

SPL Refinement

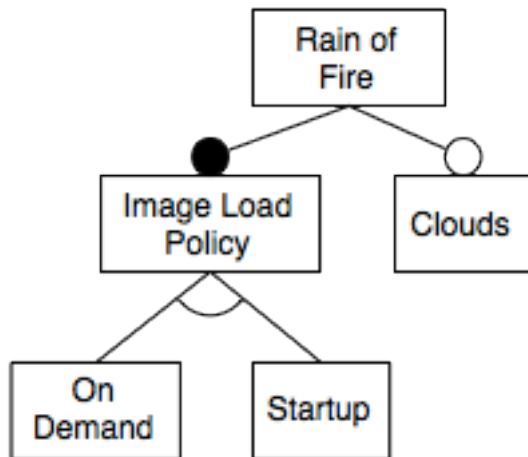
Leopoldo Motta Teixeira

Advisor: Paulo Borba
Co-Advisor: Rohit Gheyi



A PL is a tuple (F, A, K)

F



A

```

Main 1  ⇨  class Main {
           ...new StartUp(...);...
           }

Main 2  ⇨  class Main {
           ...new OnDemand(...);...
           }

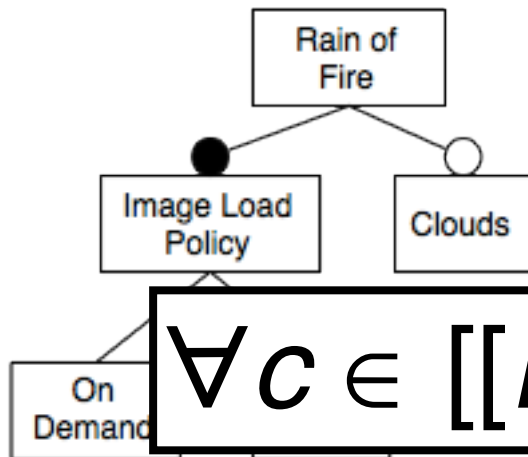
           :
  
```

K

Feature Expression	Assets
Rain of Fire	Game, GameScreen
On demand \vee Startup	CommonImgLoad
On demand	Main 2
Startup	Main 1
Clouds	Clouds

A PL is a tuple (F, A, K)

F



A

```

class Main {
Main 1  → ...new StartUp(...);...
}

class Main {
Main 2  → ...new OnDemand(...);...
}
  
```

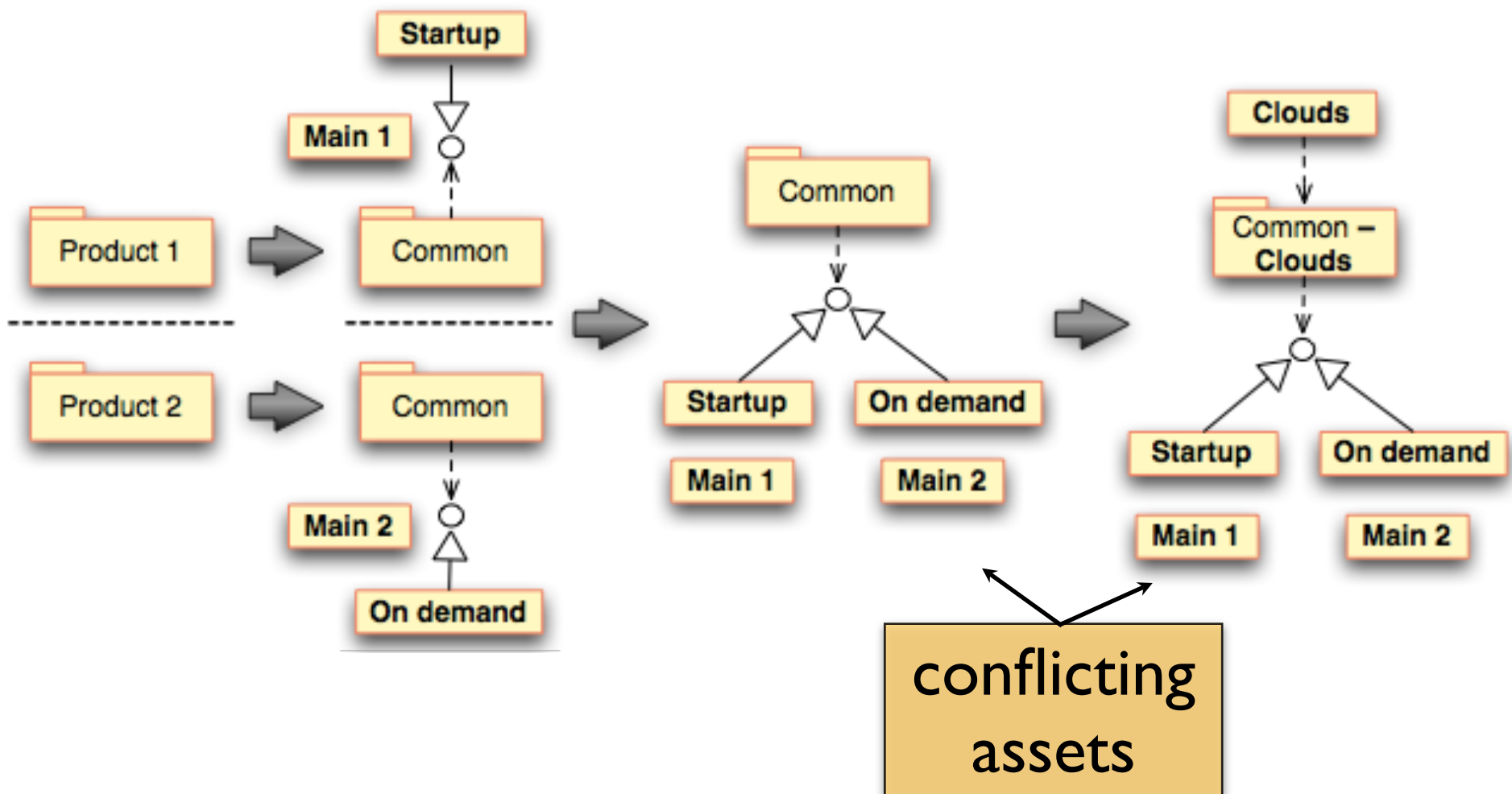
$$\forall C \in [[F]] \cdot wf([[K]]_C^A)$$

K

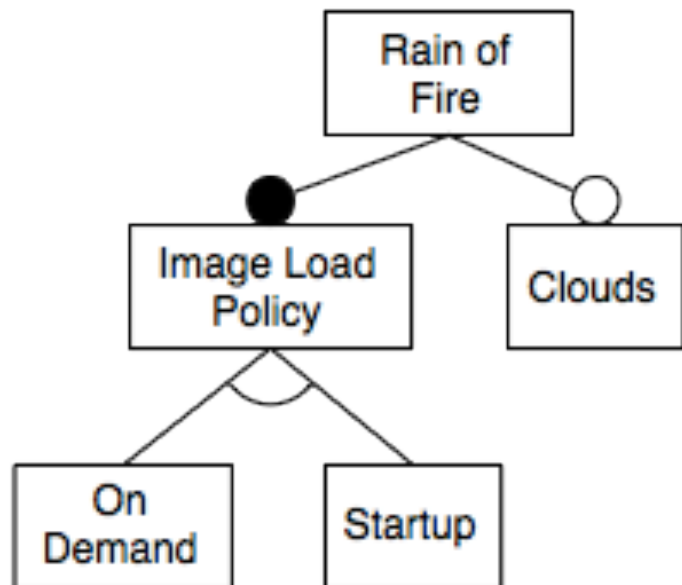
Feature Expression	Assets
Rain of Fire	Game, GameScreen
On demand \vee Startup	CommonImgLoad
On demand	Main 2
Startup	Main 1
Clouds	Clouds

**How can we safely
evolve product lines?**

Program refactoring is not enough!



and does not deal with specific PL artifacts



Feature Expression	Assets
Rain of Fire	Game, GameScreen
On demand \vee Startup	CommonImgLoad
On demand	Main 2
Startup	Main 1
Clouds	Clouds

```
Main 1    class Main {
           ...new StartUp(...);...
           }

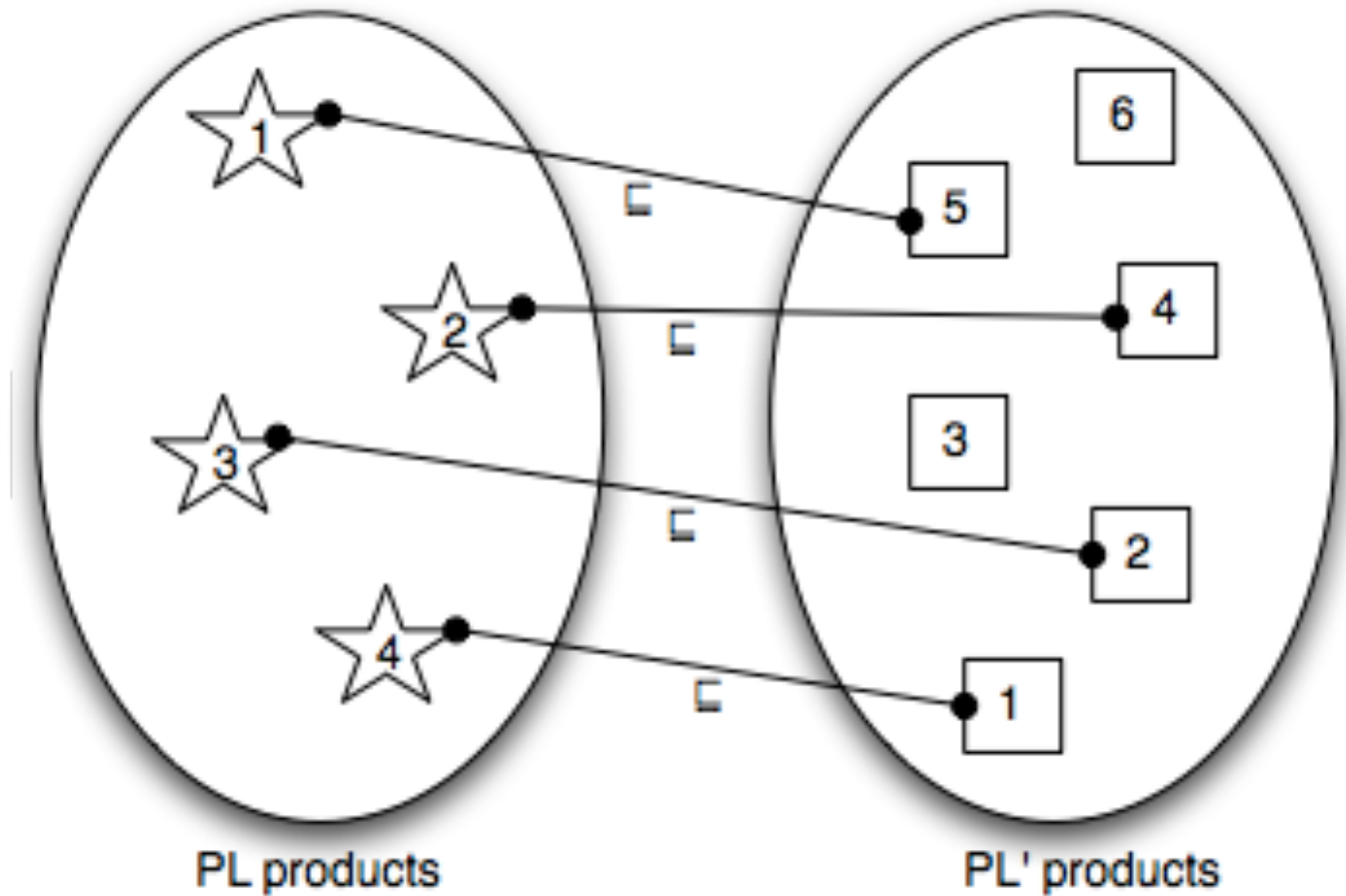
Main 2    class Main {
           ...new OnDemand(...);...
           }

           :
```

PL refinement...

- notion
- templates (catalogue)
- checkers

Product Line Refinement



Formalizing

$$PL1 \sqsubseteq PL2$$

whenever

\forall product \in PL1 .

\exists product' \in PL2 .

product \sqsubseteq product'

General Theory



F



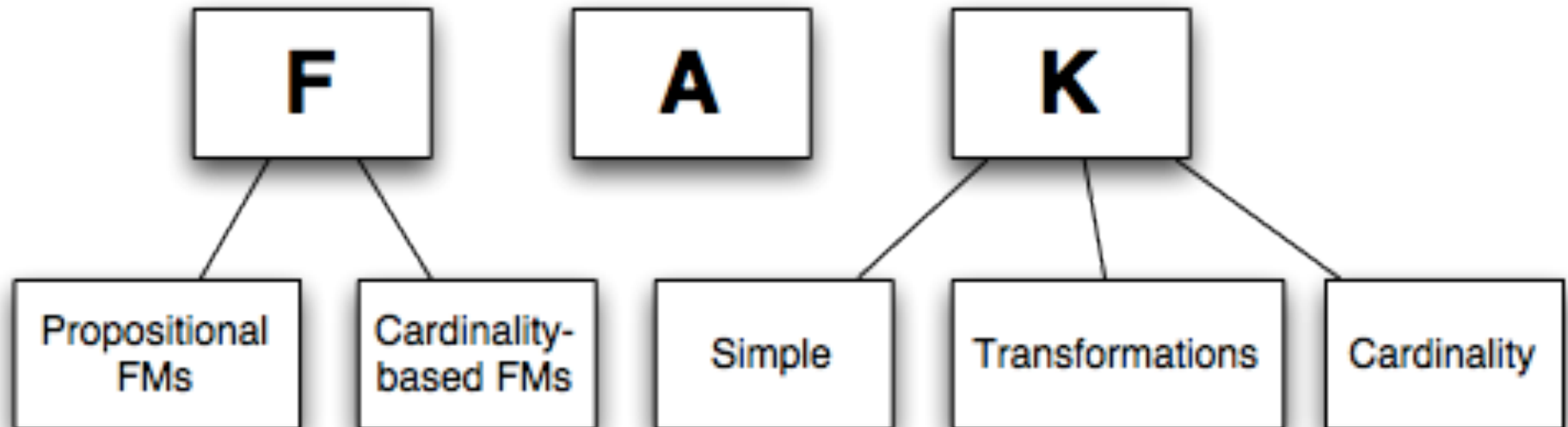
A



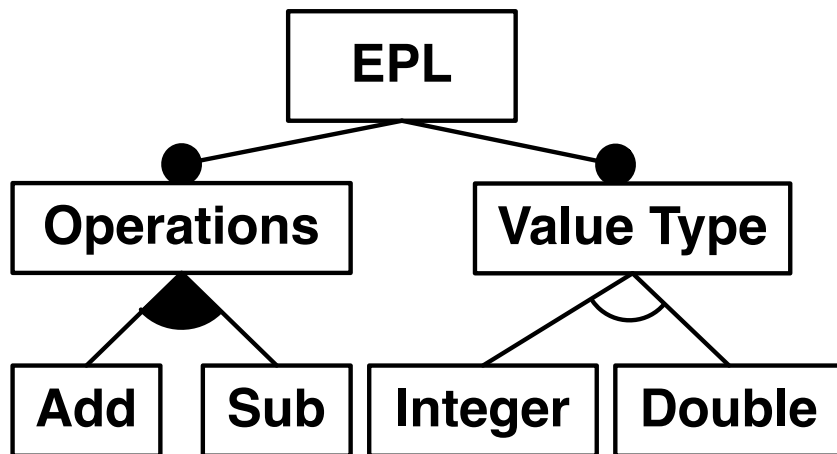
K

Interfaces for these languages

Current Status



Propositional FMs and Simple CK



Feature Expression	Assets
EPL	Program.java, Expression.java
$\text{Integer} \vee \text{Double}$	Value.java
Integer	IntegerValue.java
Double	DoubleValue.java, DoubleTypeAspect.aj
$\text{Add} \vee \text{Sub}$	BinaryExp.java
Add	AddExp.java
$\text{Add} \wedge \text{Double}$	AddDoubleTypeAspect.aj
Sub	SubExp.java
$\text{Sub} \wedge \text{Double}$	SubDoubleTypeAspect.aj

Templates

- Derived from TaRGeT SVN History Analysis
- Expressiveness analysis (Laís)
- Soundness proof

$$F \left\{ \begin{array}{c} n \mapsto a \\ \dots \end{array} \right\} \sqsubseteq F \left\{ \begin{array}{c} n \mapsto a' \\ n' \mapsto a'' \\ \dots \end{array} \right\}$$

e	n
\dots	\dots

e	n, n'
\dots	\dots

$$a \sqsubseteq a' a''$$

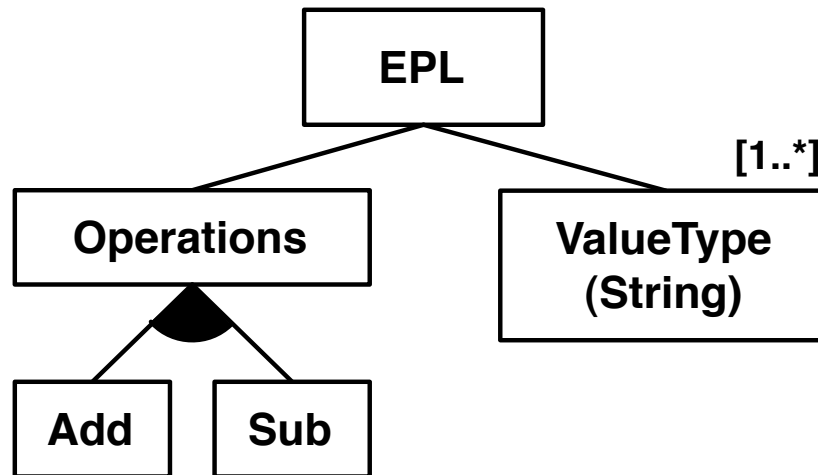
n and n' do not appear in other CK lines

n' is not an existing AM name

CK with Transformations (aka MSVCM CK)

Feature Expression	Transformations
eShop	select scenario SC01
\neg (Shopping Cart \wedge Bonus)	evaluate advice ADV01
Shopping Cart \wedge Bonus	evaluate advice ADV02
Shipping Method	bind SM to Shipping Method
...	...

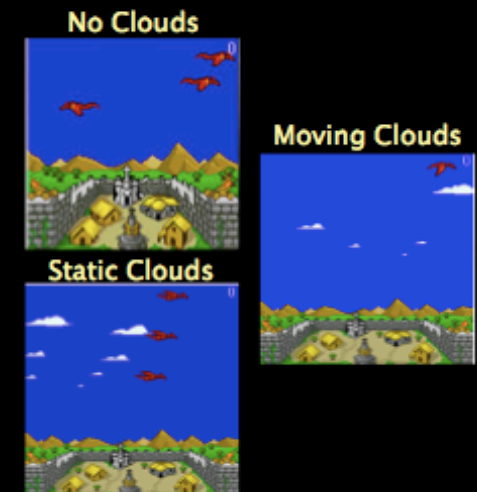
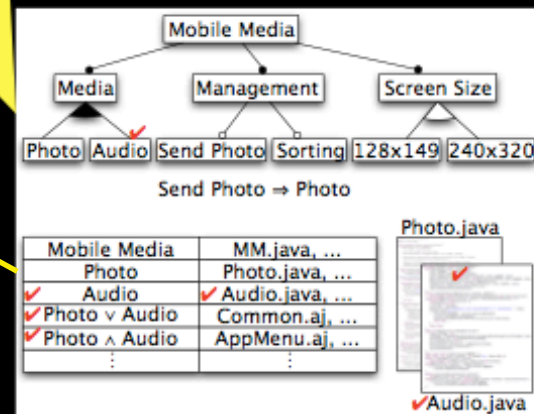
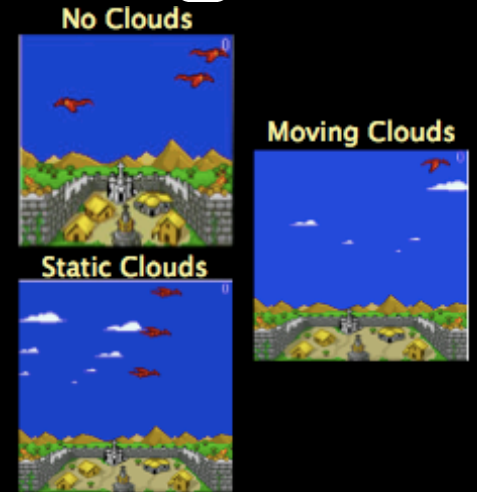
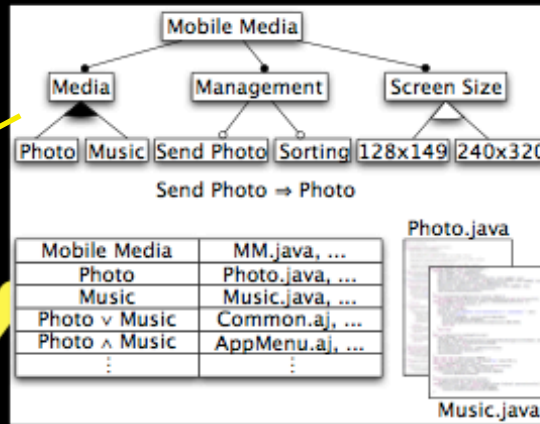
Cardinality-based



Feature Expression	Assets
"int" in ValueType	task involving int
"float" in ValueType	task involving float
"int" U "float" not in ValueType	task involving anything else than int or float
some ValueType	foreach v:ValueType do something with v
#ValueType > 1	do something when multiple types in the EPL

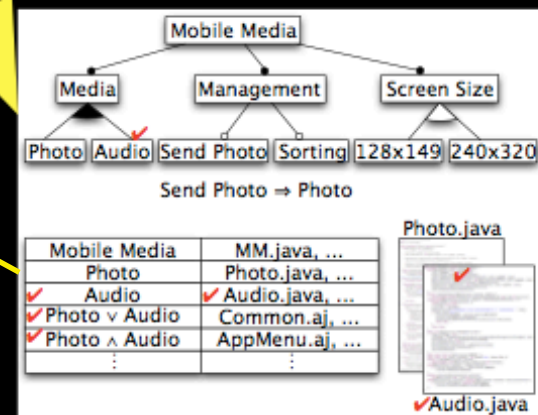
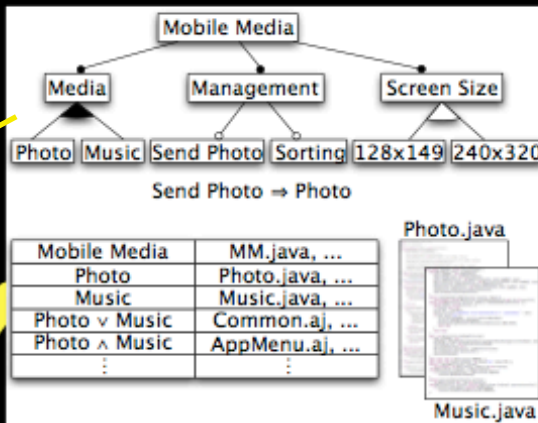
Automatic PL refinement checking

FM and CK
equivalence
checking



PL refinement approximation

testing individual assets



Current and future work

- Instantiate theory with cardinality-based FM+CK
- Analyze evolution of other SPLs
 - Discover more templates and analyze expressiveness of the proposed catalog
 - Check if verification can be automated or approximated, depending on the case
- Tool support
 - Refinement Notion and template automation

SPL Refinement

Leopoldo Motta Teixeira

Advisor: Paulo Borba
Co-Advisor: Rohit Gheyi

