

Safe Composition of Configuration Knowledge- based Software Product Lines

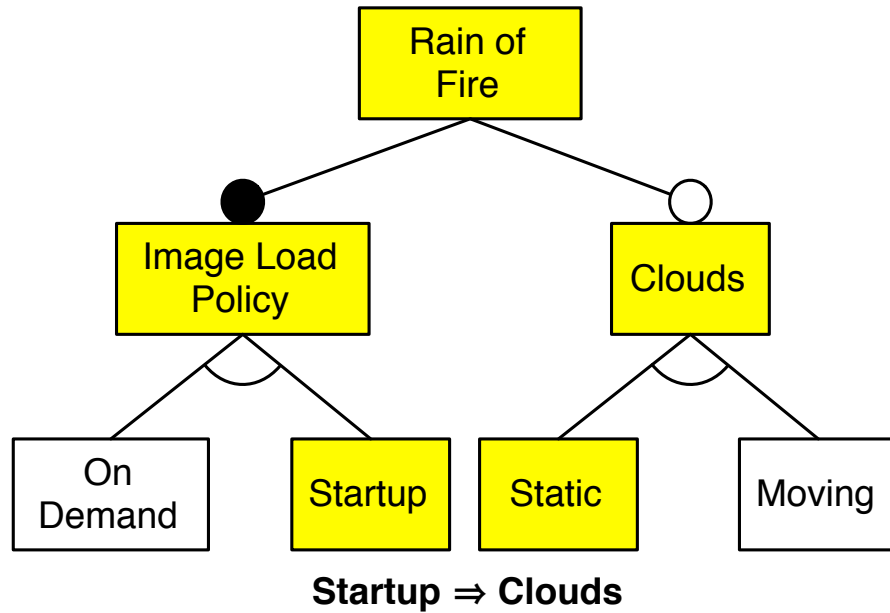
Leopoldo Motta Teixeira

Advisor: Paulo Borba

Co-Advisor: Rohit Gheyi

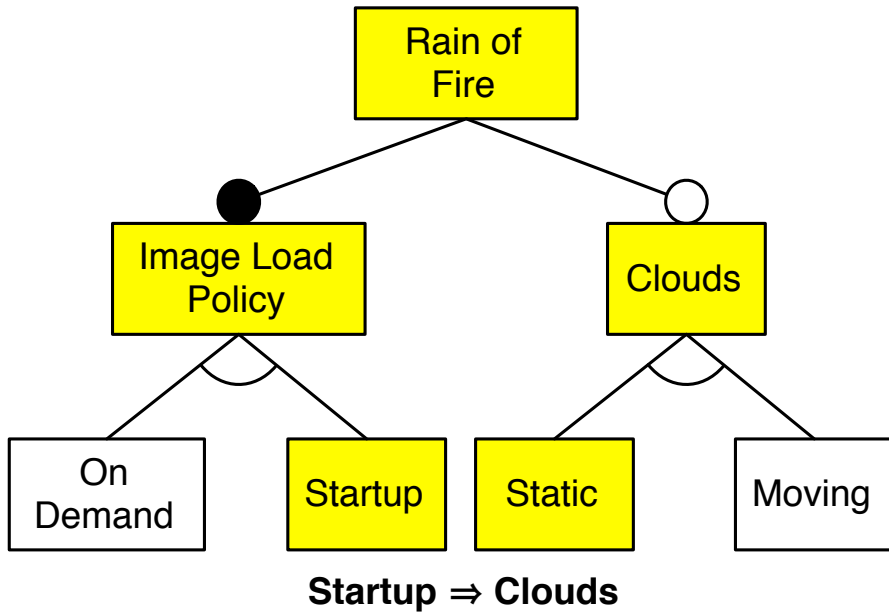


CK evaluation



| Feature Expression | Assets |
|--------------------------|----------------------------|
| Rain of Fire | Game.java, GameScreen.java |
| On demand \vee Startup | CommonImgLoad.java |
| On demand | OnDemand.aj |
| Startup | StartUp.aj |
| Clouds | Clouds.java |
| Static | CloudsStatic.aj |
| Moving | CloudsMotion.aj |

CK evaluation

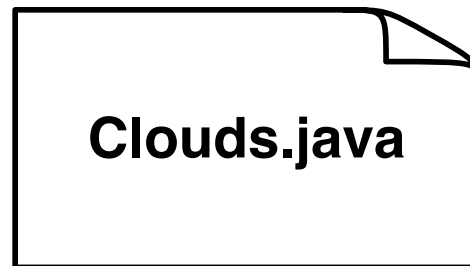


| Feature Expression | Assets |
|--------------------------|----------------------------|
| Rain of Fire | Game.java, GameScreen.java |
| On demand \vee Startup | CommonImgLoad.java |
| On demand | OnDemand.aj |
| Startup | StartUp.aj |
| Clouds | Clouds.java |
| Static | CloudsStatic.aj |
| Moving | CloudsMotion.aj |

↓ semanticsCK

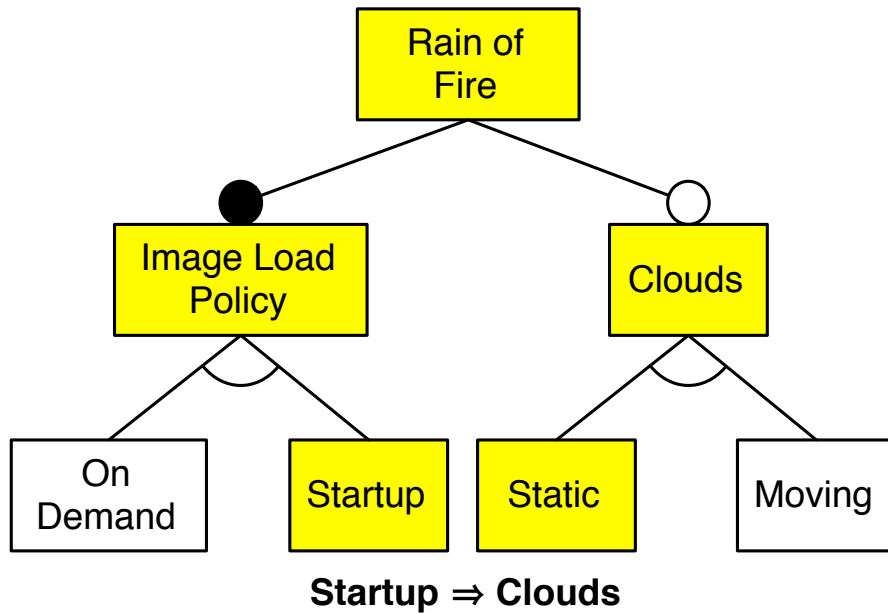


...



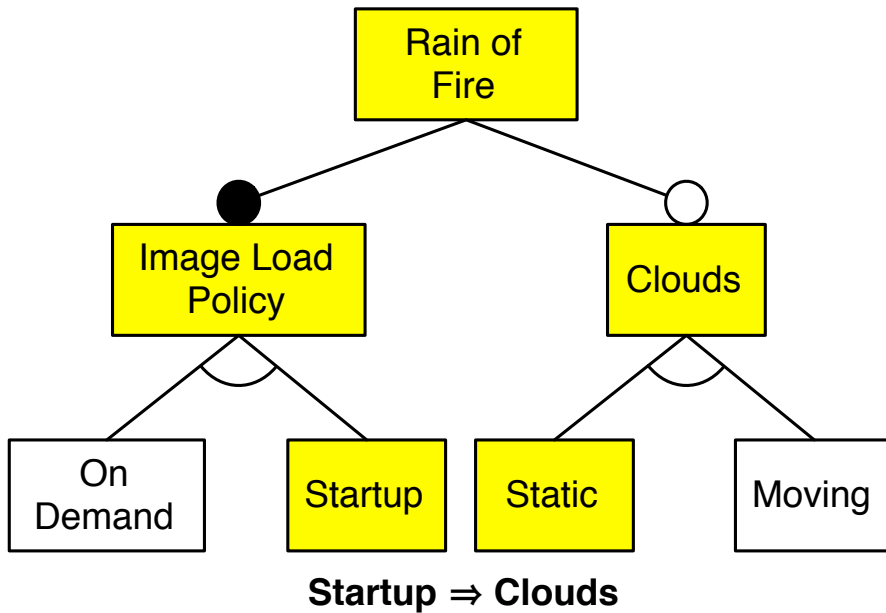
but, what if there are
problems in a CK?

missing entry, therefore, missing assets...



| Feature Expression | Assets |
|---------------------|----------------------------|
| Rain of Fire | Game.java, GameScreen.java |
| On demand ∨ Startup | CommonImgLoad.java |
| On demand | OnDemand.aj |
| Startup | StartUp.aj |
| Clouds | Clouds.java |
| Static | CloudsStatic.aj |
| Moving | CloudsMotion.aj |

missing entry, therefore, missing assets...



| Feature Expression | Assets |
|--------------------------|----------------------------|
| Rain of Fire | Game.java, GameScreen.java |
| On demand \vee Startup | CommonImgLoad.java |
| On demand | OnDemand.aj |
| Startup | StartUp.aj |
| Clouds | Clouds.java |
| Static | CloudsStatic.aj |
| Moving | CloudsMotion.aj |

↓ semanticsCK

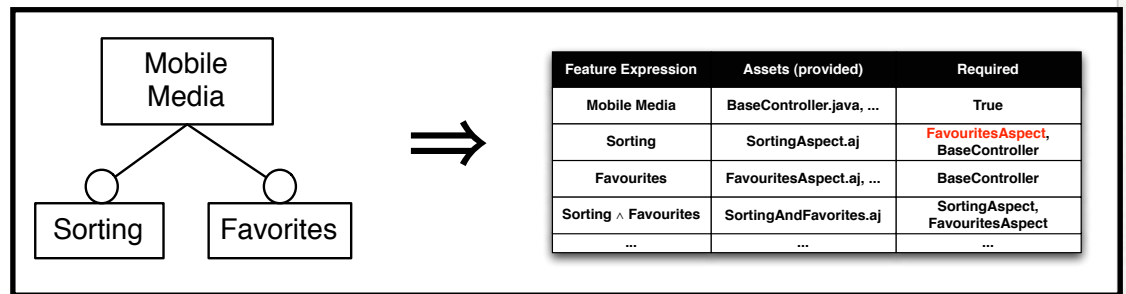


...



Proposed Solution

Approach for verification of SPLs to detect invalid products without synthesizing entire SPL

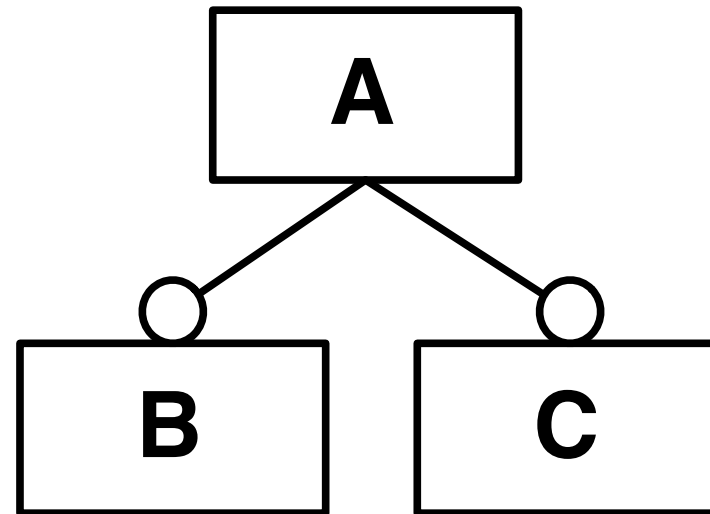


Explicit interfaces in the CK to enable verification

| Feature Expression | Assets | Provided | Required |
|--------------------------|----------------------------|------------------|---------------|
| Rain of Fire | Game.java, GameScreen.java | Game, GameScreen | -- |
| On demand \vee Startup | CommonImgLoad.java | CommonImgLoad | GameScreen |
| On demand | OnDemand.aj | OnDemand | CommonImgLoad |
| Startup | StartUp.aj | StartUp | CommonImgLoad |
| Clouds | Clouds.java | Clouds | GameScreen |
| Static | CloudsStatic.aj | CloudsStatic | Clouds |
| Moving | CloudsMotion.aj | CloudsMotion | Clouds |

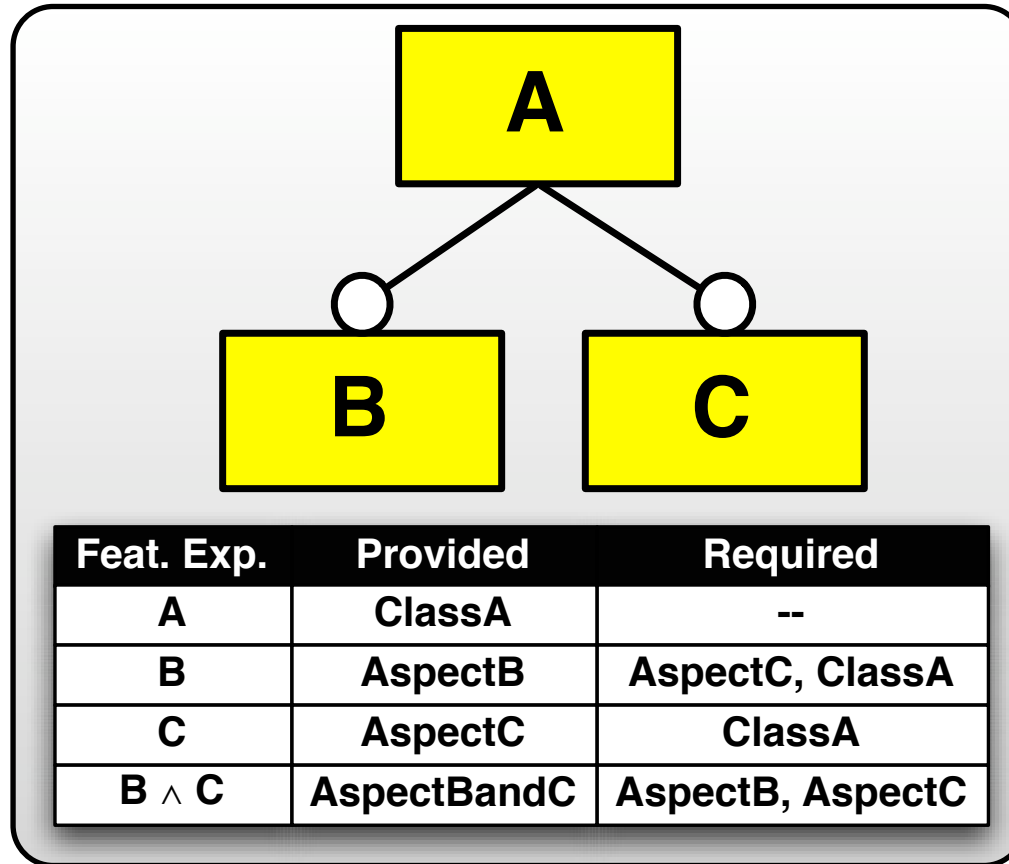
Extraction is not our focus

Toy Example

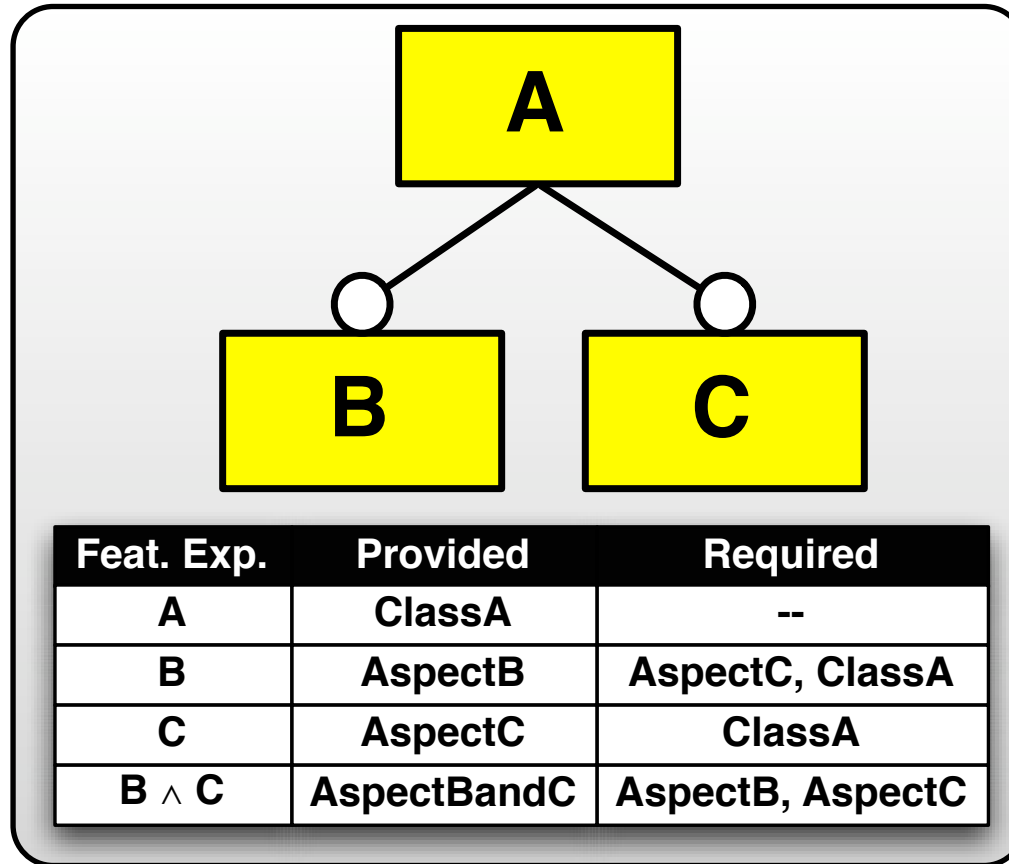


| Feature Expression | Assets (Provided) | Required |
|--------------------------------|--------------------|-------------------------|
| A | ClassA | -- |
| B | AspectB | AspectC, ClassA |
| C | AspectC | ClassA |
| B \wedge C | AspectBandC | AspectB, AspectC |

Valid Product

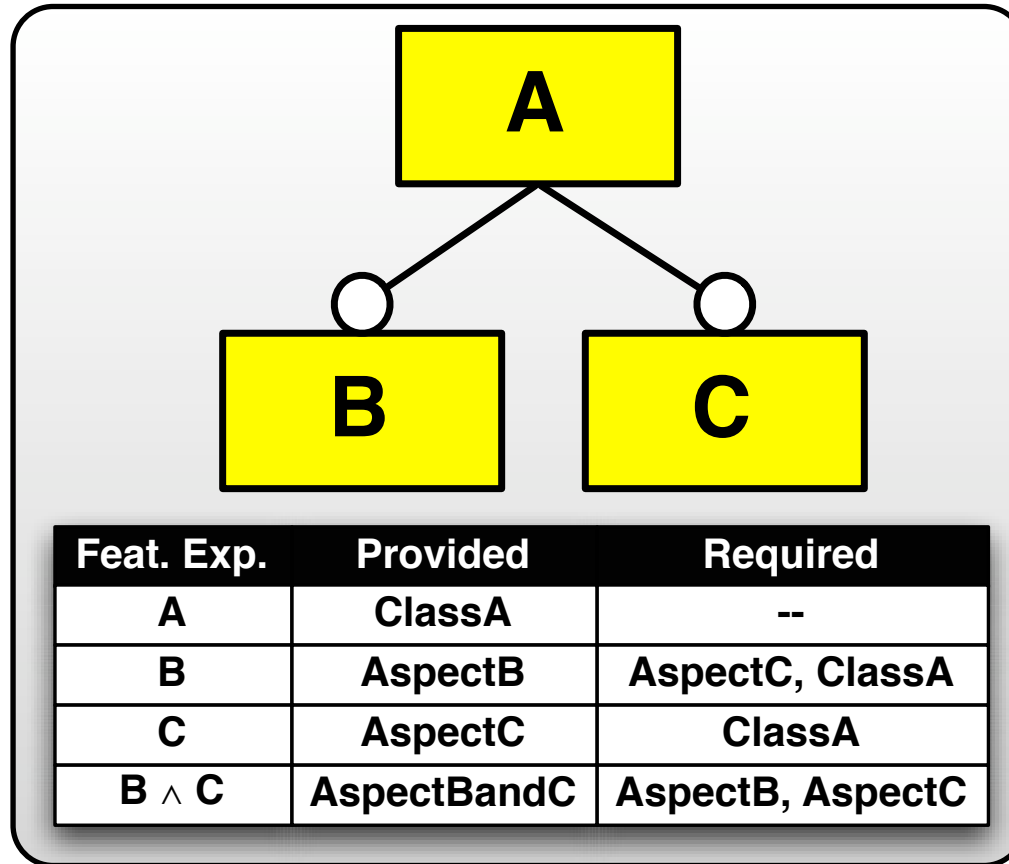


Valid Product



{ClassA, AspectB,
AspectC, AspectBandC}

Valid Product

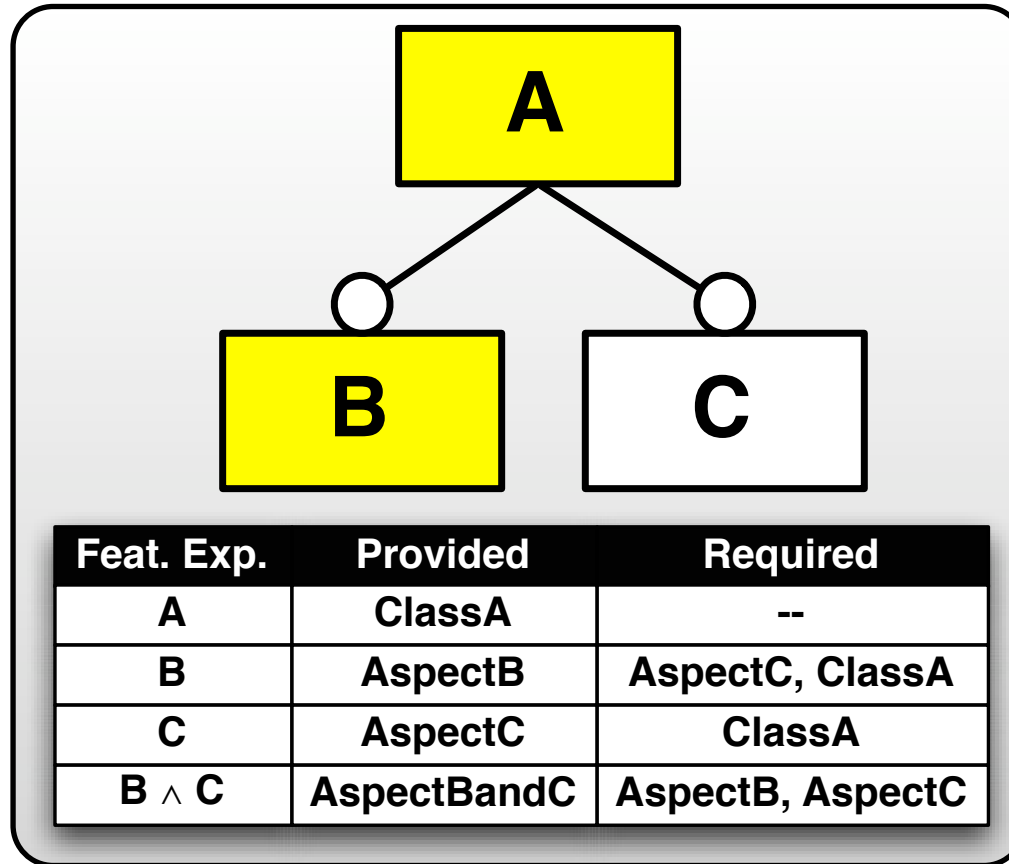


{ClassA, AspectB,
AspectC}

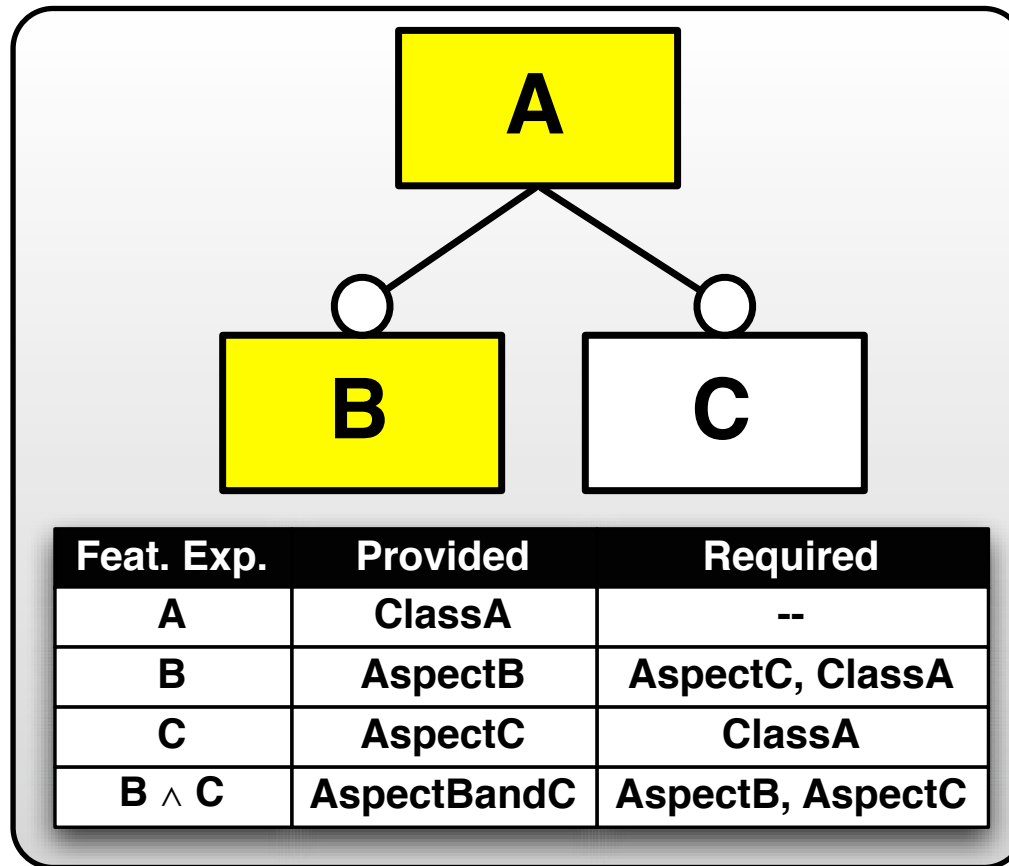
\subseteq

{ClassA, AspectB,
AspectC, AspectBandC}

Invalid Product

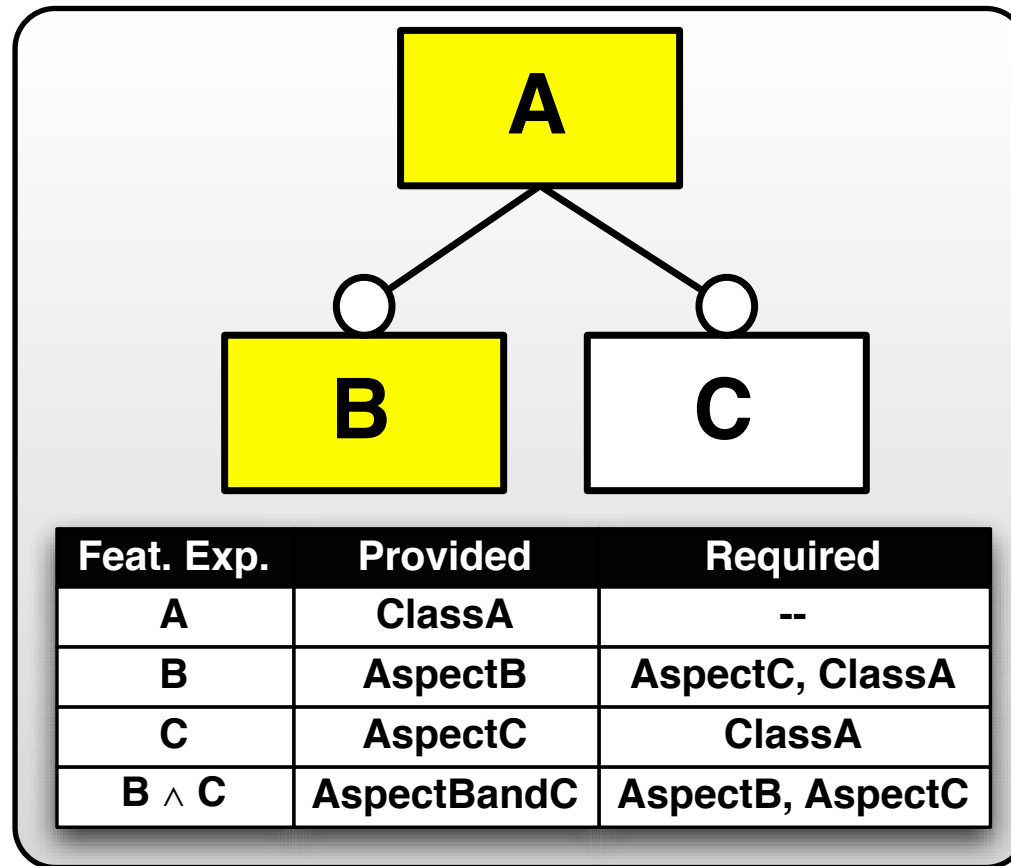


Invalid Product



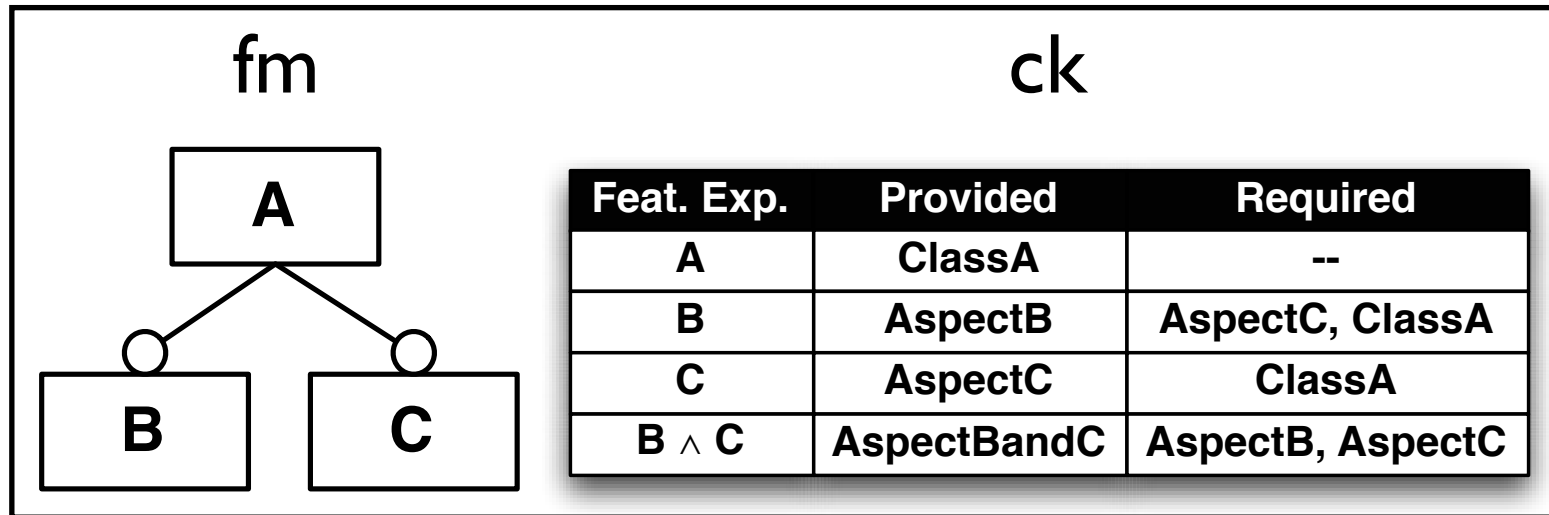
{ClassA, AspectB}

Invalid Product



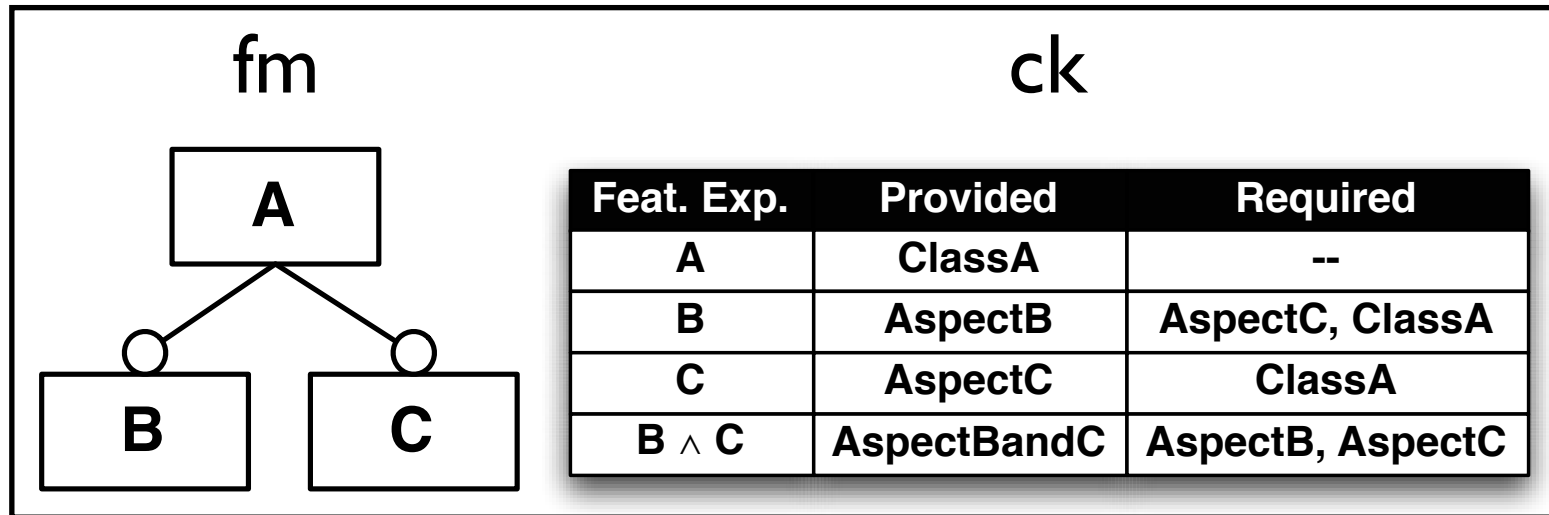
$\{\text{ClassA}, \text{AspectC}\} \not\subseteq \{\text{ClassA}, \text{AspectB}\}$

Valid SPL definition



$\forall \text{product:semantics(fm)} \mid$
 $\text{required}(\text{product}, \text{ck}) \subseteq \text{provided}(\text{product}, \text{ck})$

Valid SPL definition



$\forall \text{product:semantics(fm)} \mid$
 $\text{required}(\text{product}, \text{ck}) \subseteq \text{provided}(\text{product}, \text{ck})$

semanticsFM \Rightarrow constraintsCK

Implementation

- Specific instances of FM and CK in Alloy
- Verification using Alloy Analyzer
- Proof of concept using Java
 - Given a specification in Alloy, retrieves all invalid products if a SPL is invalid
 - Uses Alloy Analyzer API

Verification

```
assert verifyValidSPL {  
  semanticsFM[]  $\Rightarrow$  constraintsCK[]  
}  
check verifySPL for 2
```

Verification

```
assert verifyValidSPL {  
  semanticsFM[]  $\Rightarrow$  constraintsCK[]  
}  
check verifySPL for 2
```

False
(AspectBandC, AspectC, C)

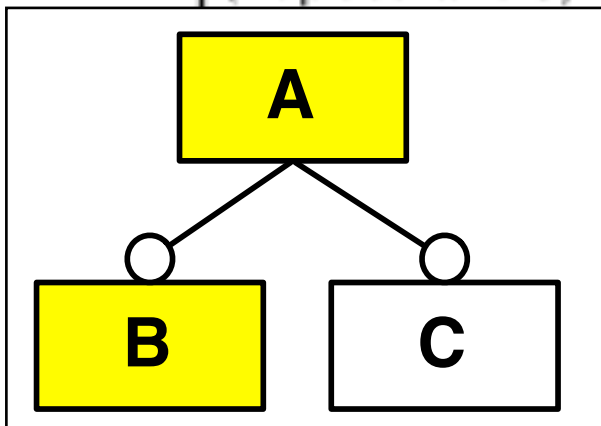
True
(A, AspectB, B, ClassA)

Verification

```
assert verifyValidSPL {  
  semanticsFM[]  $\Rightarrow$  constraintsCK[]  
}  
check verifySPL for 2
```

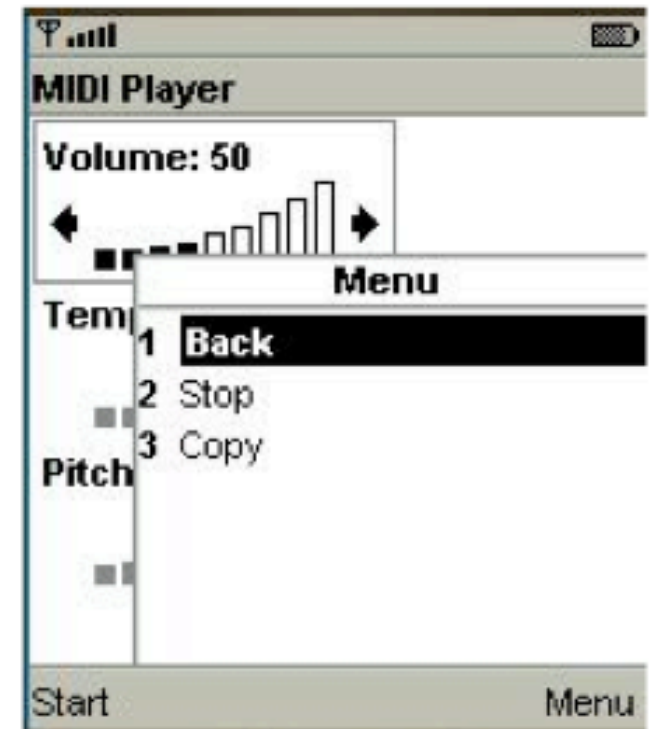
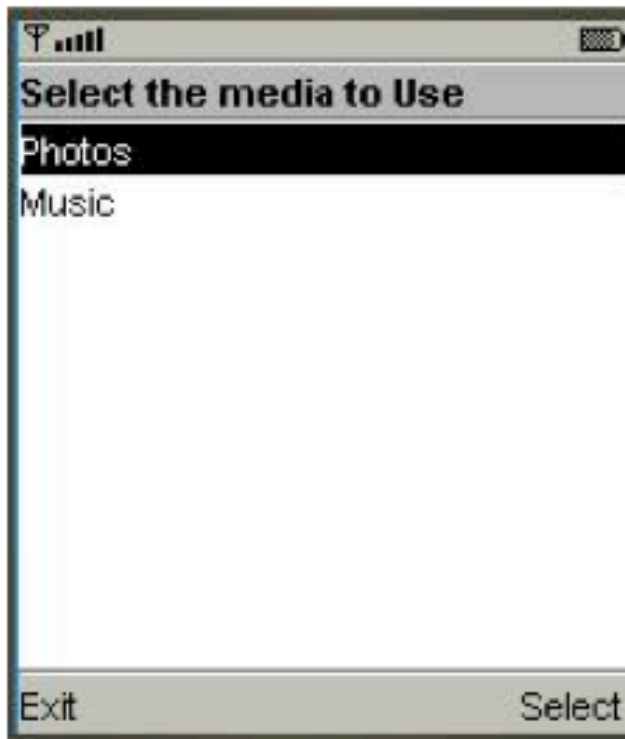
False
(AspectBandC, AspectC, C)

True
(A, AspectB, B, ClassA)

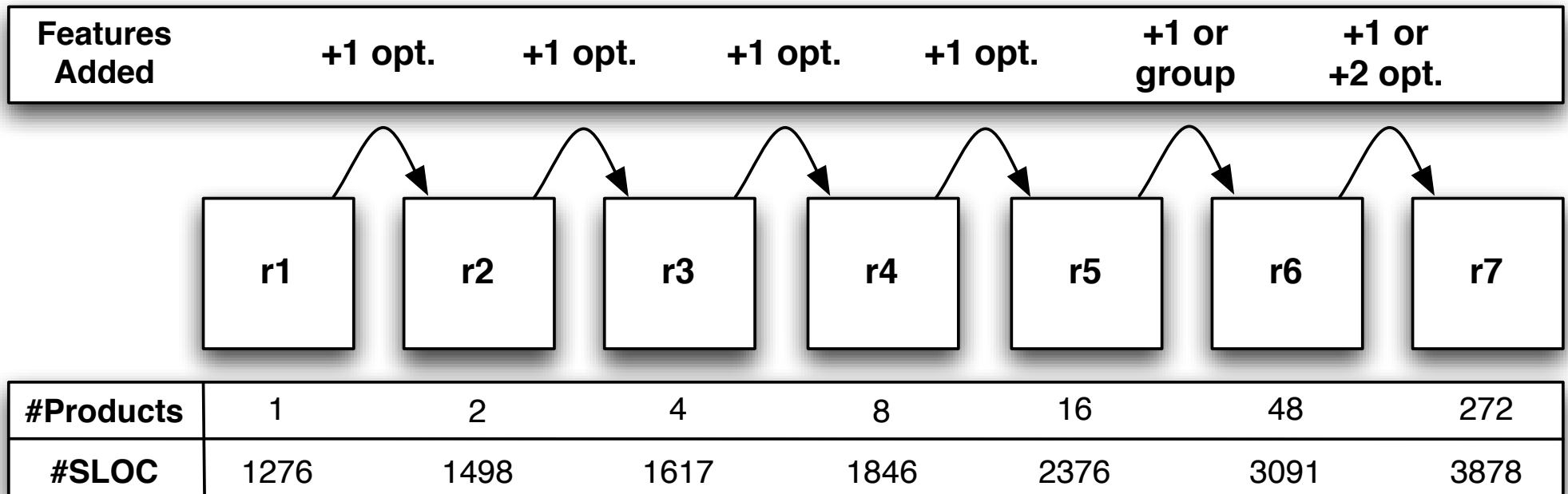


$\{ClassA, AspectC\} \not\subseteq \{ClassA, AspectB\}$

Evaluation: MobileMedia



Evolution of Releases



Provided/Required

*OptionalFeatureAspect.aj

```
1 package lancs.mobilemedia.optional;
2
3 import lancs.mobilemedia.optional.copy.CopyMultiMediaAspect;
4 import lancs.mobilemedia.optional.favourites.FavouritesAspect;
5 import lancs.mobilemedia.optional.favourites.PersisteFavoritesAspect;
6 import lancs.mobilemedia.optional.sms.SMSAspect;
7 import lancs.mobilemedia.optional.sorting.SortingAspect;
8
9 public aspect OptionalFeatureAspect {
10     declare precedence :
11         SMSAspect, CopyMultiMediaAspect, FavouritesAspect,
12         SortingAspect, PersisteFavoritesAspect;
13 }
```


Provided/Required

*OptionalFeatureAspect.aj

```
1 package lancs.mobilemedia.optional;
2
3 import lancs.mobilemedia.optional.copy.CopyMultiMediaAspect;
4 import lancs.mobilemedia.optional.favourites.FavouritesAspect;
5 import lancs.mobilemedia.optional.favourites.PersisteFavoritesAspect;
6 import lancs.mobilemedia.optional.sms.SMSAspect;
7 import lancs.mobilemedia.optional.sorting.SortingAspect;
8
9 public aspect OptionalFeatureAspect { provided
10     declare precedence :
11         SMSAspect, CopyMultiMediaAspect, FavouritesAspect,
12         SortingAspect, PersisteFavoritesAspect;
13 }
```

Provided/Required

*OptionalFeatureAspect.aj

```
1 package lancs.mobilemedia.optional;
2
3 import lancs.mobilemedia.optional.copy.CopyMultiMediaAspect;
4 import lancs.mobilemedia.optional.favourites.FavouritesAspect;
5 import lancs.mobilemedia.optional.favourites.PersisteFavoritesAspect;
6 import lancs.mobilemedia.optional.sms.SMSAspect;
7 import lancs.mobilemedia.optional.sorting.SortingAspect;
8
9 public aspect OptionalFeatureAspect { provided
10     declare precedence :
11         SMSAspect, CopyMultiMediaAspect, FavouritesAspect,
12         SortingAspect, PersisteFavoritesAspect;
13 }
```

Provided/Required

*OptionalFeatureAspect.aj

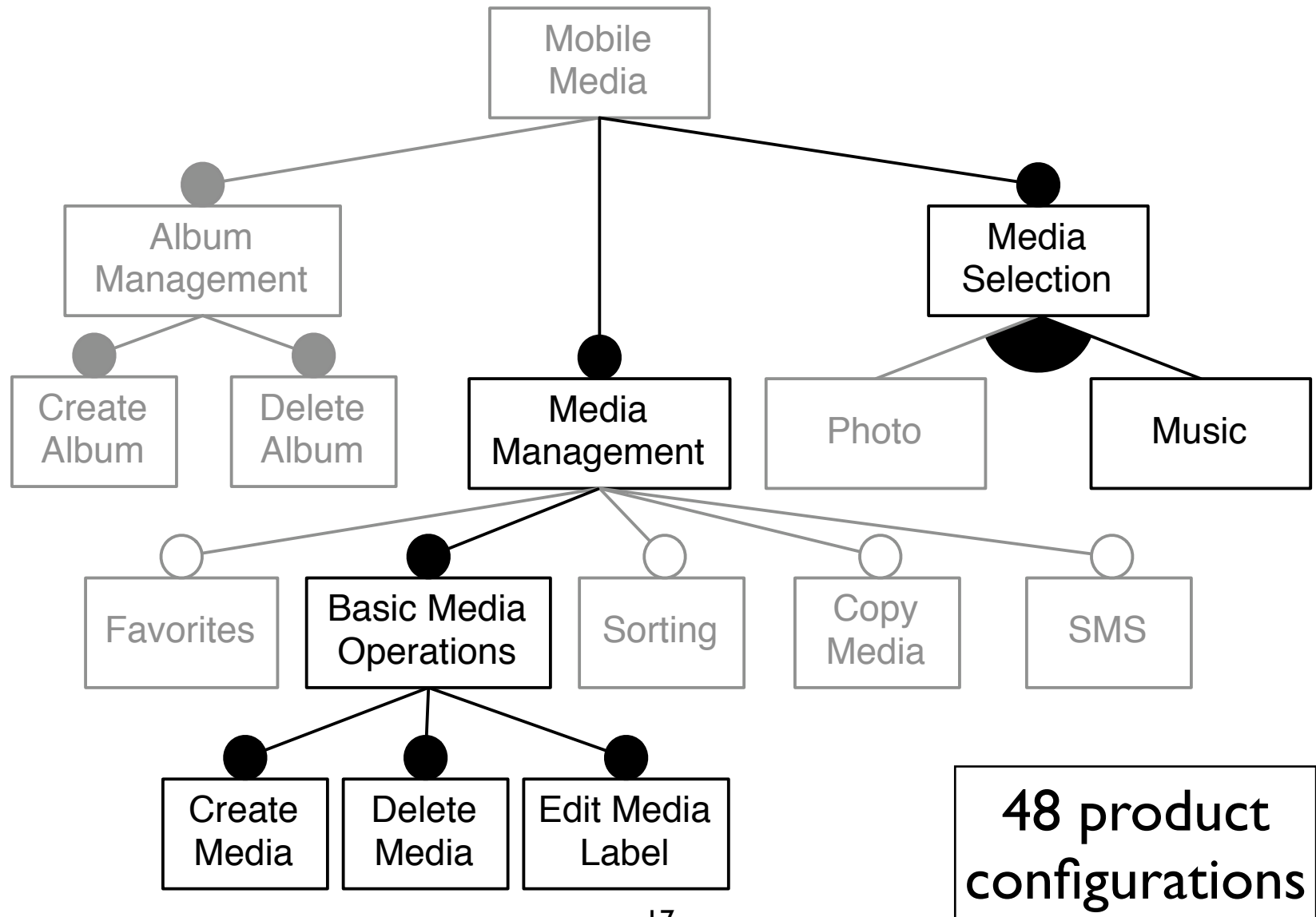
```
1 package lancs.mobilemedia.optional;
2
3 import lancs.mobilemedia.optional.copy.CopyMultiMediaAspect;
4 import lancs.mobilemedia.optional.favourites.FavouritesAspect;
5 import lancs.mobilemedia.optional.favourites.PersisteFavoritesAspect;
6 import lancs.mobilemedia.optional.sms.SMSAspect;
7 import lancs.mobilemedia.optional.sorting.SortingAspect;
8
9 public aspect OptionalFeatureAspect { provided
10     declare precedence :
11         SMSAspect, CopyMultiMediaAspect, FavouritesAspect,
12         SortingAspect, PersisteFavoritesAspect;
13 }
```

required

Releases 1-5

- Simpler releases
 - No change on media type (all deal with Photo)
- r2: 2 products --- r5: 16 products
- No problems found

Release 6



48 product configurations

SMS

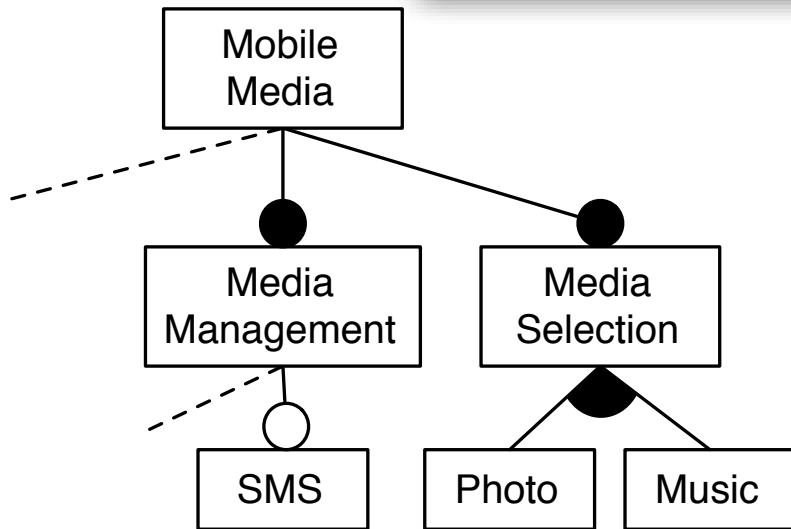
| Feature Expression | Provided | Required |
|--------------------|---|--|
| ... | ... | ... |
| SMS | SMSAspect | ..., ImageMediaAccessor , PhotoViewScreen , ... |
| Photo | ImageMediaAccessor, PhotoViewScreen, ... | MainUIMidlet, AbstractController, ... |
| ... | ... | ... |

MobileMedia01.lst
MobileMedia02a.lst
MobileMedia04.lst

SMS

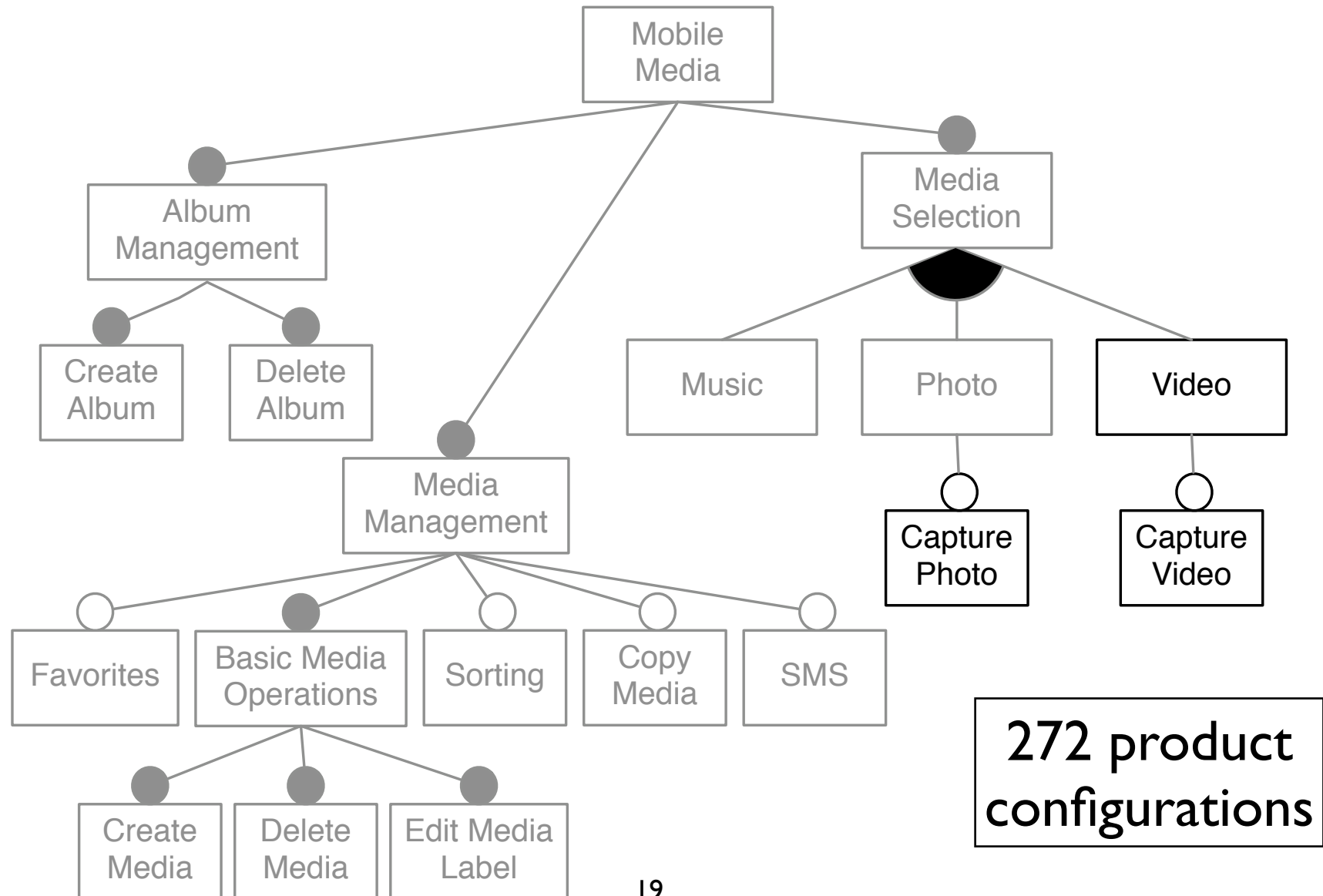
MobileMedia01.lst
 MobileMedia02a.lst
 MobileMedia04.lst

| Feature Expression | Provided | Required |
|--------------------|---|--|
| ... | ... | ... |
| SMS | SMSAspect | ..., ImageMediaAccessor , PhotoViewScreen , ... |
| Photo | ImageMediaAccessor, PhotoViewScreen, ... | MainUIMidlet, AbstractController, ... |
| ... | ... | ... |



but, feature model allows
 SMS without Photo...

Release 7



CapturePhoto

| Feature Expression | Provided | Required |
|--------------------------|---------------------|--|
| ... | ... | ... |
| Capture Photo | CapturePhotoAspect | ..., PhotoViewController , ... |
| Copy ∨ SMS | PhotoViewController | MainUIMidlet, AbstractController, ... |
| ... | ... | ... |

MobileMediaA03.lst
MobileMediaABC03.lst

CapturePhoto

| Feature Expression | Provided | Required |
|--------------------------|---------------------|--|
| ... | ... | ... |
| Capture Photo | CapturePhotoAspect | ..., PhotoViewController , ... |
| Copy ∨ SMS | PhotoViewController | MainUIMidlet, AbstractController, ... |
| ... | ... | ... |

MobileMediaA03.lst
MobileMediaABC03.lst



| Feature Expression | Provided | Required |
|--|---------------------|--|
| ... | ... | ... |
| Capture Photo | CapturePhotoAspect | ..., PhotoViewController, ... |
| Capture Photo ∨ Copy ∨ SMS | PhotoViewController | MainUIMidlet, AbstractController, ... |
| ... | ... | ... |

Photo \wedge Music \wedge Video

| Feature Expression | Provided | Required |
|---|-----------------------|---|
| ... | ... | ... |
| Photo \wedge Music \wedge Video | PhotoAndMusicAndVideo | OptionalFeatureAspect, PhotoAspect, ... |
| Sorting \wedge Favorites \wedge Copy \wedge SMS | OptionalFeatureAspect | CopyMultiMediaAspect, FavouritesAspect, ... |
| ... | ... | ... |

MobileMediaABC03.lst

Photo ^ Music ^ Video

| Feature Expression | Provided | Required |
|---|-----------------------|--|
| ... | ... | ... |
| Photo ^ Music ^ Video | PhotoAndMusicAndVideo | OptionalFeatureAspect , PhotoAspect, ... |
| Sorting ^ Favorites ^ Copy ^ SMS | OptionalFeatureAspect | CopyMultiMediaAspect, FavouritesAspect, ... |
| ... | ... | ... |

MobileMediaABC03.lst

PhotoAndMusicAndVideo.aj

```
...  
public aspect PhotoAndMusicAndVideo {  
  declare precedence :  
  MusicAspect, VideoAspect, PhotoAspect, ..., OptionalFeatureAspect;  
  ...  
}
```

Summary

| | # products | # invalid | time (s) |
|-----------|------------|------------|----------|
| release 2 | 2 | 0 | 0.7 |
| release 3 | 4 | 0 | 0.95 |
| release 4 | 8 | 0 | 1.09 |
| release 5 | 16 | 0 | 1.21 |
| release 6 | 48 | 8 | 4.12 |
| release 7 | 272 | 116 | 33.81 |

Final remarks

- It is beneficial to use a verification approach such as the proposed
- Time for verification **vs.** time for compiling a single product
- A theory for checking specific instances of FMs and CKs using Alloy

Opportunities

- Apply our verification approach with other SPLs
- Implement verification for other CK approaches

Safe Composition of Configuration Knowledge- based Software Product Lines

Leopoldo Motta Teixeira

Advisor: Paulo Borba

Co-Advisor: Rohit Gheyi

