

Software Product Line

Feature-Oriented Domain Analysis

Requirement Engineering II

Group 1 :

Elfat Esati

Sunita Soumyadarsana Mishra



University of
Zurich^{UZH}

Introduction



Topics

- Summary
- Presentation of findings
- Analysis and Interpretation
- Approaches introduced
- Limitations



Presentation of findings



Domain Analysis in Domain Engineering

Promising goal → Software reuse for improving software development life-cycle

- Can maximise profits through software reuse
- Reduce time through prior software systems
- Decreased cost is obvious even in the implementation phase
- Lines of codes reduces by 75%

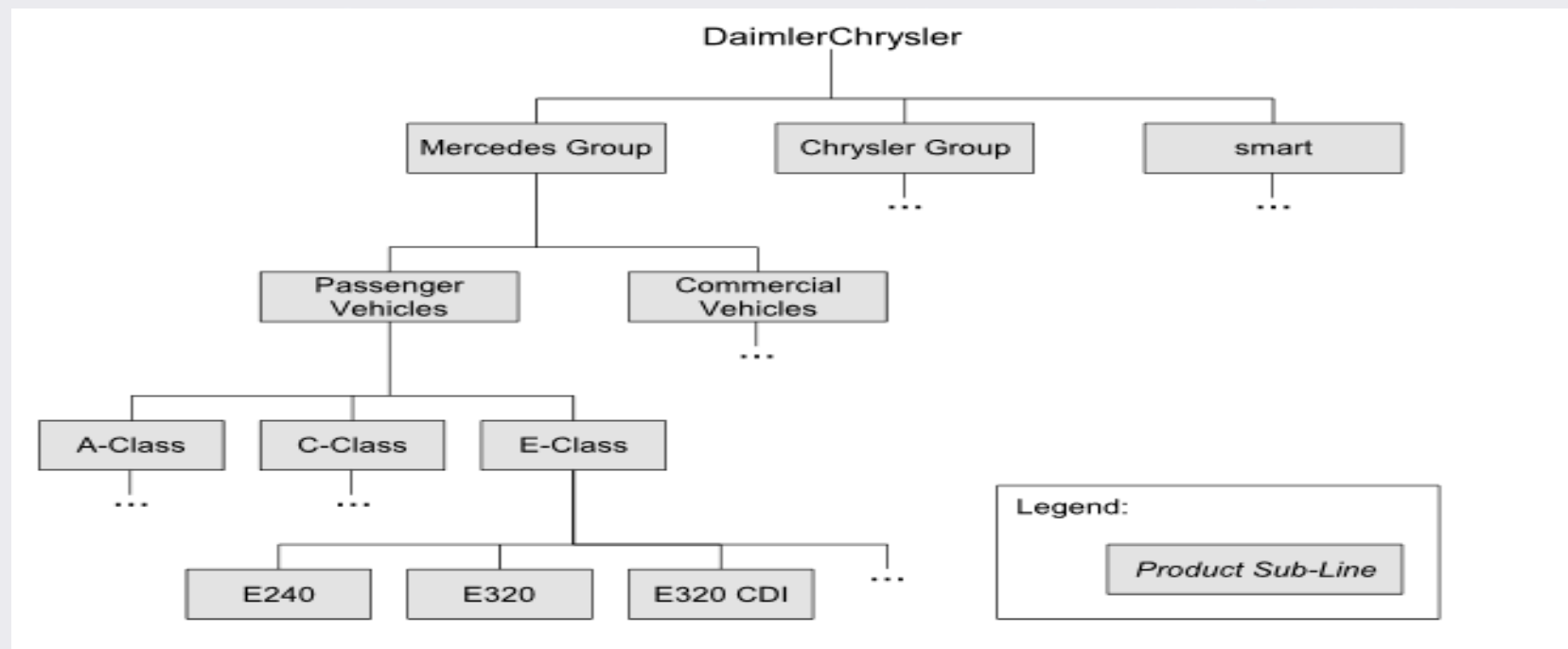


Presentation of findings



Feature modeling for Domain Analysis

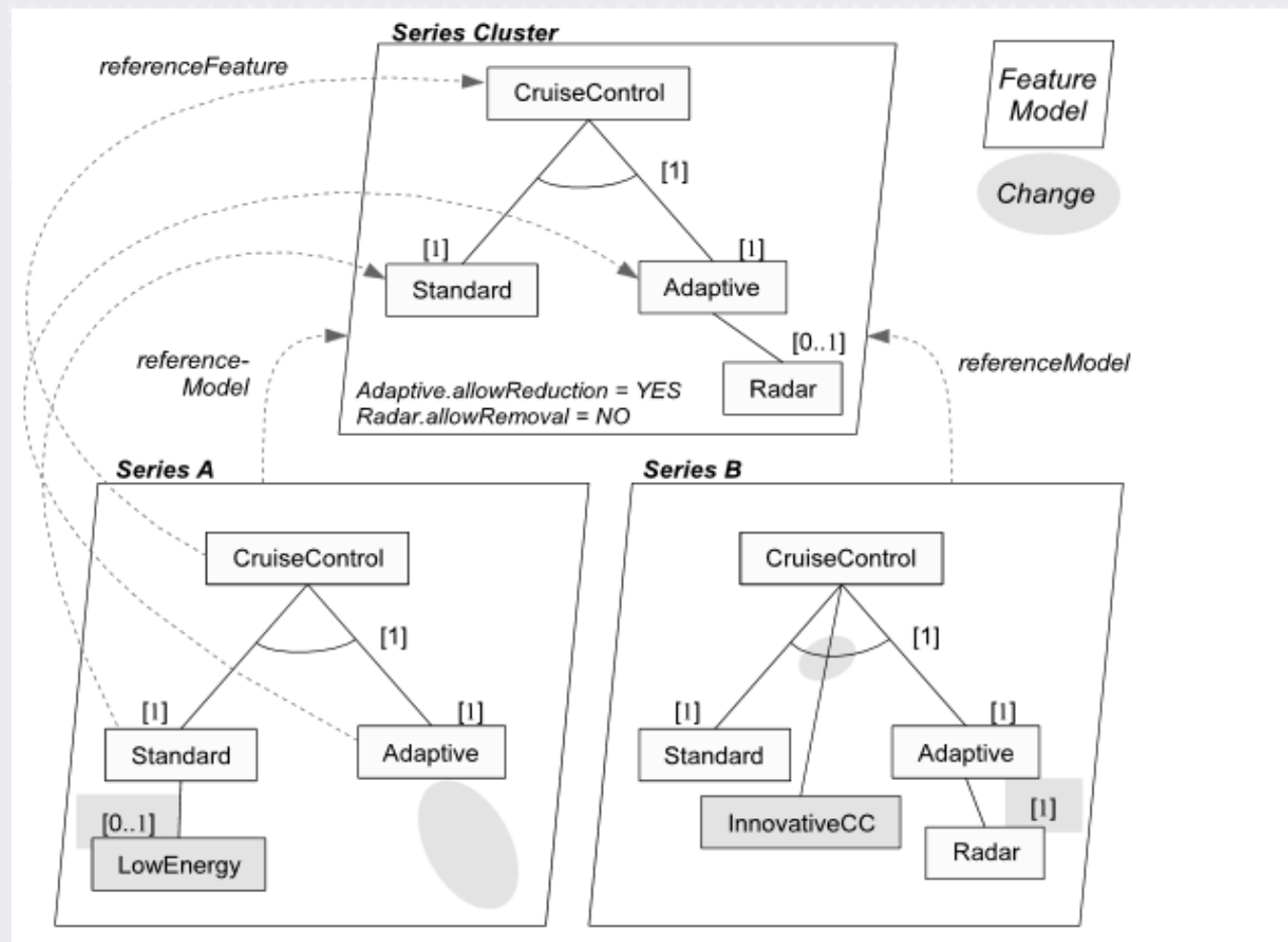
- Represented by feature diagrams
- Depicts the shared characteristics and the differences of set of products



Analysis and Interpretations



Multi Level Feature Trees as an approach to Domain modeling



Semantics matters



- The problem: Divergent semantics by different scientists lacked standardization - analysts had a tough choice of picking appropriate language first and then tackle the assignment.
- The proposal: Developing a rigorously defined generic FD language framework
- The challenge: Proving that language using the new framework is at least equally expressive, less ambiguous and contains lower redundancies than existing extensions of FD languages.



The problem



Formal definition of FD variants on top of FFD

Short name	GT	NT	GCT	TCL
OFT	1	$and \cup xor \cup \{opt_1\}$	$\emptyset$	CR
OFD	0	$and \cup xor \cup \{opt_1\}$	$\emptyset$	CR
RFD	0	$and \cup xor \cup or \cup \{opt_1\}$	$\{\Rightarrow, \}$	CR
EFD	0	$card \cup \{opt_1\}$	$\{\Rightarrow, \}$	CR
GPFT	1	$and \cup xor \cup or \cup \{opt_1\}$	$\emptyset$	CR
PFT	1	$and \cup xor \cup or \cup \{opt_1\}$	$\{\Rightarrow, \}$	$\emptyset$
CFD(OP)	0	OP	$\emptyset$	$\emptyset$
COFD	0	$and \cup xor$	$\emptyset$	$\emptyset$
VFD	0	$card$	$\emptyset$	$\emptyset$
CFT(OP)	1	OP	$\emptyset$	$\emptyset$
COFT	1	$and \cup xor$	$\emptyset$	$\emptyset$
CRFT	1	$and \cup xor \cup or$	$\emptyset$	$\emptyset$

Going granular



- GT - Graph type - Boolean 0 for DAG and 1 for Tree
- NT - Node type - set of operators allowed per arity
- GCT - Graphical constraint type, {requires, mutex, 0}
- TCL - Textual constraint language, between two primitive features

Formal definition of FD variants on top of FFD

Short name	GT	NT	GCT	TCL
OFT	1	$and \cup xor \cup \{opt_1\}$	$\emptyset$	CR
OFD	0	$and \cup xor \cup \{opt_1\}$	$\emptyset$	CR
RFD	0	$and \cup xor \cup or \cup \{opt_1\}$	$\{\Rightarrow, \}$	CR
EFD	0	$card \cup \{opt_1\}$	$\{\Rightarrow, \}$	CR
GPFT	1	$and \cup xor \cup or \cup \{opt_1\}$	$\emptyset$	CR
PFT	1	$and \cup xor \cup or \cup \{opt_1\}$	$\{\Rightarrow, \}$	$\emptyset$
CFD(OP)	0	OP	$\emptyset$	$\emptyset$
COFD	0	$and \cup xor$	$\emptyset$	$\emptyset$
VFD	0	$card$	$\emptyset$	$\emptyset$
CFT(OP)	1	OP	$\emptyset$	$\emptyset$
COFT	1	$and \cup xor$	$\emptyset$	$\emptyset$
CRFT	1	$and \cup xor \cup or$	$\emptyset$	$\emptyset$



The proposal



- Abstract vs Concrete semantics - Bidirectional conversion
- Mathematical models (set theory) vs meta model (UML)
- Primitive vs compound features
- Decomposition vs constraint edges
- Mandatory vs optional
- ...and a series of definitions to define contents, parameters and structures (including parent, son and leaf)



The challenge



Proving superiority needs comparison across multiple relevant metrics. Following parameters were used

- Expressiveness
- Embeddability
- Succinctness
- Redundancy

And winning line

“Expressively complete and polynomially succinct” a requirement for satisfiability.



THANK YOU.....

DO YOU HAVE ANY QUESTIONS ?



Discussion



Any question ?



University of
Zurich ^{UZH}