

Product Derivation with Requirements Models

Assignment 6 - Topic C

Philip Schaffner, Robert Erdin, Taya Goubran

Relation to Topic

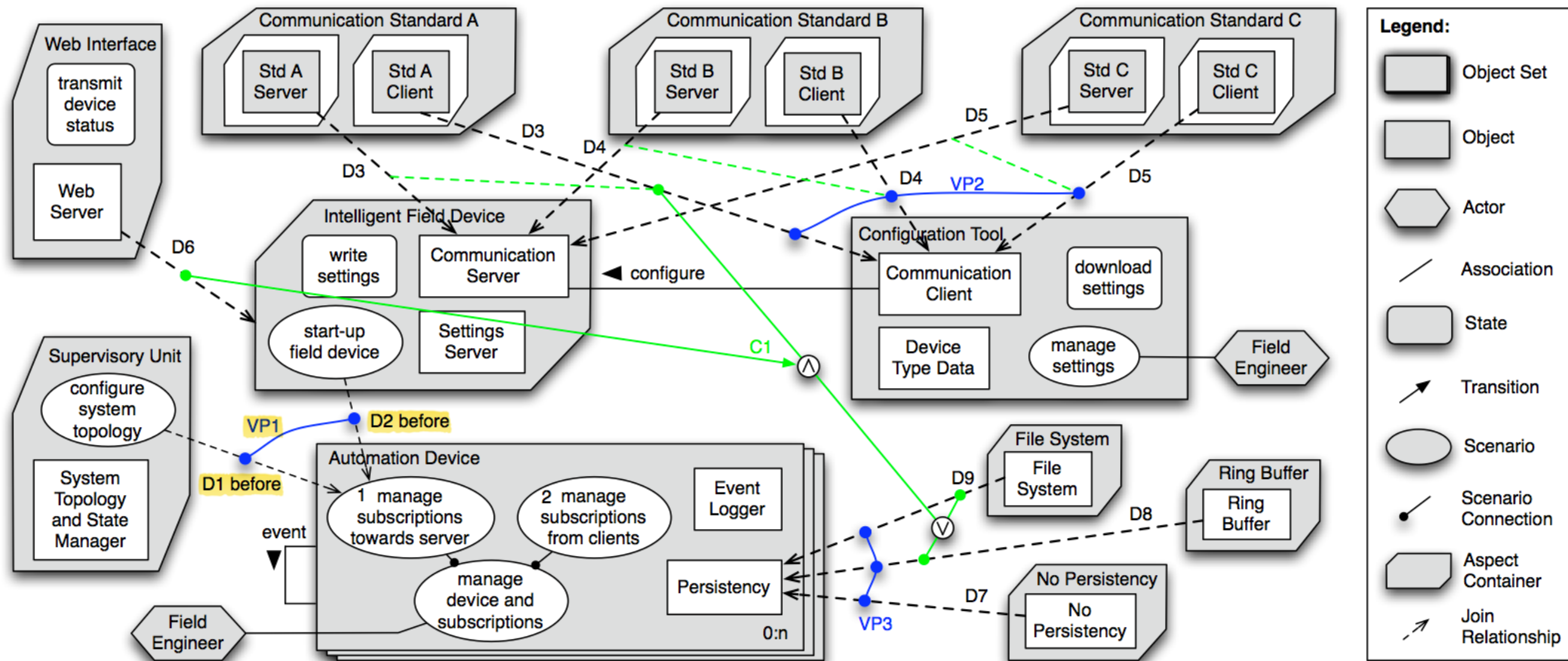
- Mapping features to Models: A Template Approach Based on Superimposed Variants - Czarnecki and Antkiewicz
- Supporting Stepwise, incremental Product Derivation in Product Line Requirements Engineering - Stoiber und Glinz

Product Derivation

- **Product Derivation:** The process of defining a single application product based on a product line variability model.
- Two challenges faced:
 - not violating the variability constraints
 - understand the effects and consequences of every variability binding decision.

The approach

- modeling based on:
 - graphical object-oriented requirements engineering language ADORA
 - ADORA's aspect-oriented modeling capabilities
 - new table-based boolean decision modeling concept
- The aim is to describe:
 - variability decision
 - constraints modeling
 - stepwise incremental derivation of products



Decision Table all decisions, details and constraints

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 cardinalities and involved decision items

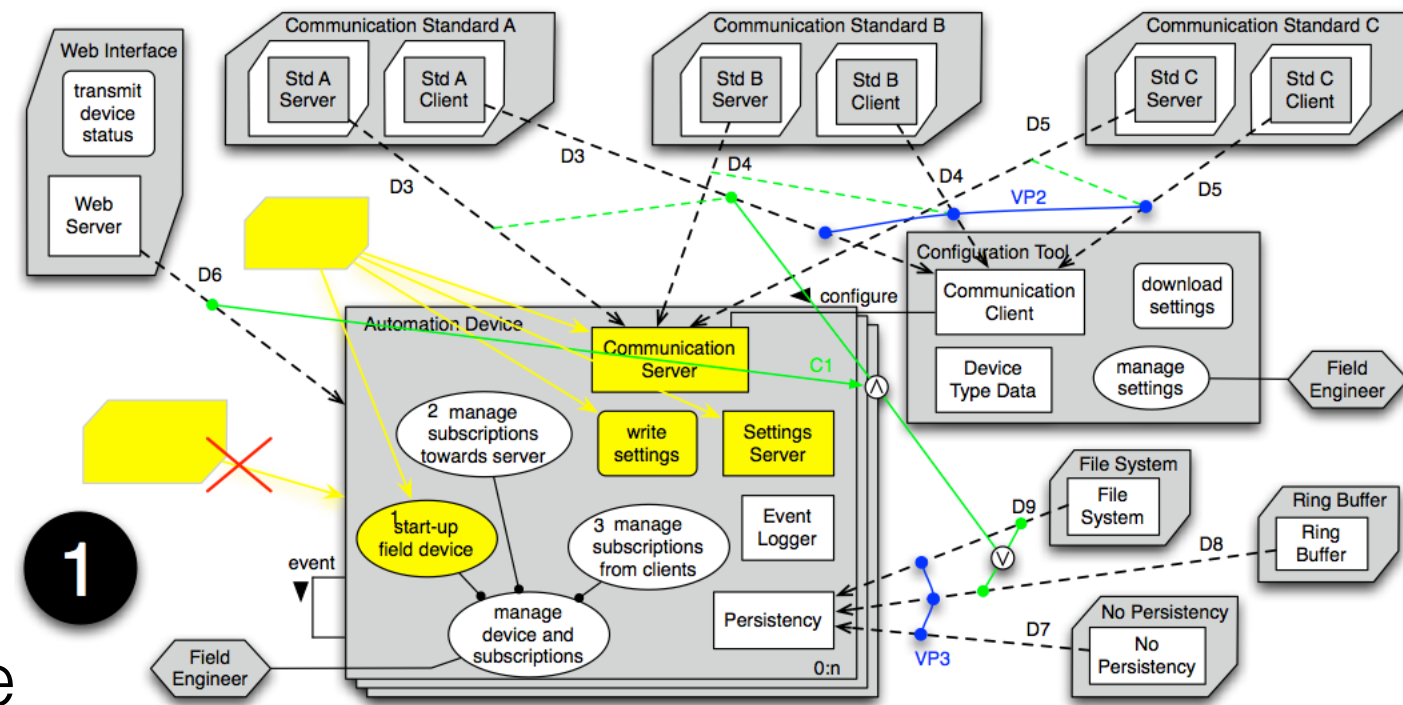
ID	Description / Rationale	minCard	maxCard	Decisions Involved
VP1	An automation device always can be either a supervisory unit or an intelligend field device.	1	1	D1, D2
VP2	There are three different communication standards to configure a field device: standard A, B, and C. At least one must be chosen.	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 specifies complex formulas in boolean logic

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)

Choose an Intelligent Field Device (D2 = true)

1. propagate the constraints:
D1= false
2. minimize by hiding the variation point VP1 (VP1= true)
3. weave the fragment of Intelligent Field Device into the Automation device
4. The join relationships receive new target join points
5. recalculations of the constraints for all remaining decisions
6. IfTrue and IfFalse values for D1 and D2 disappear



Decision Table

ID	...	Constraints	ifTrue	ifFalse	Decision
D1		VP1	-	-	false
D2		VP1	-	-	true
D3		VP2, C1		¬D6	undecided
D4		VP2			undecided
D5		VP2			undecided
D6		C1	D3 ¬D7		undecided
D7		VP3	¬D6 ¬D8 ¬D9		undecided
D8		VP3, C1	¬D7 ¬D9		undecided
D9		VP3, C1	¬D7 ¬D8		undecided

Variation Points Table

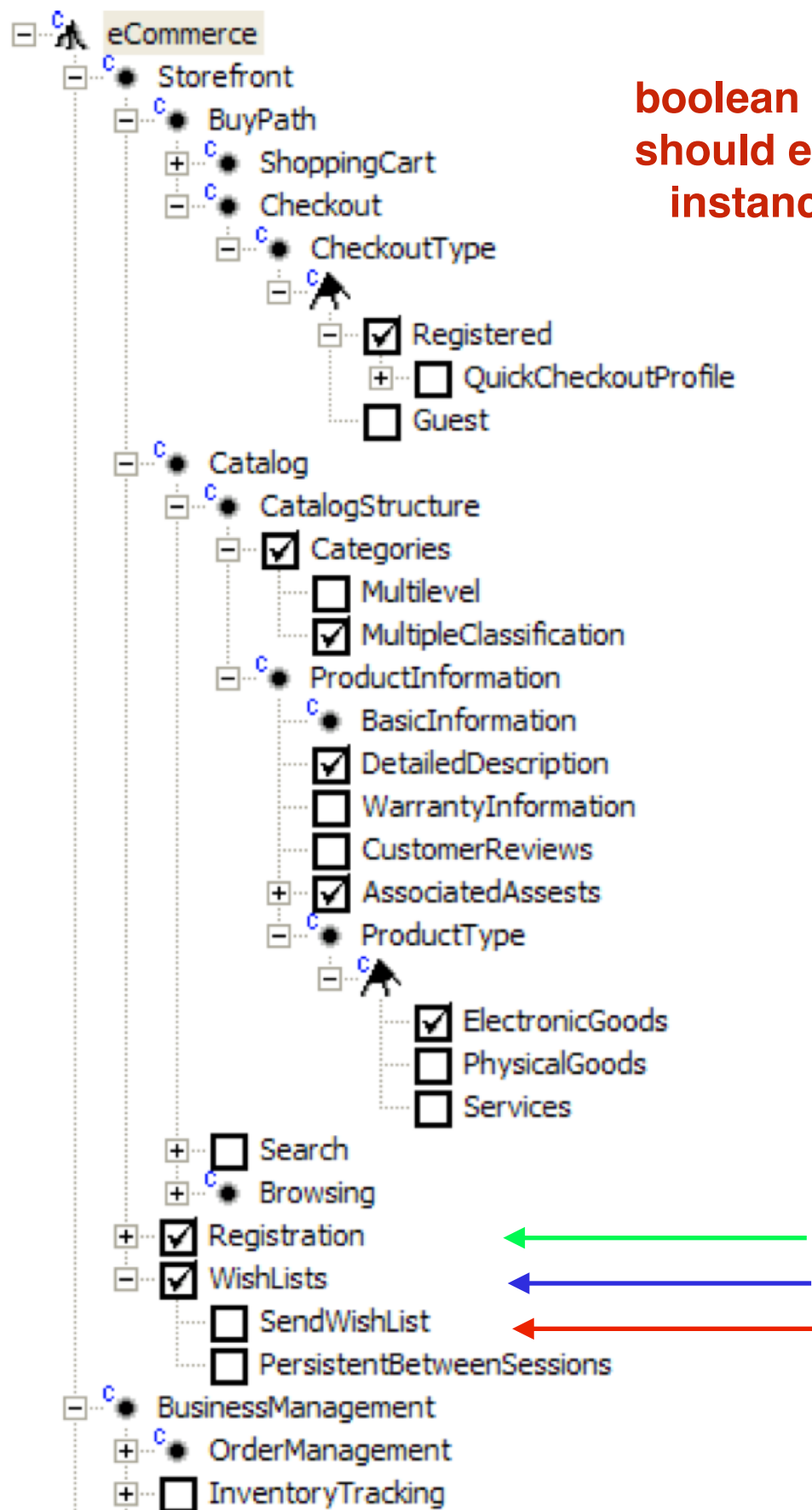
ID	Description / Rationale	minCard	maxCard	Decisions Involved
VP1	An automation device always can be either a supervisory unit or an intelligend field device.	1	1	D1, D2
VP2	There are three different communication standards to configure a field device: standard A, B, and C. At least one must be chosen.	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

Conclusion

- ADORA language allows modeling the structure, behavior and user interaction of a software system in a single hierarchical diagram
- The challenges are solved: the user is aware of the constraints and understands the effects and consequences of binding variability
- Helps keep large models comprehensible for human users

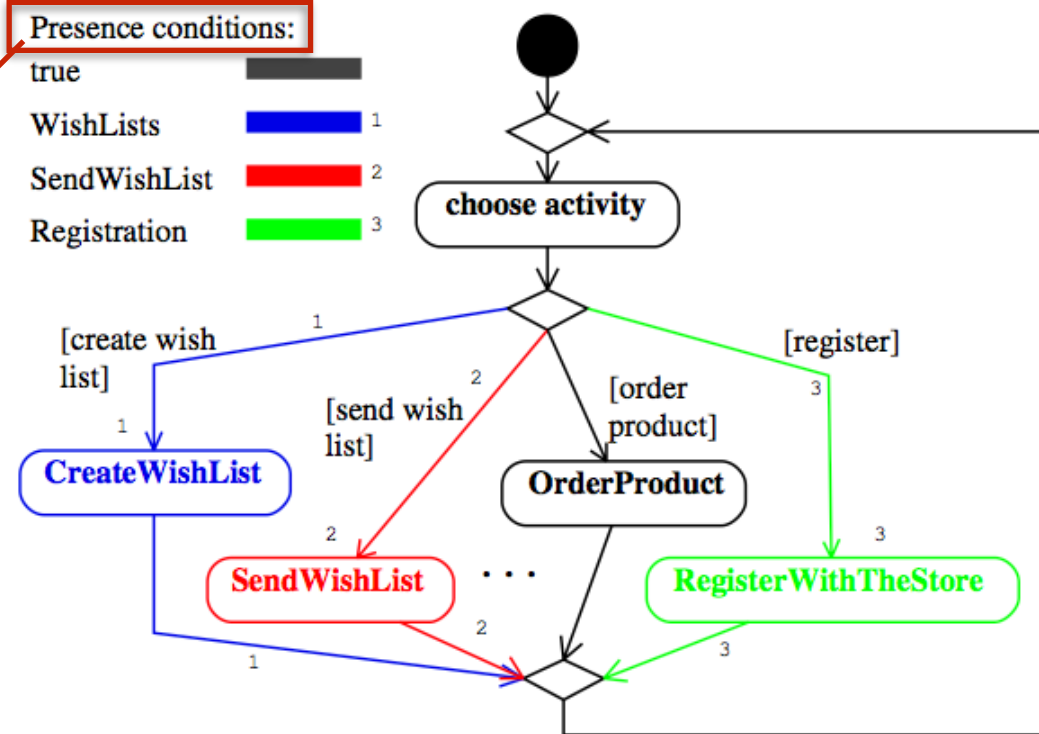
Feature Modeling

- used at any level of abstraction (e.g. requirements, design,...)
 - for any kind of artifact (e.g. code, models,...)
 - **BUT:** features in feature models are merely symbols!
- > propose a general approach for mapping feature models to other kind of models (showing the impact of selecting a given feature on the resulting model)



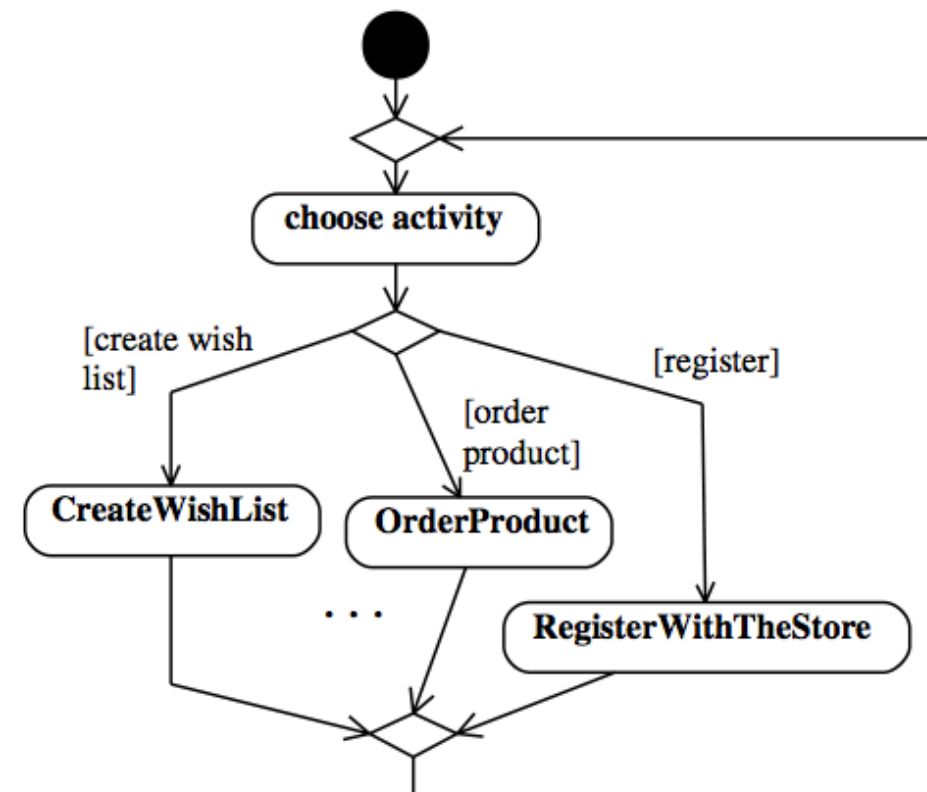
Presence conditions:

true	
WishLists	1
SendWishList	2
Registration	3



(a) Storefront template

union of all model elements

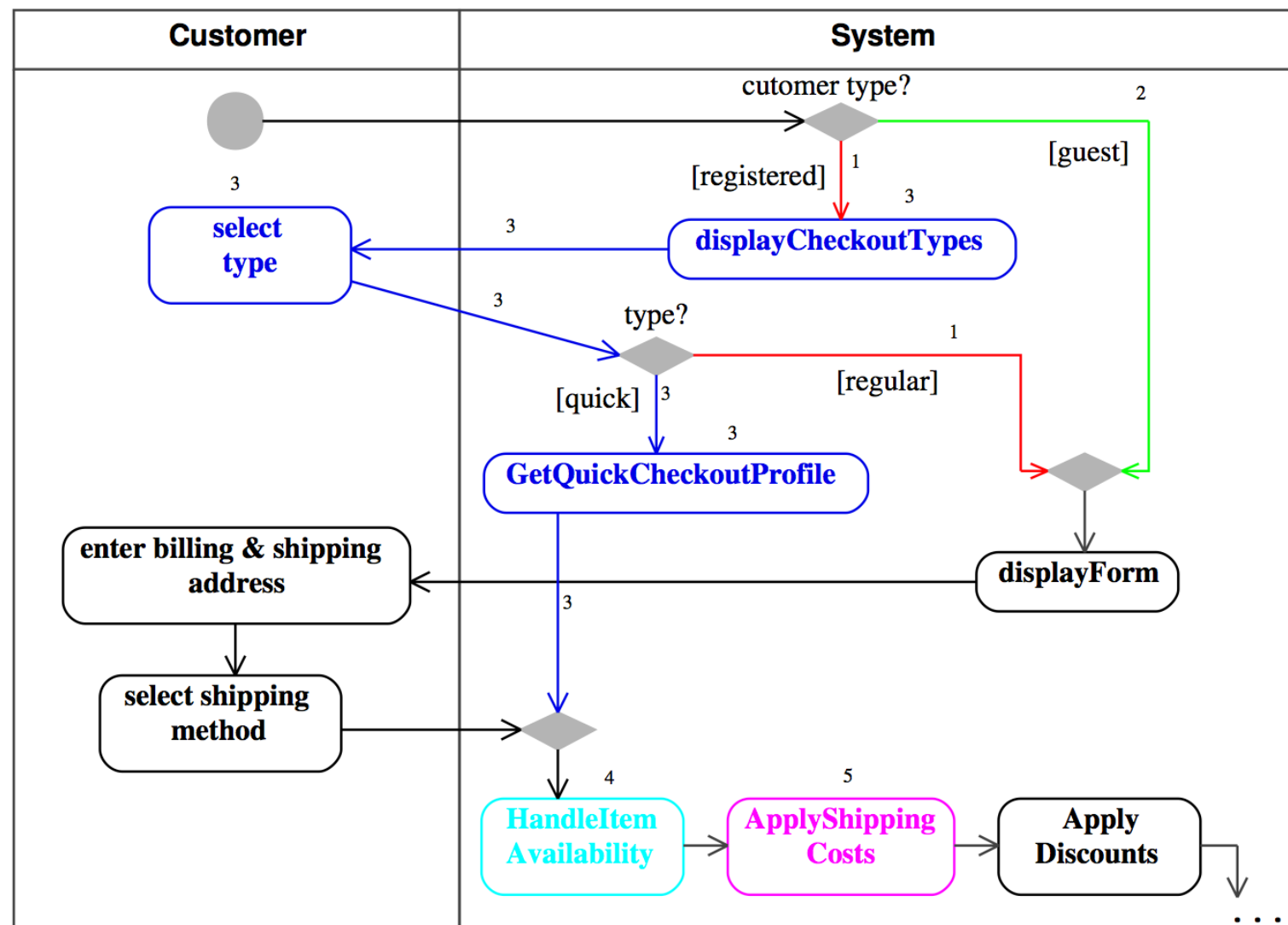


(b) Storefront instance

business-to-customer solutions -
feature configuration of a feature model

Template instantiation algorithm

1. Evaluation of ME's and PC's - check the feature configuration



Presence conditions:

true	
Registered	1
Guest	2
QuickCheckoutProfile	3
InventoryTracking	4
PhysicalGoods	5

- according to the feature configuration the only checkout method is "**registered**".

—> Set Guest, QuickCheckoutProfile, InventoryTracking and Physical Goods to **False**

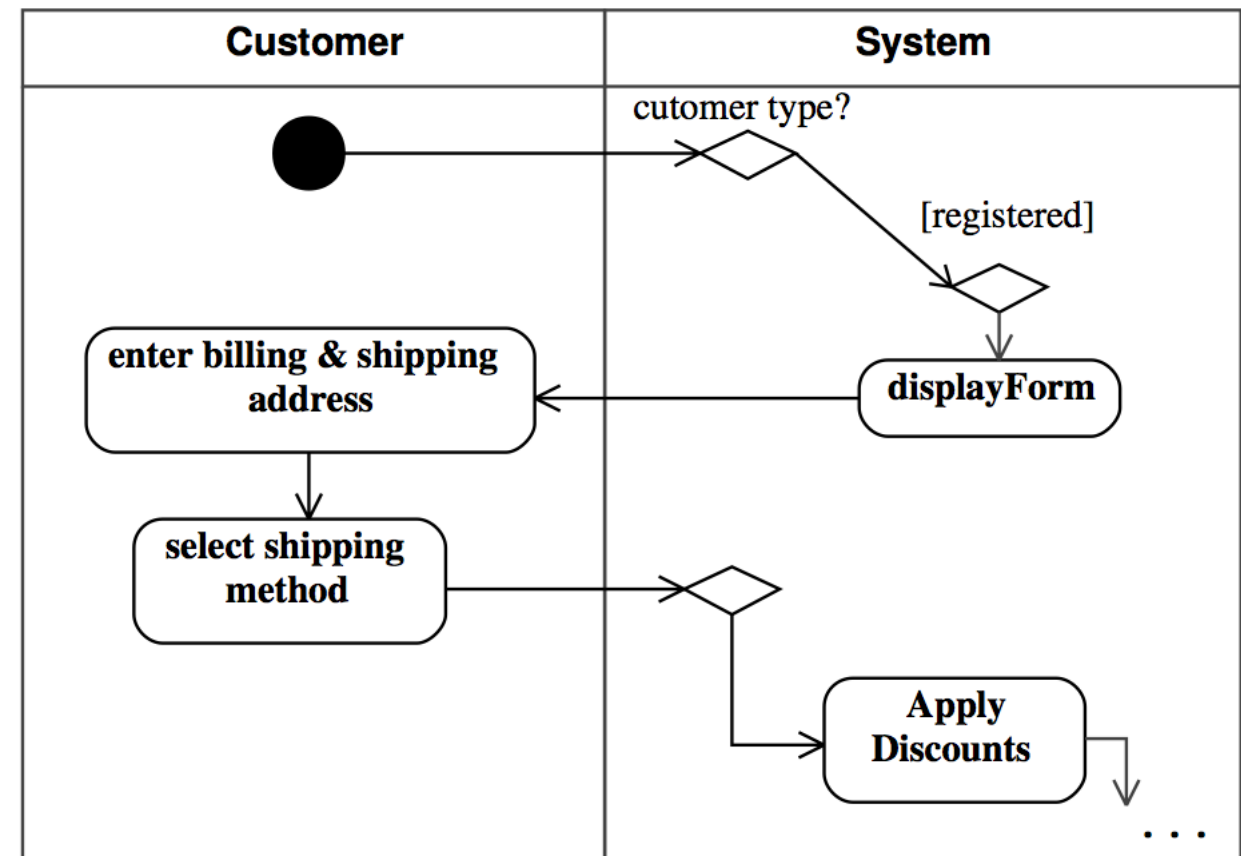
Template instantiation algorithm

2. Removal analysis

- identifies situations where flows interrupted by removed elements can be closed.
- The identifies regions are closable and could be removed.

3. Element removal and patch application

- fixing problems which may result from removing elements

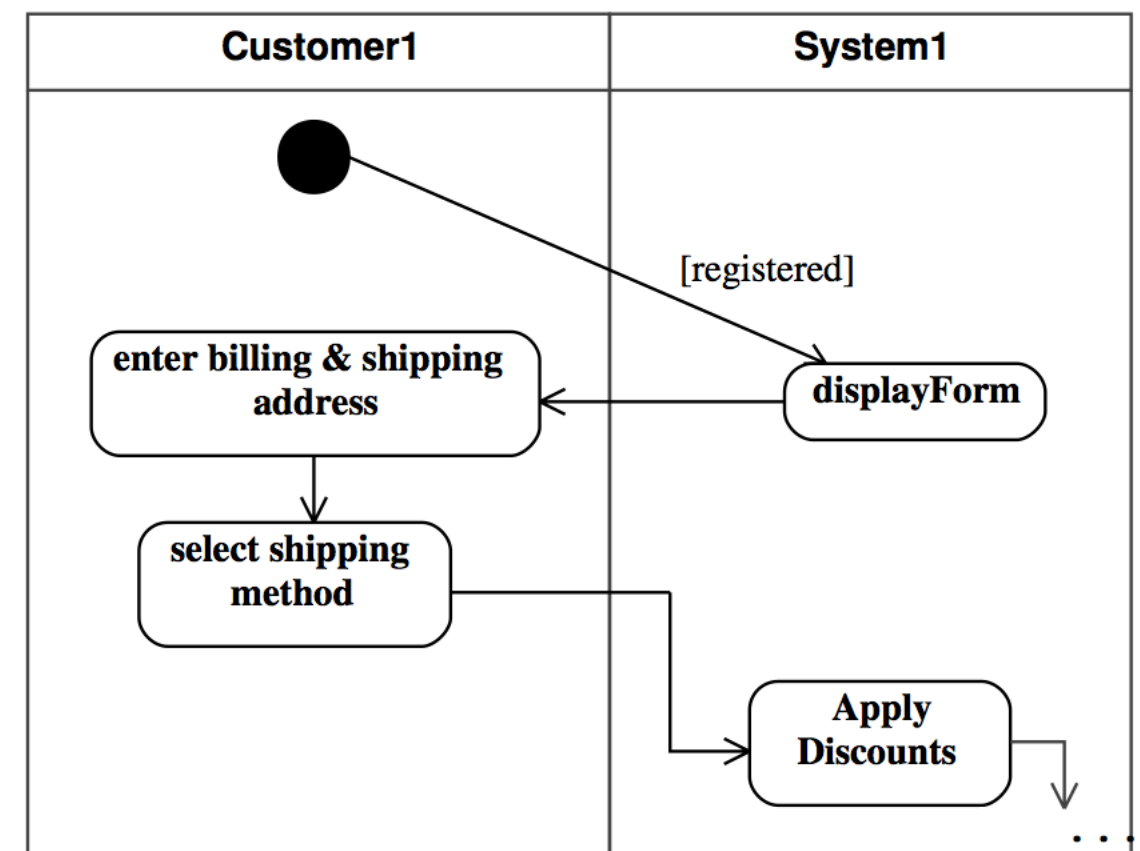


(a) Checkout instance after region removal

Template instantiation algorithm

4. Simplification

- removing elements that have become redundant after removing other elements (e.g. decision nodes, merge nodes,...)



(b) Checkout instance after simplification

Discussion

- Do you think these approaches are applicable in practice?
- Do you think these approaches could scale to large projects?