

Feature-Oriented Domain Analysis

Requirements Engineering for Software Product Lines
Group 2 - Benedikt Steger, Lucien Heitz

Overview

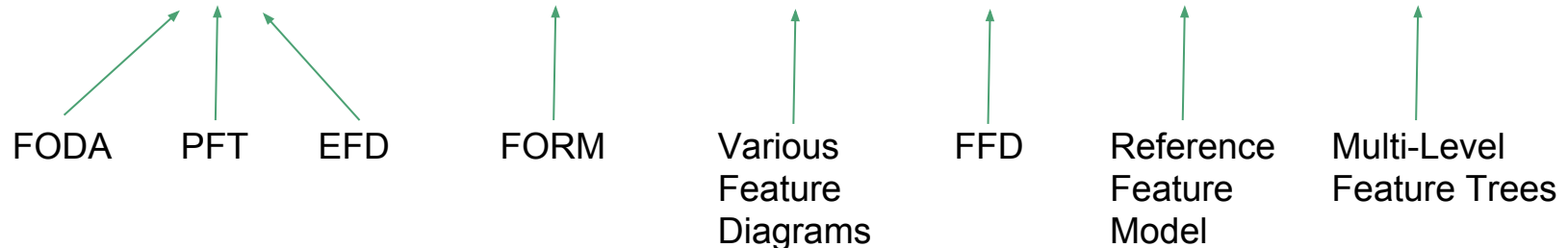
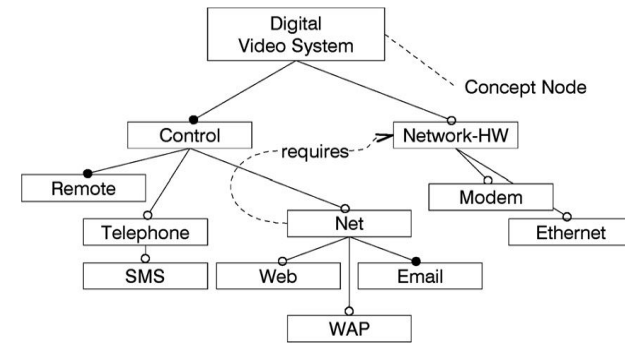
1. Approaches and Goals to Domain Analysis
2. Introduction to Feature Models
3. Domain Analysis with Feature Models
4. Feature Models in Software Product Lines
5. Questions, Remarks and Discussion Topics

1. Approaches and Goals to Domain Analysis

- Systems have multiple domain-related incarnations because
 - they need to adapt quickly due to competition (telephony systems) or
 - have to incorporate many product specific variations (automotive industry).
- Systems need management of desired behaviour
 - to forbid domain feature combinations which lead to interference and
 - to prevent problems with evolving features (new may interfere with old).
- Express complex feature combinations through feature diagrams (**Domain Engineering**).
- Goals of Domain Analysis:
 - guide the selection process of features for new products,
 - facilitate **reuse** and **evolution** of components implementing certain features and
 - provide a mean of communication and coordination between engineers, subcontractors, the management and other stakeholders.

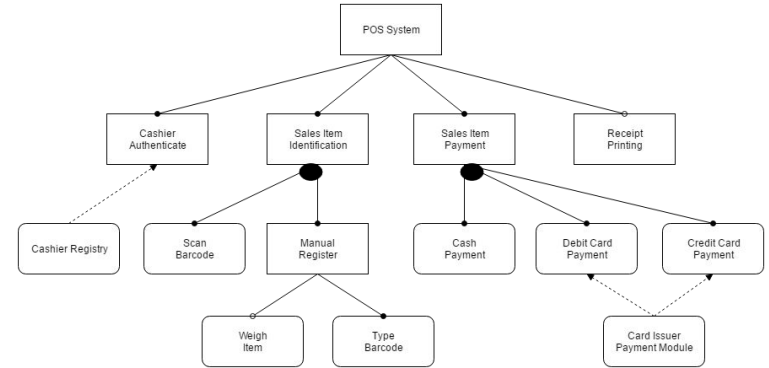
2. Introduction to Feature Models

- Feature models express
 - the dependencies of feature with one another and
 - the variabilities between variants in the same family / **Software Product Line**.
- The purpose of feature models is to provide a tool for
 - quickly adapting to changing requirements,
 - documenting characteristics/dependencies/differences of feature instances and to
 - serve as coarse-grained guideline in requirements specification.
- Various feature diagrams using free feature diagram construction exist.
- Evolution: multiple syntax → expressive → succinct → formal → deviation → continuity



3. Domain Analysis with Feature Models

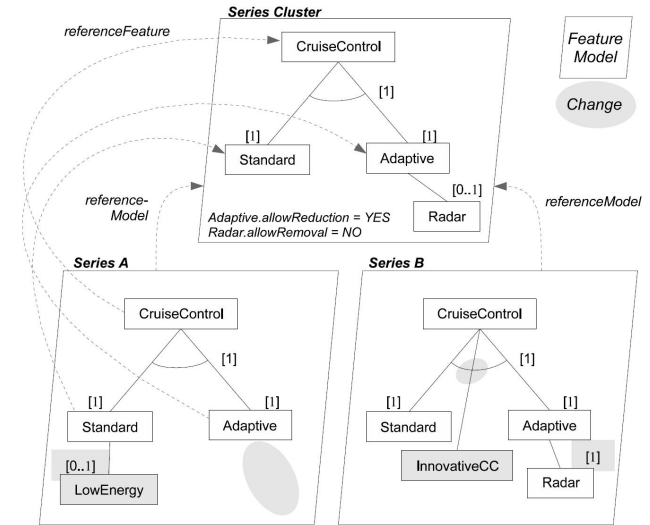
- Challenges and Benefits of Feature Models (Automotive Industry):
 - Bring structure to system components and their interaction.
 - Allow for evolution while keeping track of legacy features.
 - Provide an overview to all type specific variations.
- Limitations of Feature Models:
 - Global, monolithic trees become unmanageable and cannot account for any local variations.
 - Small or type-specific solutions cannot keep track of overall commonalities/differences.



The challenge of domain analysis with feature models is finding “a compromise between **one global** feature model and **many independent** models.” – Reiser, Weber (2006)

4. Feature Models in Software Product Lines

- Multi-level feature trees
 - provide a reference model for FDs,
 - take into consideration local variants and
 - keep track of feature across product families.
- Feature models support engineering process by
 - allowing for both bottom-up as well as top-down reuse of new features,
 - offering horizontal feature propagation (i.e., inheriting features of siblings) and
 - provide a step-by-step integration for feature refinement/evolution over time.



Example of a multi-level tree.

5. Questions, Remarks and Discussion

Dieseldiegate: Volkswagen had intentionally programmed diesel engines to cheat in the emissions controls during laboratory emissions testing.



1. The management denied any knowledge of such software. If that is true, what type of feature model might have been in use?
2. Assume Volkswagen now uses multi-level feature trees. Is it for FDs even possible to have any influence at all? Could this happen again?
3. You buy a competitor for clean fuel engines and enter a similar but new market. Would you integrate their FDs or manage them separately?

References:

- Reiser, M.-O., M. Weber (2006). Managing Highly Complex Product Families with Multi-Level Feature Trees. 14th IEEE International Conference on Requirements Engineering (RE'06), Minneapolis, MN, USA. 149-158.
- Schobbens, P.-Y., P. Heymans, J.-C. Trigaux, Y. Bontemps (2007). Generic Semantics of Feature Diagrams. Computer Networks 51, 2 (Feb. 2007). 456-479.