

An Approach for Aspect-Oriented Use Case Modelling

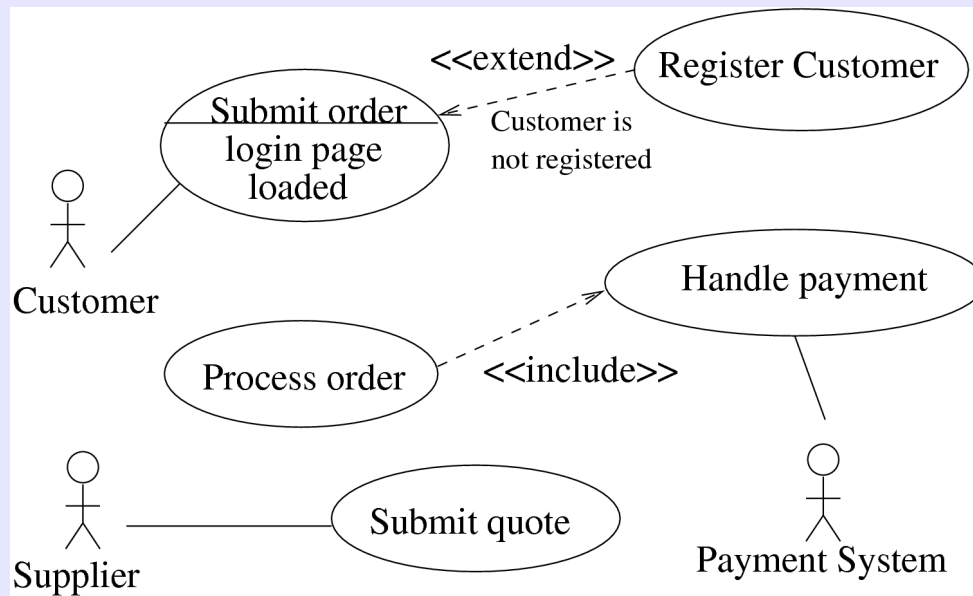
Stéphane S. Somé, Pauline Anthonysamy
SITE, University of Ottawa, Canada

Motivation

- Use Case Model
 - Set of use cases
 - description of a system external behavior
 - interactions between a system and its environment
 - sequential succession of steps
 - Relationships between use cases
 - UML <<include>>, <<extend>>, generalisation
- Missing from use Case Models
 - Effective way for modeling re-occurring concerns
 - Ability to integrate non-functional requirements

Motivation

- Example



Cross-Requirements

(1) The system shall log all transactions for auditing

(2) The system shall ensure idle sessions are closed

Motivation

Title: Submit order

Primary Actor: Customer

Goal: Allows a Customer to place an online order to be sent to a supplier.

Precondition: The OrderProcessing System is online

Postcondition: An Order has been sent

STEPS

1. The Customer loads the login page

2. The OrderProcessing System asks for the Customer's login information

3. The Customer enters a login information

4. The OrderProcessing System checks the provided login information

5. The OrderProcessing System asks the Customer's to create an Order

6. The Customer creates a new Order

7.....

(1) The system shall log all transactions for auditing

Use Case model is a specification

- needs to be understood
- needs to be managed and evolved

Motivation

- Limitations of UML relationships
 - <<*include*>> denotes explicit call to included behavior
 - <<*extend*>> introduces strong dependencies between the extending and extended use case
 - need for maintaining ***extension points***
 - *generalization* inapplicable when there is no clear abstract/concrete relation

Requirements for expressing crosscutting concerns in use case models

- **Transparency** in identification of use cases and locations where crosscutting concerns apply
 - based on high-level goals
- **Flexibility** in maintaining and evolving use case models
 - concerns independence, modularity
- **Executability** for early validation
 - ability to observe/animate composite model

Aspect-oriented approach for modeling concerns

- $\ll aspect \gg$ relation linking concerns (advice use cases) to affected base use cases
 - Concepts inspired from *AspectJ*
- Formally: $(aduc, acond, apcuts, baseUcs)$
 - *aduc* - advice use case
 - *acond* – relation condition
 - *apcuts* – identification of points of application of concerns
 - *baseUcs* – set of base use cases affected

Aspect-oriented approach for modeling concerns

Cond: logs activated
Pcut: after operation Customer *



Title: Submit order
STEPS

- 1.The Customer loads the login page
- 2.The OrderProcessing System asks for the Customer's login information
- 3.The Customer enters a login information
- 4.The OrderProcessing System checks the provided login information
- 5.The OrderProcessing System asks the Customer's
to create an Order
- 6.The Customer creates a new Order
- 7.....

Title: Create log

Goal: To allow for audits by logging entered data
STEPS

- B1.The OrderProcessing System initiates a log with the Customer entered data
- B2.The OrderProcessing System records the log on an external file system

ALTERNATIVES

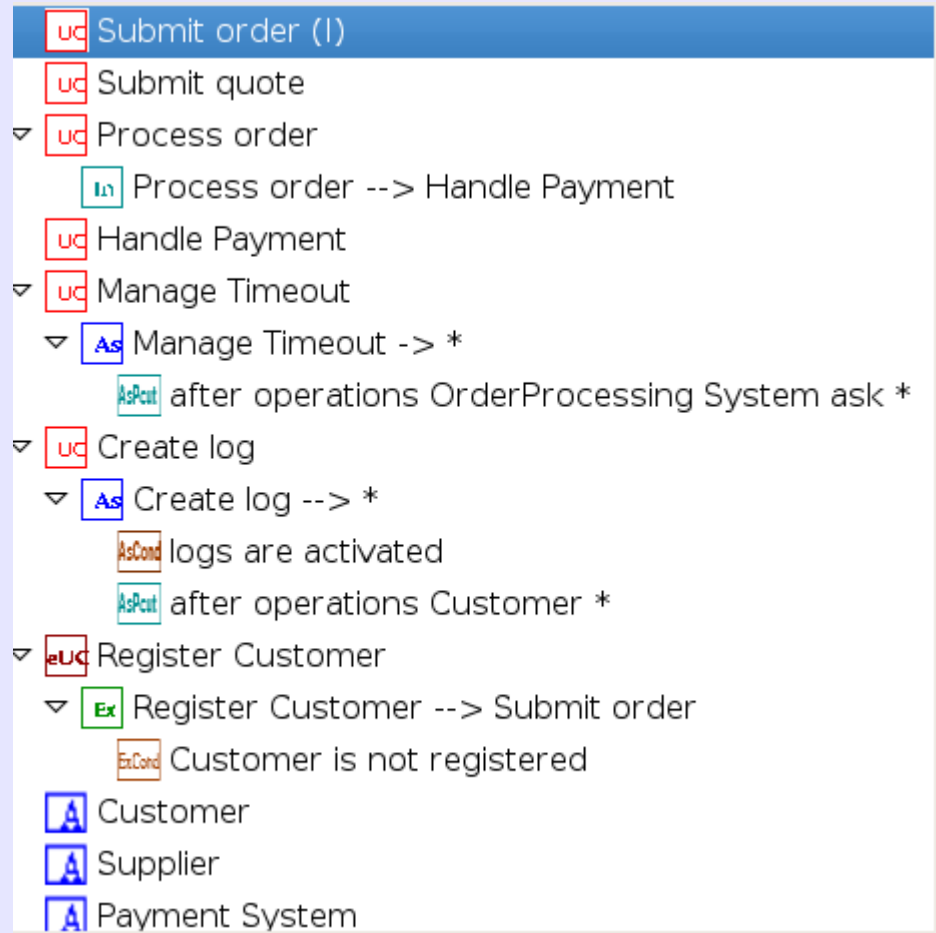
- B1.a.There is no more space left on the device
- B1.a.1.The OrderProcessing System aborts and displays an error page

Aspect-oriented approach for modeling concerns

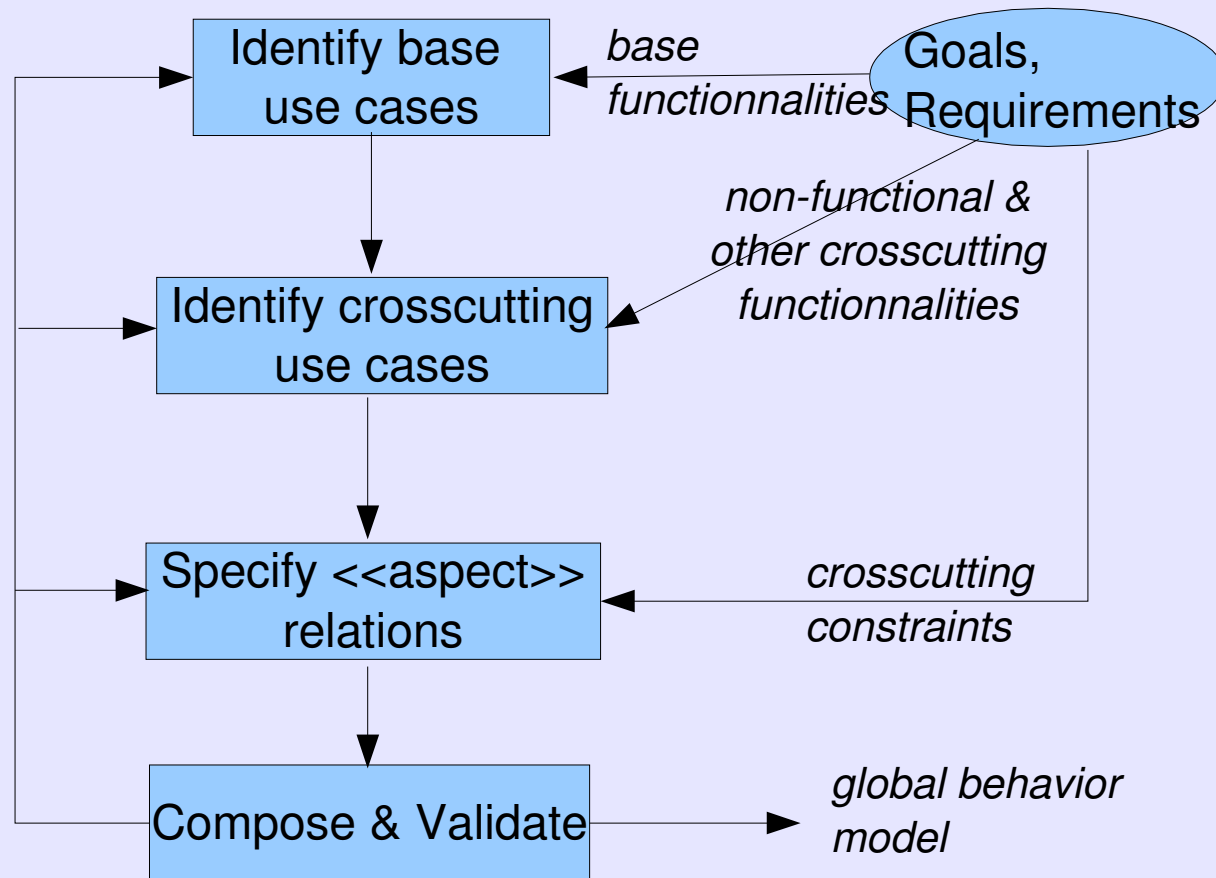
- Related use cases identified by pattern matching
 - “* order *”
 - “Primary Actor Customer”
- Definition of pointcuts relative to any use case description elements
- Identified from use case description meta-model
 - domain operations
 - steps
 - alternatives
 - conditions
 - ...

Aspect-oriented approach for modeling concerns

- Implementation in *UCEd*
- *Generation of state model from use cases*
 - Petri net model
 - Weaving of crosscutting use cases with base use cases
- Detection of interactions
- Animation



Aspect-oriented approach for modeling concerns



Conclusions

- Adaptation of “classical” aspect approach to use case description
 - still retaining use case modeling concepts (e.g. relationships)
- Approach implemented in a tool
- Validation with various case studies
 - Arcade Game Maker
 - Health Watcher
 - ...