

Product Derivation with Requirements Models

Team Presentation in **Requirements Engineering II**

Team:

Brändle, Heimgartner, Liu, Melwin, Schneider, Schneider

Presentation:

Moritz Schneider (11-490-455),
Christian Schneider (10-606-002)

Date:

November 23, 2015

Product Derivation with Requirements Models -- Agenda

- Challenges addressed by the approaches
- Approach 1:
Requirements model derivation from a product-line specification
- Approach 2:
Mapping Features to Models
- Comparison of the approaches
- Discussion

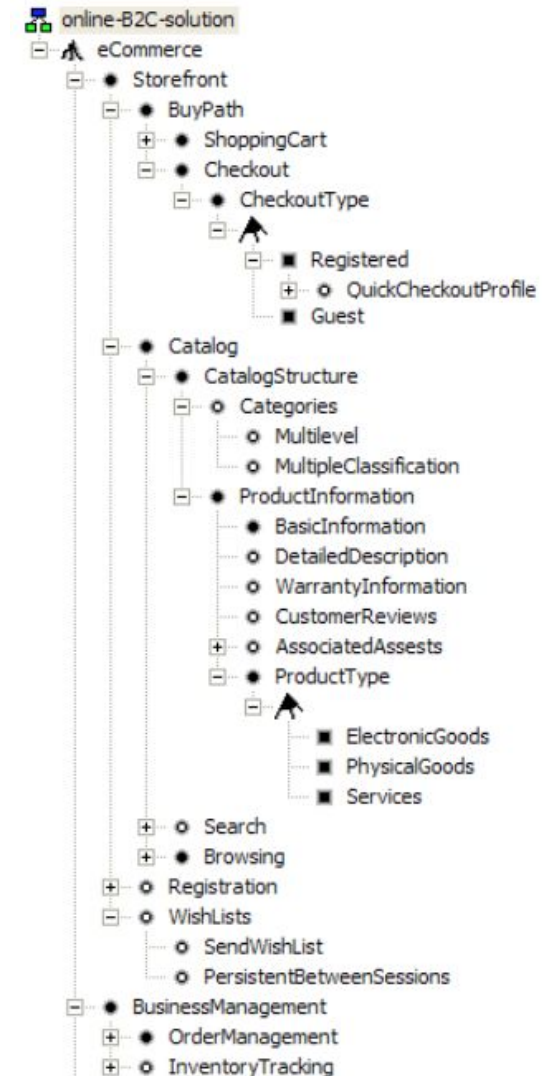
Challenges addressed by the approaches

Feature Models

- Can represent
 - commonalities and
 - variables.
- But merely consists of symbols.
- Functions
 - product-line scoping
 - derive domain-specific languages

Challenges

- Maintaining consistencies
- Connecting with requirements



(a) Online B2C Feature Model

Requirements model derivation from a product-line specification

How can a requirements model be derived from a product-line specification?

Two approaches

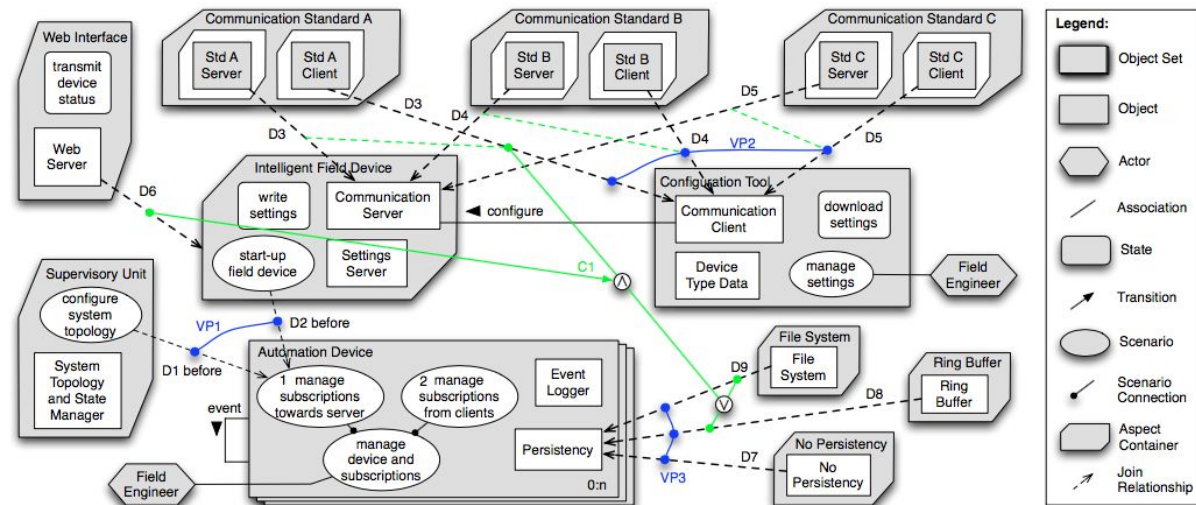
- **Incremental Product Derivation**
(Stoiber and Glinz, 2010a)
- **Mapping Features to Models / Superimposed Variants**
(Czarnecki and Antkiewicz, 2005)

Incremental Product Derivation in Product Line Requirements Engineering

Adora requirements specification

Extended with join relationships

- Features are modeled in tables
- Can be set true or false



Decision Table

ID	Description / Derivation Question	Design Rationale	Constraints	ifTrue	ifFalse	Decision
D1	Should the automation device be a supervisory unit?	The automation device platform is used for cost and quality reasons.	VP1	~D2	D2	undecided
D2	Should the automation device be an intelligent field device?	The automation device platform is used for cost and quality reasons.	VP1	~D1	D1 ~D6	undecided
D3	Should the intelligent field device support communic. std. A?	Standard A is the newest and most performant one.	VP2, C1		~D6	undecided
D4	Should the intelligent field device support communic. std. B?	Is a must in some countries.	VP2			undecided
D5	Should the intelligent field device support communic. std. C?	C is still required by many legacy systems.	VP2			undecided
D6	Should the intelligent field device a web interface?	The web interface allows access over the world wide web.	C1	D3 ~D7		undecided
D7	Is there no memory in the automation device?	Might be interesting for non-critical systems.	VP3	~D6 ~D8 ~D9		undecided
D8	Is there a ring buffer in the automation device?	Ring buffers are cheap to implement and fast.	VP3, C1	~D7 ~D9		undecided
D9	Is there a file system in the automation device?	File system offer huge amounts of storage space.	VP3, C1	~D7 ~D8		undecided

Variation Points Table

ID	Description / Rationale	minCard	maxCard	Decisions Involved
VP1	An automation device always can be either a supervisory unit or an intelligent field device.	1	1	D1, D2
VP2	There are three different communication standards to configure a field device: standard A, B, and C. At least one must be chosen.	1	3	D3, D4, D5
VP3	There exist three different persistency types: no persistency, a ring buffer, or a file system. One of these must be selected.	1	1	D7, D8, D9

Constraints Table

ID	Description / Rationale	Antecedent	Operator	Consequent
C1	A web interface always requires the communication standard A and a local persistency installed (either ring buffer or FS).	D6	=>	D3 and (D8 or D9)

Fig. 1. An automation device product line specified in the ADORA language.

Incremental Product Derivation in Product Line Requirements Engineering

Problem:

Changes can not be made at once
(inconsistencies!)

Therefore:

Step-by-step deciding
(as of this statechart)

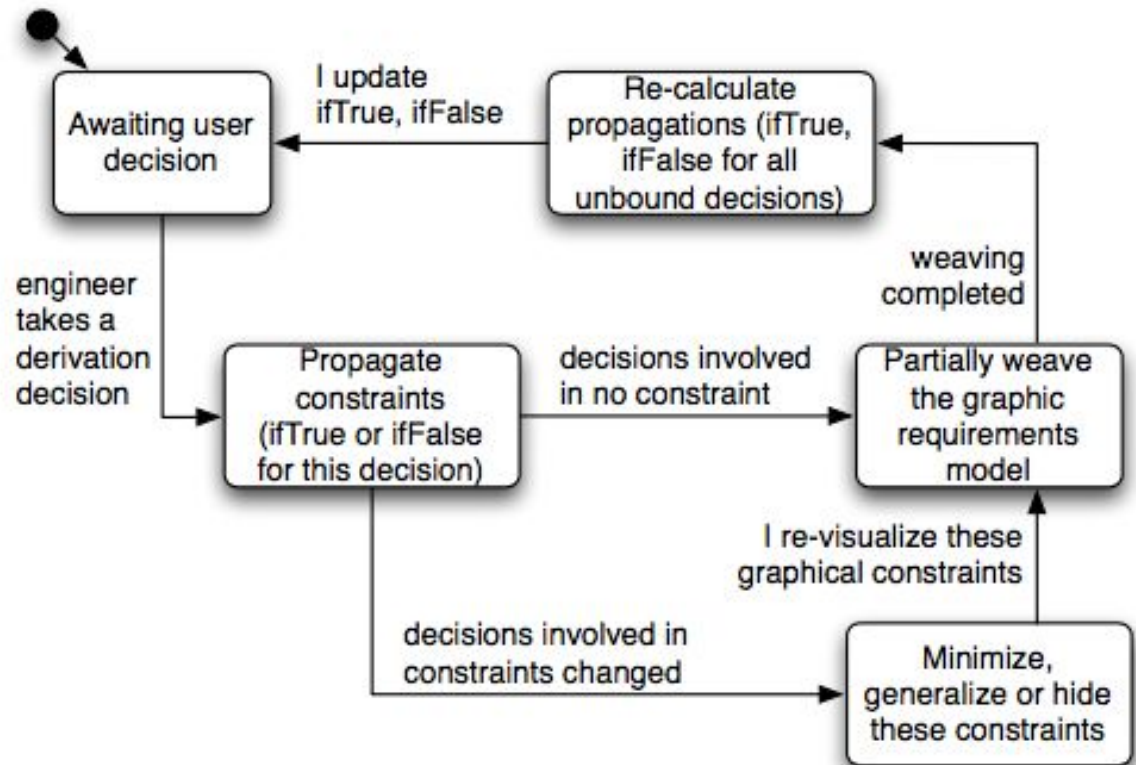
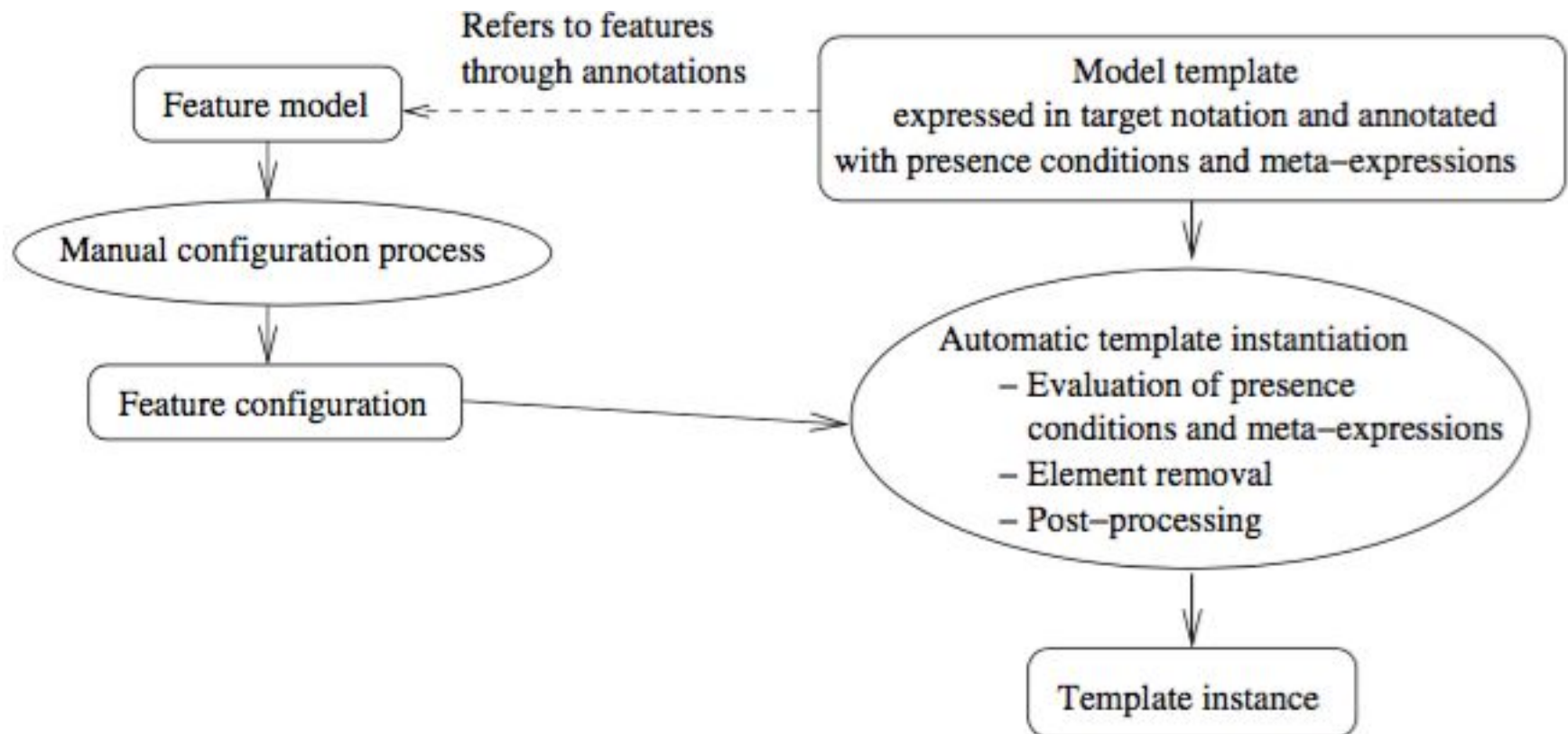


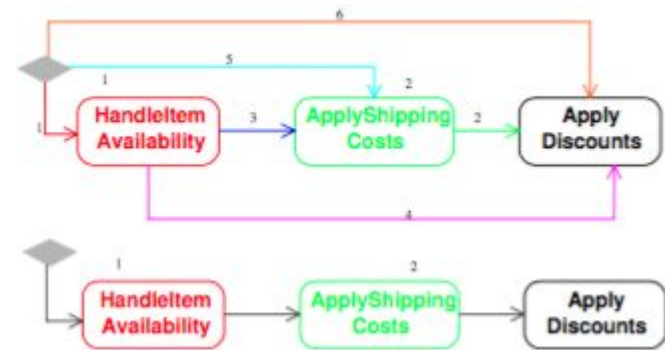
Fig. 3. A statechart describing the behavior of the ADORA tool when executing a variability binding decision taken in a product derivation process.

Mapping Features to Models



Mapping Features to Models - Template Instantiation

- Patch application
 - Is a transformation that automatically fixes a problem which may result from removing elements
 - e.g. Automatic flow closure
- Simplification
 - Involves removing elements that have become redundant after removing other elements
 - e.g. Removal of redundant control nodes



Comparison of the approaches

	Incremental Product Derivation	Mapping Features to Models
Advantages	One single integrated hierarchical model keep large models comprehensible for human users (Adora) Due to incremental process complexity reduction Allowing to model complex product lines	Show directly the impact of selecting a given feature on the resulting model Model templates can be created incrementally and simultaneously with feature model. Patching and simplification results in simpler templates Can be applied for models at any level
Disadvantages	In terms of runtime performance no satisfactory implementation yet (work in progress)	Annotation is not always simple and require few iterations

Discussion

Which of the two presented approaches is more advantageous in practice?

Can you think of practical use cases for Product Derivation?
Where can this be used?