Enhancing Feature Interfaces for Supporting Software Product Line Maintenance

Bruno B. P. Cafeo

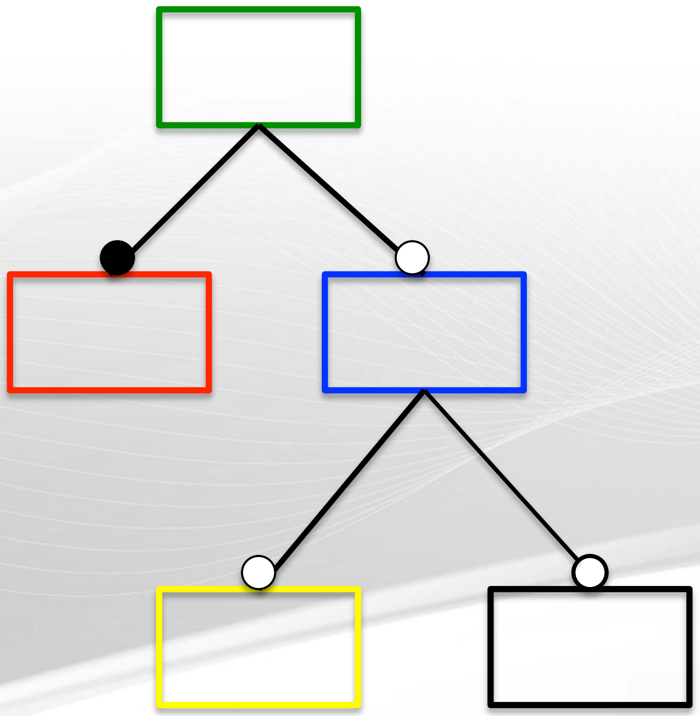
bcafeo@inf.puc-rio.br

OPUS Group

LES | DI |PUC-Rio - Brazil

Motivation

Software Product Line (SPL)

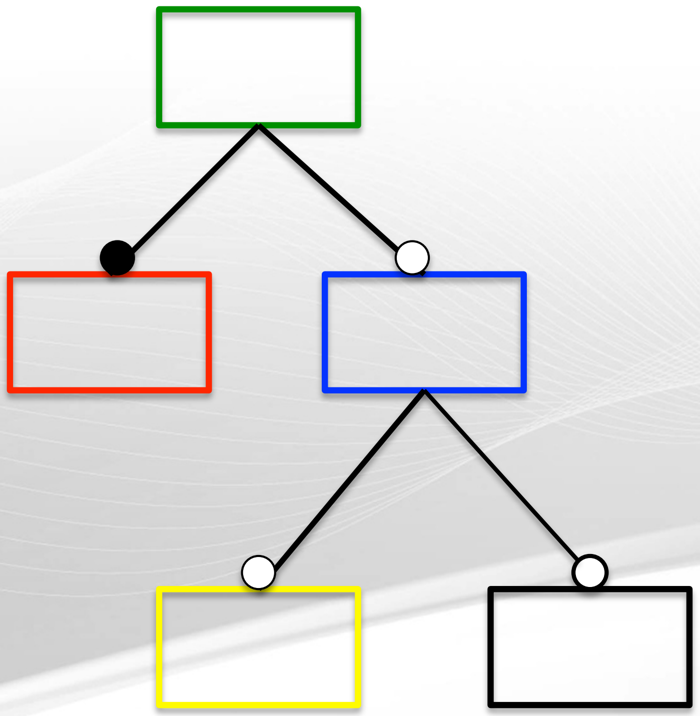


Feature Model

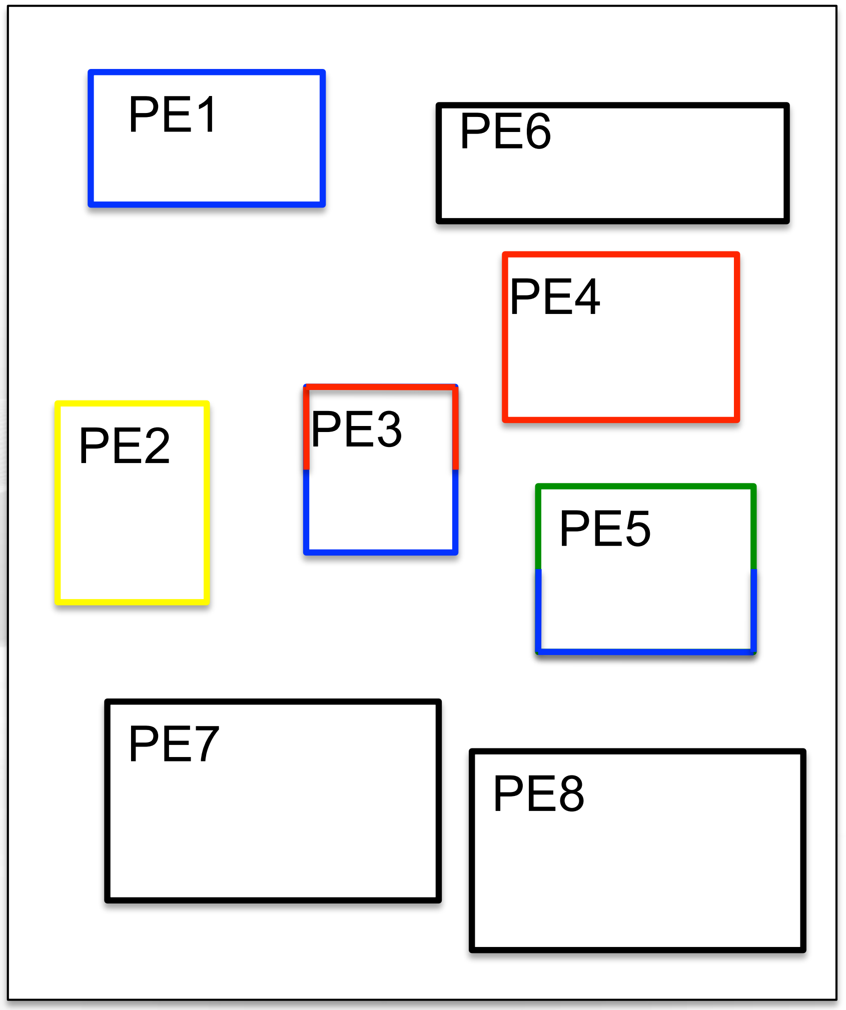
Motivation

Boundaries of features and modules of a programming language

Legend: PE – Program Element



Feature Model

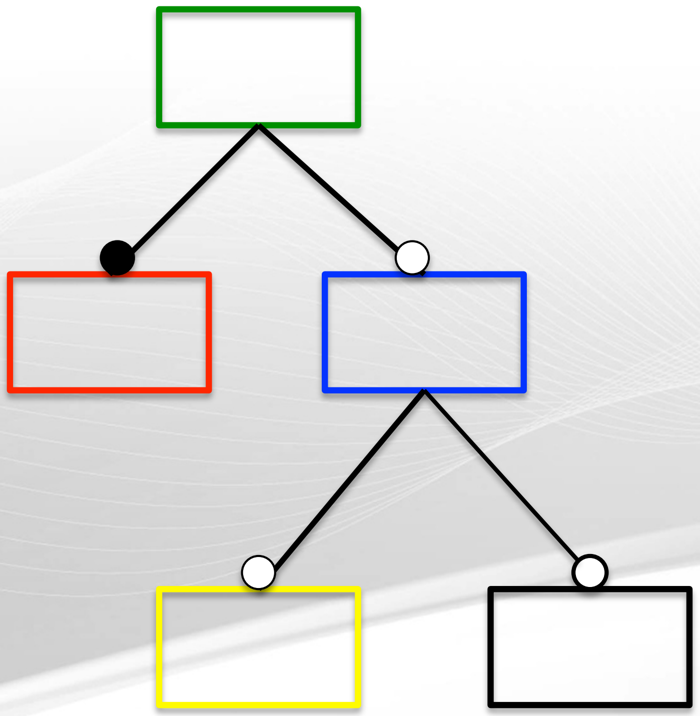


Source code

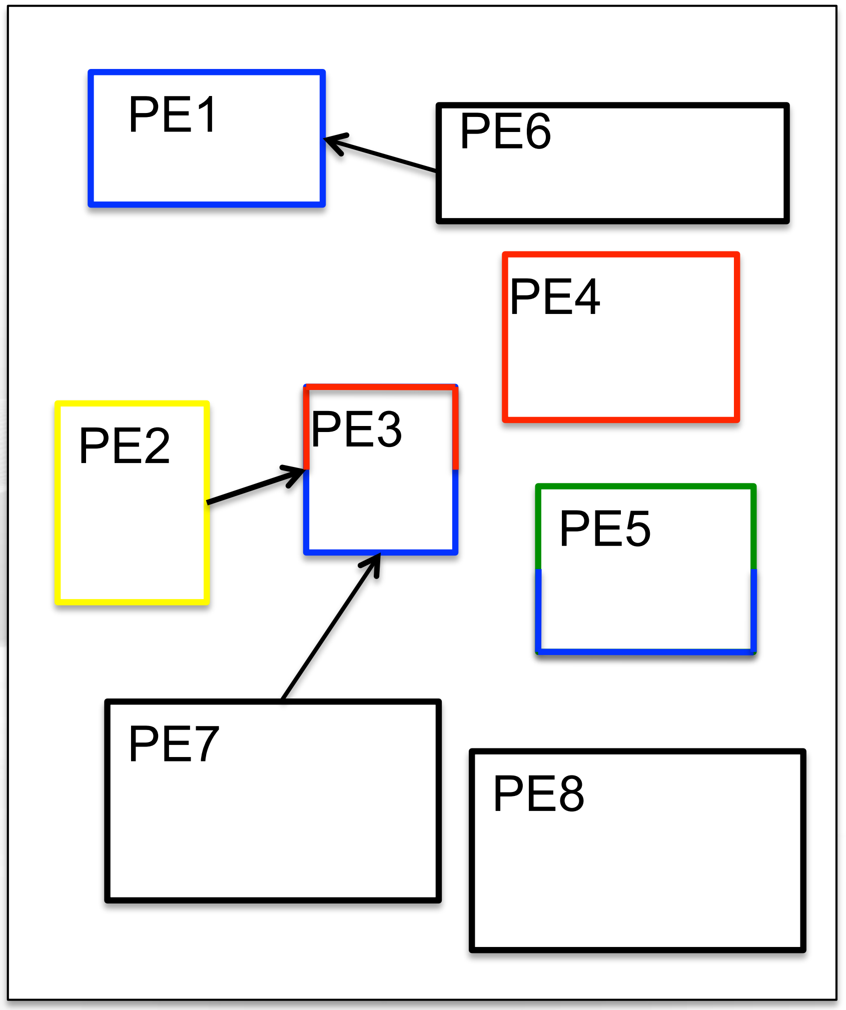
Motivation

Feature Dependencies in the source code of a SPL

Legend: PE – Program Element



Feature Model

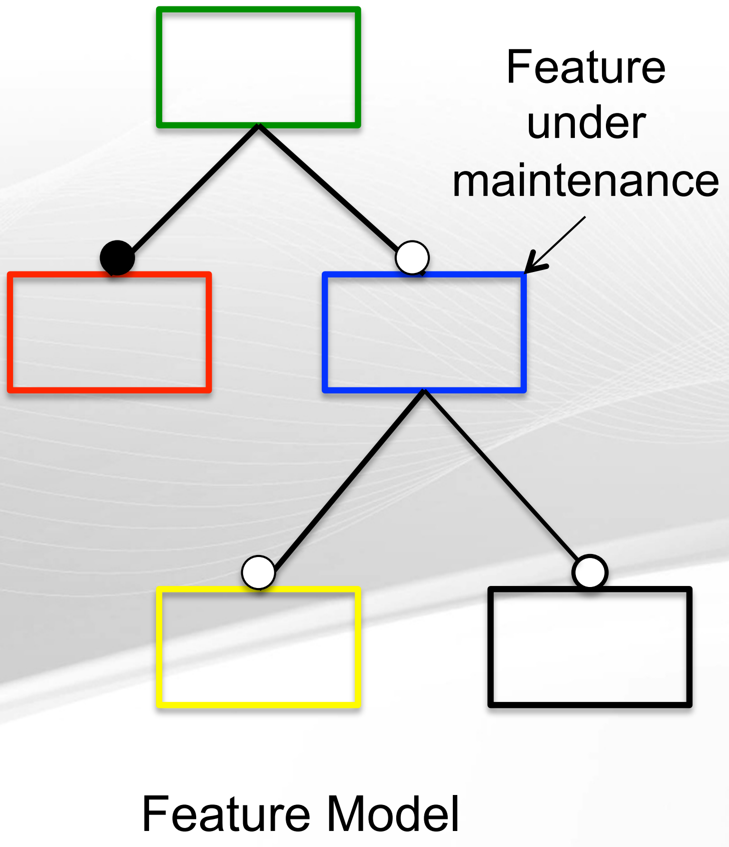


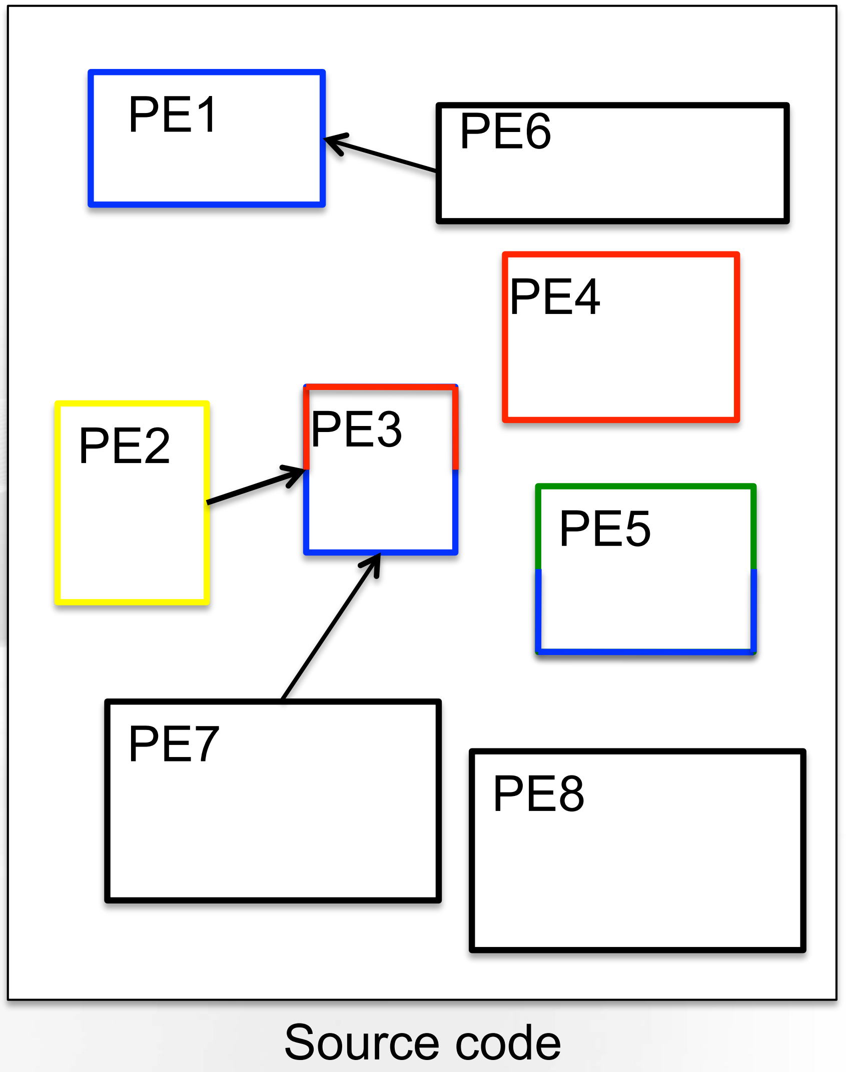
Source code

Motivation

Feature Dependencies in the source code of a SPL

Legend: PE – Program Element

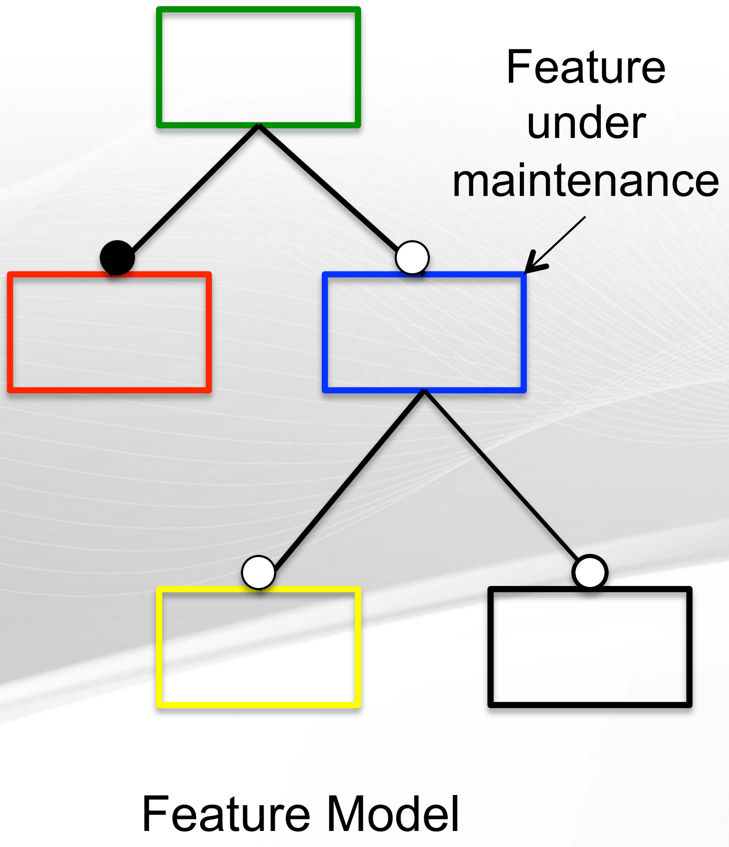


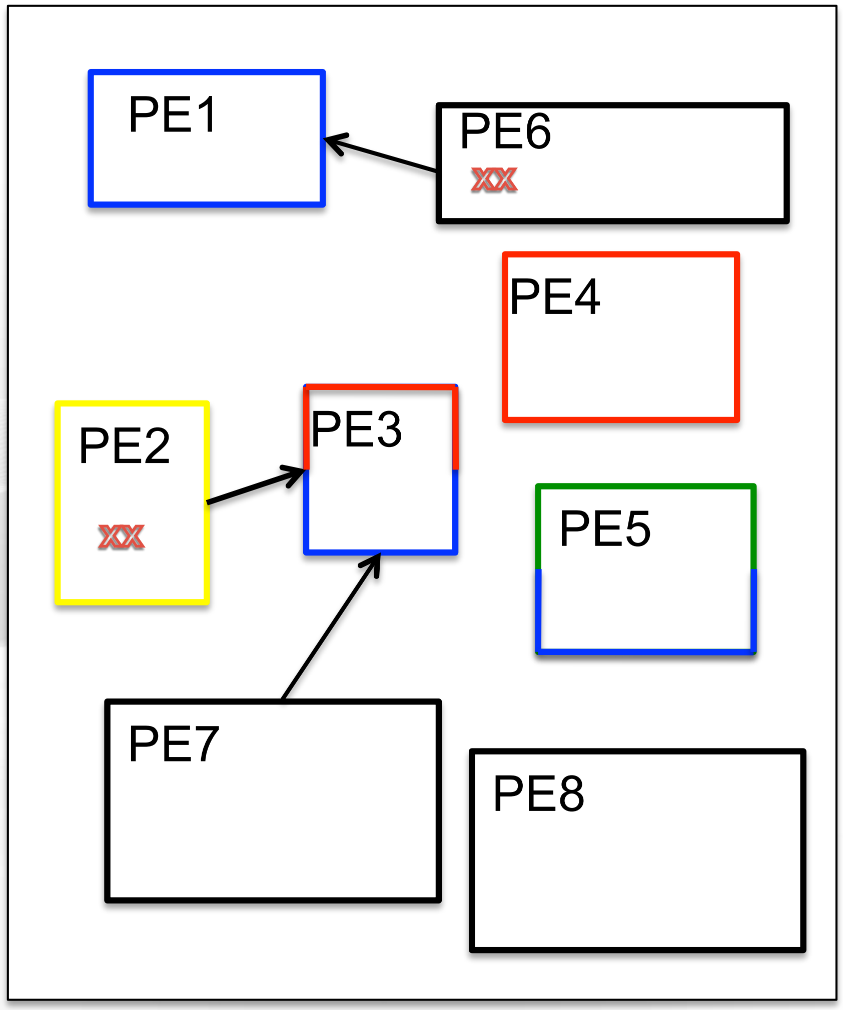


Motivation

Feature Dependencies in the source code of a SPL

Legend: PE – Program Element XX Maintenance problems related to the presence of a feature dependency





Source code

Two points to explore

- Identification and understanding of **feature dependency properties** and their **impact** on SPL **maintenance**
- **Feature modularity** improvement

Two points to explore

- Identification and understanding of **feature dependency properties** and their **impact** on SPL **maintenance**

- **Feature modularity** improvement

Feature Dependency Properties

- There is no conceptual framework that characterizes feature dependency properties in the source code
- Revealing properties that may exert an impact on SPL maintenance would be interesting

Feature Dependency Properties

Results

- Identification of properties and definition of metrics based on these properties
- Correlation of these metrics with SPL maintenance problems
- Comparison of our metrics with conventional metrics

Two points to explore

- Identification and understanding of **feature dependency properties** and their impact on SPL maintenance

- **Feature modularity** improvement

Feature Interface

Looking at the source code level

- As in a conventional stand-alone software (i.e. non-SPLs), **interfaces** should **help the understanding** of the **communication** between features in SPLs

- **Program elements** configuring dependencies are part of an implicit **feature interface**

Challenge

Complex feature interfaces

REGULAR_ACCOUNT

account_number
type
sort_code
balance
paid_in[]
paid_out[]
getEndofMonthBalance()
getBalance()
makeAPayment()
getFullStatement()
viewAccountDetailts()
getTaxes()

- Feature interfaces may become **large**
  - **Several elements** are member of an implicit interface
- Implicit feature interfaces are **not cohesive**
  - **Groups** of elements **act together** for the purpose of a dependency

Towards a Solution

- The proposed solution relies on the idea of *Interface Segregation Principle (ISP)* that states that

“clients should not be forced to depend upon interfaces that they do not use”

- We observe the idea of ISP from the point of view of SPL maintenance and we argue that

Developers should not be forced to understand parts of an interface that are not useful to their tasks

Towards a Solution (simple example)

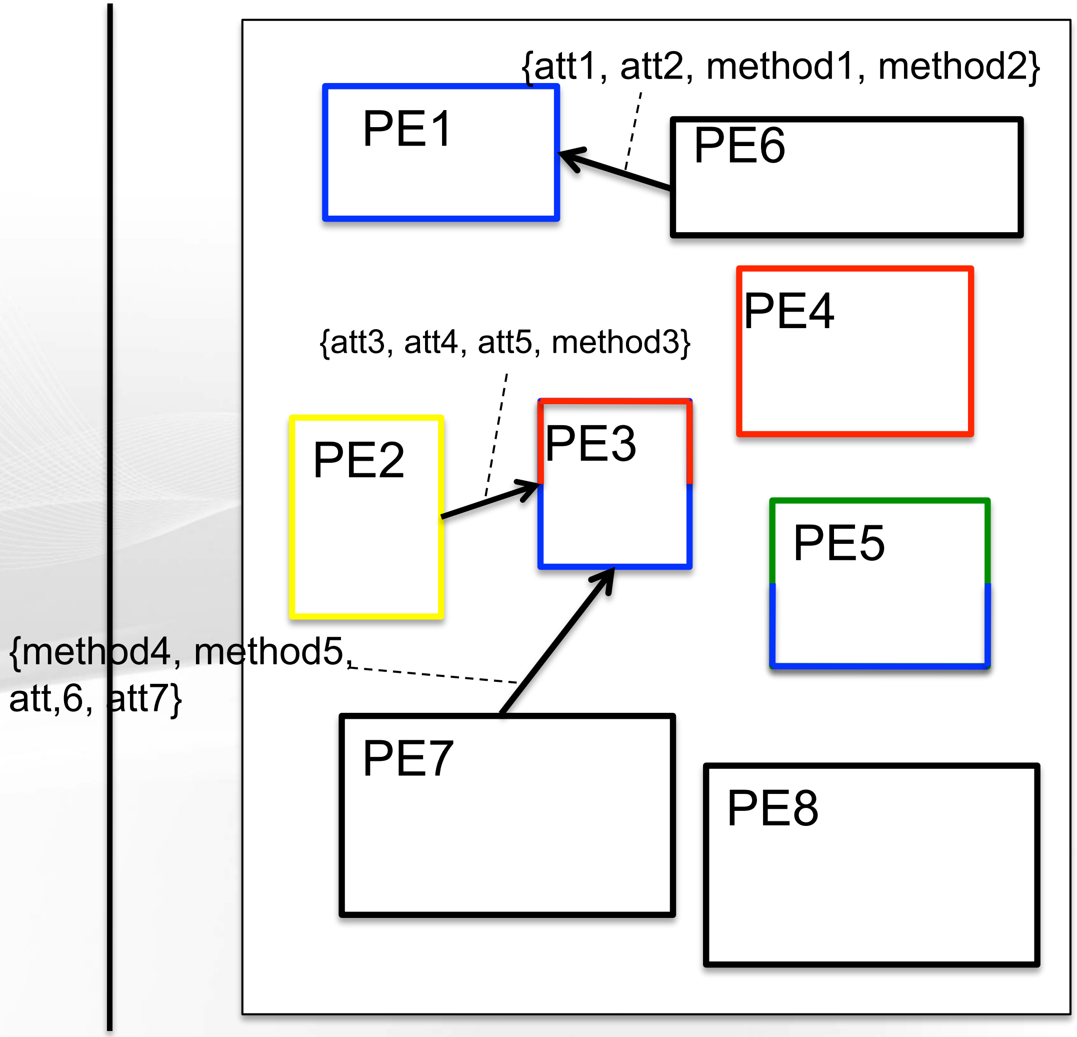
Identifying feature interface elements

Legend: PE – Program Element

Feature Interface

att1
att2
att3
att4
att5
att6
att7
method1
method2
method3
method4
method5

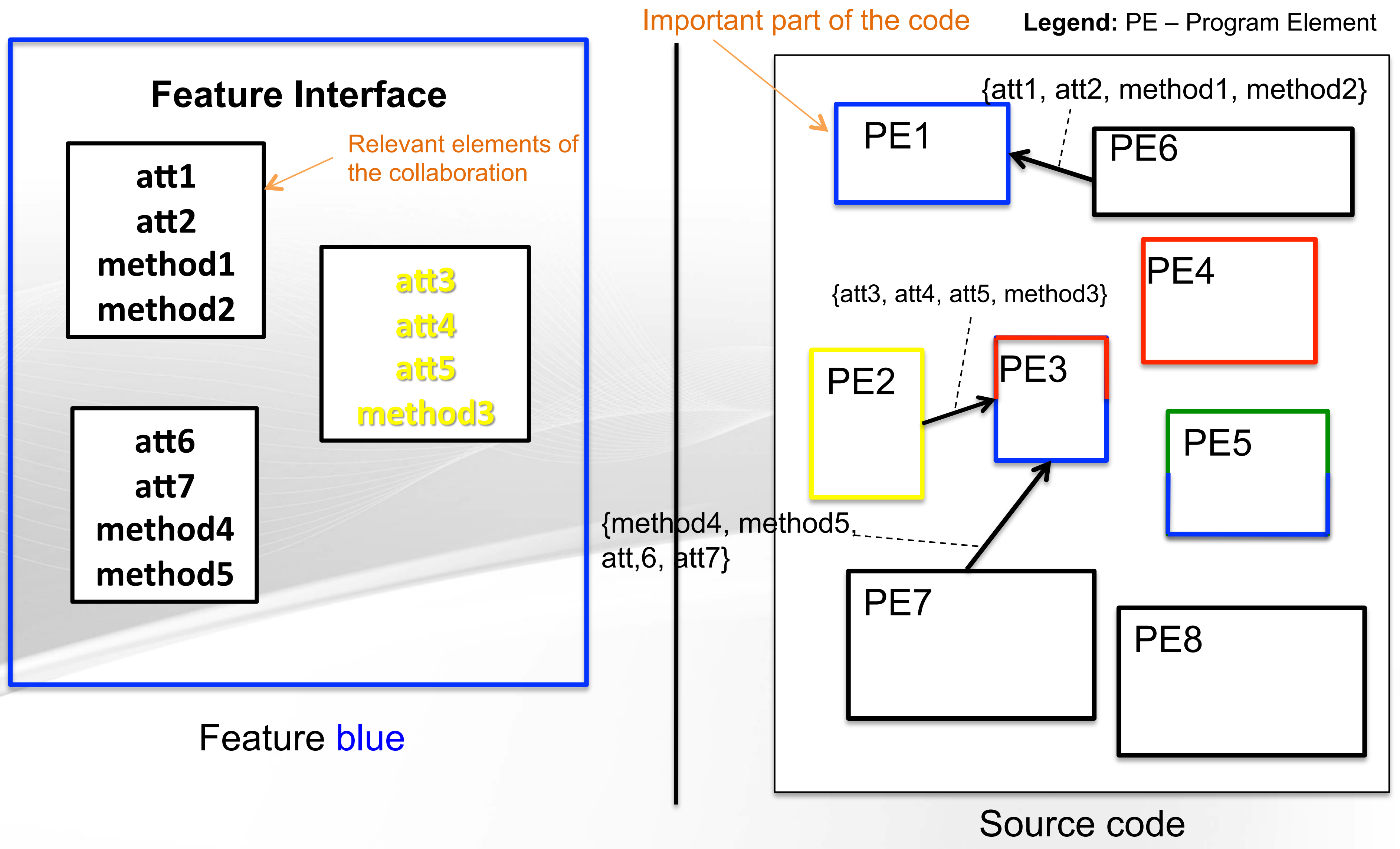
Feature blue


{att1, att2, method1, method2}
PE1
PE6
PE4
{att3, att4, att5, method3}
PE2
PE3
PE5
{method4, method5, att,6, att7}
PE7
PE8

Source code

Towards a Solution (simple example)

Understanding the feature interface



Important part of the code
Legend: PE – Program Element
Feature Interface
att1
att2
method1
method2
Relevant elements of the collaboration
att3
att4
att5
method3
att6
att7
method4
method5
{att1, att2, method1, method2}
PE1
PE6
PE4
{att3, att4, att5, method3}
PE2
PE3
PE5
{method4, method5, att,6, att7}
PE7
PE8

Feature blue

Source code

Feature Interface

Evaluation

- Conduct empirical studies to compare
  - The reduction of maintenance side effects
  - The SPL maintenance effort

Thanks!

Enhancing Feature Interfaces for Supporting Software Product Line Maintenance

Bruno B. P. Cafeo

bcafeo@inf.puc-rio.br

LES | DI |PUC-Rio - Brazil

OPUS Group