

SPL

Domain Engineering

Aims at understanding the target domain and developing reusable artifacts

Application Engineering

Receives the reusable artifacts developed in the domain engineering phase and creates an appropriate application instance for the given requirements
FEATURE

• A characteristic of a system relevant for some stakeholders
• An abstract concept for describing commonalities and variabilities
• Needs to be decided for each Product Line
A set of features + relationships between features.
- presents common + variable features
- feature forming nodes of the tree. feature variability is represented by the arcs and grouping of features
- It consists of a feature diagrams and some additional information such as short semantic description of each feature, rationale for each feature, constraints, default dependency rules.

A hierarchical form

A tree structure

Feature model in RE
RELATIONSHIP IN THE FEATURE MODEL

- Parent-child features relationships
- Cross-tree constraints
COMMON TYPES OF FEATURE MODELLING NOTATION

- Basic feature models
- Cardinality-based feature models
- Extended feature models
# BASIC FEATURE MODELS

Relationships between a parent feature and its child features (or subfeatures) are categorized as:

<table>
<thead>
<tr>
<th>Relationship</th>
<th>Diagram</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandatory</td>
<td><img src="image" alt="Diagram" /></td>
<td>child feature is required</td>
</tr>
<tr>
<td>Optional</td>
<td><img src="image" alt="Diagram" /></td>
<td>child feature is optional</td>
</tr>
<tr>
<td>Or</td>
<td><img src="image" alt="Diagram" /></td>
<td>at least one of the sub-features must be selected</td>
</tr>
<tr>
<td>Alternative(xor)</td>
<td><img src="image" alt="Diagram" /></td>
<td>one of the sub-features must be selected</td>
</tr>
</tbody>
</table>
BASIC FEATURE MODELS

In addition to the parental relationships between features, cross-tree constraints are allowed. The most common are:

<table>
<thead>
<tr>
<th>A requires B</th>
<th>The selection of A in a product implies the selection of B.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A excludes B</td>
<td>A and B cannot be part of the same product.</td>
</tr>
</tbody>
</table>
BASIC FEATURE MODELS
EXAMPLE

Feature model in RE
CARDINALITY-BASED FMS

- Extending basic feature models with UML-like multiplicities of the form \([n,m]\) with \(n\) being the lower bound and \(m\) the upper bound.

- Feature cardinality: a feature cardinality is a sequence of intervals denoted \([n...m]\). This relationship can be used as mandatory and optional in FODA.

- Group cardinality. Interval denoted \(<n...m>\), with \(n\) as lower bound and \(m\) as upper bound limiting the number of child features that can be part of a product when its parent feature is selected thus, an alternative relationship is equivalent to a \(<1...1>\) and or can be represent as \(<1..N>\) being \(N\) the number of features in the relationship.
CARDINALITY-BASED FMS

- Extending basic feature models with UML-like multiplicities of the form $[n,m]$ with $n$ being the lower bound and $m$ the upper bound.

- Feature cardinality: a feature cardinality is a sequence of intervals denoted $[n...m]$. This relationship can be used as mandatory and optional in FODA.

- Group cardinality: interval denoted $<n...m>$, with $n$ as lower bound and $m$ as upper bound limiting the number of child features that can be part of a product when its parent feature is selected. Thus, an alternative relationship is equivalent to a $<1...1>$ and or can be represent as $<1..N>$ being $N$ the number of features in the relationship.
EXTENDED FEATURE MODELS

• Extended feature models will modified by adding into the model is called feature attributes
• An attribute should consist at least a name, a domain, and a value.
• non-functional features + functional features
EXTENDED FEATURE MODELS

Feature model in RE
AN EXAMPLE OF FEATURE MODEL

- ERP (Enterprise resource planning) systems
- Data base application
- Aim to link and manage the enterprise data in a central database.
- A hierarchical structure
GENERIC STRUCTURE OF ERP SYSTEM
• Method: mapping functional requirements of ERP on a Extended Form of Feature Model
• Main: Merge the ERP hierarchical structure with the extended form of the Feature model
• Consider: Feature types.
• A set of attributes for each type of features
## An Example of Feature Model

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>Level</th>
<th>ID</th>
<th>Name</th>
<th>Parent ID</th>
<th>Category</th>
<th>Length</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERP System</td>
<td>0</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modules</td>
<td>1</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sub Modules</td>
<td>2</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Forms</td>
<td>3</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Reports</td>
<td>3</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
<td></td>
</tr>
<tr>
<td>String</td>
<td>4</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number</td>
<td>4</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>4</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td>4</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>List</td>
<td>4</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boolean</td>
<td>4</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
MAPPING FUNCTIONAL REQUIREMENTS OF ERP SPL ON AN EXTENDED FORM OF FEATURE MODEL
CONCLUSION

• Feature model is the core of SPL
• Feature model capture the commonalities and variabilities in the system
• Three common types of Feature modelling notation
• Modifying feature modelling notation method according to system requirements
THANK YOU

Questions?