

Requirements Engineering II – Assignment 6

Feature-Oriented Domain Analysis

Arthur T.
Jeton M.
Michael S.
Rita P.
Robert R.

Literature

- Managing Highly Complex Product Families with Multi-Level Feature Trees
[Reiser, Weber, 2006]
- Generic semantics of feature diagrams
[Schobbens, Heymans, Trigaux, Bontemps, 2006]

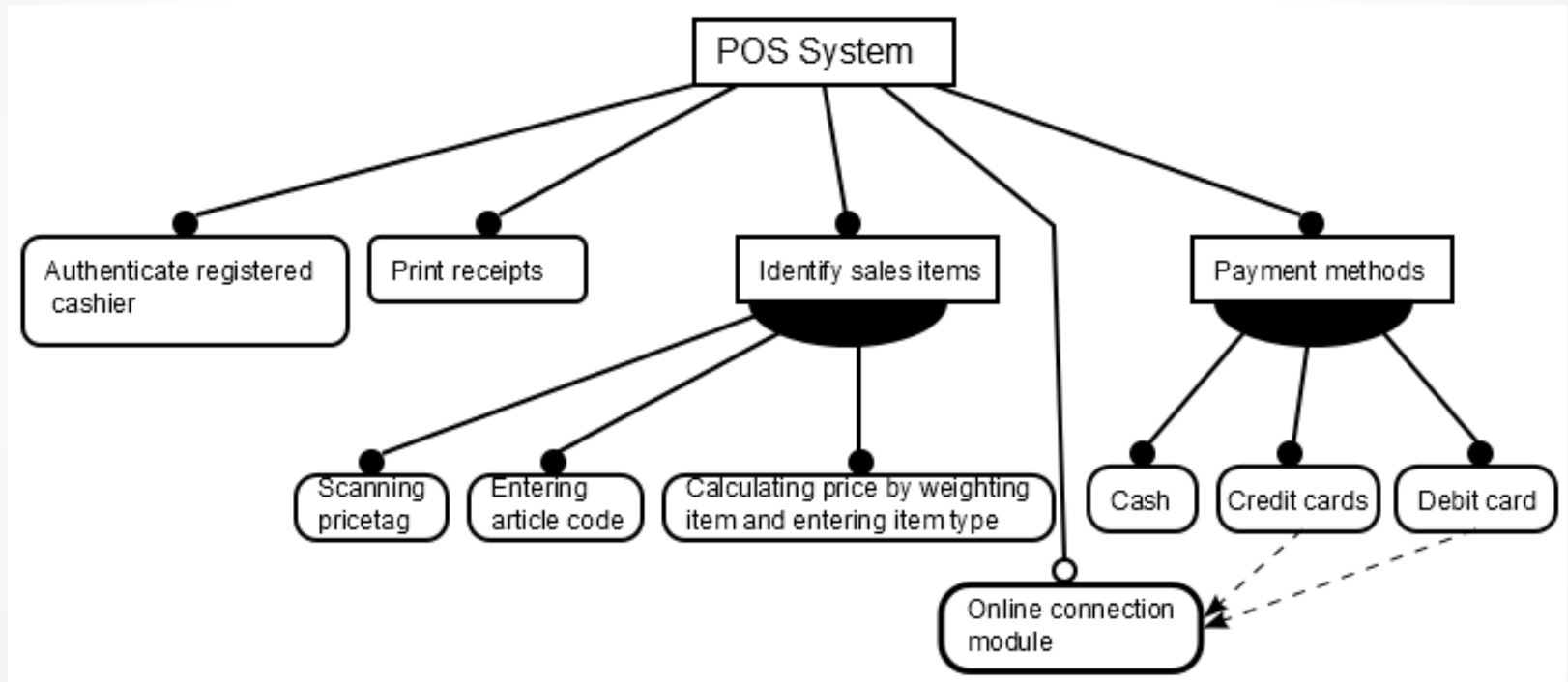
Features (1)

- Product line: Set of products which share characteristics but also show certain differences
- Feature: Distinguishable characteristic of product
- Two processes in software product line engineering:
 - Domain engineering
 - Application engineering
- Domain engineering: Features are units of evolution
- Application engineering: Product instance is defined by selection of features
- Features are link between both processes

Features (2)

- Challenges
 - Feature interaction: New features modify operation of existing features
 - Feature interference: Undesirable feature interaction
- True for domain- and application engineering: Features and the interactions must be well identified and known
- Feature oriented domain analysis (FODA)
 - Need for artefact-independent modelling of features
 - Introduced concept of feature models
 - Introduced feature diagrams
- Feature diagrams used for modelling / visualisation

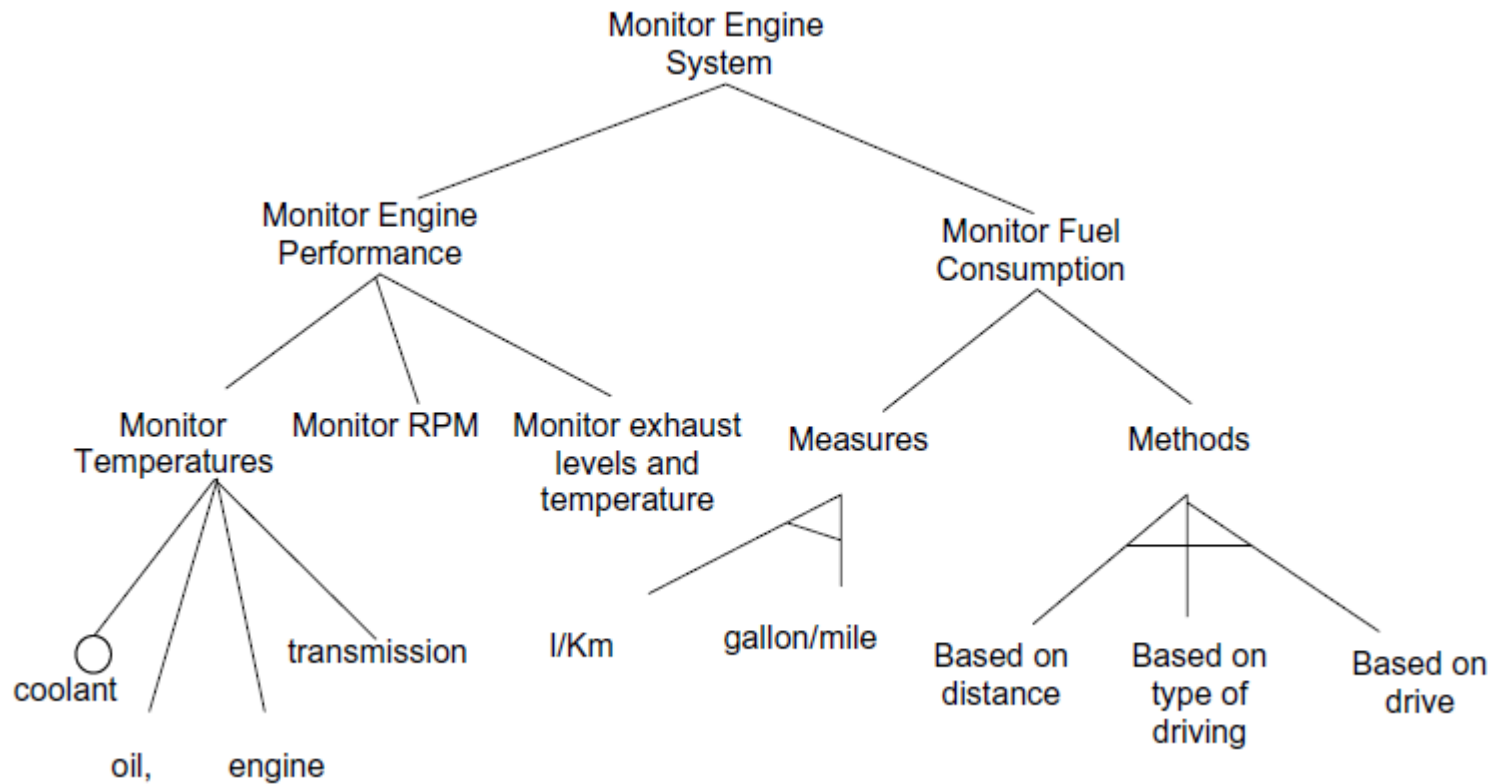
Example FD (POS)



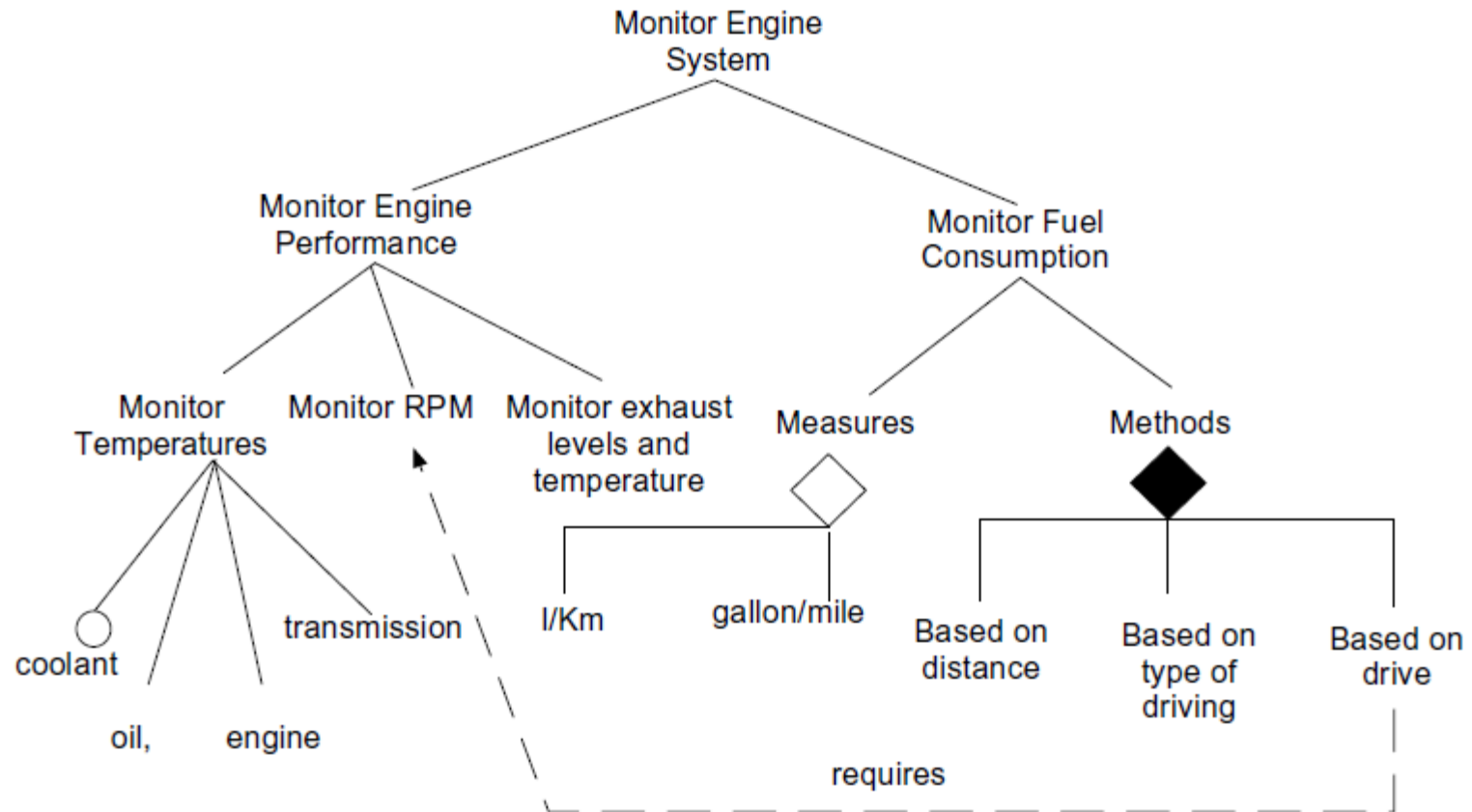
Feature Diagrams (1)

- FDs provide a concise and explicit way to
 - Describe allowed variabilities between products of the same line/family
 - Represent feature dependencies
 - Guide the selection of features allowing the construction of a specific product
 - Facilitate the reuse and the evolution of software components implementing these features
- FD: Graphs or trees
- Different kinds exist

Variations of FDs (1)



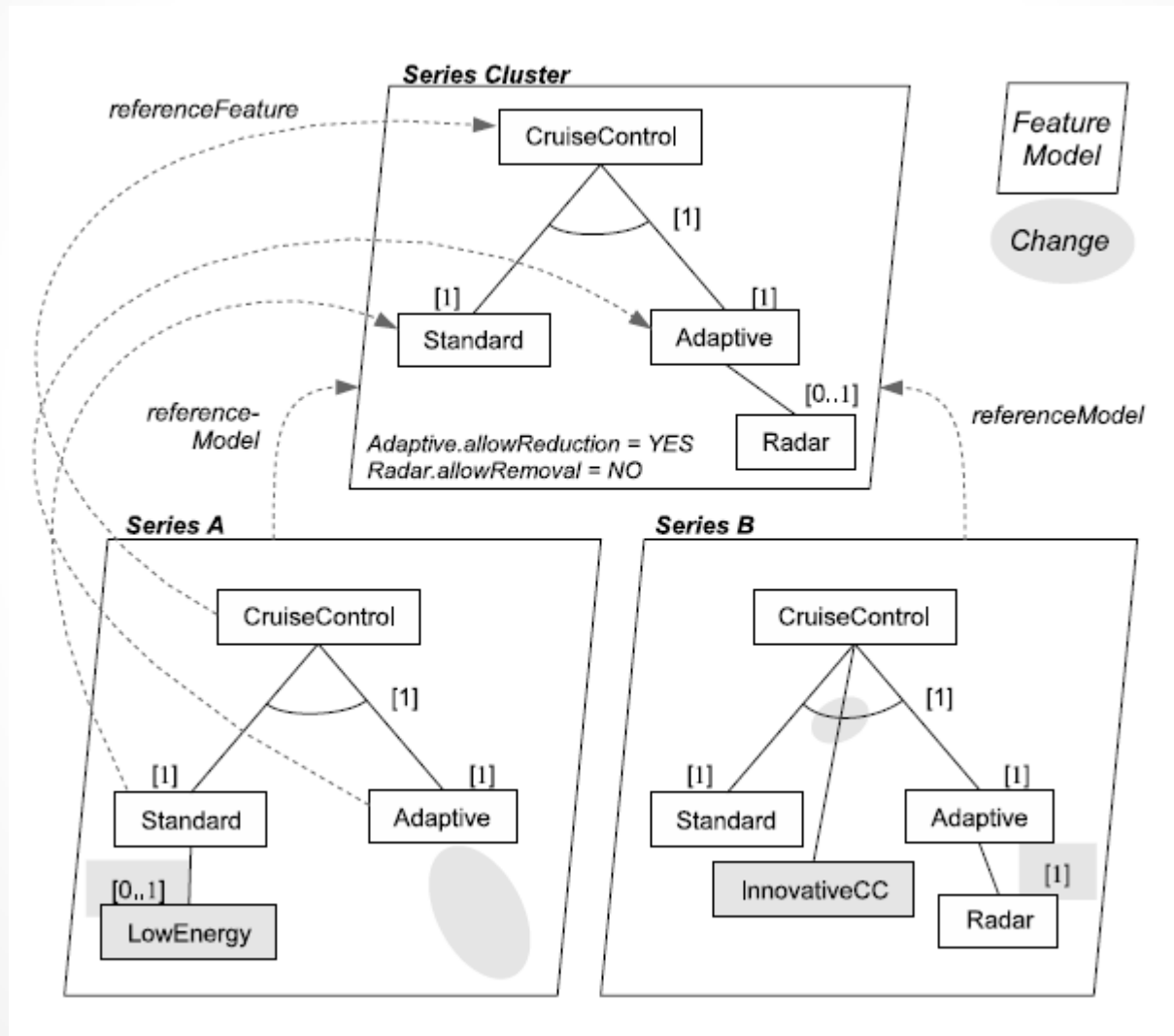
Variations of FDs (2)



Multi-Level Feature Trees (1)

- FD are useful but:
 - Can become very large, complex and inflexible
 - Problems with complex product families
- Use of Multi-Level feature trees to tackle problem
 - Tree of feature models
 - Parent models serve as a reference feature model for children
 - Introduction of changes locally at first and then made globally visible in step-by-step process

Multi-Level Feature Trees (2)



Formal Description of FD (1)

- Multiple forms of FDs
- Similar semantics but different notation
- Formal problems:
 - How to transform between representations?
 - How to compare different FDs?
 - How to support / analyse FDs with tools?
 - How prevent misinterpretation of FDs?
 - How to proof attributes of FDs?

Formal method for describing / defining / comparing FD necessary.

Formal Description of FD (2)

- Free Feature Diagrams (FFD) as generalization of all FD variants
- FDs differ in 4 parameters
 - Tree or diagram?
 - Types of operators on nodes
 - Types of graphical constraints
 - Types of textual constraints
- Use this 4 Parameters for FFD to define each FD language

Summary

- Features are central point in product line development
 - Interface between domain- and application engineering
 - Define the actual product instances
- Feature Diagrams / Trees are useful to describe features of product line
 - But: Complexity / Size problems / Multiple styles
 - Multi-Level Features Trees can help with first two problems
- Formal methods / framework necessary to enable proof of and tool support for Feature Trees

Questions?