

# Specifying Product Line Variability in Requirements Models

Marc-Alain Chételat  
Emanuel Oehri

# Agenda

- What is product line variability?
- Why is there a need to specify product line variability in requirements specifications?
- Two approaches to extending existing modeling languages (e.g. UML) to support variability modeling
  - Flexible Variant Configuration Tool (Jarzabek et al., 2013)
  - Feature Unweaving (Stoiber & Glinz, 2010)
- Conclusion
- Discussion

# What is product line variability?

- A product line comprises a group of similar systems that together correspond to a particular product segment
- The emphasis is put on product flexibility
- Fulfill the promise of tailor-made systems built specifically for the needs of particular customers
- Commonalities shared by the product can be exploited to achieve economies of production
- Product lines enhance quality: Each new system takes advantage of all the defect elimination of previous system development
- Costs for development and maintenance decrease due to economies of scale

# Why is there a need to specify product line variability in requirements specification?

- One variant is often reflected in many different views of a domain
- One model is often affected by many variants
- Tracing multiple occurrences of the same variant is a major challenge
- The dependencies among the variants are difficult to identify

→ Domain models become difficult to understand undermining the major role of domain analysis which is understanding

→ UML models of a single system may be too complicated

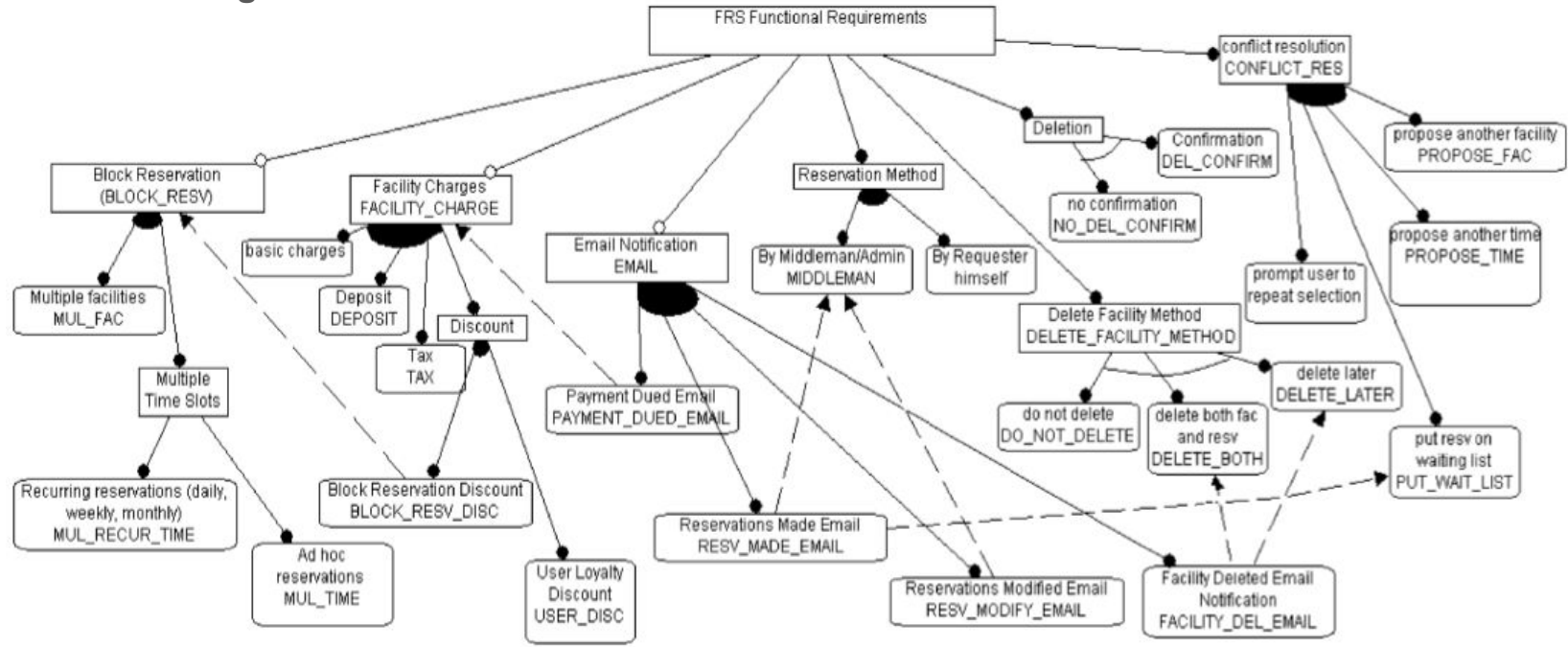
→ With variants, models soon become unmanageable

# Flexible Variant Configuration Tool (Jarzabek et al., 2003)

- Approach:
  - Modeling requirements of a typical system in a domain in UML
    - **Domain Defaults**
  - Description of variants as deltas in respect to domain defaults
    - **Feature Diagram**
    - **f-defaults**
  - FCV interprets variants to produce customized domain model views that meets specific requirements
    - Synthesizes customization scripts and f-defaults
    - Performs customization by calling a **frame processor**
    - Outputs **customized domain model views**
- Next: Illustration of approach with a facility reservation system family

# Illustration: Facility Reservation System Family (1/4)

- **Domain Defaults:** Make / Delete / Modify a Reservation, Search / Retrieve Reservations, Add / Delete / Modify a Facility, Search / Retrieve Facilities
- **Feature Diagram:**



# Illustration: Facility Reservation System Family (2/4)

- Frame commands:

```
.COPY f-default
.END-COPY f-default

.INSERT name
.INSERT-BEFORE name

.INSERT-AFTER name

.BREAK name

.SELECT variable
.WHEN value

.ORWHEN value

.OTHERWISE
.END-SELECT variable

.REPLACE variable BY value
```

- f-default use case *Make Reservation*

F-DEFAULT Make\_Reservation\_uc // suffix 'uc' indicates that the contents of this f-default is a use case

.REPLACE FACILITY BY "meeting room"

.REPLACE RESPONSETIME BY "30 secs"

1. Introduction

Allows a user to select FACILITY and make new reservation.

2. Flow of events

2.1 The user is identified.

.BREAK MIDDLEMAN //a breakpoint at which variant "reservation by middlemen" can be addressed

2.2 The system shows existing "FACILITY"s.

2.3 The user selects a "FACILITY" and supplies the time period of reservation.

.BREAK BLOCK\_RESV\_FAC //a breakpoint at which variant "select multiple facilities" can be addressed

2.4 If the "FACILITY" is reserved at the specified time, a warning message is prompted and user repeats step 2.2. Otherwise, the system confirms the new reservation.

3. Special Requirement

System should respond to user input within "RESPONSETIME".

# Illustration: Facility Reservation System Family (3/4)

- Frame commands:
- A partial customization script for the MIDDLEMAN variant

```
.COPY f-default
.END-COPY f-default

.INSERT name
.INSERT-BEFORE name

.INSERT-AFTER name

.BREAK name

.SELECT variable
.WHEN value

.OTHERWISE
.END-SELECT variable

.REPLACE variable BY value
```

```
CS MIDDLEMAN // Customization Script for MIDDLEMAN variant
```

```
// customize Make_Reservation_uc
```

```
.COPY Make_Reservation_uc
```

```
.INSERT-AFTER MIDDLEMAN
```

If the user is a middleman, then enter the requester's information.

```
.END-COPY Make_Reservation_uc
```

```
...
```

```
// customize Retrieve_Reservation_uc
```

```
.COPY Retrieve_Reservation_uc
```

```
.INSERT-AFTER MIDDLEMAN
```

If the user is a middle man, then when searching/retrieving reservations by requester, the user enters userid of the requester on behalf of the requester

```
.END-COPY Retrieve_Reservation_uc
```

# Illustration: Facility Reservation System Family (4/4)

- Customized *Make Reservation* use case:

Make Reservation use case with MIDDLEMAN variant:

1. Introduction

Allows a user to select **meeting room** and make new reservation.

2. Flow of events

2.1 The user is identified.

**If the user is a middleman, then enter the requester's information.**

2.2 The system shows existing **meeting rooms**.

2.3 The user selects a **meeting room** and supplies the time period of reservation.

2.4 If the **meeting room** is reserved at the specified time, a warning message is prompted, and user repeats

step 2.2. Otherwise, the system confirms the new reservation.

3. Special Requirement

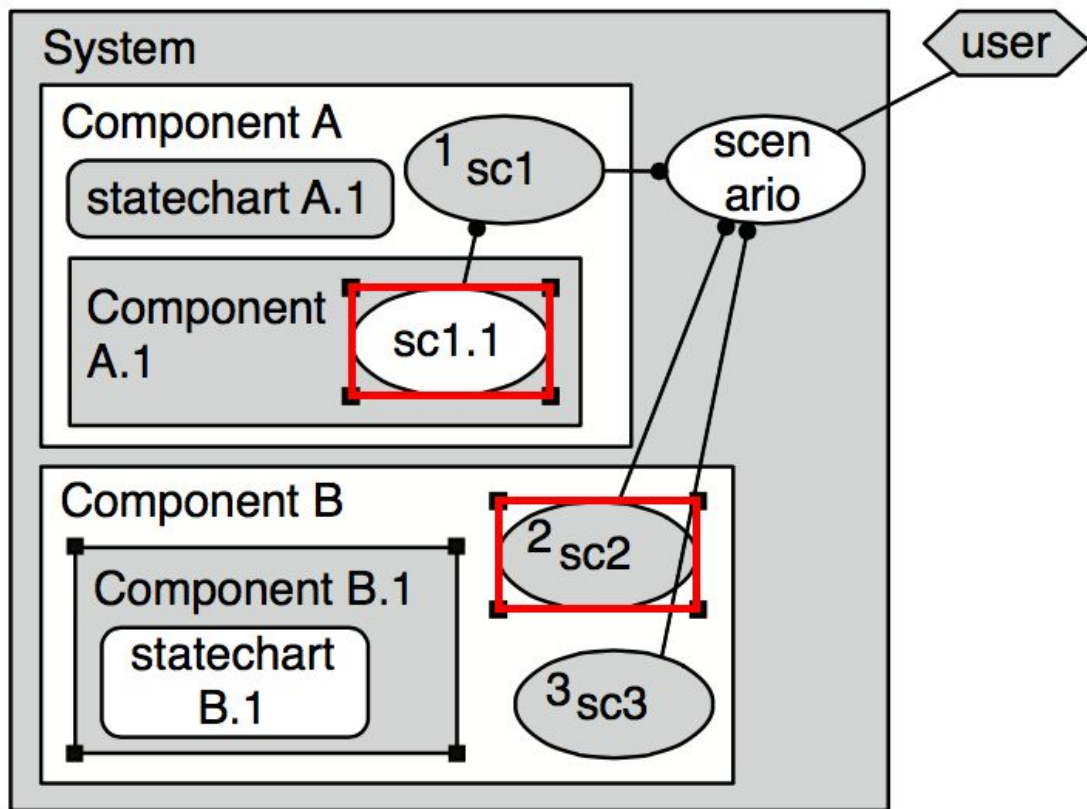
System should respond to user input within **30secs**.

## Note:

Approach shown for *use cases*. Other UML notation also possible, e.g. similar approach for activity diagrams when converting diagram into equivalent textual representation (with the help of an XMI tool).

# Feature Unweaving (Stoiber & Glinz, 2010)

- Approach (based on aspect-oriented modeling):
  - Manual identification of the elements of the model that vary
  - Automatic extraction and refactoring of these elements into features (feature unweaving function)
  - Built on the ADORA model
- Next: Illustration of the approach



# Feature Unweaving Algorithm

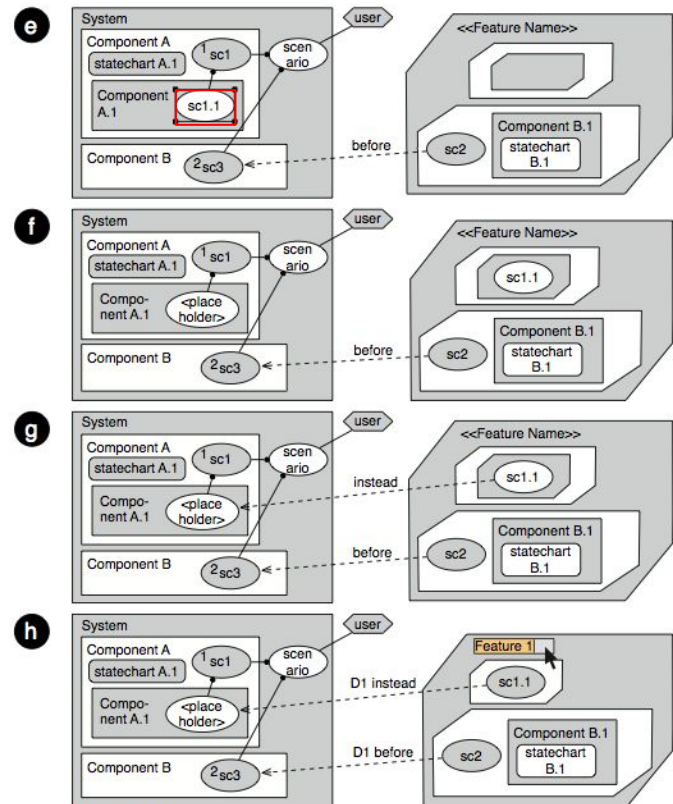
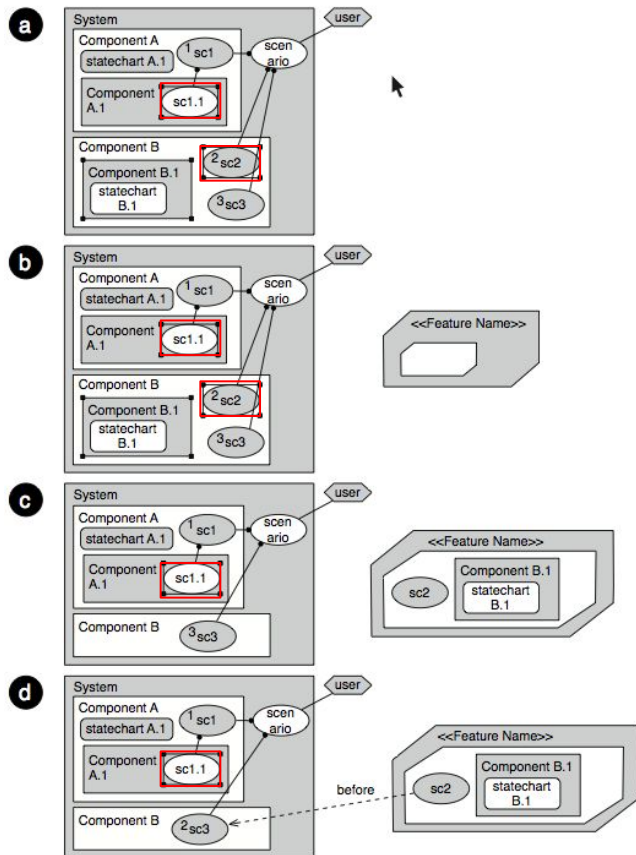
- The feature unweaving algorithm
- The recursive extraction function

1. *Create the BaseAC*
2. *Call the recursive extraction function (parameters: the ADORA model, the BaseAC)*
3. *Remove all redundant ACs in the BaseAC*
4. *Fine-tune the graphic layout*
5. *Set the BaseAC as a variability (this adds a unique decision item and annotates the outgoing join relationships)*

*Input-parameters: a PC, an AC*

- 2.1 *Find the BasePC*
- 2.2 *Extract all directly nested selected elements of the BasePC into this AC*
- 2.3 *Calculate and set the join relationship(s)*
- 2.4 *For every nested PC that contains any selected element*
  - 2.4.1 *Create a new AC inside this AC*
  - 2.4.2 *Call the recursive extraction function (parameters: this BasePC, the new AC)*

# Feature Unweaving Illustrated



Decision Table

| ID | Description | Rationale | Constraints | FeatureName | Decision  |
|----|-------------|-----------|-------------|-------------|-----------|
| D1 | TBD         | TBD       | -           | Feature 1   | undecided |

# An Overview

- Feature Unweaving is...
  - ... a structured, semi-automatic approach to modularize selected variable elements as a feature...
  - ... using aspect-oriented modeling
  - Compared to ad hoc spreadsheets (AND/OR tables)
    - Constraints between variable features are specified explicitly
    - For every feature, traceability links to the corresponding requirements are defined
- Benefits of Feature Unweaving are...
  - Effort and cost reduction for creating variable features
  - Guarantee of semantical equivalence of the original and the new, refactored model

# Conclusion

- Product line variability in requirements specification has major advantages:
  - Development costs are reduced
  - Products can be developed faster
  - Software quality will increase because of reuse
- In order to keep requirements specification manageable an approach other than usual domain specification is needed

Questions?

# Discussion

Questions:

- What are the strength and weaknesses of each of the two approaches?  
(Flexible Variant Configuration Tool & Feature Unweaving)
- Are the tools easy to understand? How could you improve ease of use?