

Requirements Engineering II

Feature-Oriented Domain Analysis

Contents

- Domain Analysis
- Feature Models
- Comparison of advanced feature models
- Multi Level Feature Trees for Product Line engineering

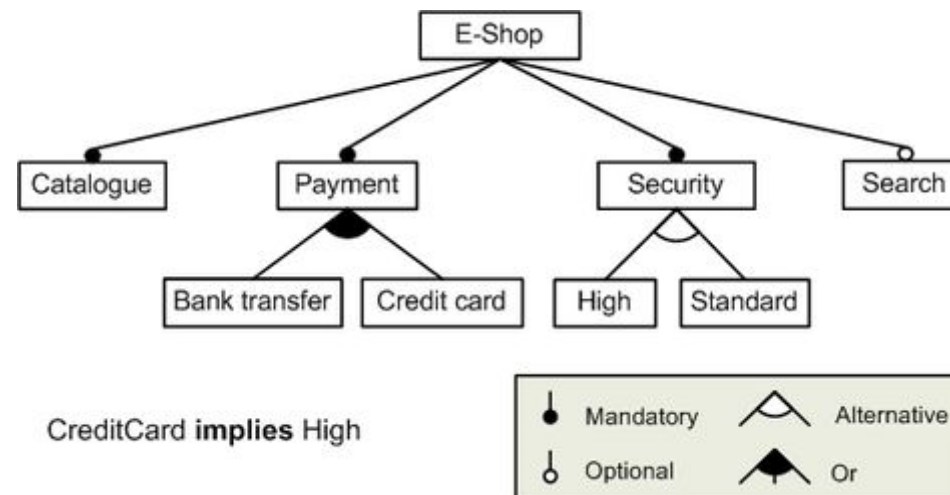
Domain Analysis

- First stage in Feature-Oriented Domain Analysis
- Collect information about domain
- Define the domain
- Result: Domain model
- Aim: Identify common and varying points in a domain
- Enables building product lines / configuration
- Difference to traditional RE: categorization into similarities and differences

Feature Models

- Defines features
 - Graphical representation as feature diagram
 - ... or tabular (relational data)
- ... and their dependencies
- ... within a product line
- Provide an overview for feature construction
- Enable reuse and evolution of software product lines

Feature Models (Basic Example)



Source: Wikipedia

Advanced Feature Models

- OFT
 - Part of Feature-oriented domain analysis
- OFD
 - Same as OFT, but can be directed acyclic graph
- FeatuRSEB
 - Directed acyclic graph
 - xor, or, and
- van Gurp et al. (VBFD)
 - Extend FeatuRSEB: binding times, external features
- Generative programming

Advanced Feature Models (2)

- Riebisch et al. (EFD)
 - Redundancies in FDs
 - Limit them by replacing or, xor, and with UML-like cardinalities and mandatory/optional edges
- Product Line Use case modelling for System and Software engineering (PLUSS)
 - Based on FeatuRSEB
 - Combines Use-cases and FDs
- Multi level feature trees
 - More later

Differences

- Syntax and graphical notation
- Expressiveness (what features they can model)
- Redundancy / Ambiguities
- Various “specializations”

Benefits / Drawbacks

- Analysis
 - “legality” of the diagram
 - Dead feature detection
 - Number of products
 - Model comparisons
- Configuration management
- Creation overhead

Multi-Level Feature Trees

- Feature Models in Product Line Engineering
- Addition of Reference Model, a “parent” of the current model
 - Model deviations in the “child” model
 - Store changes to gain insight on differences (added, deleted, moved features)
- Formal restrictions (not enforced) as to what changes can be made

Multi-level feature trees

- Benefits:
 - Simplifies working in distributed environments
 - Gradual implementation of feature trees within a big company
 - Local modifications/adaption easily possible
 - Semi-structured using restrictions, not enforced to give freedom for practical problems

Questions / Discussion

