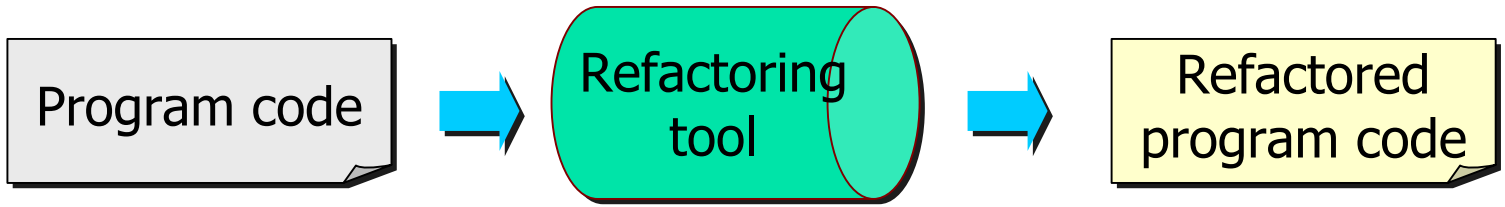
GPCE'04 Software Transformation Systems Workshop
(Vancouver, Canada, November 24th, 2004)

Transforming Object-Oriented Programs into Structurally Reusable Components

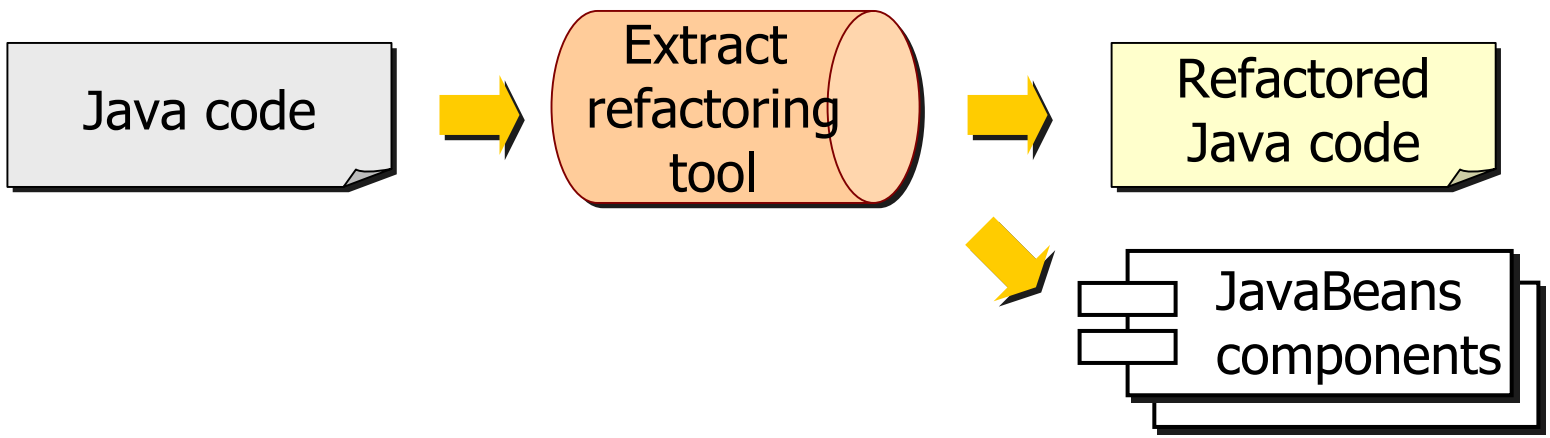
Hironori Washizaki
National Institute of Informatics, Tokyo, Japan
Yoshiaki Fukazawa
Waseda University, Tokyo, Japan
http://www.washizaki.net/

Refactoring as Transformation

- Refactoring



- Extract Component Refactoring



Class Relation Graph (CRG)

- Multigraph that represents relations among OO classes

```
public abstract class A {}
```

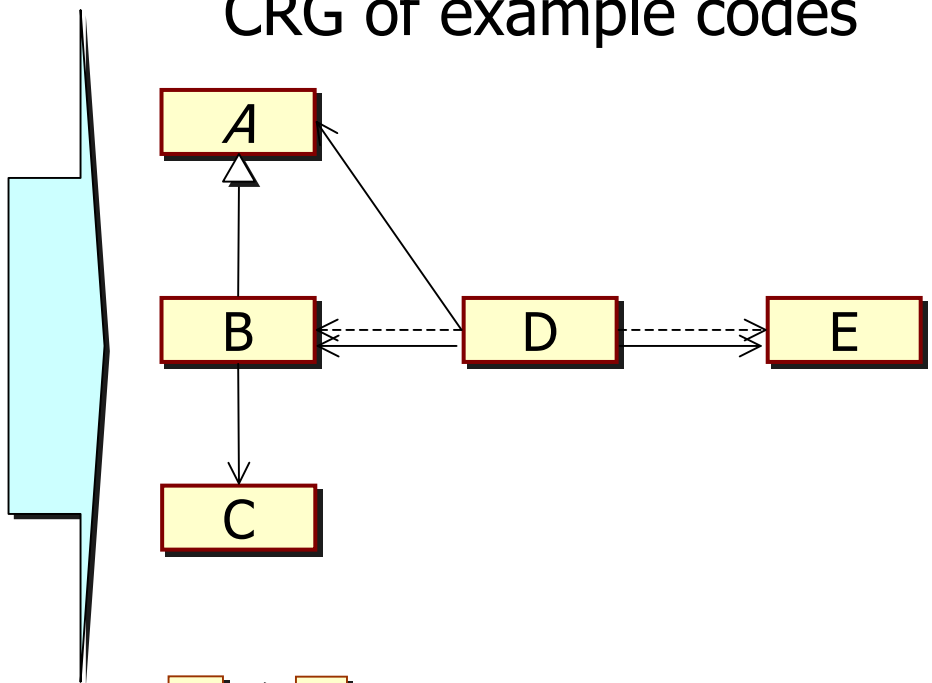
```
public class B extends A {
  public B(){ C c; }
}
```

```
public class C {
  private C(){ }
}
```

```
public class D {
  public D() {
    A a = new B();
    E e = new E(); }
}
```

```
public class E {}
```

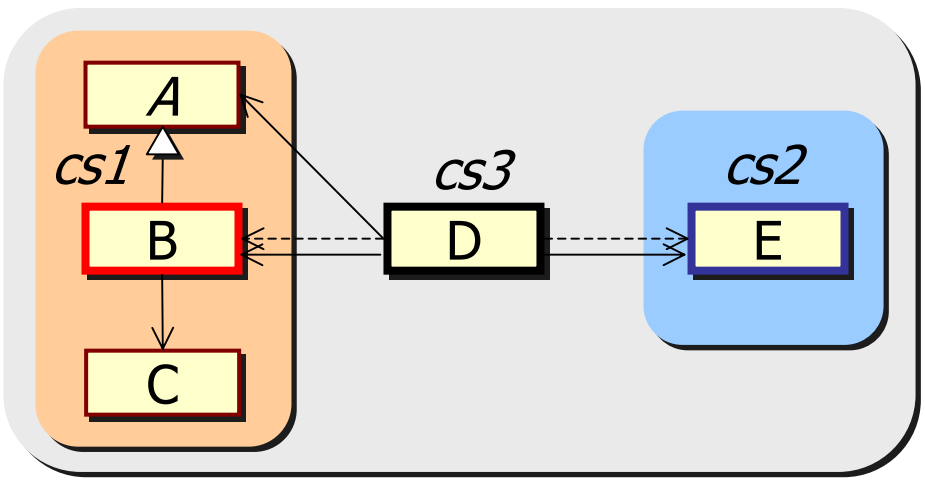
CRG of example codes

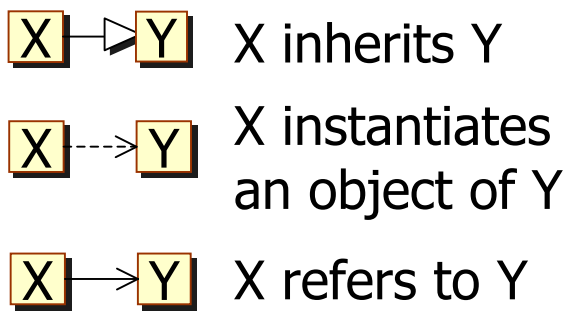


Detecting Component Candidates

- Structurally reusable component
  - Standalone and executable JavaBeans component
  - Component's interface is separated from implementation
  - All classes necessary for instantiation are packaged
- Detecting all clusters (component candidates) on CRG
  - Cluster: *cs* = (<Facade class>, <Set of participants>)

cs1 = (B, { A, B, C }) *cs2* = (E, { E }) *cs3* = (D, { A, B, C, D, E })

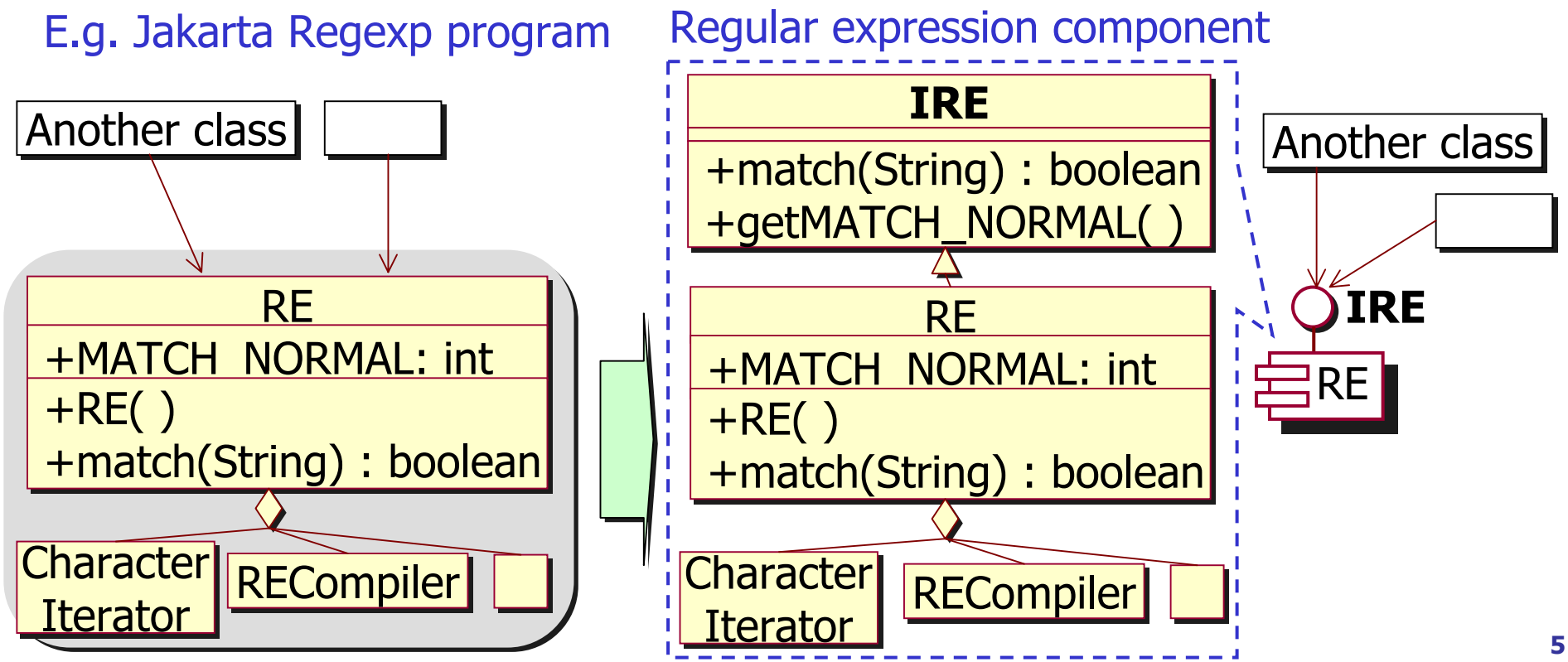




Extract Component Refactoring

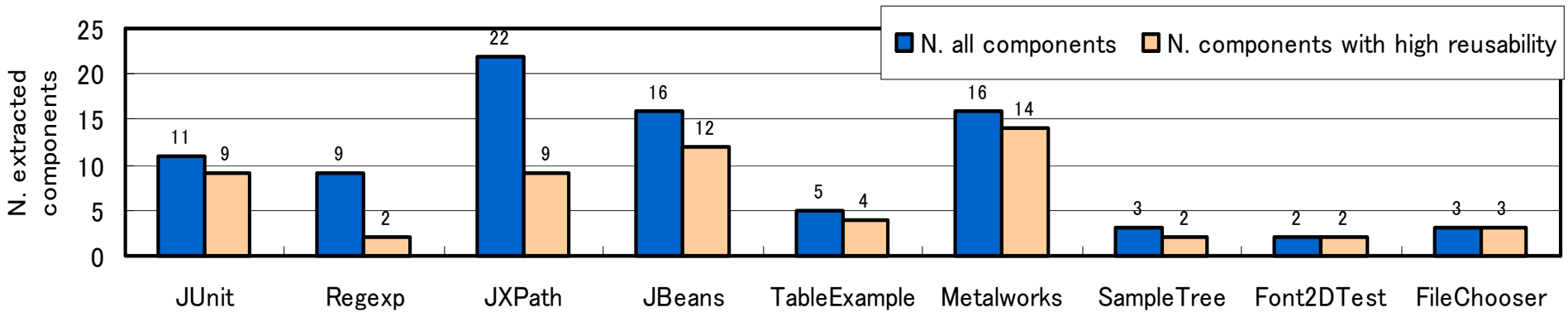
- Transforming Clusters into Components:
  - 1. Creates a new Facade interface
  - 2. Adds declarations of all public methods
  - 3. Adds declarations of setter/getter methods corresponding to public fields
  - 4. Compiles and packages all class files into JAR archive
  - 5. Modifies the surrounding parts to use extracted component

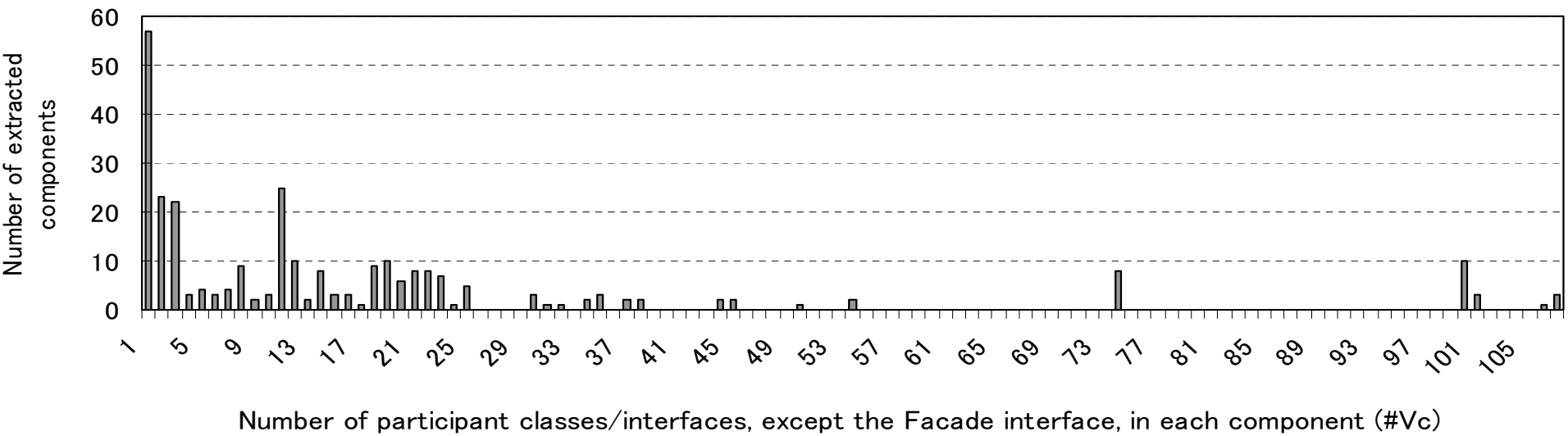
E.g. Jakarta Regexp program



Extraction Experiment

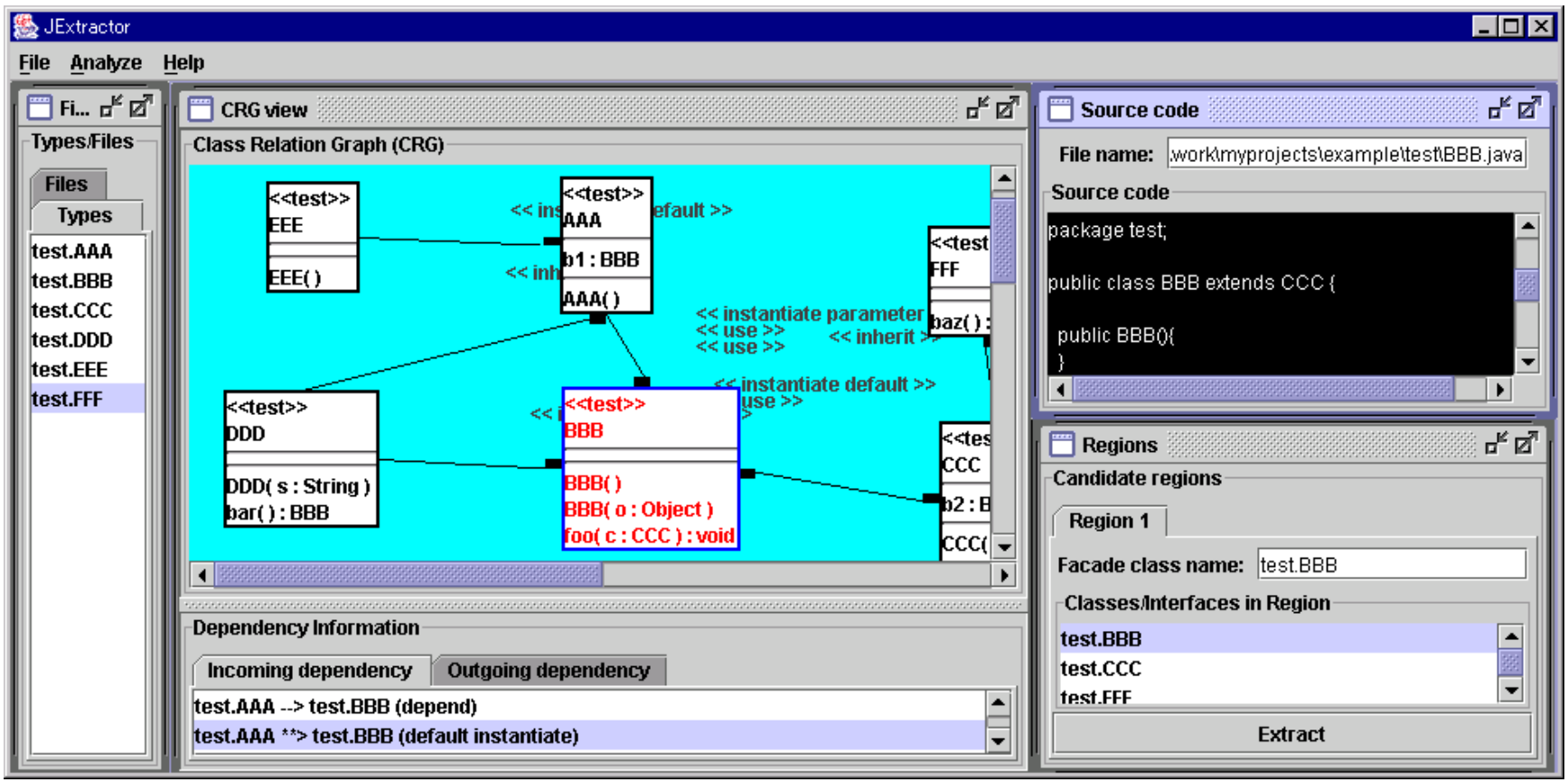
- Target: 4 open source programs + 6 Java demo programs
- Result: We extracted 282 components
  - 72% of all extracted components are high reusable
  - 80% of all extracted components are composed of three or more classes





Automated Extraction Tool

- Implementation: Java2 SDK1.4.0, JavaCC (parser generator)
- Analyzes Java program source codes, and displays CRG
- Performs all necessary steps of the Extract Component Refactoring



What does "reusable" mean?

- Extracted components are structurally reusable
  - Internal structure is hidden from outside
  - Component can be instantiated and executed alone
- However, NOT semantically reusable
  - Not always reusable in all possible contexts
  - How to guarantee semantic reusability?
  - Are some generative techniques (e.g. template programming) helpful for this objective?
- Towards "generative reuse"
  - Target of reuse has been abstracted.
  - Target can be easily customized to match the new contexts.
  - Target has capability of generating a part of program by specifying parameters.

Thank you.

National Institute of Informatics
国立情報学研究所

GPCE'04 Software Transformation Systems Workshop
(Vancouver, Canada, November 24th, 2004)

Transforming Object-Oriented Programs into Structurally Reusable Components

Hironori Washizaki
National Institute of Informatics, Tokyo, Japan
Yoshiaki Fukazawa
Waseda University, Tokyo, Japan
http://www.washizaki.net/