

Feature-Oriented Domain Analysis

Philipp Huber, Michael Schneider & Appiah-Bonna Emmanuel Twumasi

Assignment Questions:

- ??? What is the basic approach for domain analysis?
- ??? What does a feature model express?
What is the purpose of a feature model?
- ??? What advanced forms of feature models exist?
How do they differ?
- ??? Benefits and limitations of feature modelling for domain analysis?
- ??? How are feature models used in the software product line engineering?

Feature-Oriented Domain Analysis

Agenda:

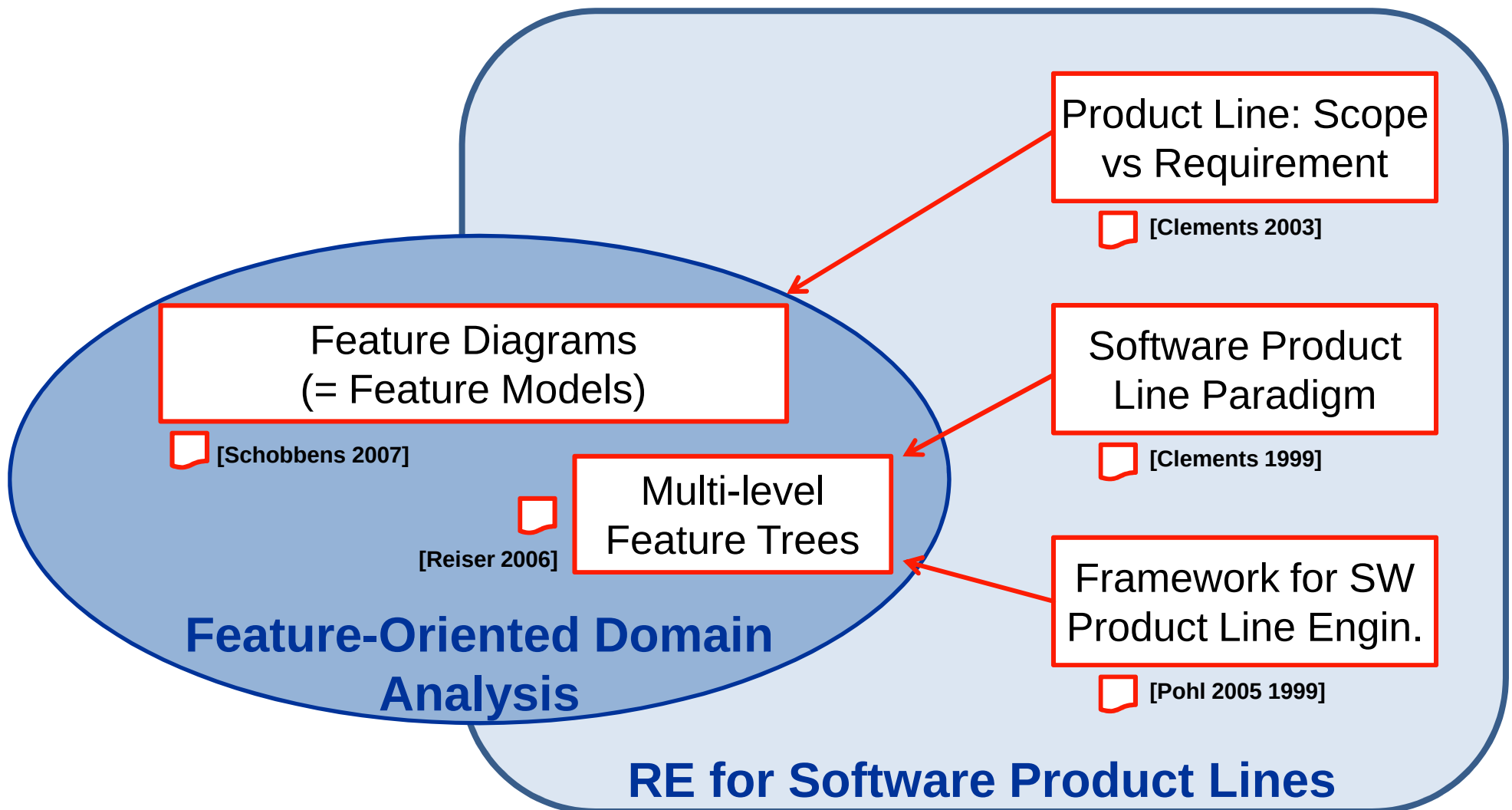
1. Relate Session Topics

2. Feature Models

- An Approach for Domain Analysis
- Advanced Forms of Feature Models
- Benefits and Limitations of Feature Modelling

3. Feature Modelling for Software Product Lines

PHILIPP HUBER III MICHAEL SCHNEIDER III APPIAH-BONNA EMMANUEL TWUMASI IIIIIIII FEATURE-ORIENTED DOMAIN ANALYSIS



Feature-Oriented Domain Analysis

Agenda:

1. Relate Session Topics

2. Feature Models

- An Approach for Domain Analysis
- Advanced Forms of Feature Models
- Benefits and Limitations of Feature Modelling

3. Feature Modelling for Software Product Lines

A feature is ...

“a distinguishable characteristic of a concept (e.g., system, component and so on)

that is relevant to some stakeholder of the concept.”

Features have dependencies ...

- can be decomposed in tree structures
- may have dependencies between each other
- are not used for one-shot developments, but for systems that must have multiple incarnations, because
 - either: they need to evolve through time (SW releases)
 - and/or: variants must be deployed in many contexts (different products of a software product line)

Feature Models (Feature Diagrams)

- Popular approach to manage software product lines
 - Documentation about their features = characteristics
 - Which are common to all product instances?
 - Which differ from one product instance to another?
 - Dependencies between characteristics?
- Feature Diagrams (FDs)
are a family of popular modelling languages

Advanced Forms of Feature Models

- Feature Diagrams
- Feature Diagrams + Generative Programming
 - decomposition constraints
 - graphical constraints
 - staged configuration
- Feature Diagrams + Abstract Syntax
 - (e.g. Formal specification languages, see RE I chapter 10).
 - define tree including all rules
 - notation for all dependencies/constraints
 - between features
 - products within product lines, config parameters
 - etc. (e.g. Releases)
- Generic formalization of feature diagrams + abstract syntax
(see further details in [Schobbens 2007])
- Multi-level Feature Tree

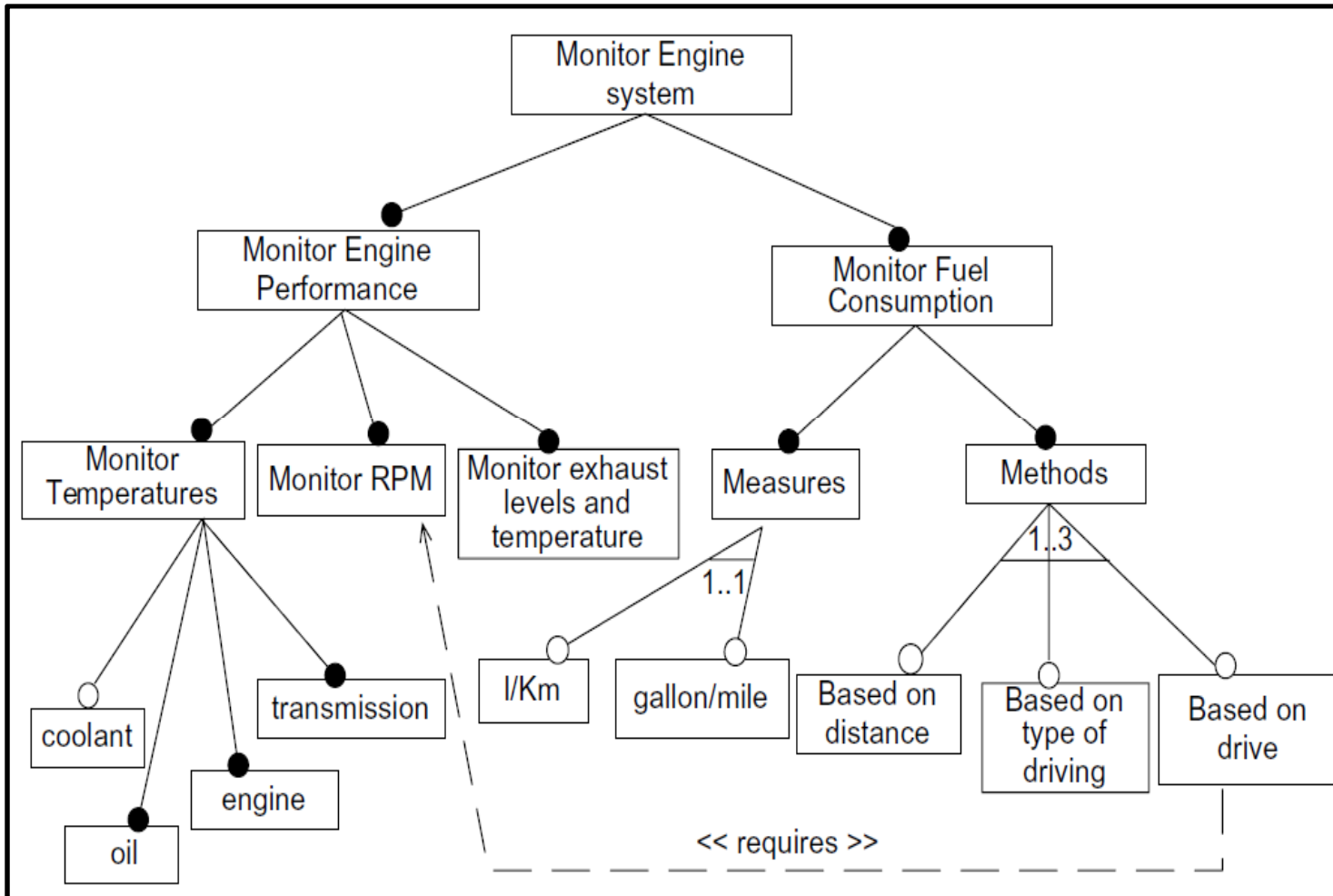
(derived from Reuse-Driven Software Engineering Business method)



Optional Features

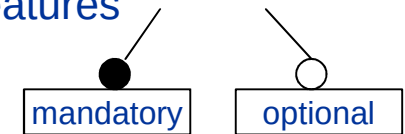


“requires” / “mutex”

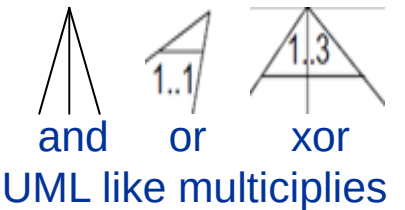


Comments:

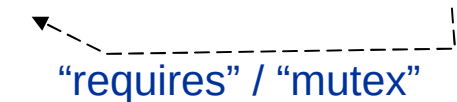
Features



Decomposition



Constraints:



(Product Line Use Case Modelling for System and Software Engineering Method)

12





 mandatory optional

“requires” / “mutex”

Benefits of Feature Models

- Describe allowed variabilities between products of the same product line
 - 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.
-
- Define the scope per product and release
 - Formal notation as base for feature management especially also for automated tools

Limitation of Feature Modelling

- Complexity, especially for complex product lines
 - Diagrams: dito.
 - Abstract Syntax for rules/dependencies: Hard to understand (for normal stakeholders or when not deeply involved)
 - Efforts for mass changes and re-design of architecture
 - Beside of tree, name and dependencies, also other information is needed (feature ID, description, releases/dates, priority, status, etc.)
- Visualise and discuss specific aspects
- Base for Tools, especially for product line with mass customisation
complex standard software (COTS)
- But often: Spreadsheet with tree control and all required info will do

Feature-Oriented Domain Analysis

Agenda:

1. Relate Session Topics

2. Feature Models

- An Approach for Domain Analysis
- Advanced Forms of Feature Models
- Benefits and Limitations of Feature Modelling

3. Feature Modelling for Software Product Lines

Feature Modelling of Product Lines

→ Possible Approaches

- One single model for all individual products within the software product line
 - As presented in the models above
 - But: having views/filters
(e.g. per delivered individual product or per selected combination of other parameters)
- “Multi-level Feature Trees”
Reference Model <-> Individual Models per product
 - As applied by DaimlerChrysler [Reiser 2006]
 - See next slides
- Special case: Hierarchical Product Lines

Context Variability of a Software Product Line

A silver Mercedes-Benz E-Class sedan is shown from a front-three-quarter view, parked on a light-colored surface. The car is sleek and modern, with the Mercedes-Benz logo prominently displayed on the front grille.

Tourer



Age Module | AMG

Tourer

T-Modelle

Coupés

Shooting Brakes

Cabriolets/Roadster

Offroader

Vans & Reisemobile

PHILIPP HUBER III MICHAEL SCHNEIDER III APPIAH-BONNA EMMANUEL TWUMASI IIIIIIII FEATURE-ORIENTED DOMAIN ANALYSIS

Change



Benefits of “Multi-level Feature Trees”

Top-down

- Central control and re-use of common features
- Speed to implement new features in all individual products or to implement a new product

Bottom-up

- Allows independent views, features and priorities for implementation per individual product
- Speed and flexibility for innovation
- Can still be taken over into reference model (even best solution from parallel developments)