BX4RE: Synchronizing Feature Models and Use Cases

Haiyan Zhao

Peking University

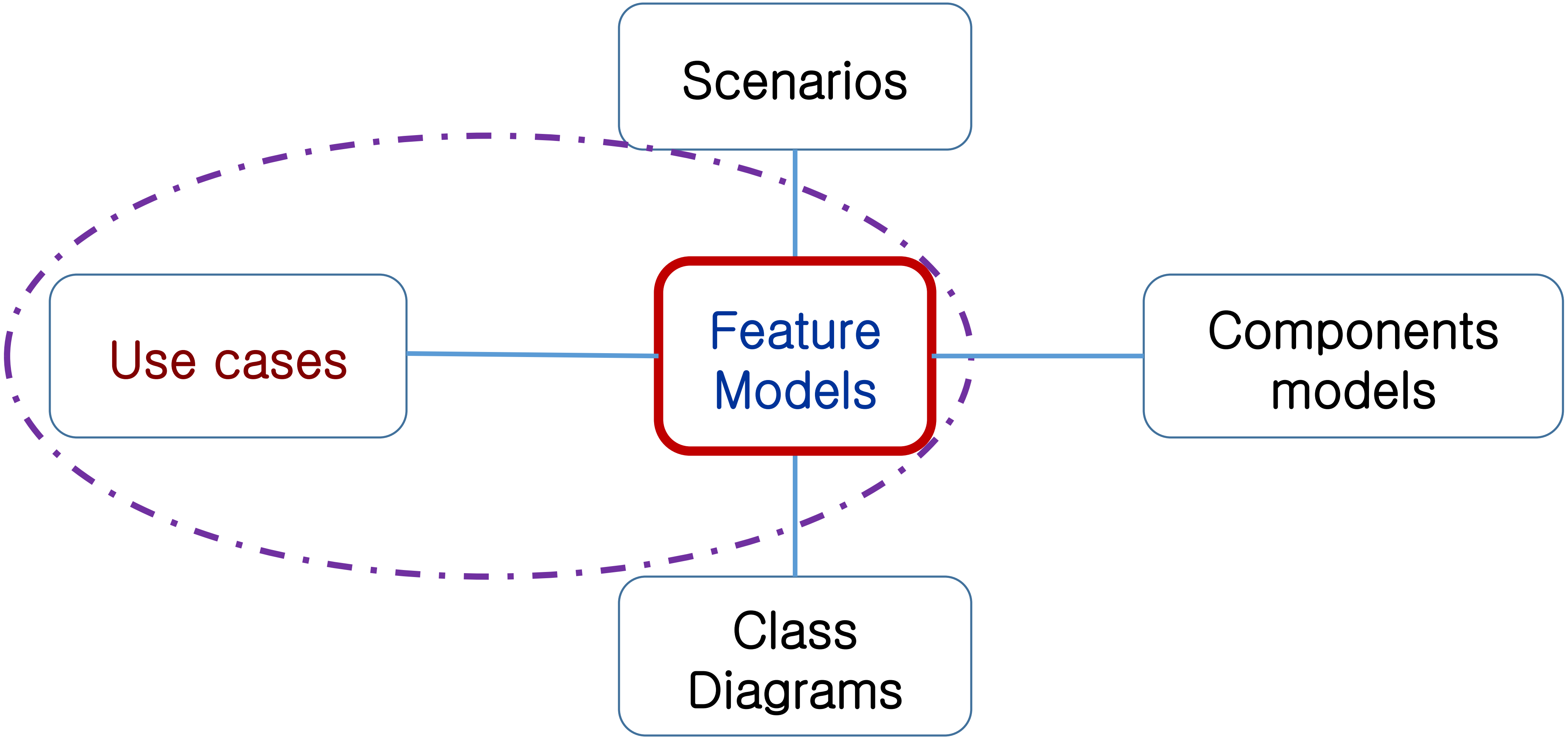
zhhy.sei@pku.edu.cn

Joint work with

Weize Zhao (PKU), Zhenjiang Hu (NII)

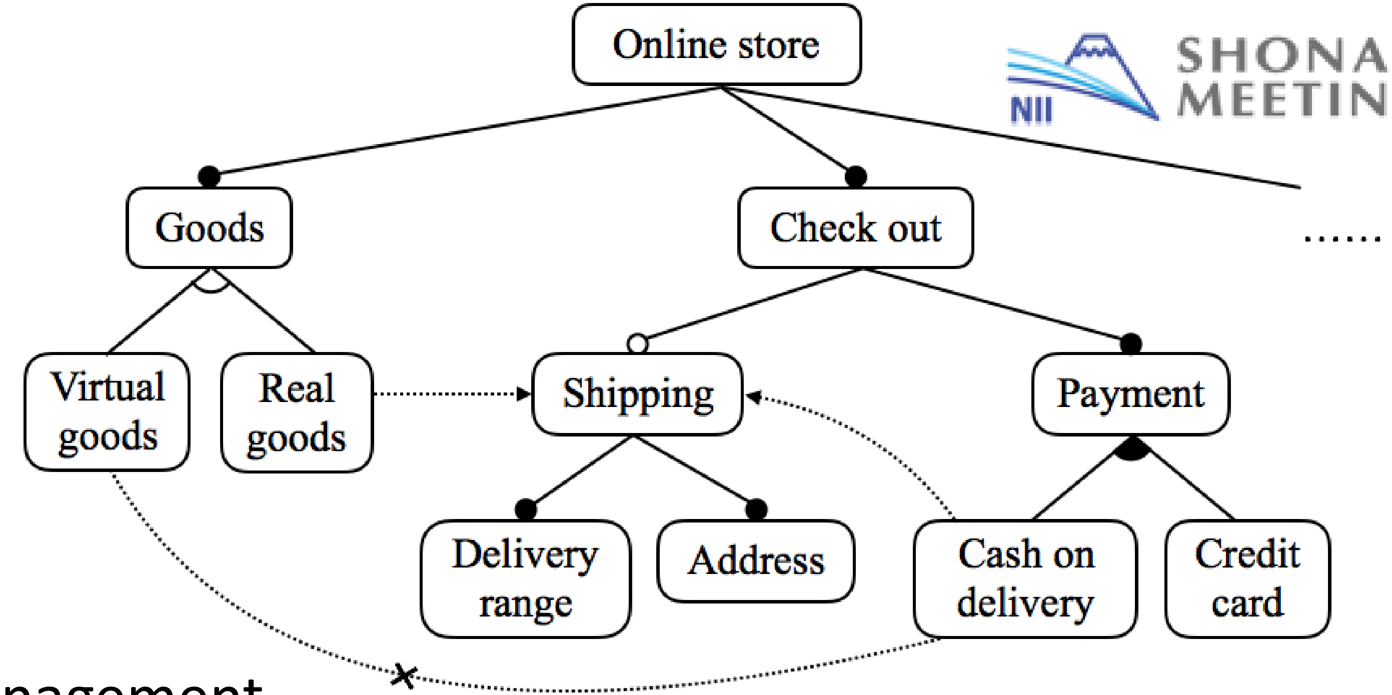
Shonan BX, 25-29 Sept. 2016

Feature Model Centric Approach to RE/SPL/SW Adaptation



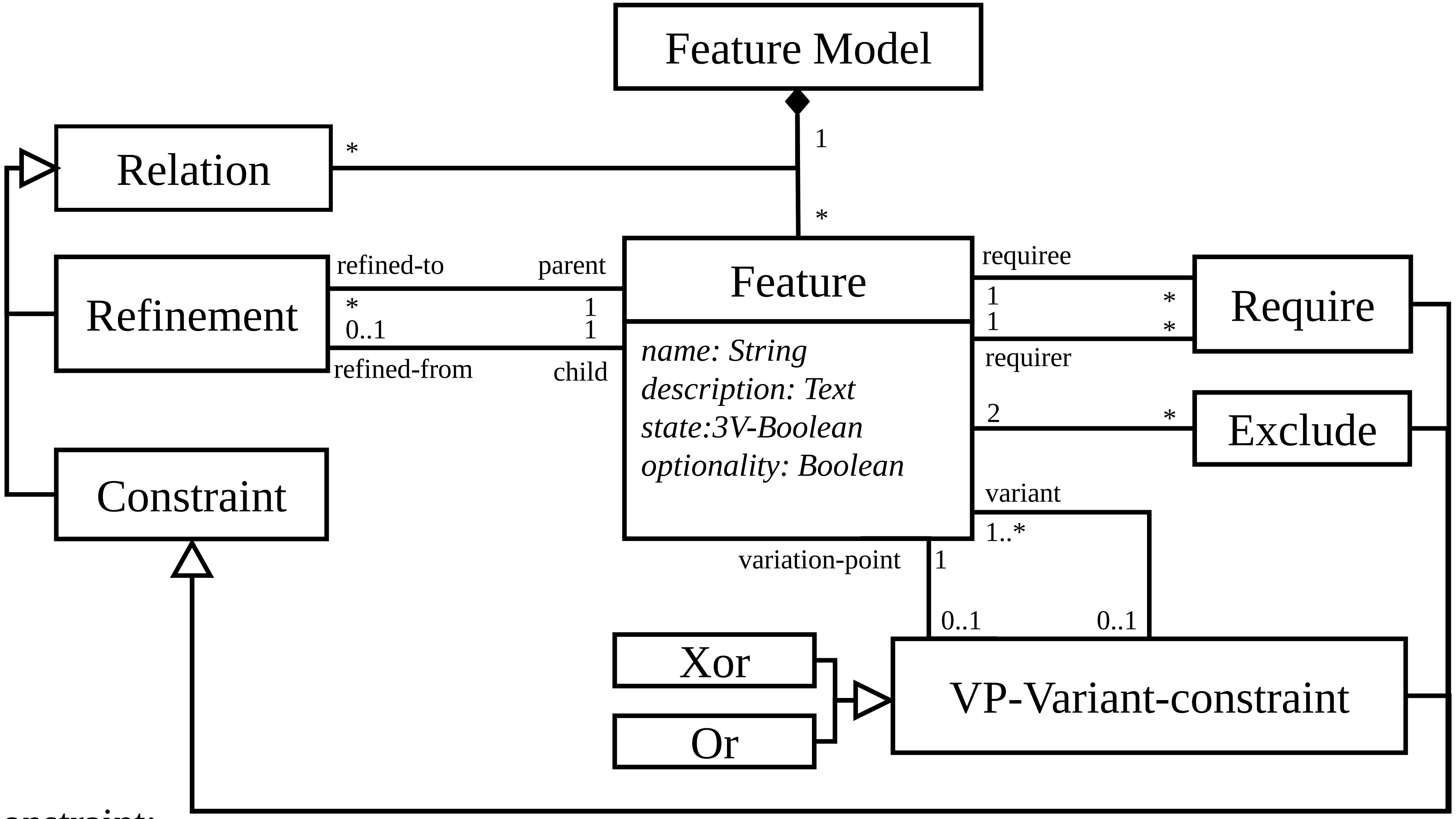
+ Traceability between feature models and other artifacts

Feature Model



- Fulfills the need of variability management
  - Commonality : Mandatory features
  - Variability : Optional features , Variant features
- Provides an efficient way to abstract from requirements
  - Group a set of related requirements in one feature
  - Narrow the gap between the end-user and the developer perspective of a software system

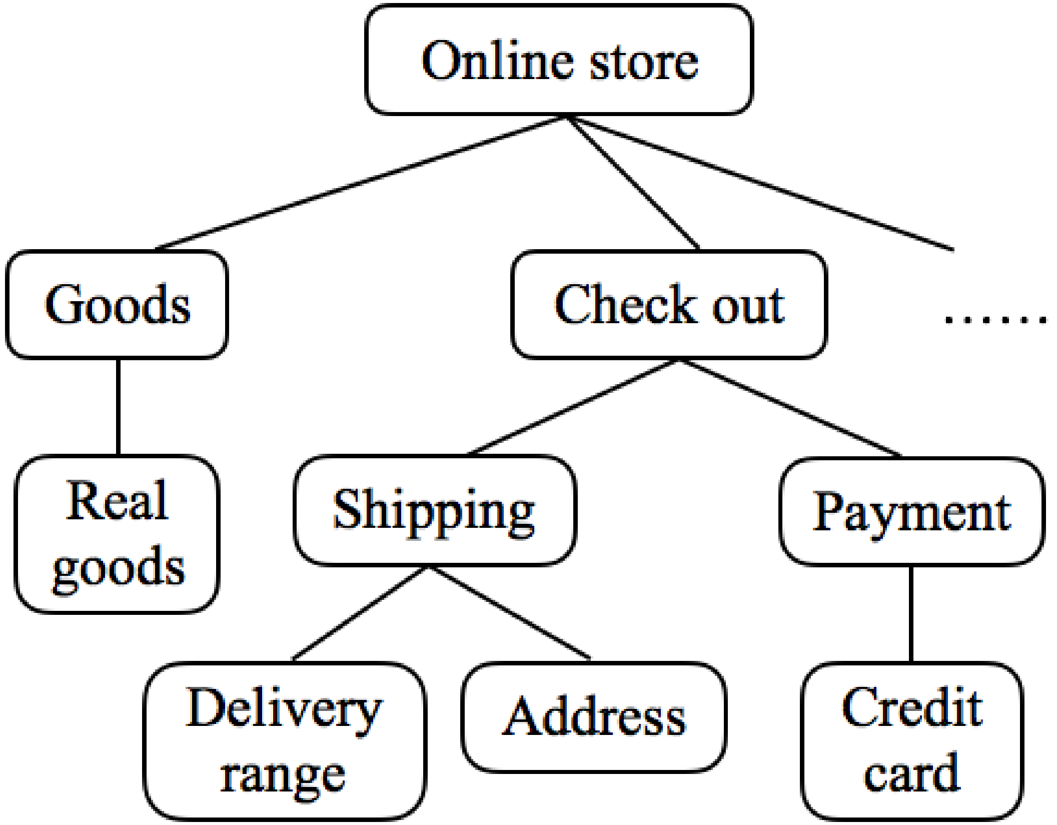
Meta-model of Feature Models



OCL constraint:
context c: VP-Variant-Constraint
inv: c.variant->forAll(refined-from.parent=c.variation-point)

Feature Model Configuration

- A set of features selected from the feature model, which describes a specific system (software), as well as satisfies the constraints imposed by the model

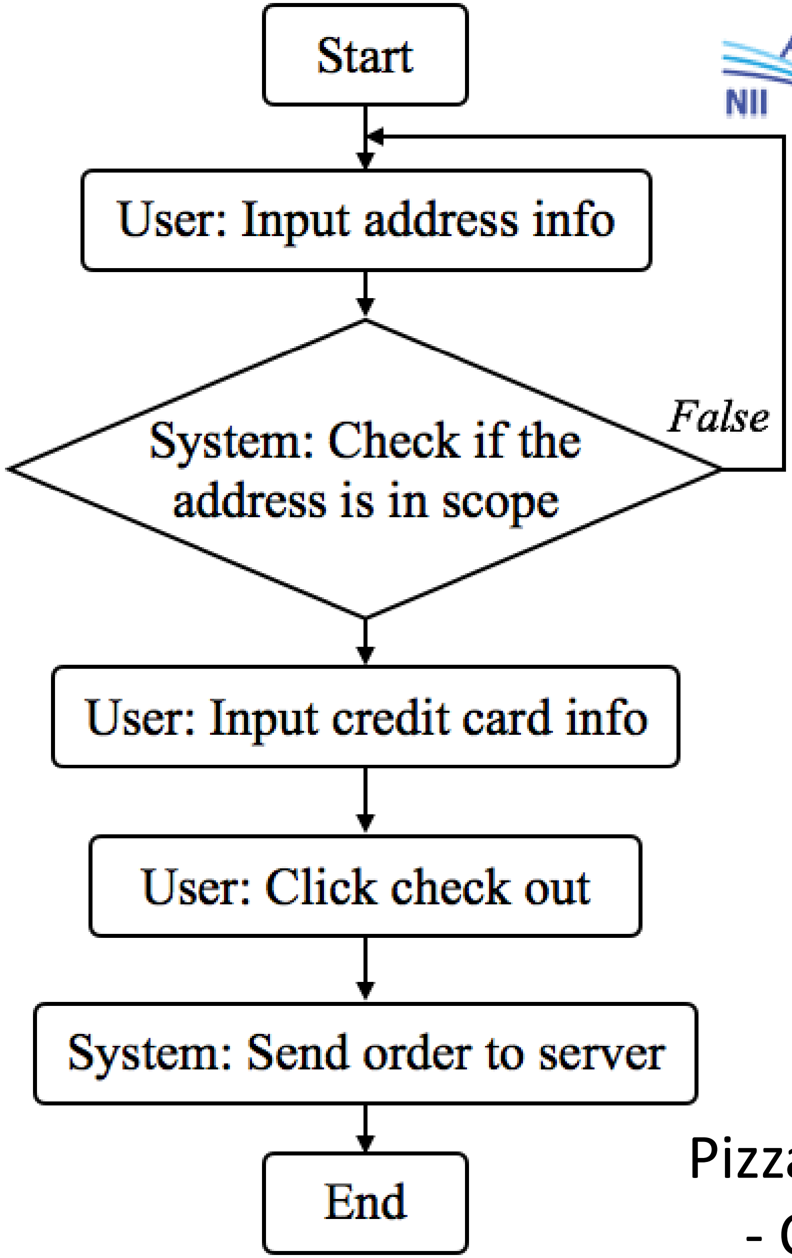


Online Pizza Store

Use Case

- Consists of a list of actions, typically defining the interactions between a role (actor) and a system to achieve a substantial goal

- Easy to communicate with stakeholders/end-users

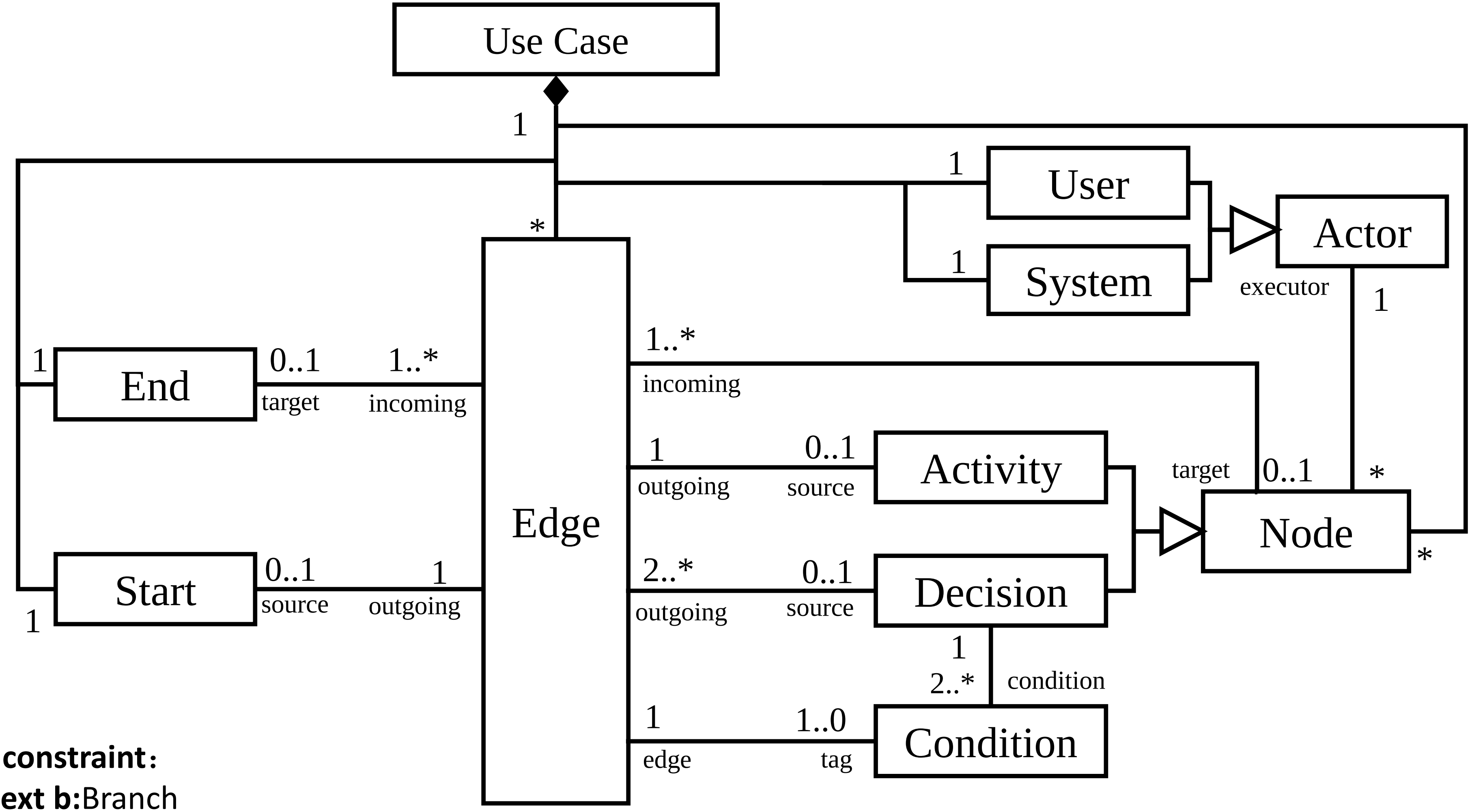


Start
User: Input address info
System: Check if the address is in scope
False
User: Input credit card info
User: Click check out
System: Send order to server
End

Pizza Online Store

- Check Out

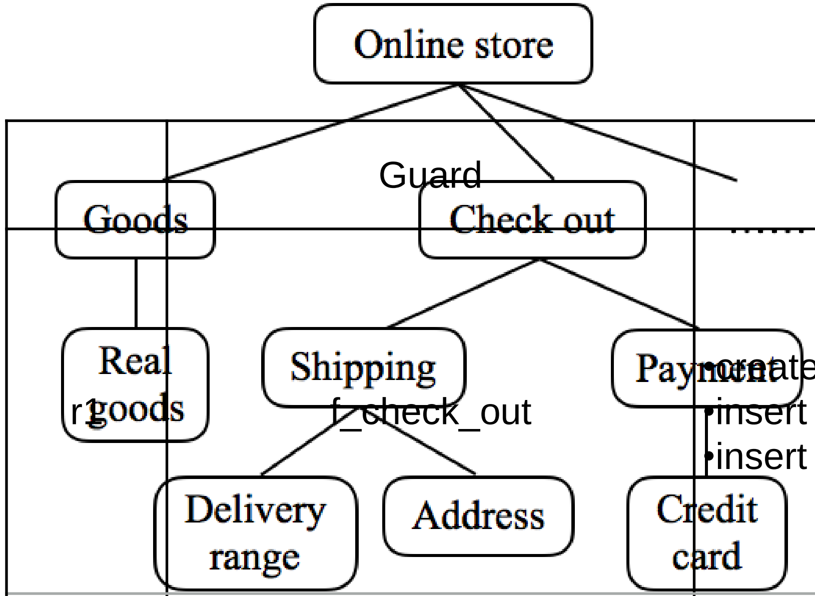
Meta-model of Use Cases

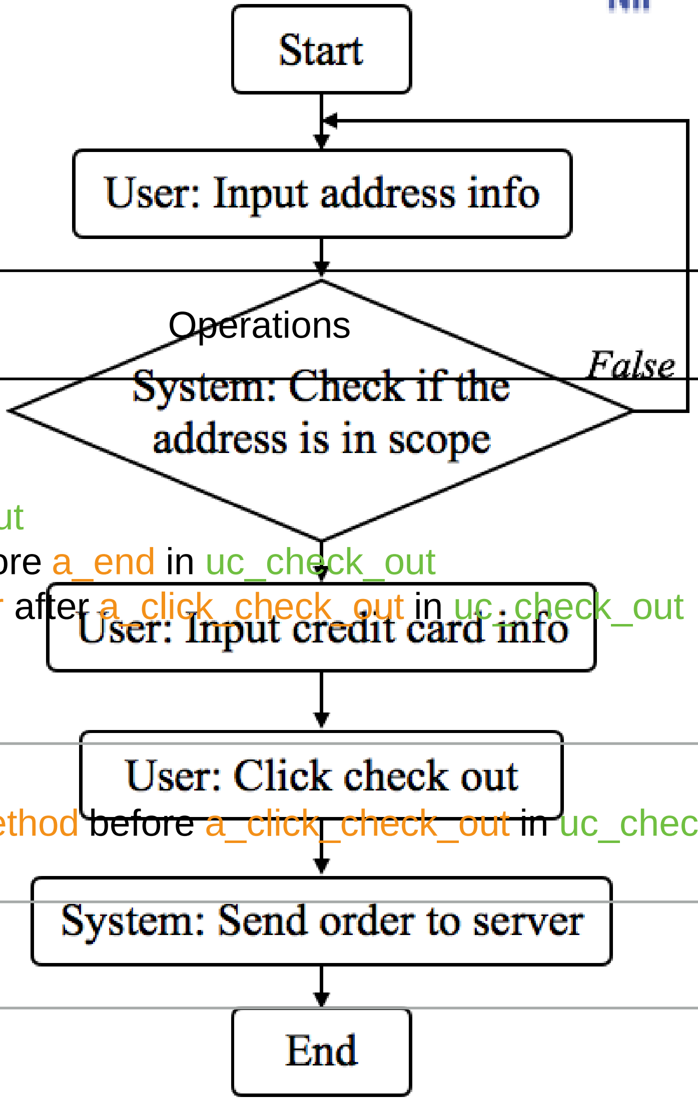


OCL constraint：
context b: Branch
inv: b.outgoing->forAll (e : Edge | e.tag->size()=1 **and** b.condition->includes(e.tag)) **and**
b.condition->forAll (c : Condition | c.edge->size()=1 **and** b.outgoing->includes(c.edge))

TDL4UC: X-rules

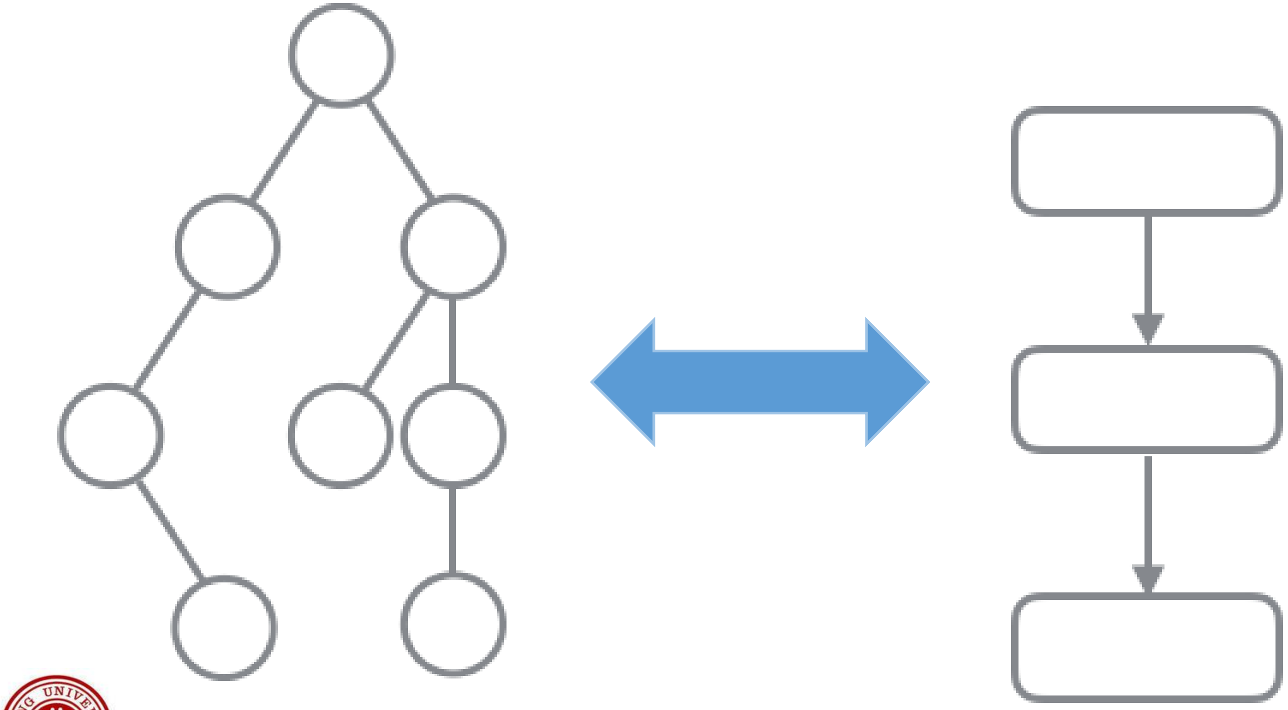
	Guard	Operations
r1	f_check_out	•create use case uc_check_out •insert a_click_check_out before a_end in uc_check_out •insert a_send_order_to sever after a_click_check_out in uc_check_out
r2	f_payment	•insert b_pick_a_payment_method before a_click_check_out in uc_check_out
…	…	…

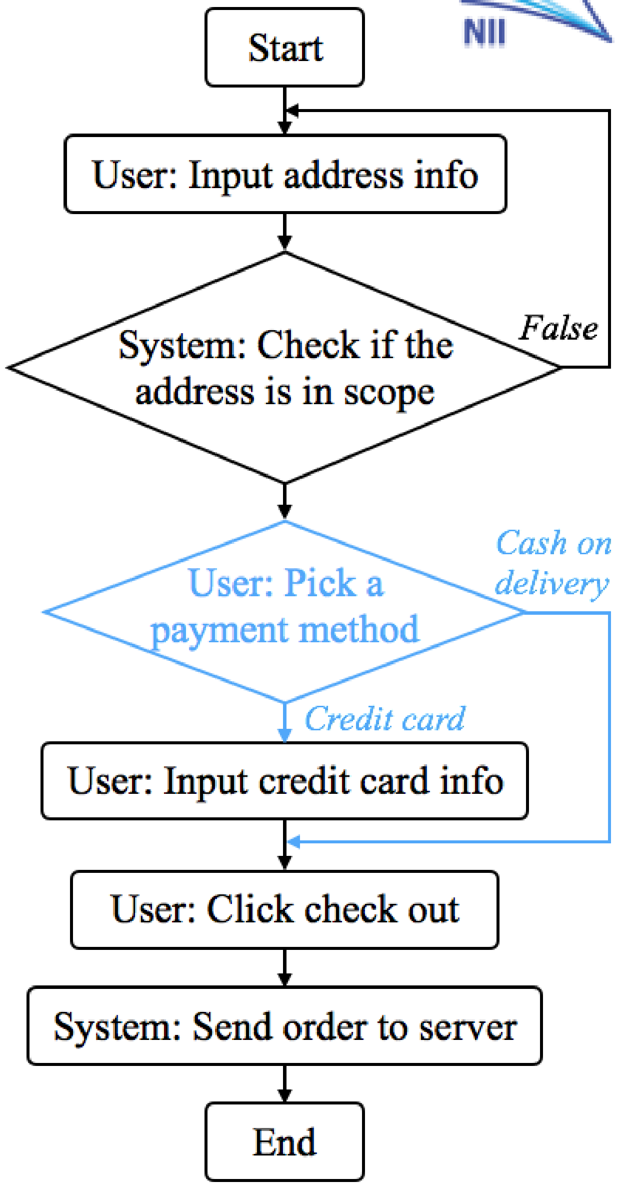



Start
User: Input address info
System: Check if the address is in scope
False
User: Input credit card info
User: Click check out
System: Send order to server
End

Synchronization UC with FMC

Need to propagate the modification in use case to feature model configuration, and may further to feature model itself





Start

User: Input address info

System: Check if the address is in scope

False

User: Pick a payment method

Cash on delivery

Credit card

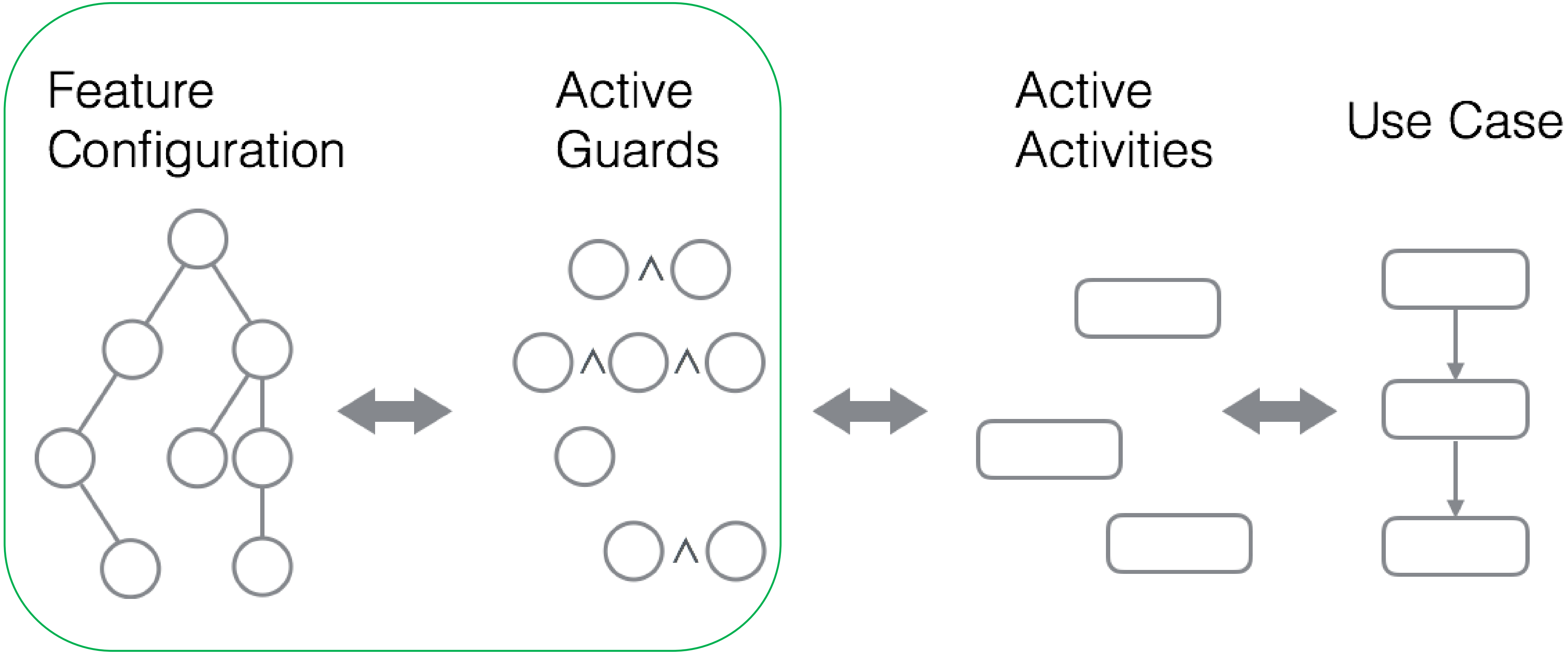
User: Input credit card info

User: Click check out

System: Send order to server

End

X-rules vs UC & FMC



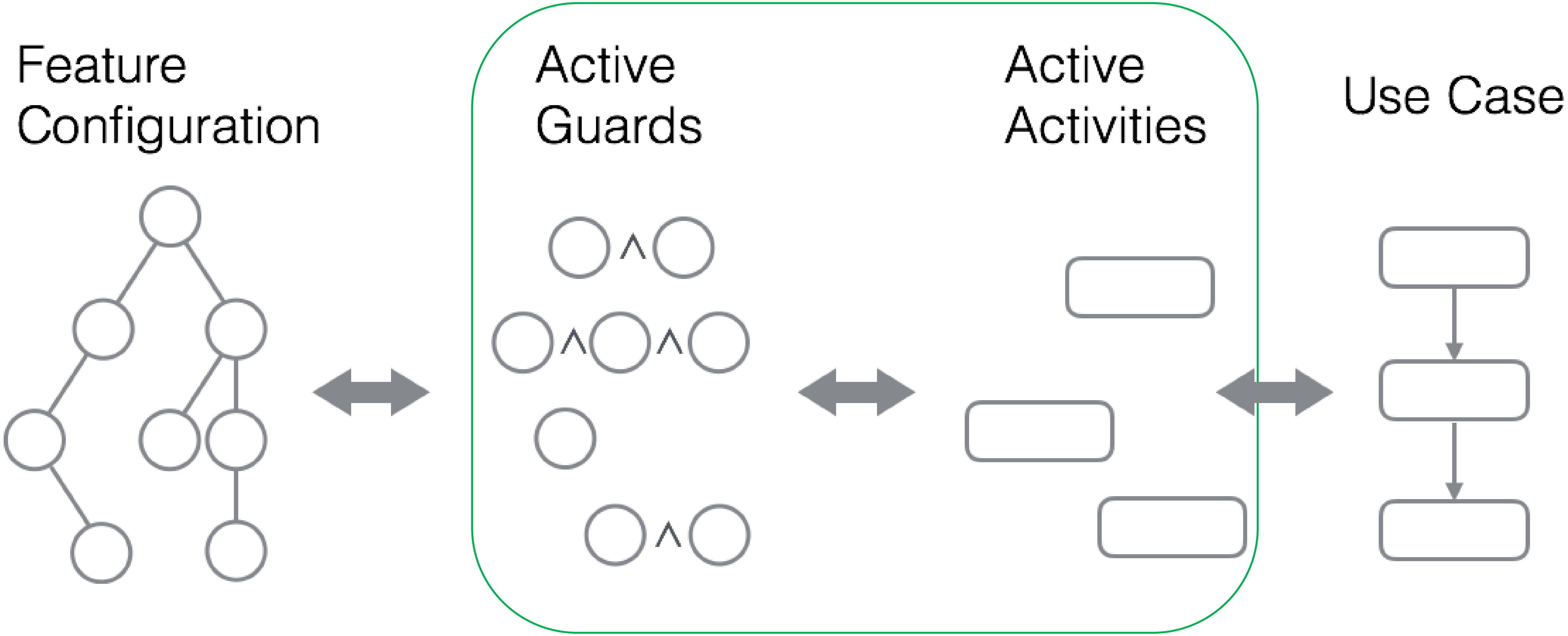
Feature Configuration

Active Guards

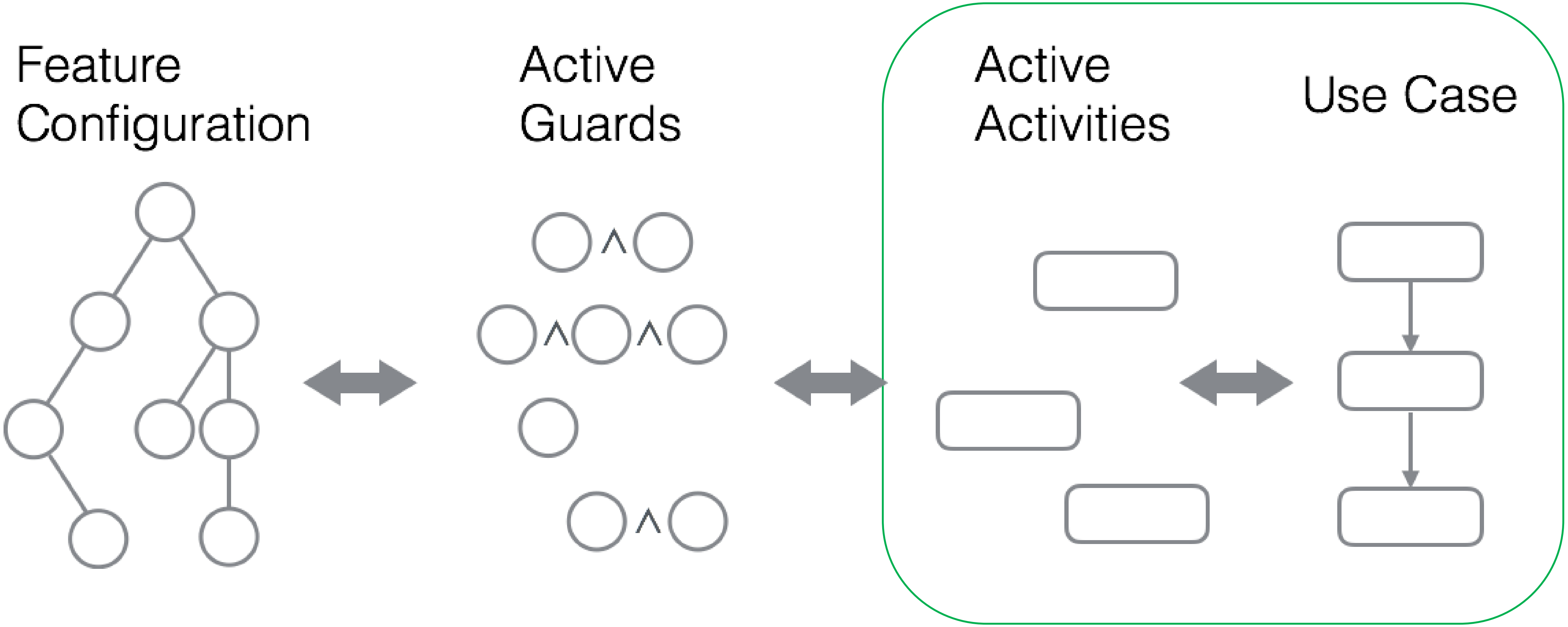
Active Activities

Use Case

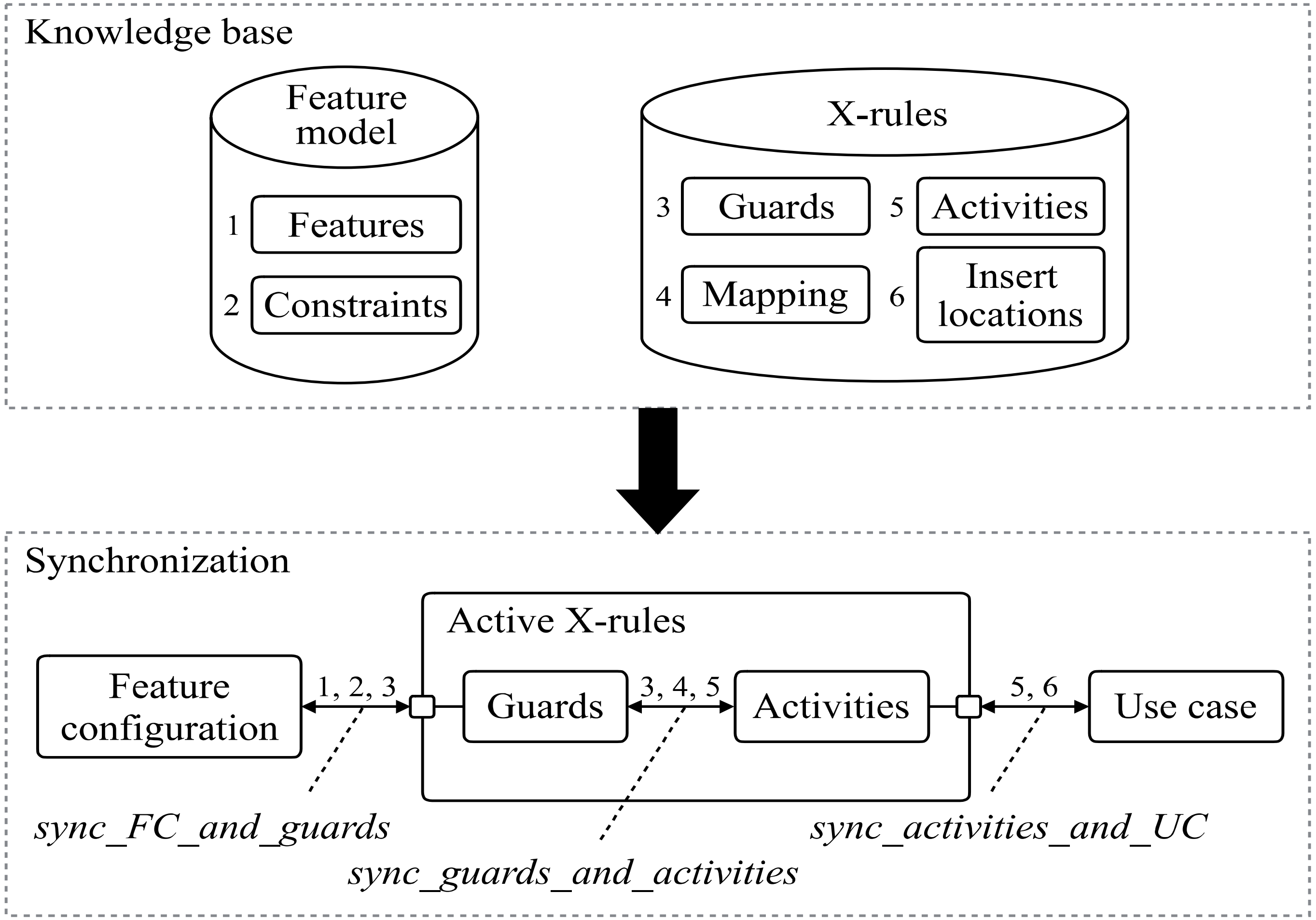
X-rules vs UC & FMC



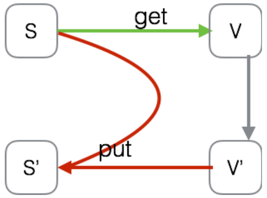
X-rules vs UC & FMC

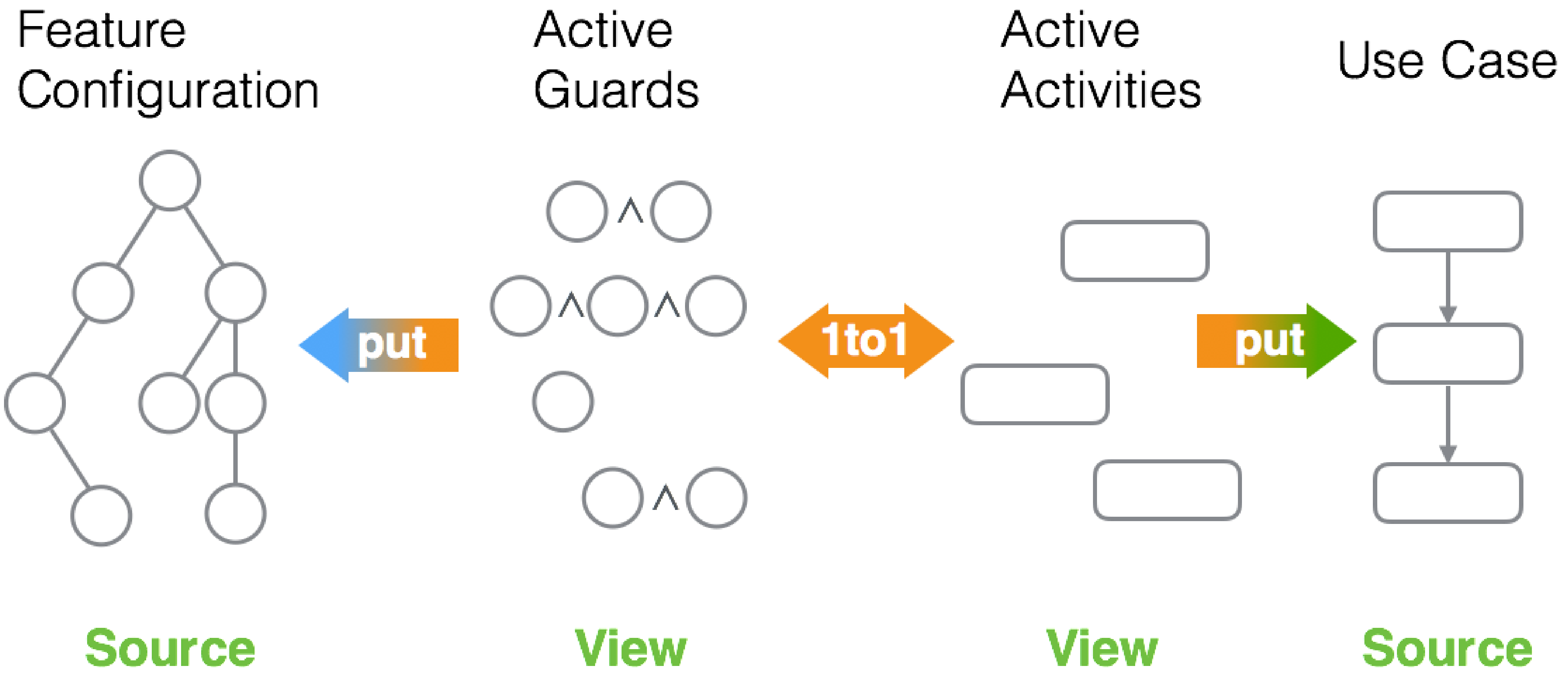


Synchronization Framework

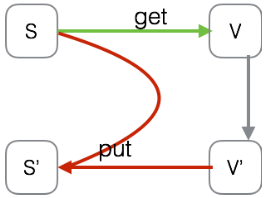


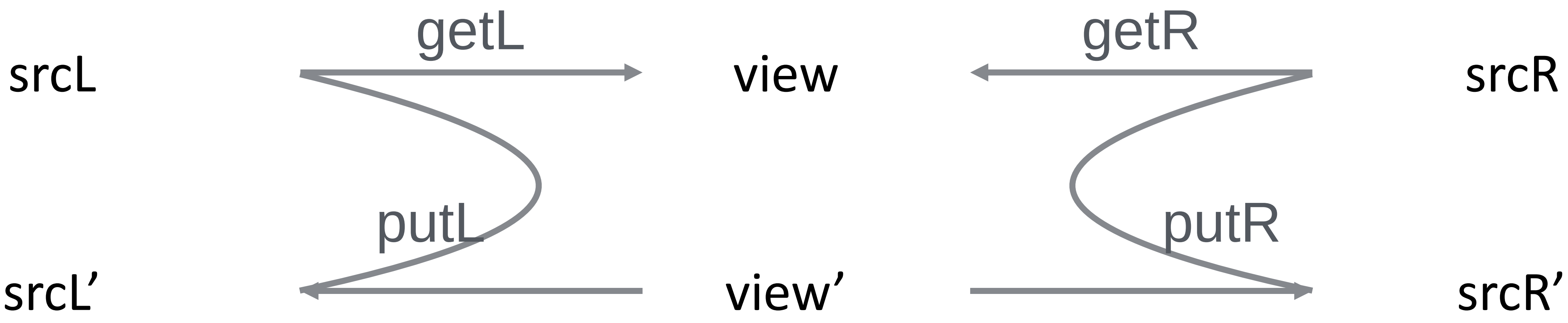
Synchronization Using BiGUL





Synchronization Using BiGUL




srcL

getL

view

getR

srcR

putL

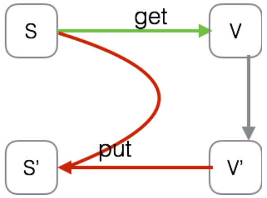
putR

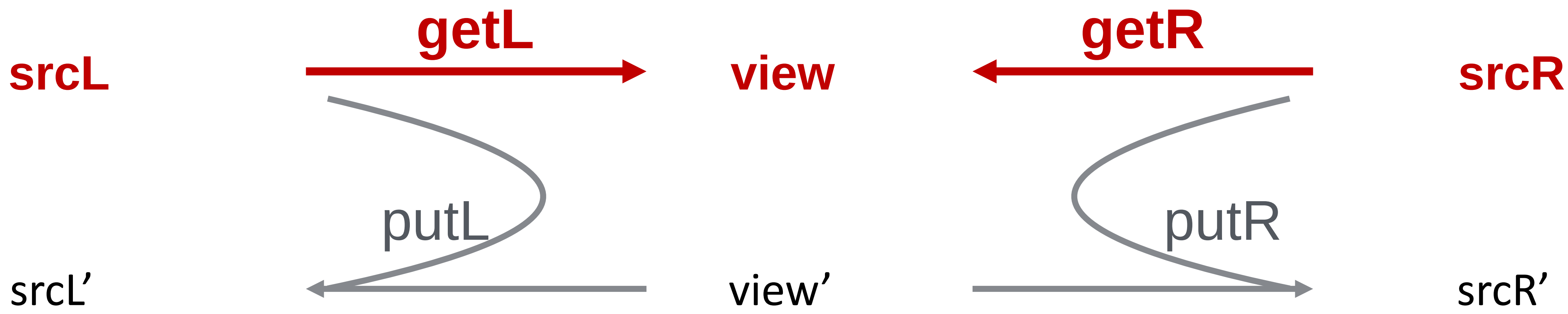
srcL’

view’

srcR’

Synchronization Using BiGUL





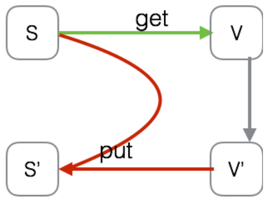
srcL getL view getR srcR

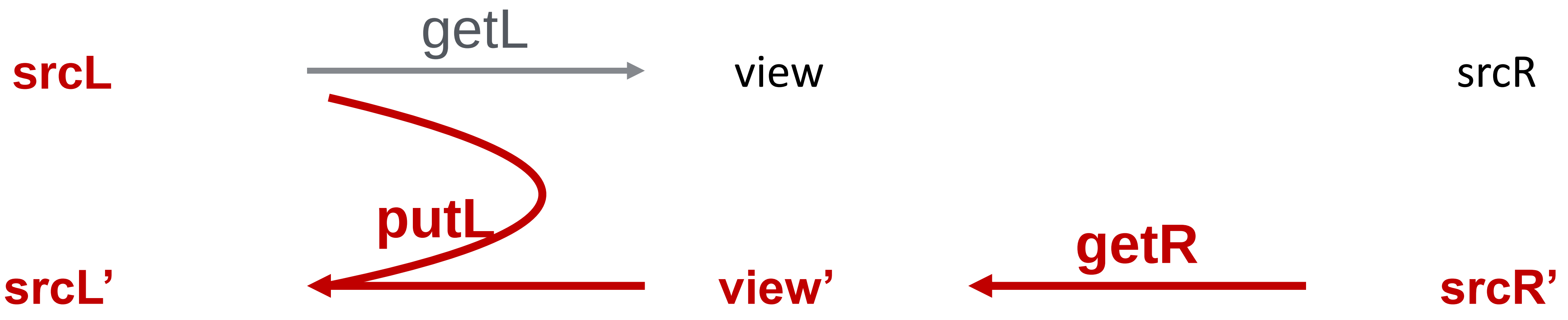
putL putR

srcL’ view’ srcR’

Consistency: srcL~srcR $\overset{\text{def}}{=}$ **getL srcL = getR srcR**

Synchronization Using BiGUL

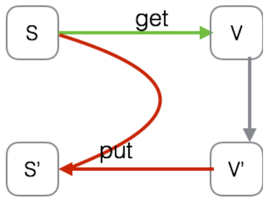


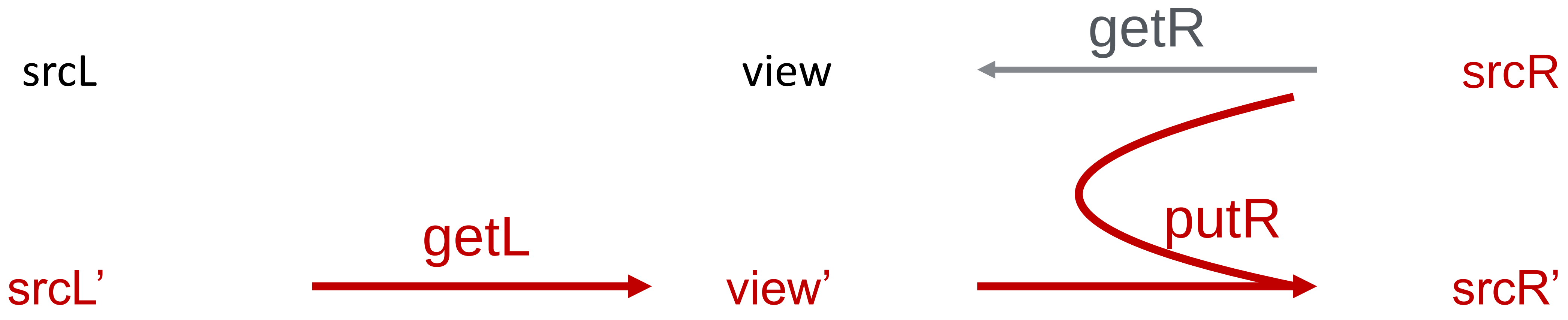


Consistency: srcL~srcR $\overset{\text{def}}{=}$ **getL srcL = getR srcR**

# 1: putR2L srcL srcR’= putL (srcL (getR srcR))

Synchronization Using BiGUL





srcL view getR srcR

putR

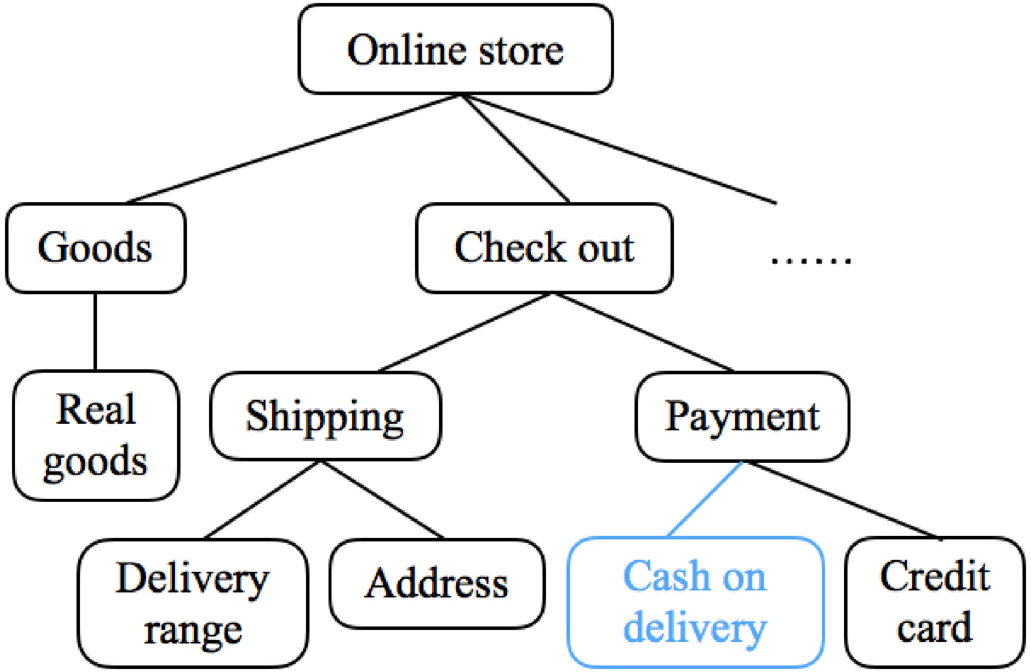
srcL' getL view' srcR'

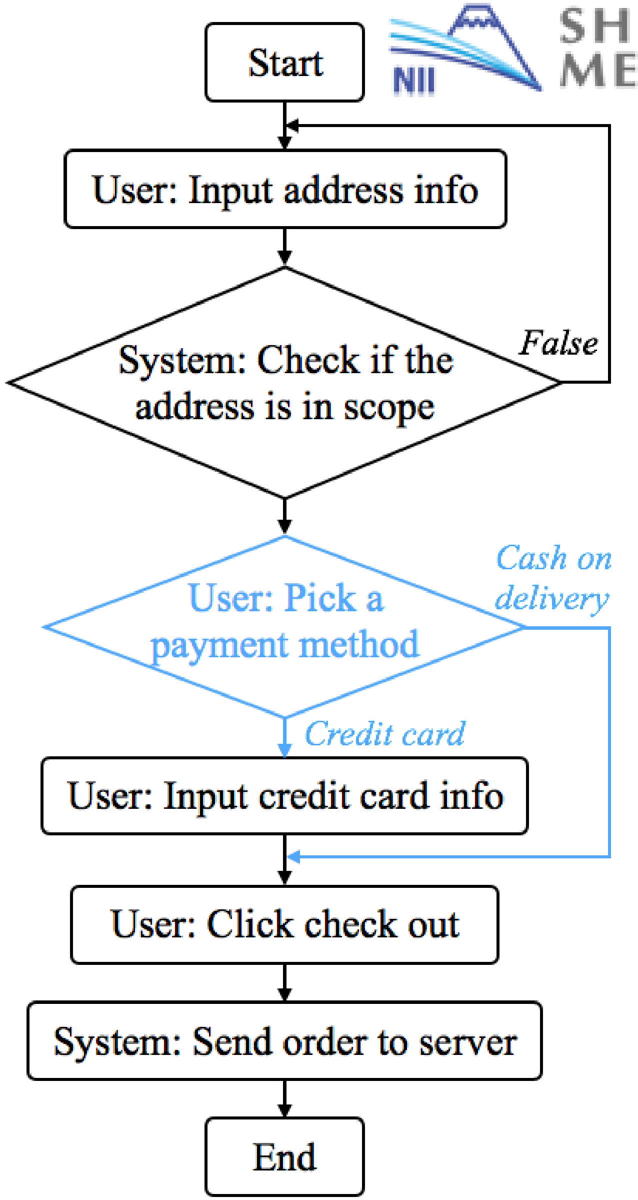
Consistency: srcL~srcR $\overset{def}{=}$ getL srcL = getR srcR

#1: putR2L srcL srcR'= putL (srcL (getR srcR'))

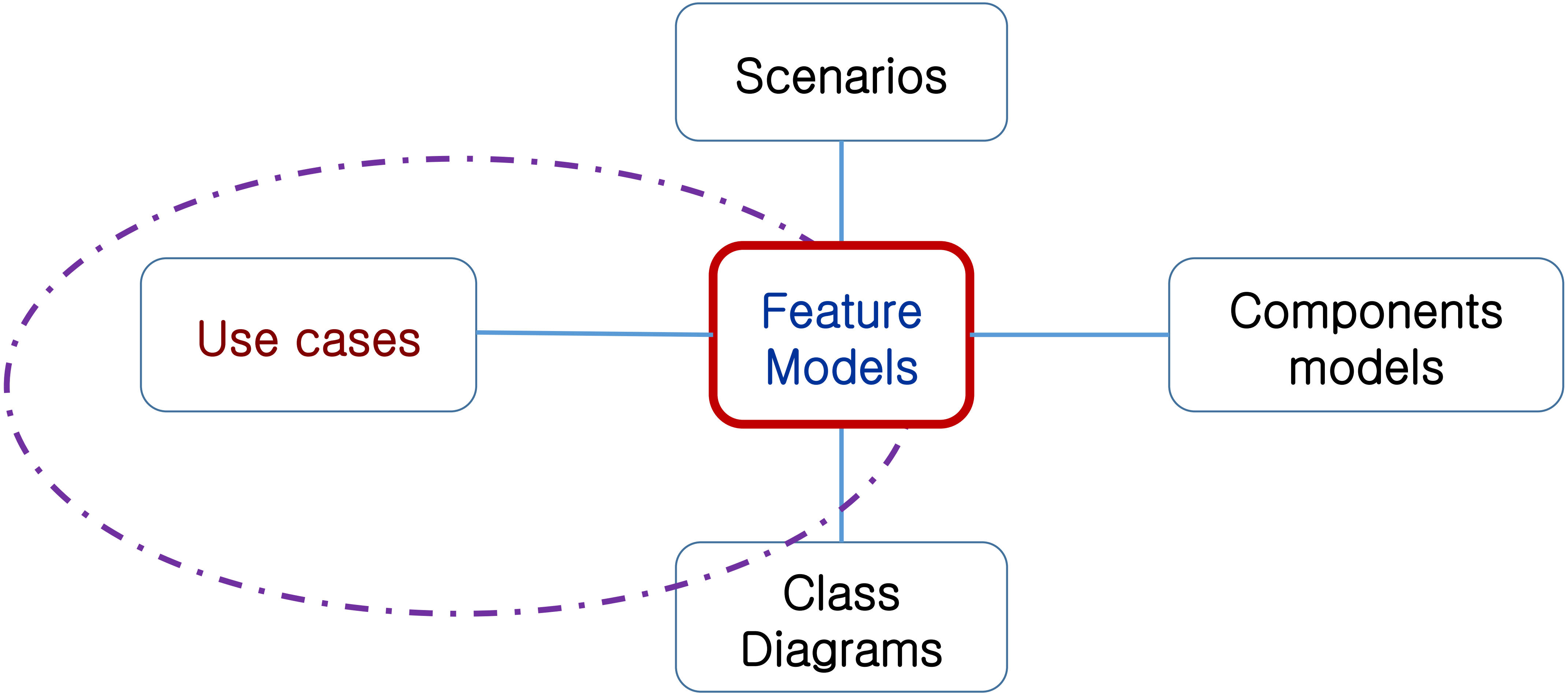
#2: putL2R srcR srcL' = putR (srcR (getL srcL'))

Synchronization Using BiGUL



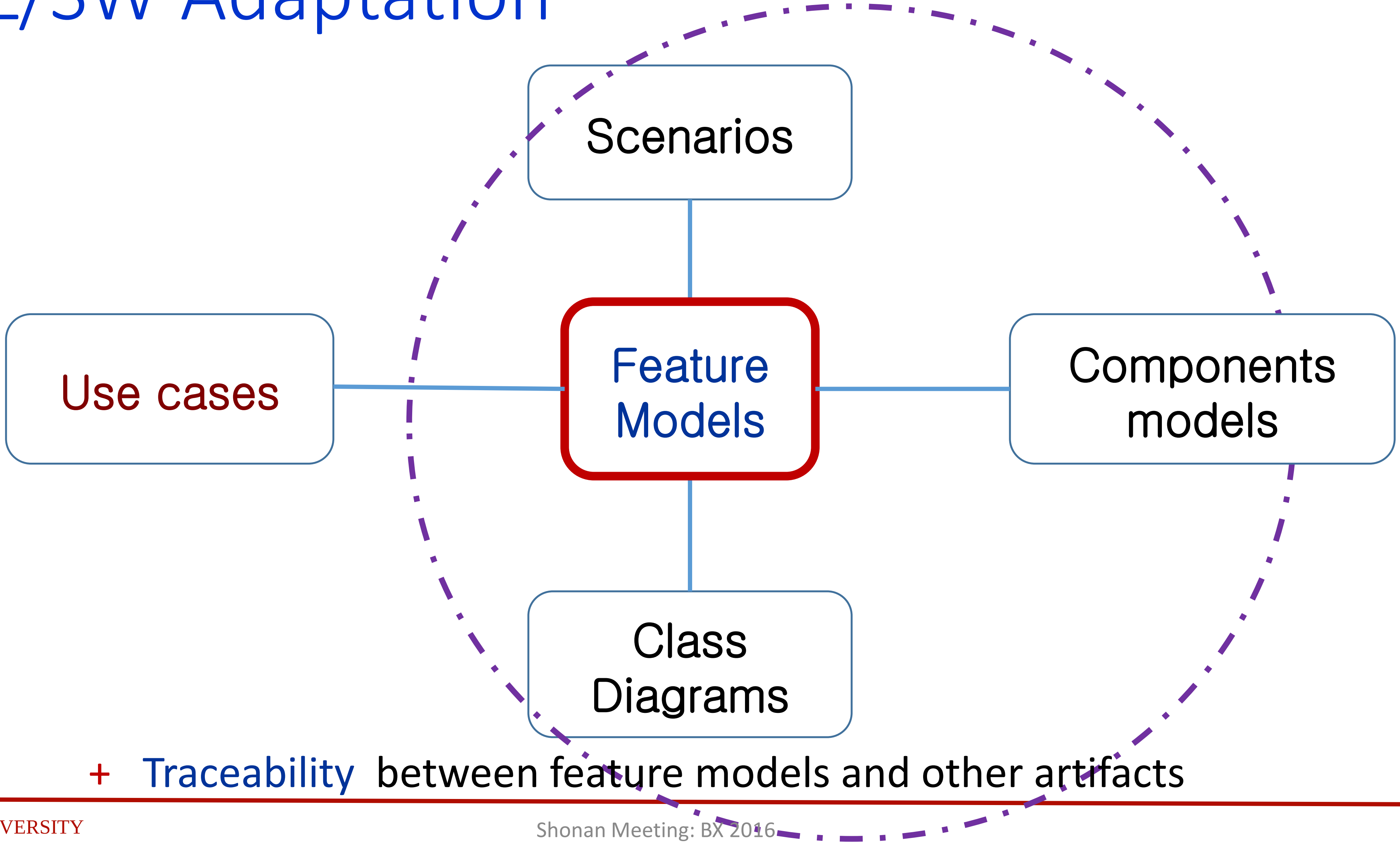

Start
User: Input address info
System: Check if the address is in scope
False
User: Pick a payment method
Cash on delivery
Credit card
User: Input credit card info
User: Click check out
System: Send order to server
End

Feature model centric approach to RE/SPL/SW Adaptation



+ Traceability between feature models and other artifacts

Feature model centric approach to RE/SPL/SW Adaptation



+ Traceability between feature models and other artifacts

Conclusion

- A small step : FMC & UC
  - Modification to UC is limited
  - Evolution of feature model / X-rule

- Behavioral models vs Structural models
  - Ordered list / set

- Collective models merging/synchronization

THANKS!