

REQUIREMENTS ENGINEERING



Tanvi Singh

Ilya Ryzhov

10.12.2012



Handling variant requirements in domain modeling

Problems of classical modelling:

- Suitable only for small examples (doesn't scale up);
- Complexity grows exponentially with the growth of model.

Solution:

- Domain defaults modelling;
- Flexible Variant Configuration tool (FVC).



Facility Reservation System (FRS)

- Functionalities covered by FRS defaults:

Make a Reservation;

Delete a Reservation;

Modify a Reservation;

Search/Retrieve Reservations;

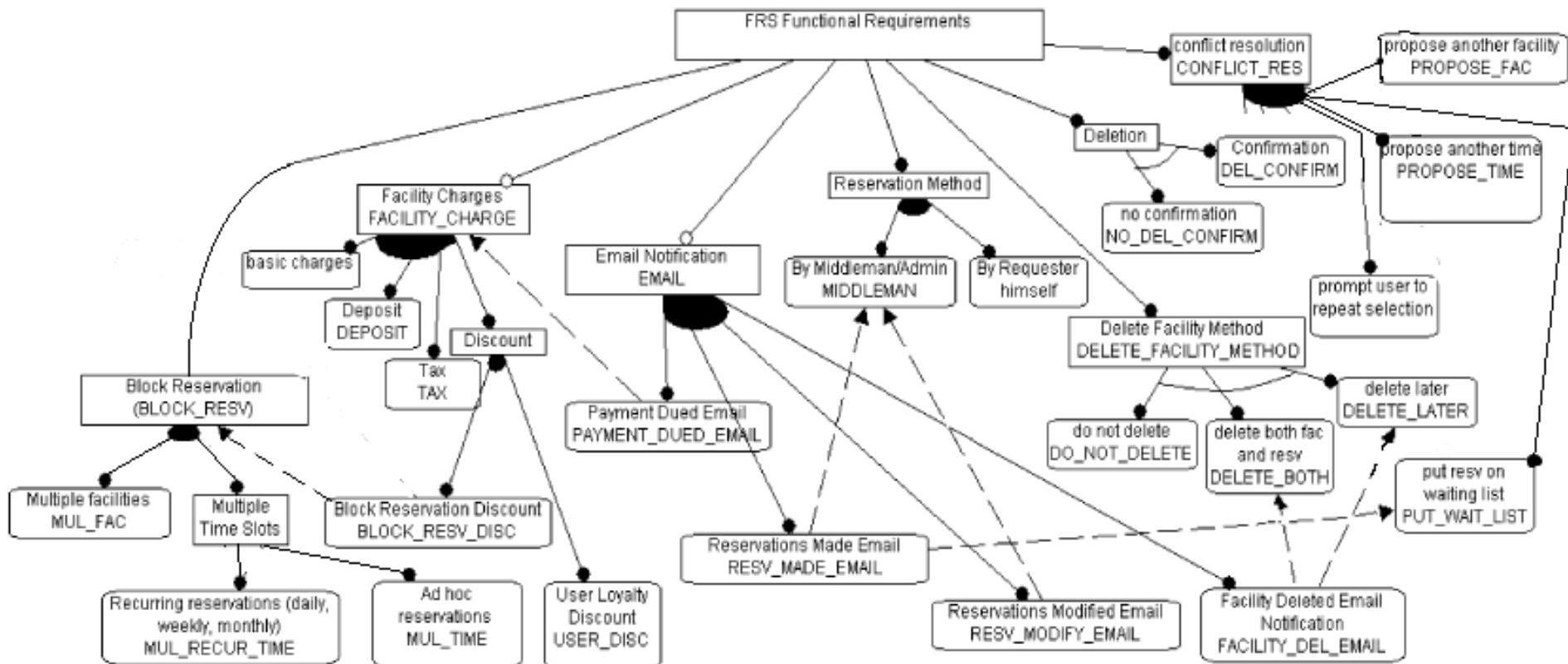
Add a Facility;

Delete a Facility;

Modify a Facility;

Search/Retrieve Facilities.

FRS feature diagram with dependencies



Default use case “Make Reservation”

DEFAULT Make_Reservation_uc // *suffix ‘uc’ indicates that the contents of this-default is a use case*

1. Introduction

Make Reservation allows a user to select “FACILITY” and make new reservation.

2. Flow of events

2.1 The user is identified.

2.2 The system shows existing “FACILITY”s.

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

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".

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".

A partial customization script for the MIDDLEMAN variant

```
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
```

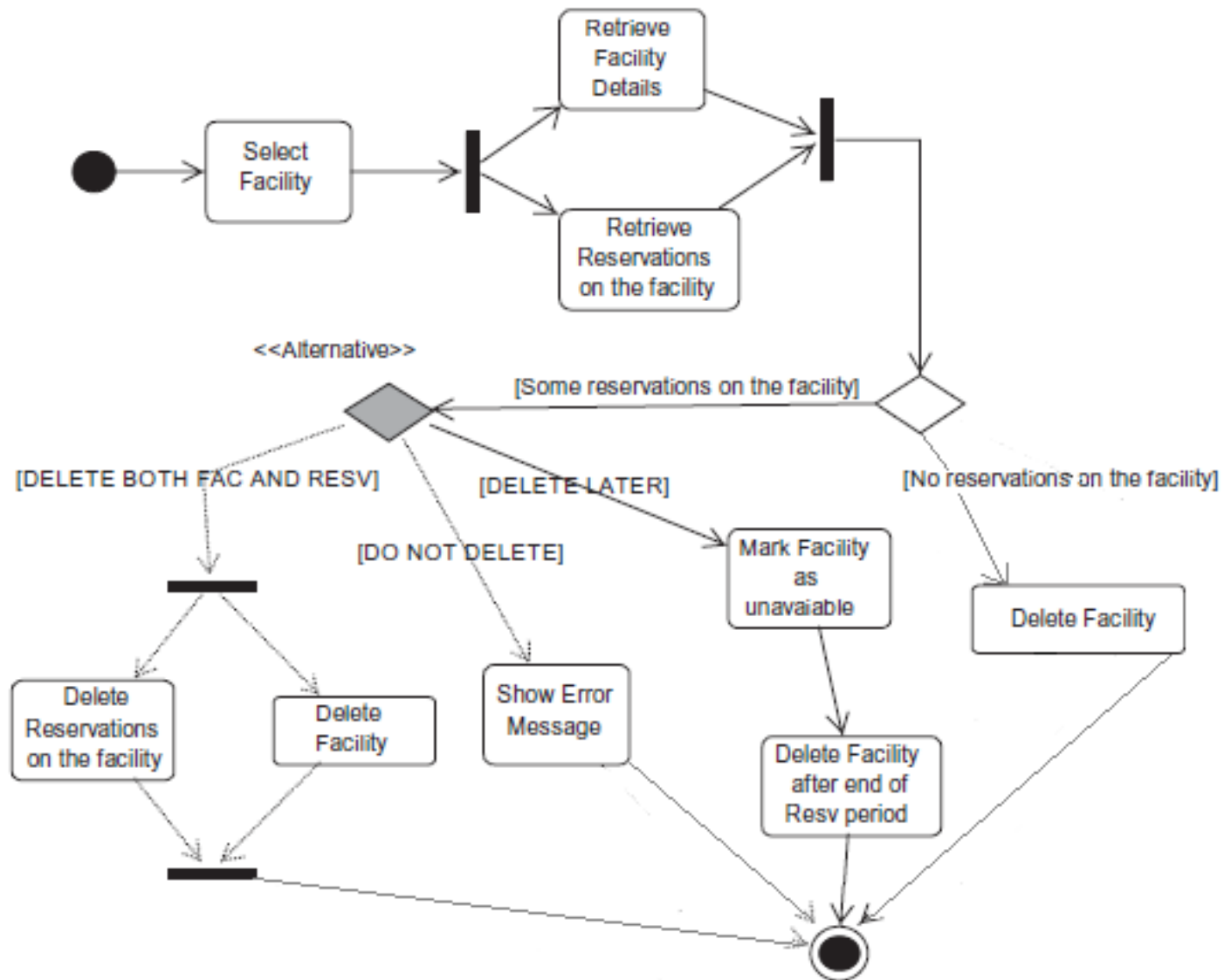
```
.COPY Retrieve_Reservation_uc
```

```
.INSERT-AFTER MIDDLEMAN
```

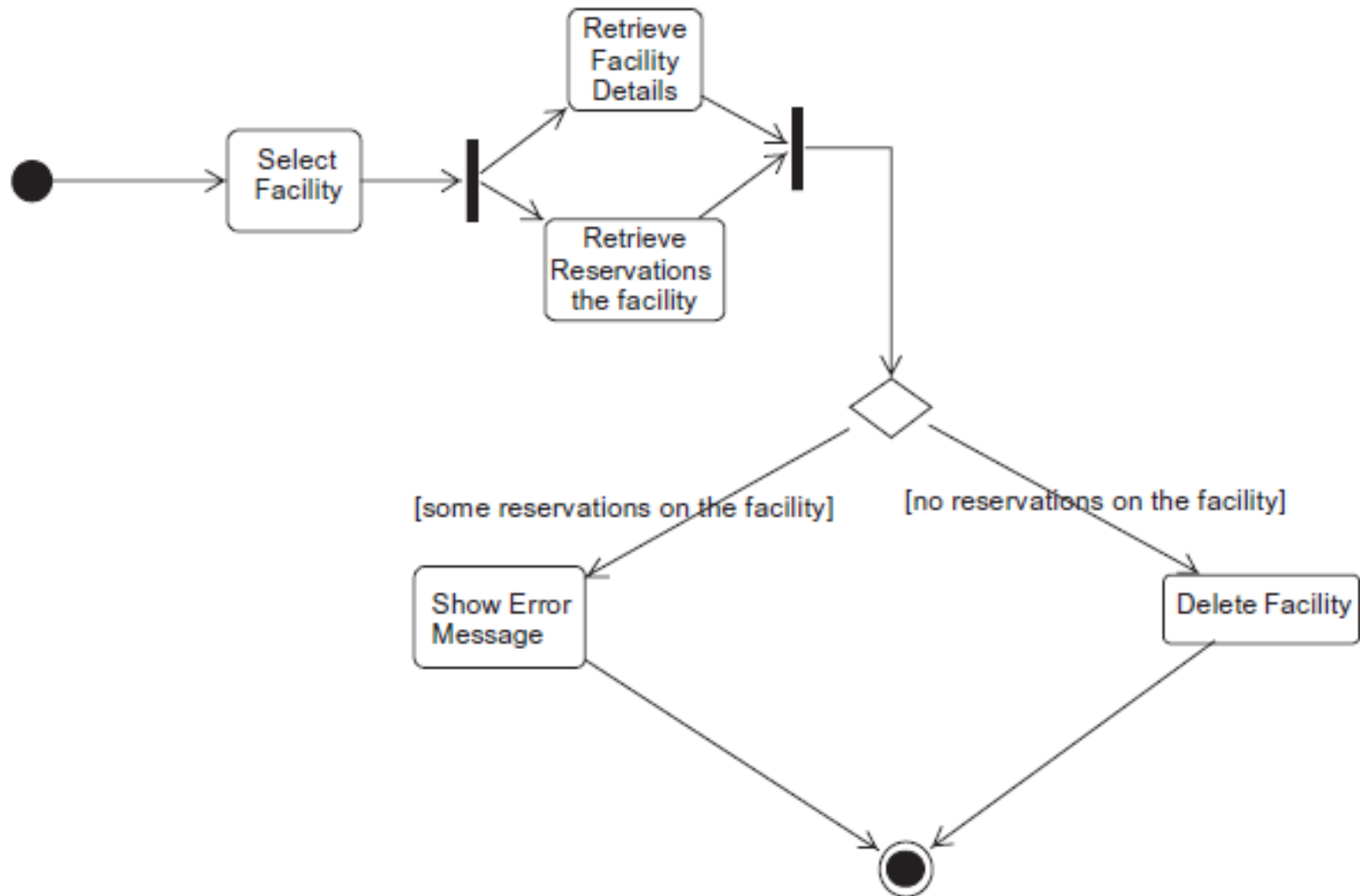
If the user is a middle man, then when searching/retrieving reservations by requester, the user enters userid of the requester on behalf of the requester

```
.END-COPY Retrieve_Reservation_uc
```

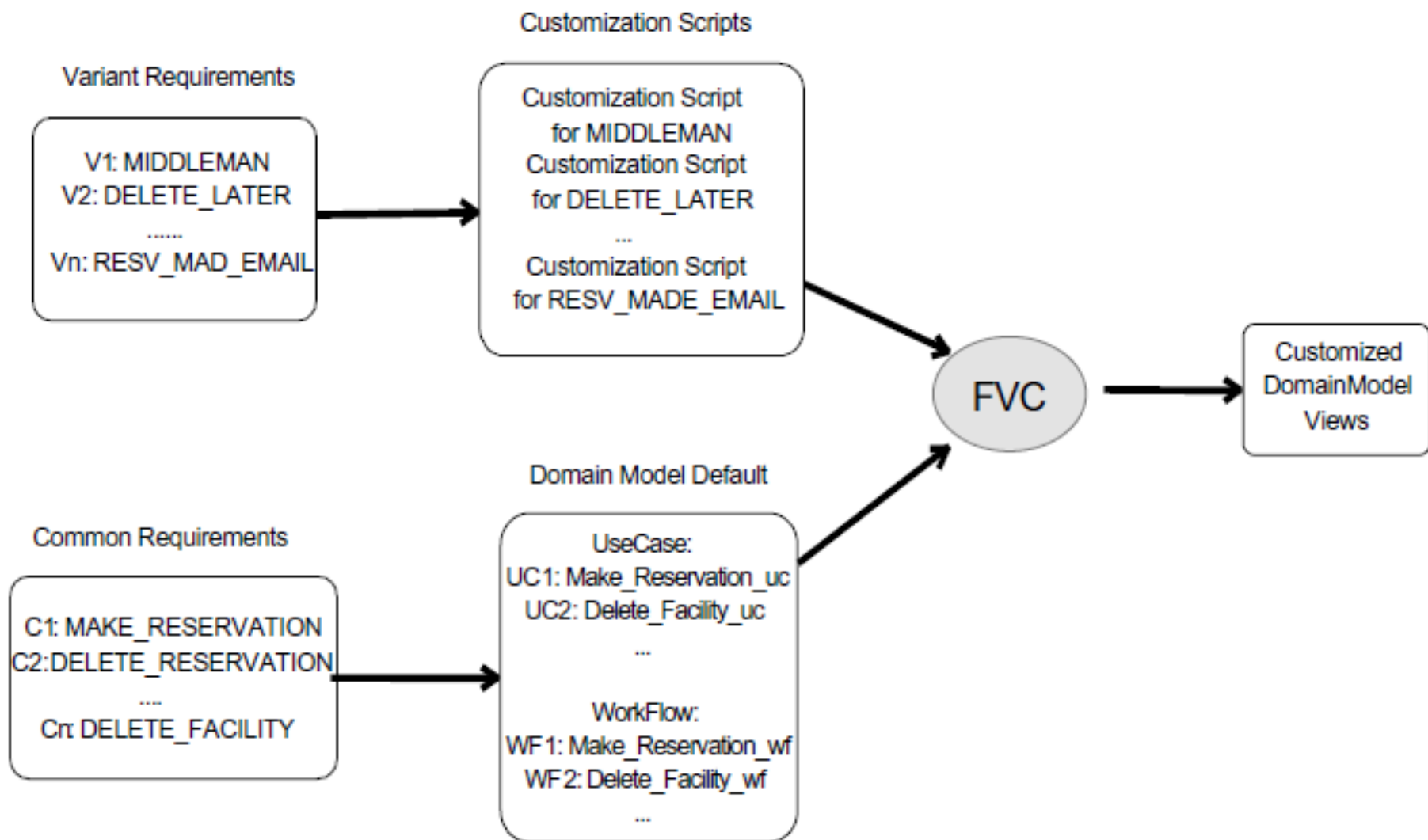
Delete Facility workflow with variation point



A customized *Delete Facility* workflow



An overview of the customization process





Benefits of domain modelling approach and FVC tool

- Reduction of complexity of a domain modeling;
- Reduction of the amount of errors;
- Better support for customization;
- Ease of evolution;
- Exclusion of hand work;
- Time economy.



Feature Unweaving :: Efficient Variability Extraction and Specification for Emerging Software Product Lines.

BY :Reinhard Stoiber, Martin Glinz

What is “FEATURE UNWEAVING”?

- Framework of the ADORA language and tool.
- Aspect oriented modeling.
- Extraction and Refracting- fully automated.
- No manual effort for manual consistency checking.

Traditional Variability modeling solutions:

- Spreadsheets
- AND/OR Tables
- Feature modeling with UML

THE FEATURE UNWEAVING FUNCTION- ***TERMS***

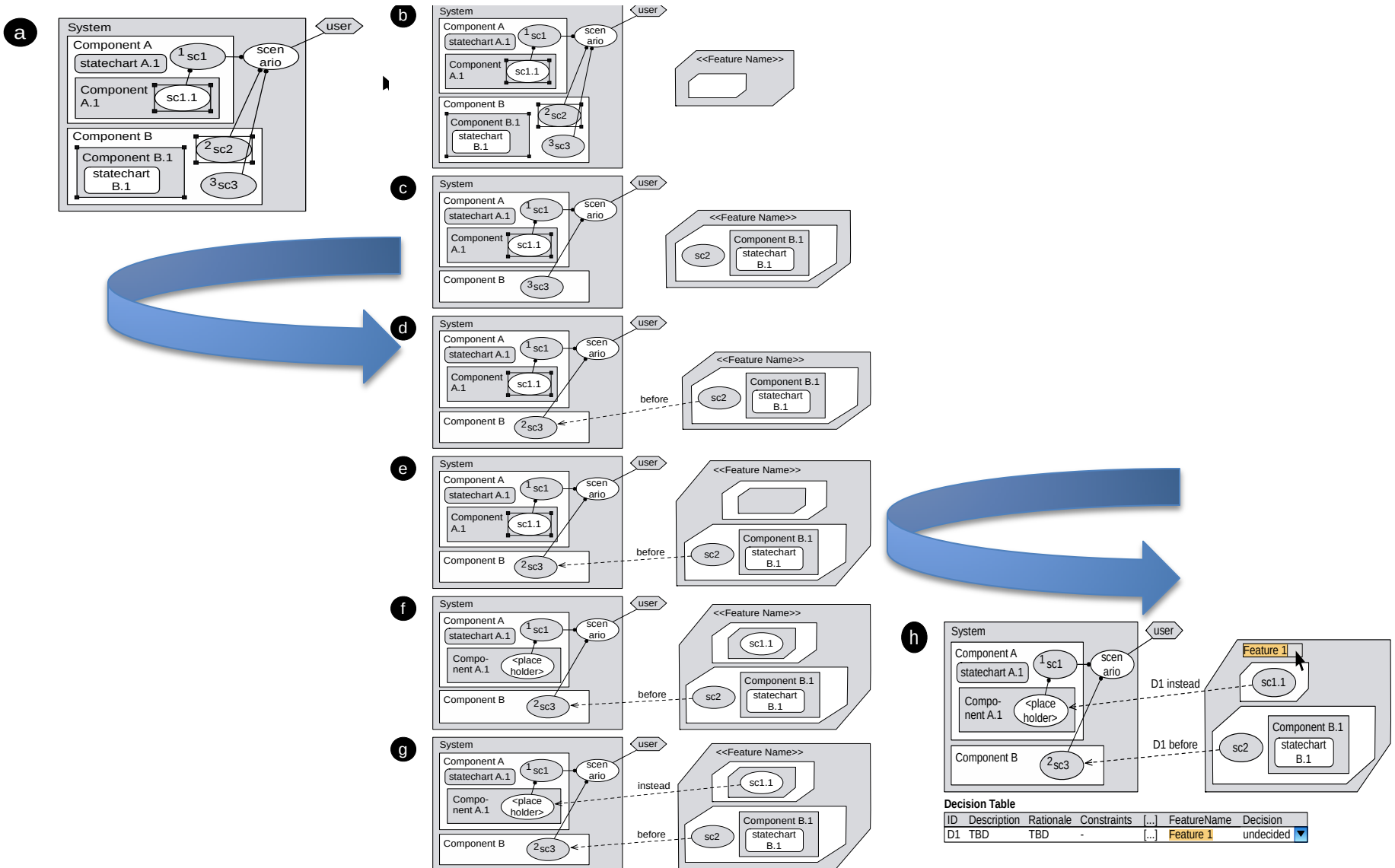
Types of Variability

- *Local Variability:*
- *Homogeneously Cross-Cutting Variability:*
- *Heterogeneously Cross-Cutting Variability:*

The Feature Unweaving Algorithm

1. *Create the BaseAC*
2. *Call the recursive extraction function (parameters: the*
 1. *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)*

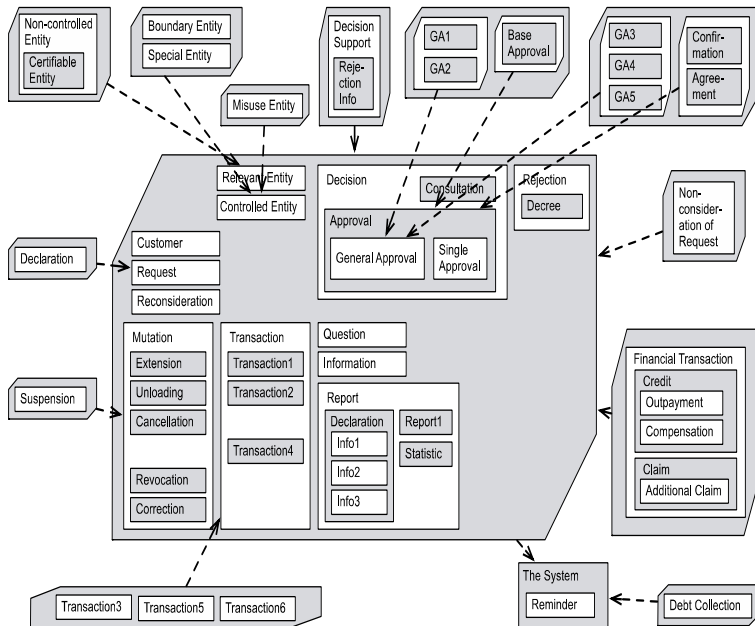
THE FEATURE UNWEAVING- EXECUTION TRACE



Feature unweaving a.) Reference Model b-g.) Steps h.) Final state

CASE STUDY I- GOVERNMENT DECISION SYSTEM - EXEMPLAR

COMPARISON OF SPECIFICATION EFFORT.



Work Result	Feature Extraction	AND/OR Table
Model requirements of Prod1		
Prod1	Specify Prod1: add 47 reqs.	Specify Prod1: add 47 reqs.
Add variability		
Var12	Extract Prod1: add 5 aspects (15 reqs.) Add Prod2: add 6 aspects (14 reqs.)	Add Prod2: add 14 reqs. Expand table: add 72 cells
Derive products from Var12		
Prod1 Prod2	Weave 5, remove 6 aspects Weave 6, remove 5 aspects	Remove 14 requirements Remove 15 requirements
Add variability		
Var123	Extract Prod1+2: add 1 aspect (31 reqs.) Add Prod3: add 1 aspect (1 req.)	Add Prod3: add 1 req. Expand table: add 75 cells
Derive products from Var123		
Prod1 Prod2 Prod3	Weave 6, remove 7 aspects Weave 7, remove 6 aspects Weave 2, remove 1 aspect	Remove 15 requirements Remove 16 requirements Remove 53 requirements
Total	75 specification operations 29 product derivation operations	209 spec. operations 84 prod. deriv. operations

4'100 products in ADORA

VS

262 ($4.61 \cdot 10^{18}$) possible products in AND/OR Table

CASE STUDY II- MOBILE PHONE DOMAIN- GoPhone

Comparative Evaluation

- Sparx-Systems Enterprise Architect (EA) in combination with pure-systems pure::variants,
- IBM Rational Focal Point
- ADORA

	1. Requirements		2. Requirements and Variability		
	ADORA	EA	ADORA	EA	pure::variants
# diagrams	1	17	1	17	1
# elements	174	254	261	331	68
# connections	~ 200	338	318	482	38
# cross-tree constraints	-	-	14	-	12
<i>Sub-total</i>	-	-	-	<i>830</i>	<i>119</i>
Total	375	609	594	949	

Benefits-
Integrated modeling, Comprehensive model overview &
Reduced modeling effort

Conclusions and Limitations

- Case 1- Improvements in efficiency of
 - product line specification,
 - product variant definition,
 - release planning
 - consistency maintenance.
- Case 2-Significant reduction of the
 - complexity and
 - size of the graphical product line requirements specification.

Here are the Limitations:

1. The modeler's mental map related to the concerned model elements gets disturbed.
2. ADORA specification needs to be created first.
3. Considerable amount of aspect containers and join relationships.
4. Many pseudo-elements needed because of limitations/requirements of ADORA's aspect weaving semantics.



Thank You