Software product lines

Paulo Borba

Informatics Center

Federal University of Pernambuco

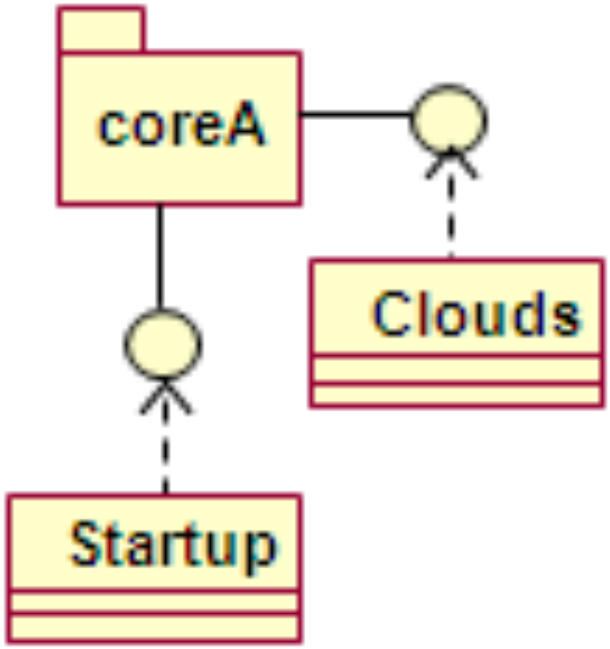
Models for software product lines

Paulo Borba

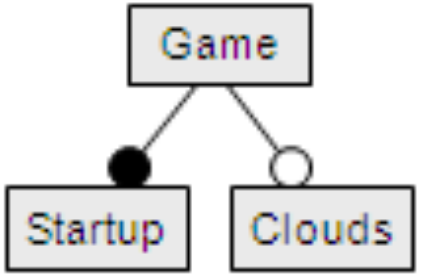
Informatics Center

Federal University of Pernambuco

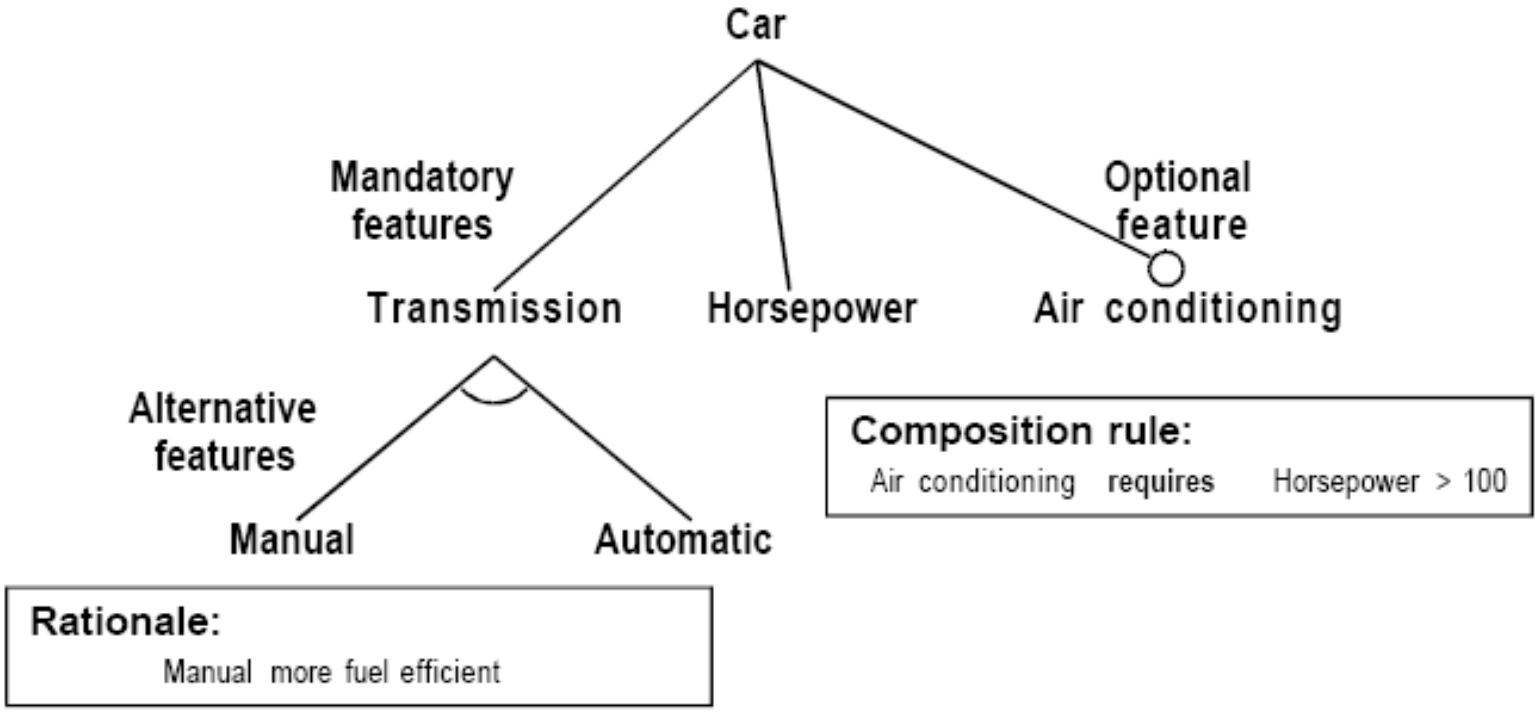
The code shows possible variation points and variations...



but feature models show more...

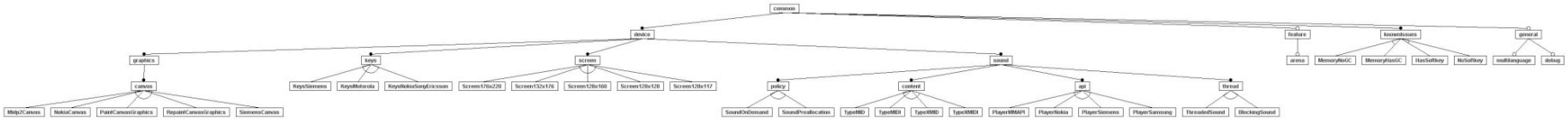


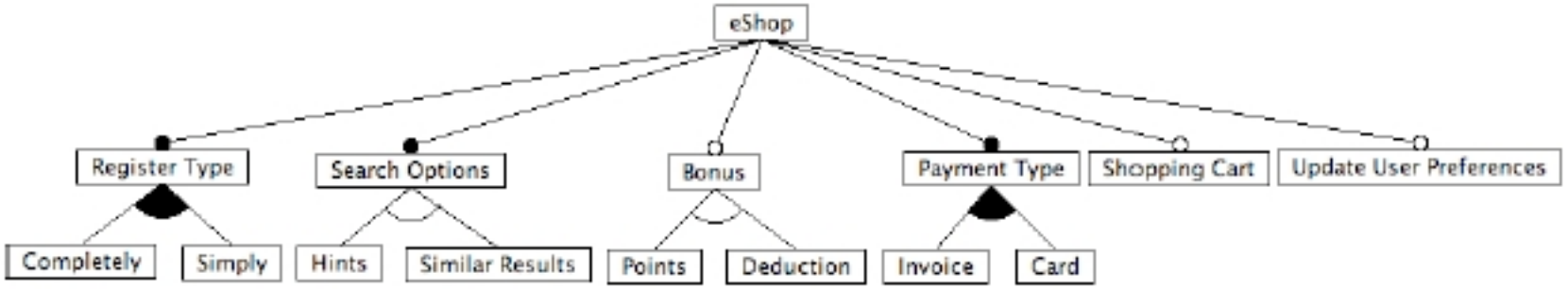
and help visualize and manage variations…



From Feature-Oriented Domain Analysis, Kang et al 1990

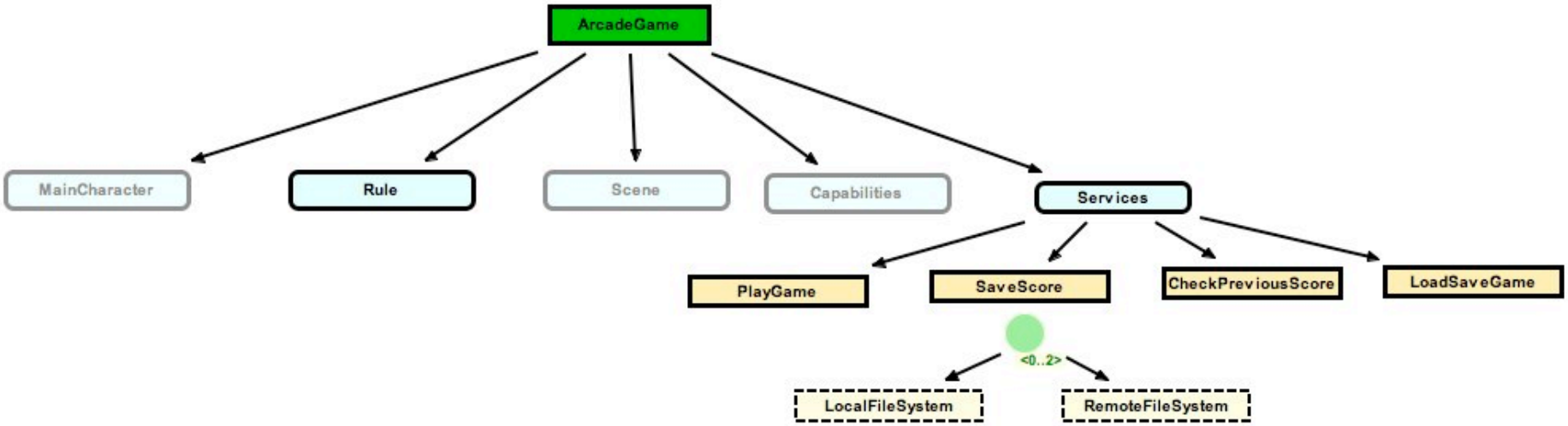
for non trivial product lines





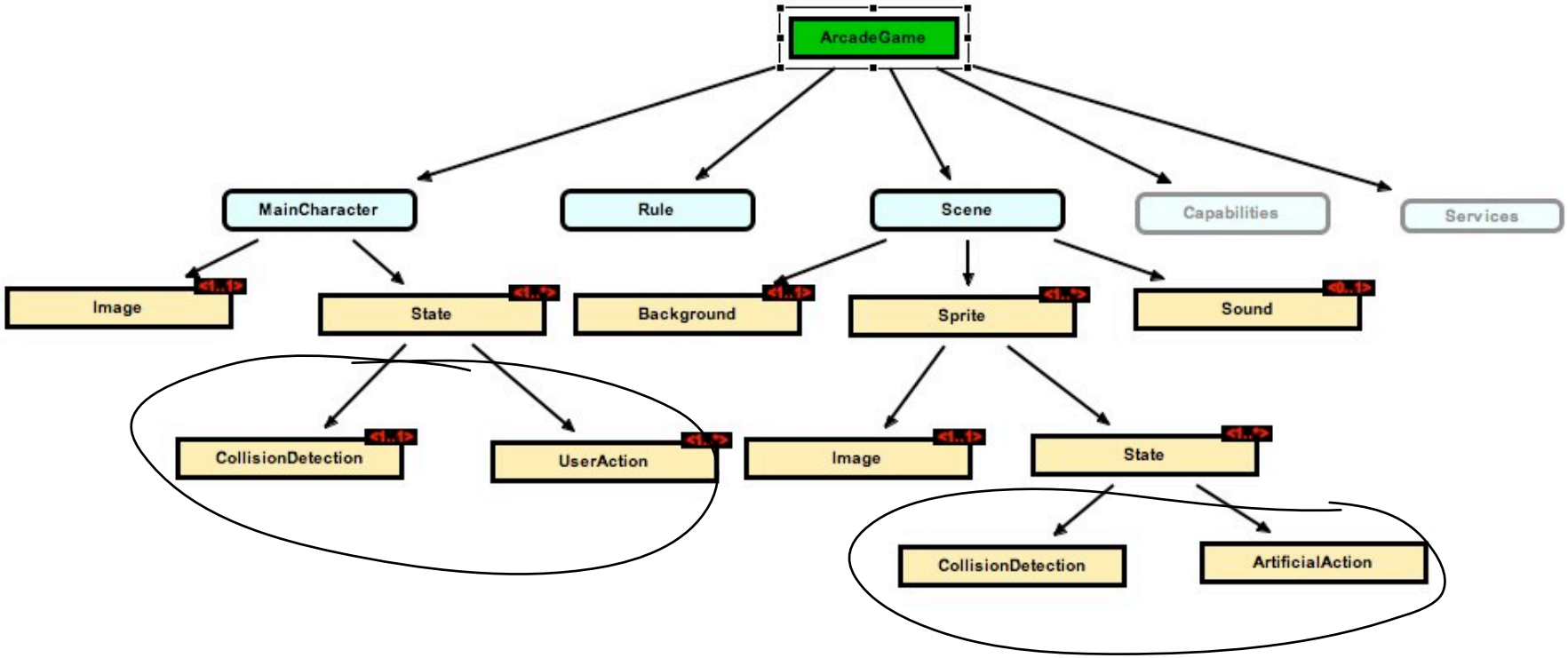
Based on http://gp.uwaterloo.ca/files/2006-lau-masc-thesis.pdf

Cardinalities and attributes yield the power of DSLs...



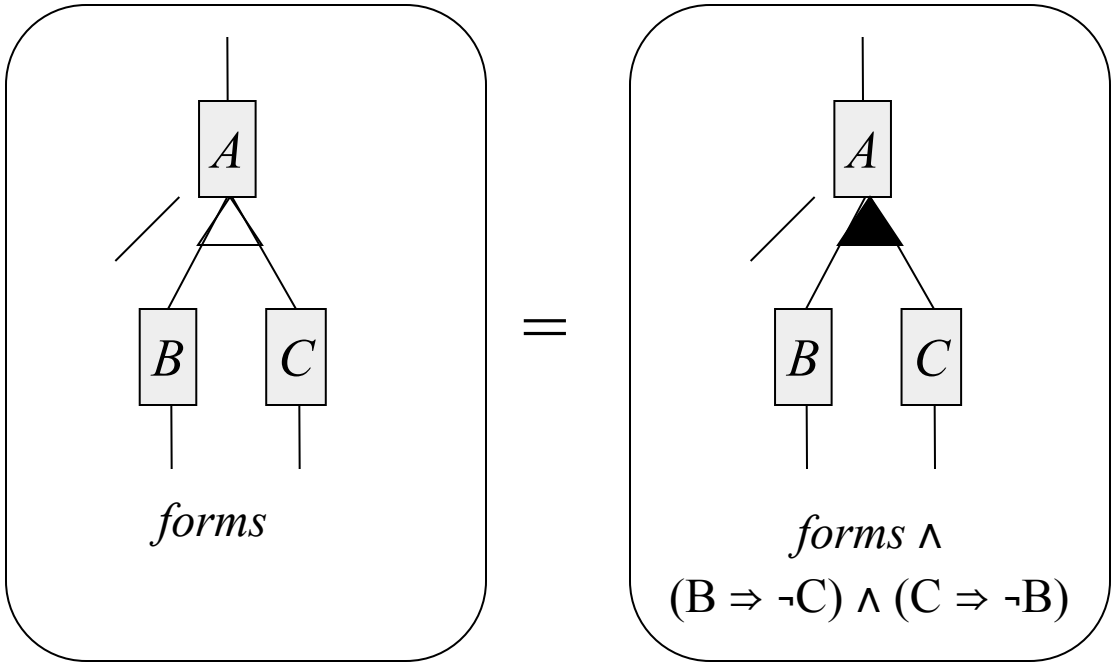
From http://www.sei.cmu.edu/productlines/ppl/

when model is targeted at configuration

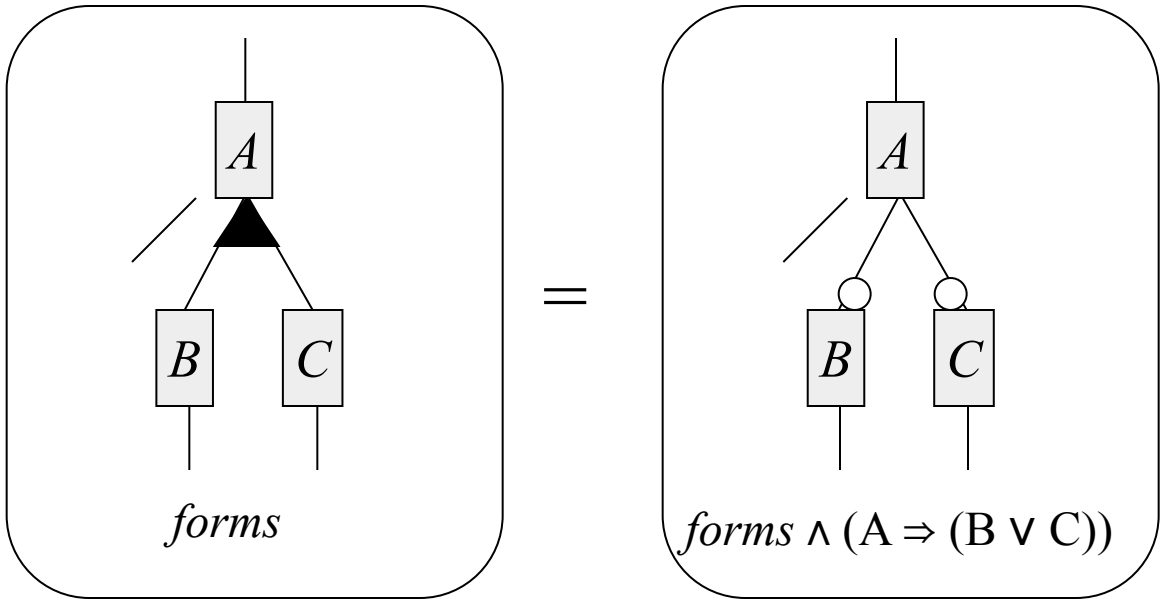


From http://www.sei.cmu.edu/productlines/ppl/

Feature model transformational semantics

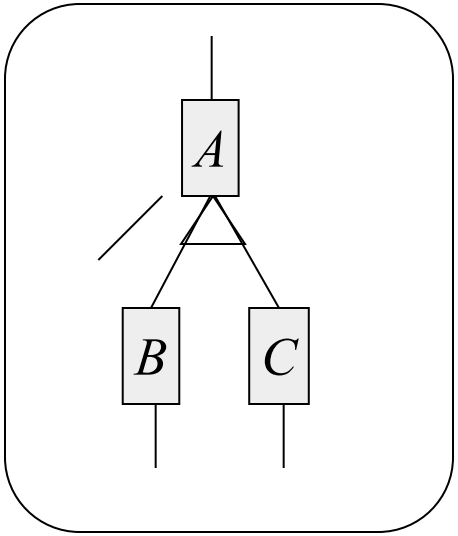


Reduction strategy to features and formulae language



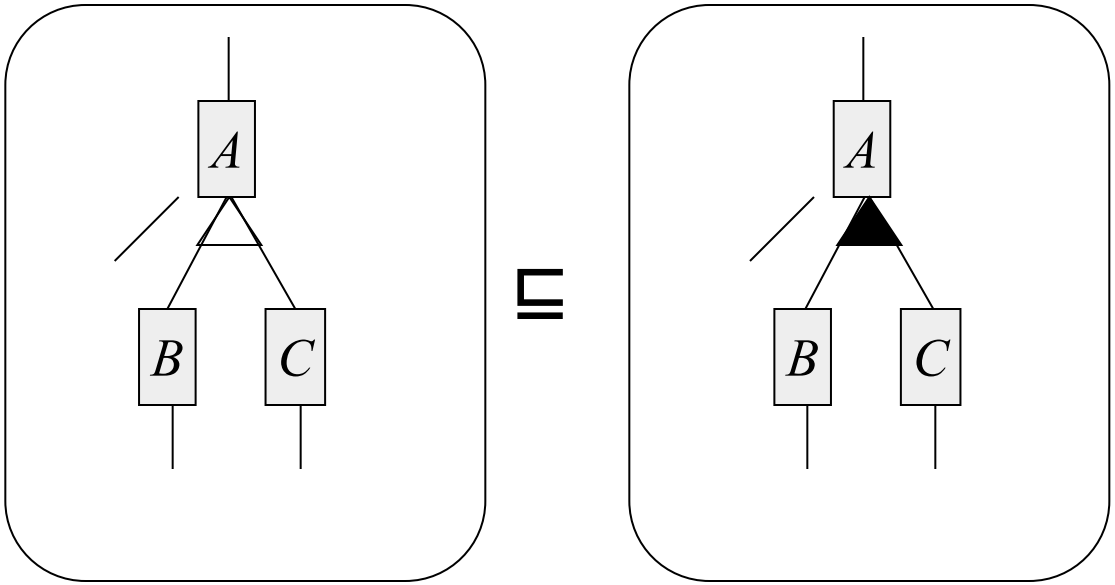
Valid configurations semantics

semantics(



) = *{A,B}*
{A,C}
…

Feature model refactorings as improved configurability



{A,B}
{A,C}
…

⊆

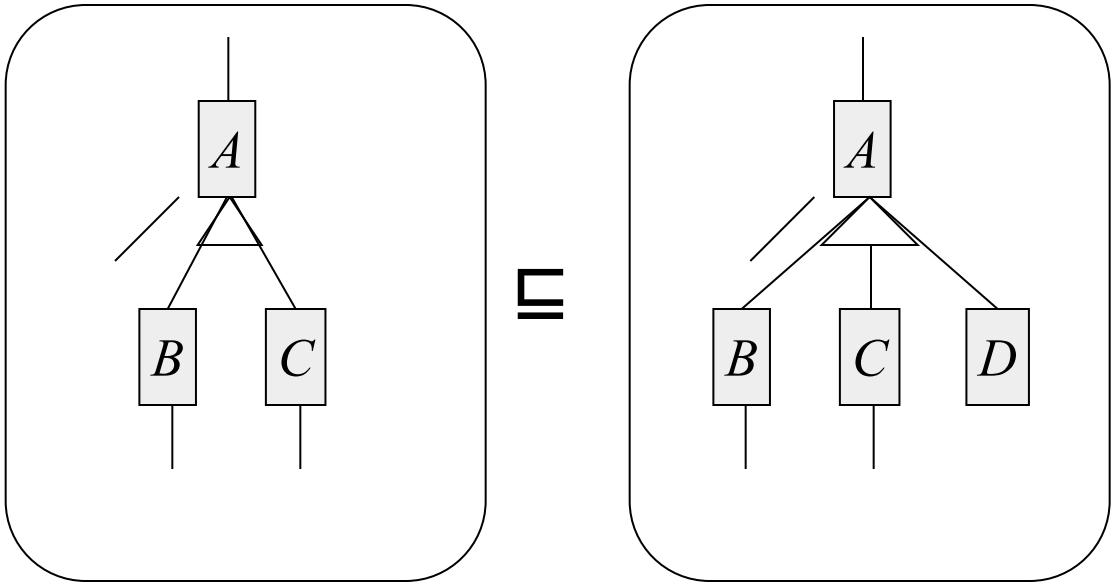
{A,B}
{A,C}
{A,B,C}
…

Feature models and configurability improvement

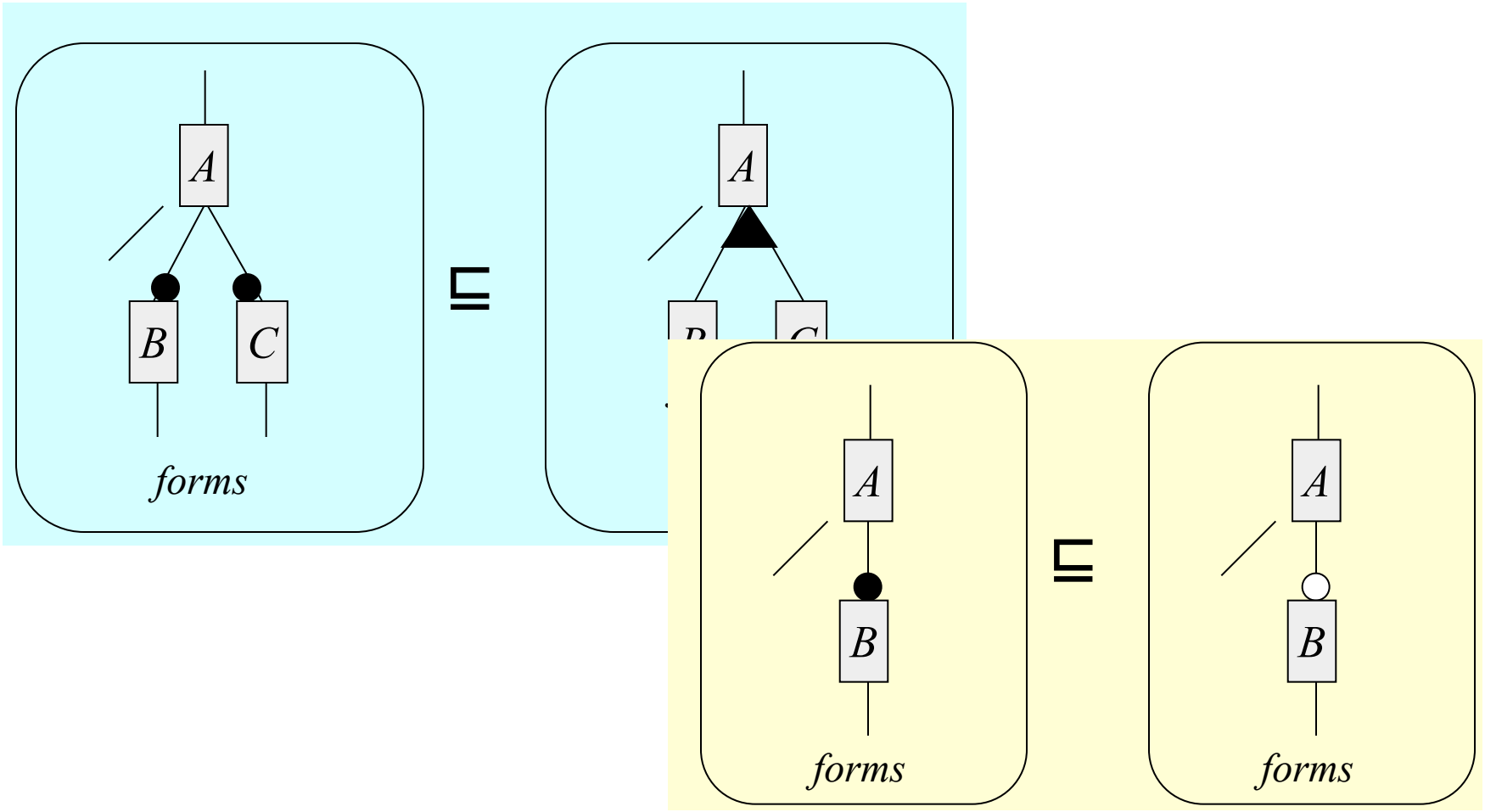
$$\textbf{\textit{semantics}}(fm) = \{c{:}Config \mid satConstraints(fm,c)\}$$

$$fm \sqsubseteq ifm = semantics(fm) \subseteq semantics(ifm)$$

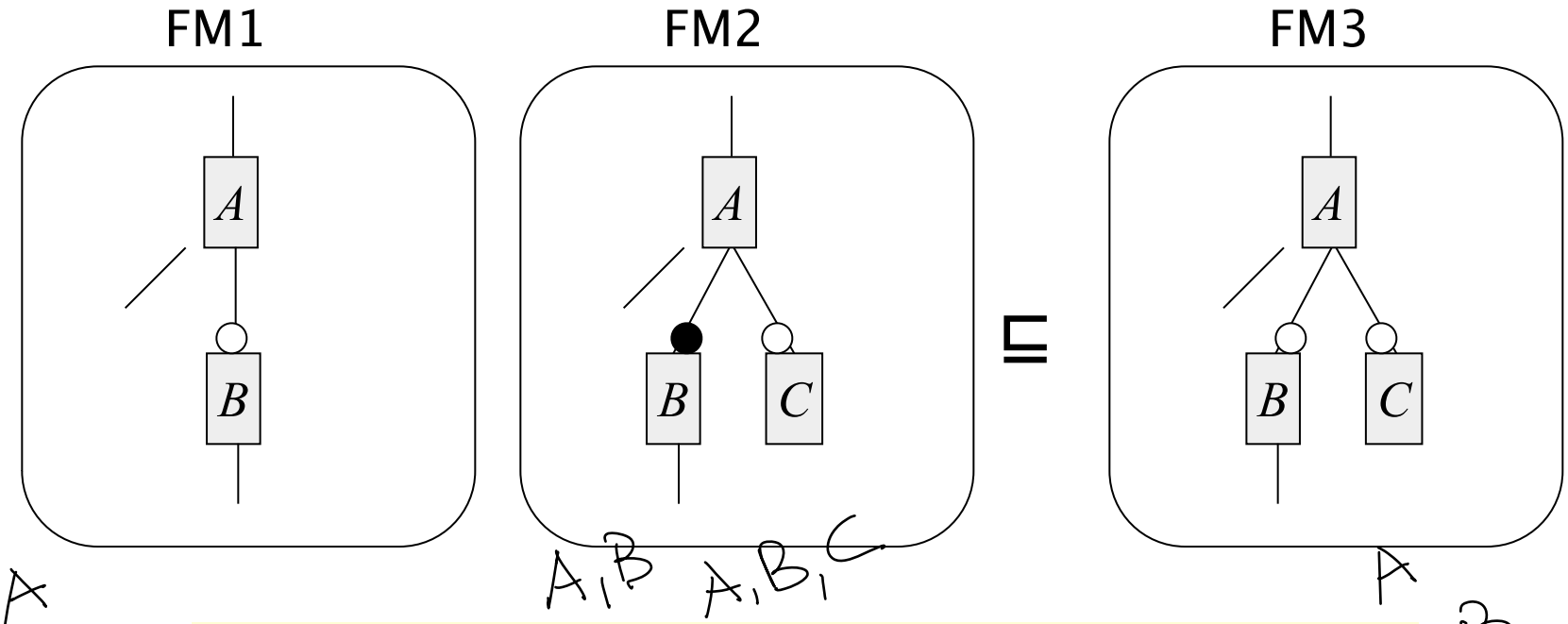
But no need to think about semantics directly: add alternative node



Refactoring catalog



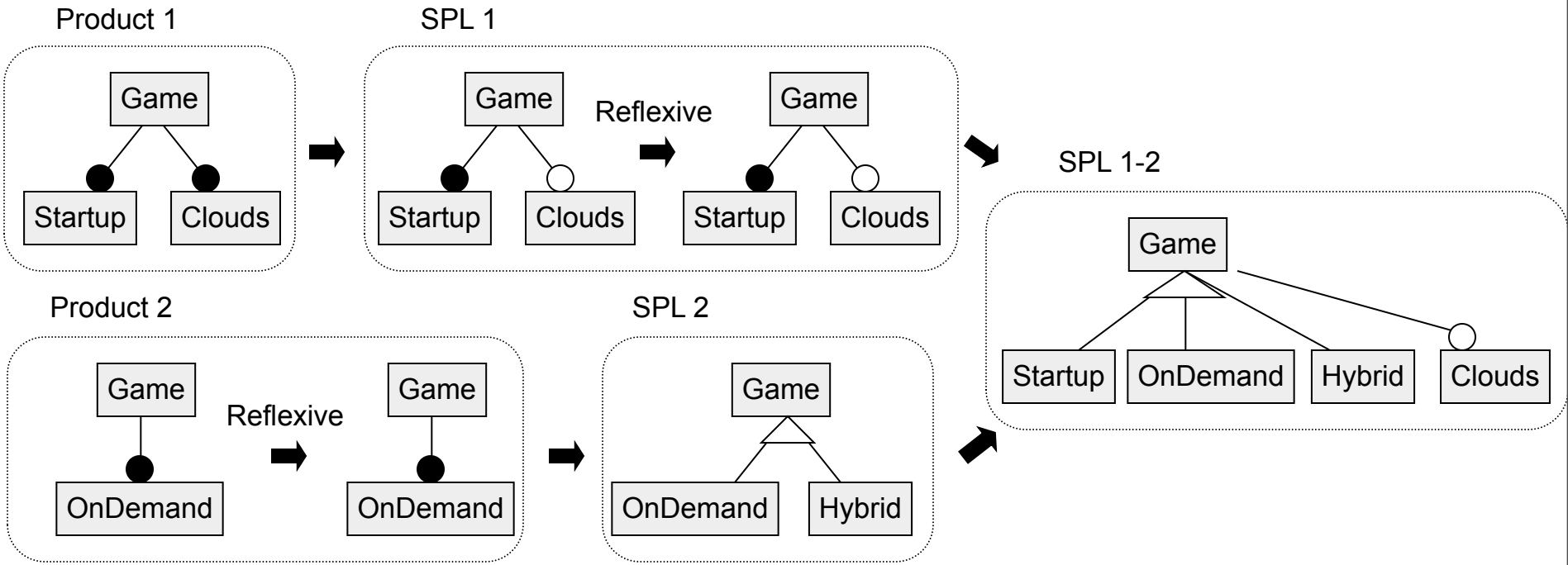
Refactoring populations and families: feature models



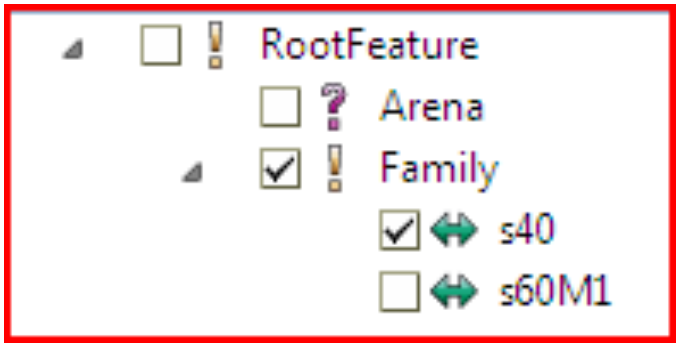
$$fm1\ fm2 \sqsubseteq fm =$$

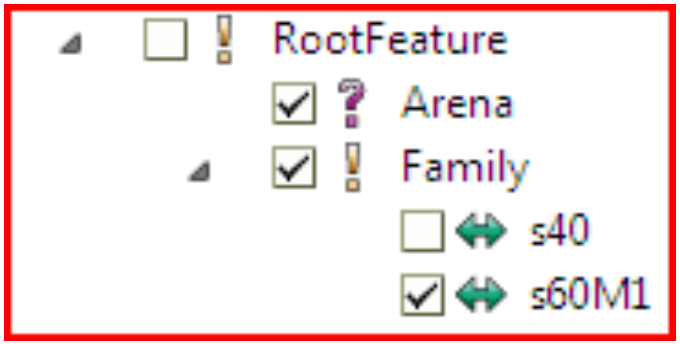
$$fm1 \sqsubseteq fm \wedge fm2 \sqsubseteq fm$$

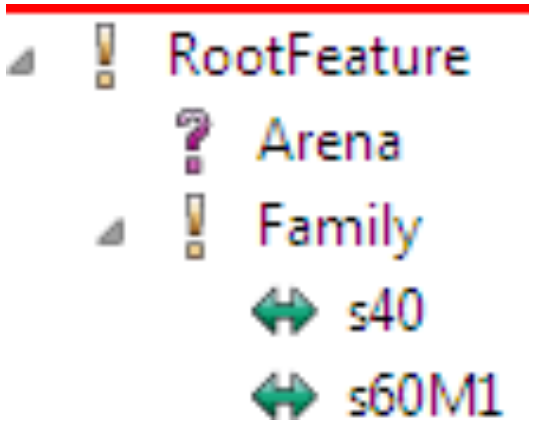
Case study: feature model refactoring



Instance models define products







Feature model instantiation might be…

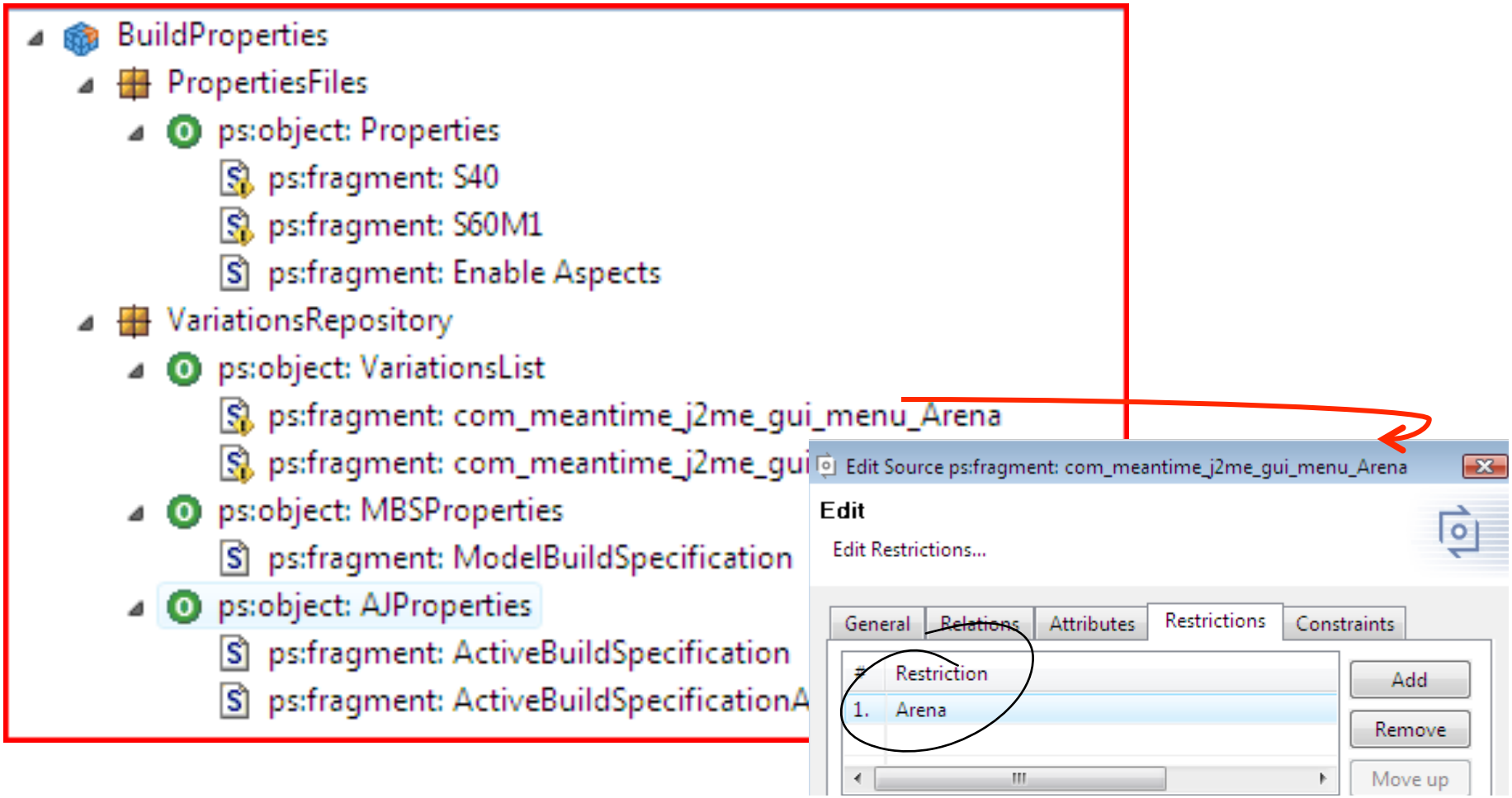
- Manual, defined by developers
- Automatic, inferred from context

Instance models as equations

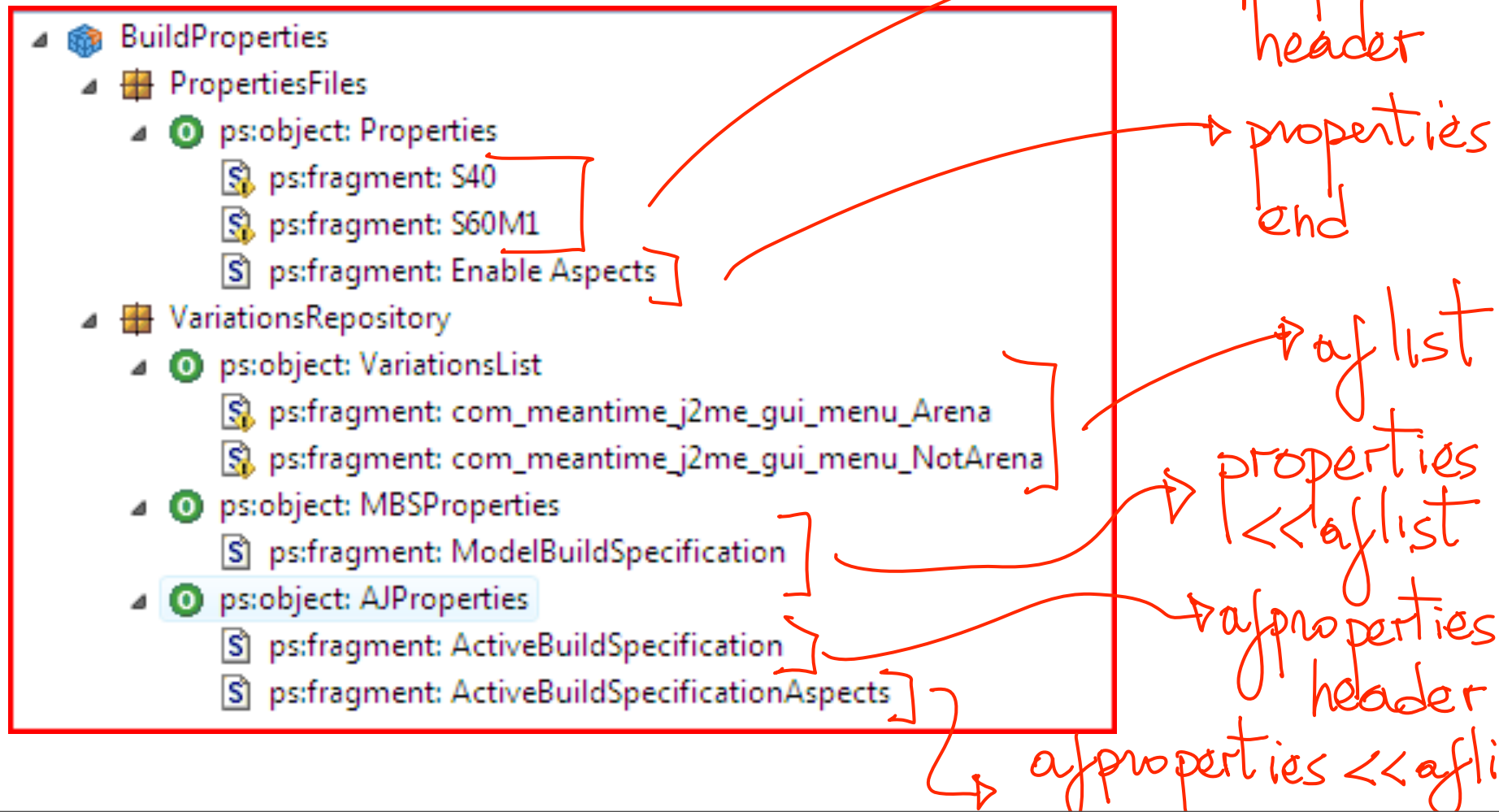
```
p1 = base + s40 + Screen128x128

p2 =    base + Arena + s60M1
      + Screen128x128

p3 = base + s60M1 + Screen128x160
```

Doesn't work for
feature interaction

Configuration knowledge builds products



In this case, property files...



that are used to build products

S40.properties

```
ppsymbols = device_screen_128x128, device_keys_nokia, device

wtk.cldc.version = 1.0
wtk.midp.version = 1.0

obfuscator.options = -dontusemixedcaseclassnames -overloadag

midlet.icon.dir = res/icons/16x16
midlet.res = res/texts/${build.lang} res/levels res/sounds/o
midlet.res.includes = lang*.txt *.ott *.bvg aS40.dat l.dat

midlet.jad.entriesfile = build/properties/S40.jad.entries

wtk.emulator.device = Nokia_7210_MIDP_SDK_v1_0
aspects.enabled = true
aspects.compiler = abc
aspects.list = src/com/meantime/j2me/gui/menu/NotArena.aj \
```

S40.ajproperties

```
src.includes = src/
src.inclusionpatterns = src/*.java,\
src/com/meantime/j2me/gui/menu/NotArena.aj \
```

Configuration knowledge essentially is...

a mapping from feature expressions to core assets

eShop configuration knowledge

Feature Expression	Transformations
eShop	select scenario SC01
	select scenario SC02
not (Shopping Cart and Bonus)	evaluate advice ADV01
Shopping Cart and Bonus	evaluate advice ADV02
Update User Preferences	evaluate advice ADV03
Shipping Method	bind SM to Shipping Method
...	...

without relationship between cart and bonus...

Feature Expression	Transformations
eShop	select scenario SC01
	select scenario SC02
not (Shopping Cart and Bonus)	evaluate advice ADV01
Shopping Cart and (not Bonus)	evaluate advice ADV–SC, ADV–notBonus
Shopping Cart and Bonus	evaluate advice ADV–SC, ADV–Bonus
Update User Preferences	evaluate advice ADV03
Shipping Method	bind SM to Shipping Method
...	...

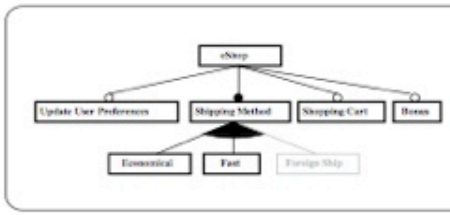
CK avoids

- scattering of configuration info
- core asset dependence on feature model
- feature model pollution, one-to-one mapping between features and components
- limitation to a single variability mechanism

SPL
Use Case Model

Feature Model

Product Configuration



Configuration Knowledge

Feature Expression	Transformations
eShop	select scenario SC01
	select scenario SC02
not (Shopping Cart and Bonus)	evaluate advice ADVO1
(ShoppingCart and Bonus)	evaluate advice ADV02
Update User Preferences	evaluate advice ADV03
Shipping Method	bind SM to Shipping Method
...	...

Weaving Process

Buy Goods

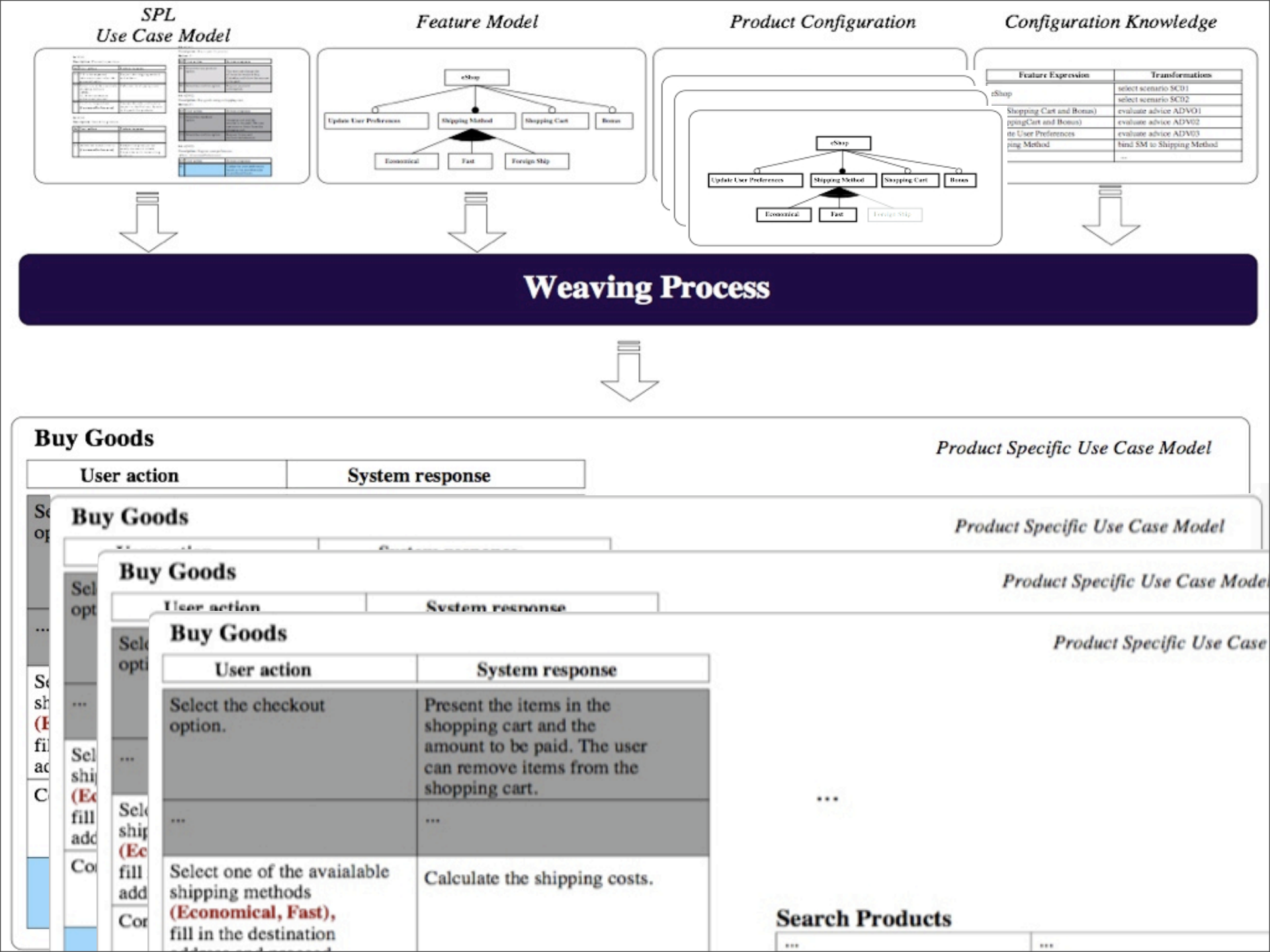
Product Specific Use Case Model

User action	System response
Select the checkout option.	Present the items in the shopping cart and the amount to be paid. The user can remove items from the shopping cart.
...	...
Select one of the avaialable shipping methods **(Economical, Fast),** fill in the destination address and proceed.	Calculate the shipping costs.
Confirm the purchase.	Execute the order and send a request to the Delivery System to dispatch the products.
	Update the user preferences based on the search results or purchased items.

...

Search Products

...	...
...	...
...	...
...	...
...	...
	Update the user preferences based on the search results or purchased items.


SPL
Use Case Model
Feature Model
Product Configuration
Configuration Knowledge
eShop
Update User Preferences
Shipping Method
Shopping Cart
Bonus
Economical
Fast
Foreign Ship
Feature Expression
Transformations
select scenario SC01
select scenario SC02
evaluate advice ADVO1
evaluate advice ADV02
evaluate advice ADV03
bind SM to Shipping Method
Weaving Process
Buy Goods
Product Specific Use Case Model
User action
System response
Select the checkout option.
Present the items in the shopping cart and the amount to be paid. The user can remove items from the shopping cart.
Select one of the avaialable shipping methods (Economical, Fast), fill in the destination
Calculate the shipping costs.
Search Products

Models for software product lines

Paulo Borba

Informatics Center

Federal University of Pernambuco