

---

---

## Requirements Engineering II HS2016 - Assignment 5: Software Product Lines

### Product Derivation with Requirements Models

Fatou Ceesay  
Christopher Wells  
28.11.2016

---

---

# Agenda

- Deriving Req. Models from Product Line Specs.
- Product Derivations Approaches
- Strength and Weakness
- Questions / Discussion



# Req. Models from Product Line Specification

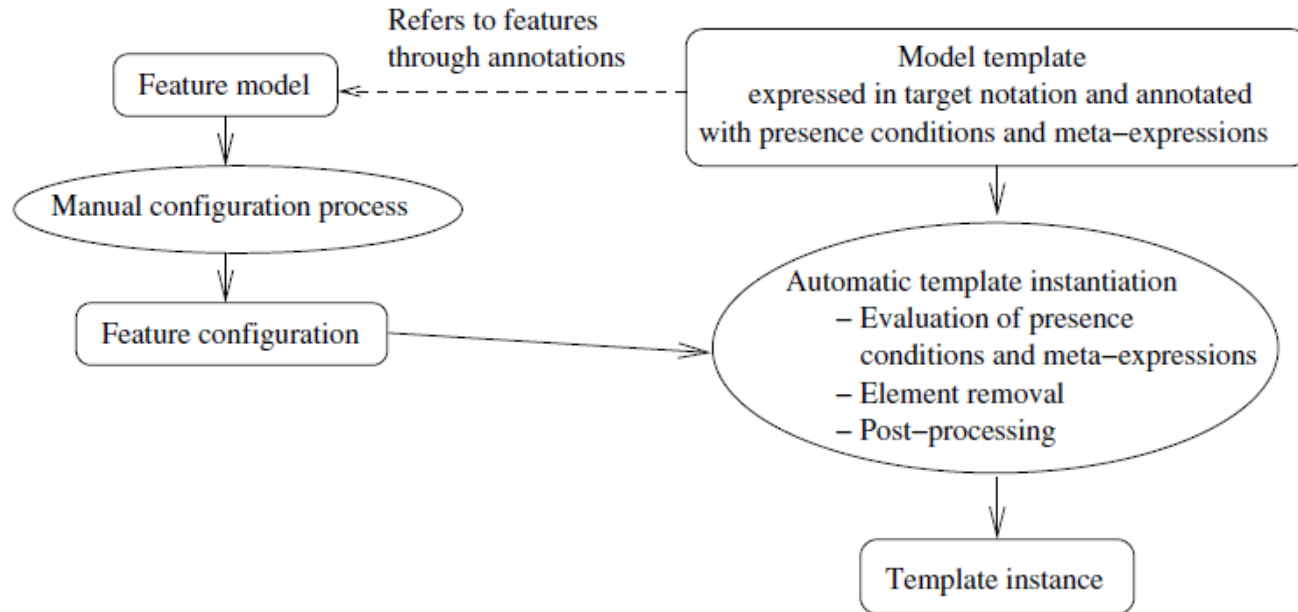
- Requirements Models embed product features
- Features exhibit commonalities and variation of software product lines
- Product line specification depict domain specific language

## → Challenge:

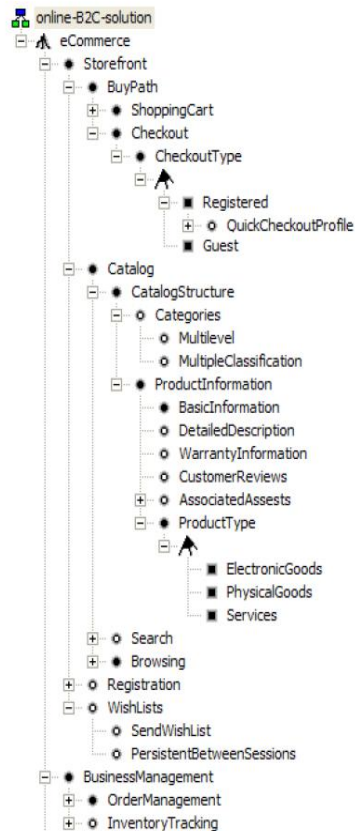
Mapping product derived from a feature model to requirements specification

# Product Derivation Approach - I

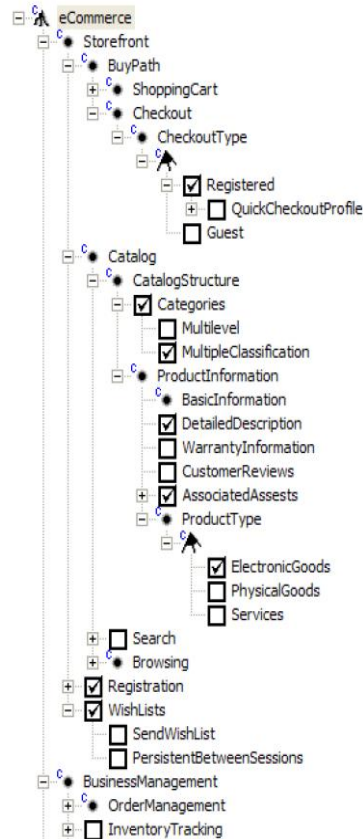
## Mapping Features to Models: UML (Czarnecki & Michal, 2005)



# Feature Configuration



(a) Online B2C Feature Model



(b) Feature configuration

## Legend:

Cardinality [1:1]



Optional [0:1]



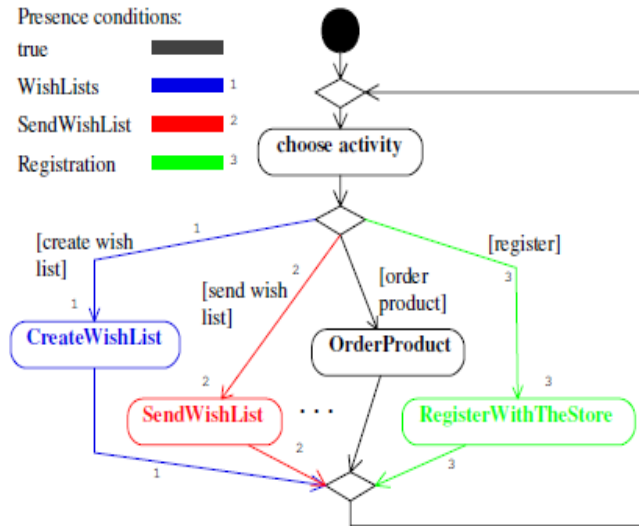
Group [1:k]



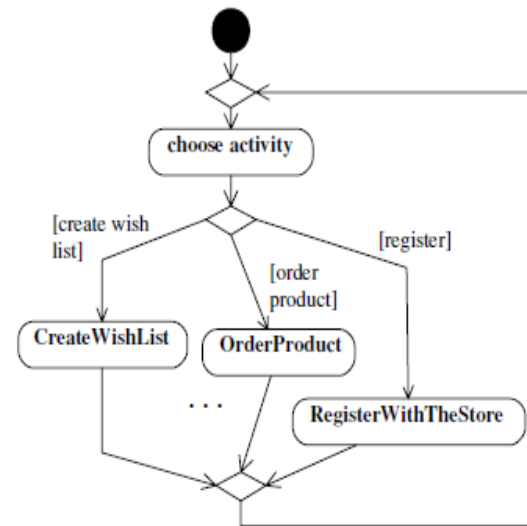
Grouped feature



# Template Instantiation: Activity Diagram



(a) Storefront template

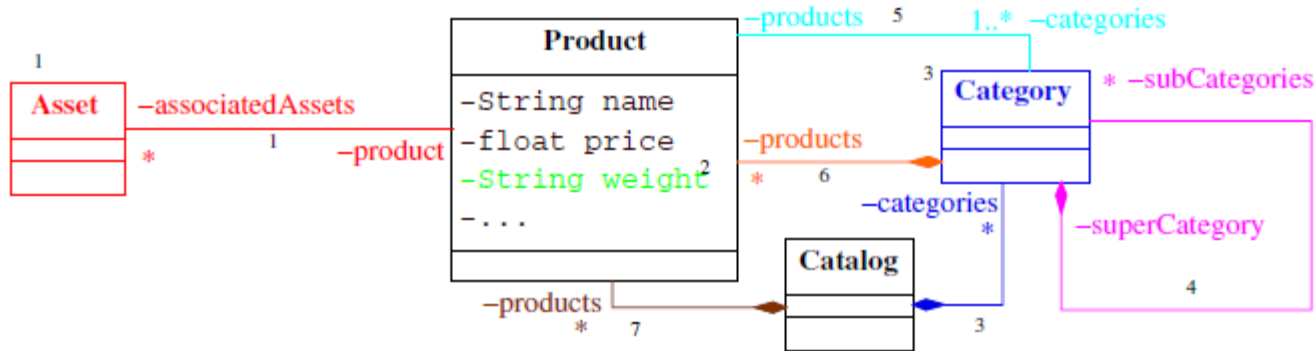


(b) Storefront instance

# Template Instantiation: Class Diagram

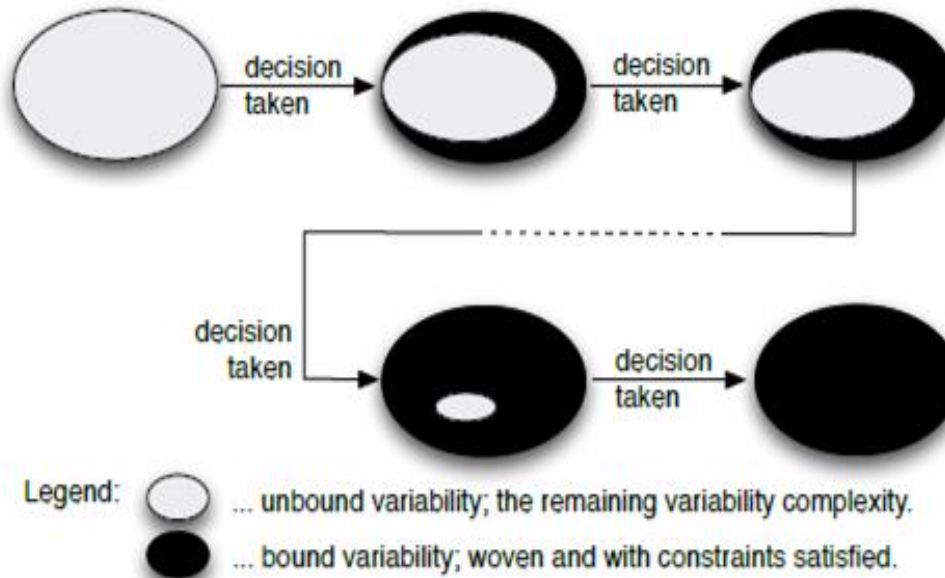
Presence conditions:

|                  |                                                                                   |            |                                                                                     |                                                                                     |   |
|------------------|-----------------------------------------------------------------------------------|------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---|
| true             |  | MultiLevel |  | 4                                                                                   |   |
| AssociatedAssets |  | 1          | MultipleClassification                                                              |  | 5 |
| PhysicalGoods    |  | 2          | Categories & !MultipleClassification                                                |  | 6 |
| Categories       |  | 3          | MultipleClassification   !Categories                                                |  | 7 |

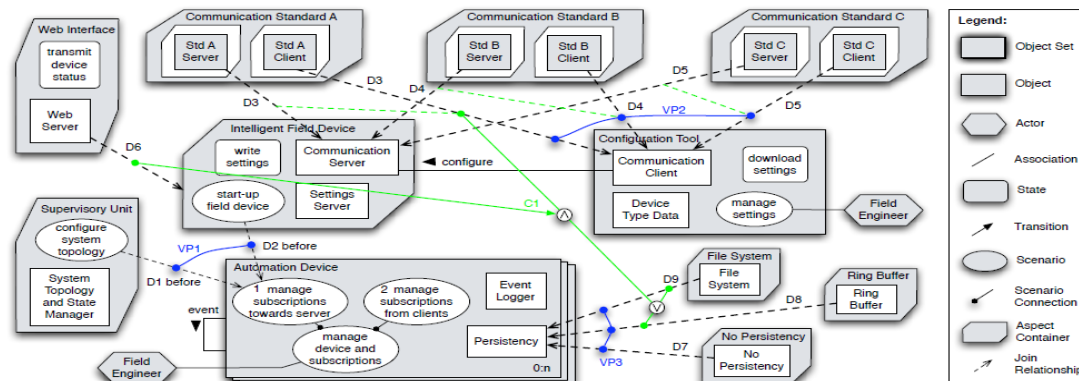


# Product Derivation Approach - II

Stepwise and Incremental: ADORA (Stoiber & Glinz, 2010)



# Stepwise Incremental Product Derivation



| 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 |

| 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         |

| 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.

# Constraint Analysis with ADORA

The approach uses Boolean satisfiability (SAT) of constraints to determine what set of decisions satisfy the model.

- 1) An undecided decision item is decided once true (a), and once false (b)
- 2) Each remaining undecided item (if any) are run through step 1 (once set to true, then to false).
- 3) When all combinations of decision are made in 2, then the SAT solver is run over each combination to see if they meet all constraints.
  - a) If yes, then this is a valid set of decisions
  - b) If no, then the first decision (of the set of decision) is checked as to whether it was true or false. The opposite of that result is applied to the next *decision*, then *it* is entered into the 'ifTrue' or 'ifFalse' column of the decision table for the item.

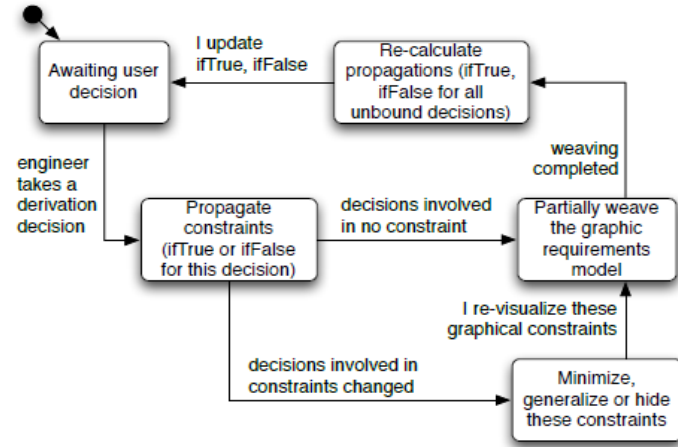


Fig. 3. A statechart describing the behavior of the ADORA tool when executing a variability binding decision taken in a product derivation process.

# Comparison of Approaches

|                   | Incremental Product Derivation                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Mapping Features to Models                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Strengths</b>  | <ul style="list-style-type: none"><li>• Provides single model of decisions, variations (of requirements) and constraints</li><li>• Provides autonomous means to determine satisfiability of decisions to constraints</li></ul>                                                                                                                                                                                                                                                                                                     | <ul style="list-style-type: none"><li>• Model templates can be incrementally created with feature models</li><li>• Expressing PCs and MEs in terms of features provides traceability between features and their realization in models</li></ul>                                                                                                                                                                                                                   |
| <b>Weaknesses</b> | <ul style="list-style-type: none"><li>• Requires domain knowledge of different graphical model entities (i.e. aspect container, object set, etc.) that must be understood by technical and non-technical team members to be effectively used</li><li>• Could require significant amounts of time to setup and utilize system</li><li>• Simple decisions to be made <i>multiplies</i> the number of paths needed for SAT (<math>O(2n)</math>) when not modeling could significantly reduce number of paths needed for SAT</li></ul> | <ul style="list-style-type: none"><li>• Not suitable for product lines with complex constraints as this approach can only define simple variability constraints</li><li>• Use of different diagrams requires context switching which is not ideal</li><li>• Annotation can be tedious, if the models are iterated the annotations will need to be modified</li><li>• Only provides names of available features, no details about concrete functionality</li></ul> |

# Questions / Discussion

- Which of the two approaches would be useful in case of tracing requirements of the models?

The background of the slide features a close-up, slightly blurred image of several interlocking metal gears. The gears are light-colored, possibly brass or steel, and their teeth are clearly visible. The lighting creates soft highlights and shadows, giving a sense of depth and mechanical complexity. The overall tone is warm and professional.

**Thank You!**