

# *Error Detection by Refactoring Reconstruction*

---

**International Workshop  
on Mining Software Repositories  
May 17, 2005  
Saint Louis**

**Carsten Görg**  
Saarland University  
goerg@cs.uni-sb.de

**Peter Weißgerber**  
Catholic University Eichstätt  
peter.weissgerber@ku-eichstaett.de

# *Refactoring*

*„Refactoring is a disciplined technique for restructuring an existing body of code, altering its internal structure without changing its external behavior.“*

*(Martin Fowler)*

# Effects of Local Refactorings

```
interface org.gjt.sp.jedit.search.SearchFileSet
```

```
getCode():String  
getFirstFile(View):String  
getNextFile(View, String):String  
getFiles(View):String  
getFileCount(View):int
```



```
org.gjt.sp.jedit.search.BufferListSet
```

```
getCode():String  
getFirstFile(View):String  
getNextFile(View, String):String  
getFiles(View):String  
getFileCount(View):int  
abstract _getFiles(Component):String[]
```

```
org.gjt.sp.jedit.search.CurrentBufferSet
```

```
getCode():String  
getFirstFile(View):String  
getNextFile(View, String):String  
getFiles(View):String  
getFileCount(View):int
```



```
org.gjt.sp.jedit.search.AllBufferSet
```

```
getCode():String  
AllBufferSet(String)  
getFileFilter():String  
_getFiles(Component):String[]
```

```
org.gjt.sp.jedit.search.DirectoryListSet
```

```
getCode():String  
getFileFilter():String  
DirectoryListSet(String, String, boolean)  
getDirectory():String  
isRecursive():boolean  
_getFiles(Component):String[]
```

# Renaming Methods to Change Event Handling

```
java.awt.event.MouseAdapter
```



```
org.gjt.sp.jedit.gui.EnhancedMenuItem$MouseHandler
```

```
mouseExited(MouseEvent):void  
mouseEntered(MouseEvent):void  
mouseClicked(MouseEvent):void  
mouseReleased(MouseEvent):void
```

```
org.gjt.sp.jedit.gui.EnhancedCheckBoxMenuItem$MouseHandler
```

```
mouseExited(MouseEvent):void  
mouseEntered(MouseEvent):void  
mouseClicked(MouseEvent):void  
mouseReleased(MouseEvent):void
```

# *Outline*

- Reconstructing refactorings
- Checking consistency
- Case studies on open source projects

# *Detecting and Visualizing Refactorings from Software Archive*

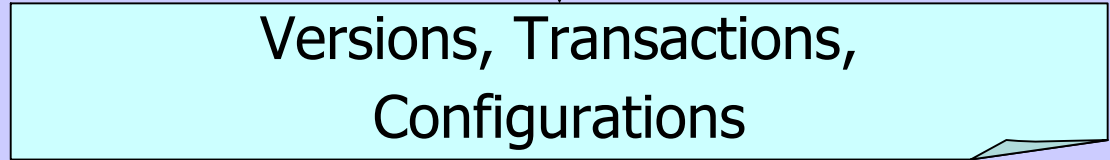
[Görg, Weißgerber: IWPC2005]

- **Search for refactorings on the level of classes and methods**
- **Structural refactorings**
  - **Move Class**
  - **Move Method (Pull up / Push down)**
- **Local refactorings**
  - **Rename Method**
  - **Add/Remove Parameter To/From Method**
  - **Hide Method**

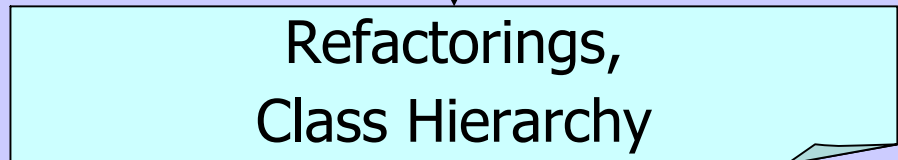
[Zimmermann, Weißgerber: Preprocessing  
CVS Data for Fine-Grained Analysis,  
MSR2004]



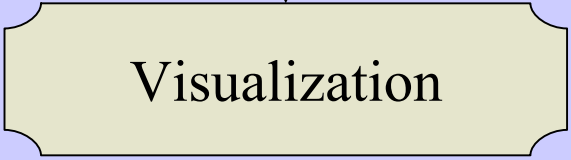
Preprocessing



Analyses



Graph drawing



# Artefacts within one transaction

Parse each revision of each transaction

Identify

- Classes
  - Full qualified name
- Fields
  - Name and type
- Methods
  - Name, parameter list, return type

*Transaction t*

main.java-1.3

```
package org.epos.core;
```

```
public class Main {
```

```
    int x_pos, y_pos;
```

```
    private void setPos(int x, int y) { ... }
```

```
    public int getXPos() { ... }
```

```
    public int getYPos() { ... }
```

```
    private int getXPosPlus1() { ... }
```

```
}
```

gui.java-1.2

```
package org.epos.gui;
```

```
public class Gui extends JFrame {
```

```
    Image image;
```

```
    public void setImage(Image i) { ... }
```

```
    public void paintImage() { ... }
```

```
}
```

# Compare artefacts of current and previous revision

added,  
removed,  
common  
artefacts

```
main.java-1.2

package org.epos.core;

public class Main {
    int x_pos;
    public void setXPos(int x) { ... }
    public int getXPos() { ... }
    public int getXPosPlus1() { ... }
    public void test() { ... }
    private void paintImage() { ... }
}
```

## Transaction t

```
main.java-1.3

package org.epos.core;

public class Main {
    int x_pos, y_pos;
    private void setPos(int x, int y) { ... }
    public int getXPos() { ... }
    public int getYPos() { ... }
    private int getXPosPlus1() { ... }
}

}
```

```
gui.java-1.1

package org.epos.gui;

public class Gui extends JFrame {
    Image image;
    public void setImage(Image i) { ... }
    public void getImage() { ... }
}

}
```

```
gui.java-1.2

package org.epos.gui;

public class Gui extends JFrame {
    Image image;
    public void setImage(Image i) { ... }
    public void paintImage() { ... }
}

}
```



# *Uncovering refactorings*

- Use simple heuristics
- Consequences:
  - Only candidates
  - Problem of impure and ambiguous refactorings

# „Rename Method“

- Method A **added** in new version
- Method B **deleted** from previous version

- same parameters
- same return type
- same method body

Core.java-1.5

```
package org.epos.core;

public class Core {

...

public int computeCoreValue() {
    int result=1;
    for (int i=0;i<this.max;i++) {
        result*=i;
    }
    return result;
}

...
}
```

## Transaction *t*

Core.java-1.6

```
package org.epos.core;

public class Core {

...

public int getCoreValue() {
    int result=1;
    for (int i=0;i<this.max;i++) {
        result*=i;
    }
    return result;
}

...
}
```

# „Add / Remove Parameter“

- Method A **added** in new version
- Method B **deleted** from previous version

- same name
- same return type
- parameter list of A is a subset of parameter list of B

Text.java-1.3

```
package org.epos.text;

public class Text {

    ...

    public void printString(String s) {
        // ...
    }

    ...

}
```

*Transaction t*

Text.java-1.4

```
package org.epos.text;

public class Text {

    ...

    public void printString(String s,
                           int size) {
        // ...
    }

    ...

}
```

# *Ambiguous Refactorings*

It is not always possible to unambiguously identify refactorings:

Old configuration:

```
Class A  
m(a,b)  
m(b,c)
```

New configuration:

```
Class A  
m(a,b,c)
```

# *Checking Consistency*

- For each refactoring compute list of other possible candidates
- 2 types of candidates
  - Methods in subclasses
  - Methods in sibling classes
- Check if missing refactoring has been applied in a later transaction

# *Checking Consistency*

Different types of possible errors:

- Subclasses
  - Interface method or abstract method is no longer implemented
  - Refactored method is inherited instead of being overwritten
- Sibling classes
  - Wrong program behavior

# *Case studies – refactorings in open source projects*

- JEdit
  - 6 developers
  - 2141 Transactions
  - Start: September'01
- Jakarta Tomcat
  - 40 developers
  - 5096 transactions
  - Start: Oktober'99

*interface org.gjt.sp.jedit.buffer.BufferChangeListener*

*contentInserted(Buffer, int, int, int, int):void*  
*c ADD PARAMETER FROM foldLevelChanged(Buffer, int):void*  
*foldLevelChanged(Buffer, int, int):void*



*org.gjt.sp.jedit.buffer.BufferChangeAdapter*

*foldLevelChanged(Buffer, int):void*  
*contentInserted(Buffer, int, int, int, int):void*  
*contentRemoved(Buffer, int, int, int, int):void*

*org.gjt.sp.jedit.textarea.JEditTextArea\$BufferChangeHandler*

*contentInserted(Buffer, int, int, int, int):void*  
*contentRemoved(Buffer, int, int, int, int):void*  
*r ADD PARAMETER FROM foldLevelChanged(Buffer, int):void*  
*foldLevelChanged(Buffer, int, int):void*

*org.gjt.sp.jedit.syntax.DefaultTokenHandler*

*init():void*  
*getFirstToken():Token*  
*handleToken(byte, int, int, TokenMarker.LineContext):void*  
*getParserRuleSet(TokenMarker.LineContext):ParserRuleSet*  
*c ADD PARAMETER FROM addToken(Token, TokenMarker.LineContext):boolean*  
*addToken(Token, TokenMarker.LineContext, boolean):boolean*



*org.gjt.sp.jedit.syntax.DisplayTokenHandler*

*init(Segment, SyntaxStyle[], FontRenderContext, TabExpander):void*  
*createToken(byte, int, int, TokenMarker.LineContext):Token*  
*addToken(Token, TokenMarker.LineContext):boolean*



### org.apache.jasper.compiler.ParseEventListener

```
beginPageProcessing():void
handleComment(Mark, Mark):void
handleDirective(String, Mark, Mark, Hashtable):void
handleBean(Mark, Mark, Hashtable):void
handleBeanEnd(Mark, Mark, Hashtable):void
handleGetProperty(Mark, Mark, Hashtable):void
handleSetProperty(Mark, Mark, Hashtable):void
handlePlugin(Mark, Mark, Hashtable, String):void
handleCharData(char[]):void
getTagLibraries():TagLibraries
handleForward(Mark, Mark, Hashtable, Hashtable):void
handleInclude(Mark, Mark, Hashtable, Hashtable):void
endPageProcessing():void
handleTagEnd(Mark, Mark, String, String, Hashtable, TagLibraryInfoImpl, TagInfo):void
handleDeclaration(Mark, Mark, Hashtable):void
handleScriptlet(Mark, Mark, Hashtable):void
h ADD PARAMETER FROM handleTagBegin(Mark, Hashtable, String, String, TagLibraryInfoImpl, TagInfo):void
handleTagBegin(Mark, Mark, Hashtable, String, String, TagLibraryInfoImpl, TagInfo):void
```

### org.apache.jasper.compiler.BaseJspListener

```
BaseJspListener(JspReader, ServletWriter)
beginPageProcessing():void
endPageProcessing():void
handleComment(Mark, Mark):void
handleDirective(String, Mark, Mark, Hashtable):void
handleBean(Mark, Mark, Hashtable):void
handleBeanEnd(Mark, Mark, Hashtable):void
handleGetProperty(Mark, Mark, Hashtable):void
handleSetProperty(Mark, Mark, Hashtable):void
handlePlugin(Mark, Mark, Hashtable, String):void
handleCharData(char[]):void
handleForward(Mark, Mark, Hashtable, Hashtable):void
handleInclude(Mark, Mark, Hashtable, Hashtable):void
getTagLibraries():TagLibraries
handleTagEnd(Mark, Mark, String, String, Hashtable, TagLibraryInfoImpl, TagInfo):void
handleDeclaration(Mark, Mark, Hashtable):void
handleScriptlet(Mark, Mark, Hashtable):void
h ADD PARAMETER FROM handleTagBegin(Mark, Hashtable, String, String, TagLibraryInfoImpl, TagInfo):void
handleTagBegin(Mark, Mark, Hashtable, String, String, TagLibraryInfoImpl, TagInfo):void
```

### org.apache.jasper.compiler.DelegatingListener

```
beginPageProcessing():void
endPageProcessing():void
handleComment(Mark, Mark):void
handleDirective(String, Mark, Mark, Hashtable):void
handleBean(Mark, Mark, Hashtable):void
handleBeanEnd(Mark, Mark, Hashtable):void
handleGetProperty(Mark, Mark, Hashtable):void
handleSetProperty(Mark, Mark, Hashtable):void
handlePlugin(Mark, Mark, Hashtable, String):void
handleCharData(char[]):void
handleForward(Mark, Mark, Hashtable, Hashtable):void
handleInclude(Mark, Mark, Hashtable, Hashtable):void
getTagLibraries():TagLibraries
DelegatingListener(ParseEventListener, Parser.Action)
doAction():void
handleTagEnd(Mark, Mark, String, String, Hashtable, TagLibraryInfoImpl, TagInfo):void
handleDeclaration(Mark, Mark, Hashtable):void
handleScriptlet(Mark, Mark, Hashtable):void
h ADD PARAMETER FROM handleTagBegin(Mark, Hashtable, String, String, TagLibraryInfoImpl, TagInfo):void
handleTagBegin(Mark, Mark, Hashtable, String, String, TagLibraryInfoImpl, TagInfo):void
```

### org.apache.jasper.compiler.DumbParseEventListener

```
handleComment(Mark, Mark):void
handleDirective(String, Mark, Mark, Hashtable):void
handleBean(Mark, Mark, Hashtable):void
handleBeanEnd(Mark, Mark, Hashtable):void
handleGetProperty(Mark, Mark, Hashtable):void
handleSetProperty(Mark, Mark, Hashtable):void
handleCharData(char[]):void
getTagLibraries():TagLibraries
DumbParseEventListener(JspReader, ServletWriter)
handlePlugin(Mark, Mark, Hashtable):void
handleForward(Mark, Mark, Hashtable):void
handleInclude(Mark, Mark, Hashtable):void
handleTagBegin(Mark, Hashtable, String, String, TagLibraryInfoImpl, TagInfo):void
handleTagEnd(Mark, Mark, String, String, Hashtable, TagLibraryInfoImpl, TagInfo):void
handleDeclaration(Mark, Mark, Hashtable):void
handleScriptlet(Mark, Mark, Hashtable):void
handleExpression(Mark, Mark, Hashtable):void
```

### org.apache.jasper.compiler.JspParseEventListener

```
beginPageProcessing():void
endPageProcessing():void
handleComment(Mark, Mark):void
handleDirective(String, Mark, Mark, Hashtable):void
handleBean(Mark, Mark, Hashtable):void
handleBeanEnd(Mark, Mark, Hashtable):void
handleGetProperty(Mark, Mark, Hashtable):void
handleSetProperty(Mark, Mark, Hashtable):void
handlePlugin(Mark, Mark, Hashtable, String):void
handleCharData(char[]):void
handleForward(Mark, Mark, Hashtable, Hashtable):void
handleInclude(Mark, Mark, Hashtable, Hashtable):void
getTagLibraries():TagLibraries
addGenerator(Generator):void
generateAll(Class):void
generateHeader():void
generateFooter():void
handleTagEnd(Mark, Mark, String, String, Hashtable, TagLibraryInfoImpl, TagInfo):void
handleDeclaration(Mark, Mark, Hashtable):void
handleScriptlet(Mark, Mark, Hashtable):void
handleExpression(Mark, Mark, Hashtable):void
J ADD PARAMETER FROM handleTagBegin(Mark, Hashtable, String, String, TagLibraryInfoImpl, TagInfo):void
handleTagBegin(Mark, Mark, Hashtable, String, String, TagLibraryInfoImpl, TagInfo):void
```

### interface org.apache.tomcat.core.RequestInterceptor

```
contextMap(Request):int  
requestMap(Request):int  
beforeBody(Request, Response):int  
preService(Request, Response):int  
beforeCommit(Request, Response):int  
afterBody(Request, Response):int  
postService(Request, Response):int  
authenticate(Request, Response):int  
getMethods():String[]  
r ADD PARAMETER FROM authorize(Request, Response):int  
authorize(Request, Response, String[]):int
```

### org.apache.tomcat.core.BaseInterceptor

```
log(String):void  
setContextManager(ContextManager):void  
setDebug(int):void  
requestMap(Request):int  
contextMap(Request):int  
beforeBody(Request, Response):int  
BaseInterceptor()  
preService(Request, Response):int  
beforeCommit(Request, Response):int  
afterBody(Request, Response):int  
postService(Request, Response):int  
contextInit(Context):void  
preServletInit(Context, ServletWrapper):void  
postServletInit(Context, ServletWrapper):void  
preServletDestroy(Context, ServletWrapper):void  
postServletDestroy(Context, ServletWrapper):void  
engineInit(ContextManager):void  
addContext(ContextManager, Context):void  
contextShutdown(Context):void  
engineShutdown(ContextManager):void  
removeContext(ContextManager, Context):void  
getMethods():String[]  
addSecurityConstraint(Context, String, Container):void  
authenticate(Request, Response):int  
addContainer(Container):void  
removeContainer(Container):void  
newSessionRequest(Request, Response):int  
r ADD PARAMETER FROM authorize(Request, Response):int  
authorize(Request, Response, String[]):int
```

### org.apache.tomcat.request.JDBCRealm

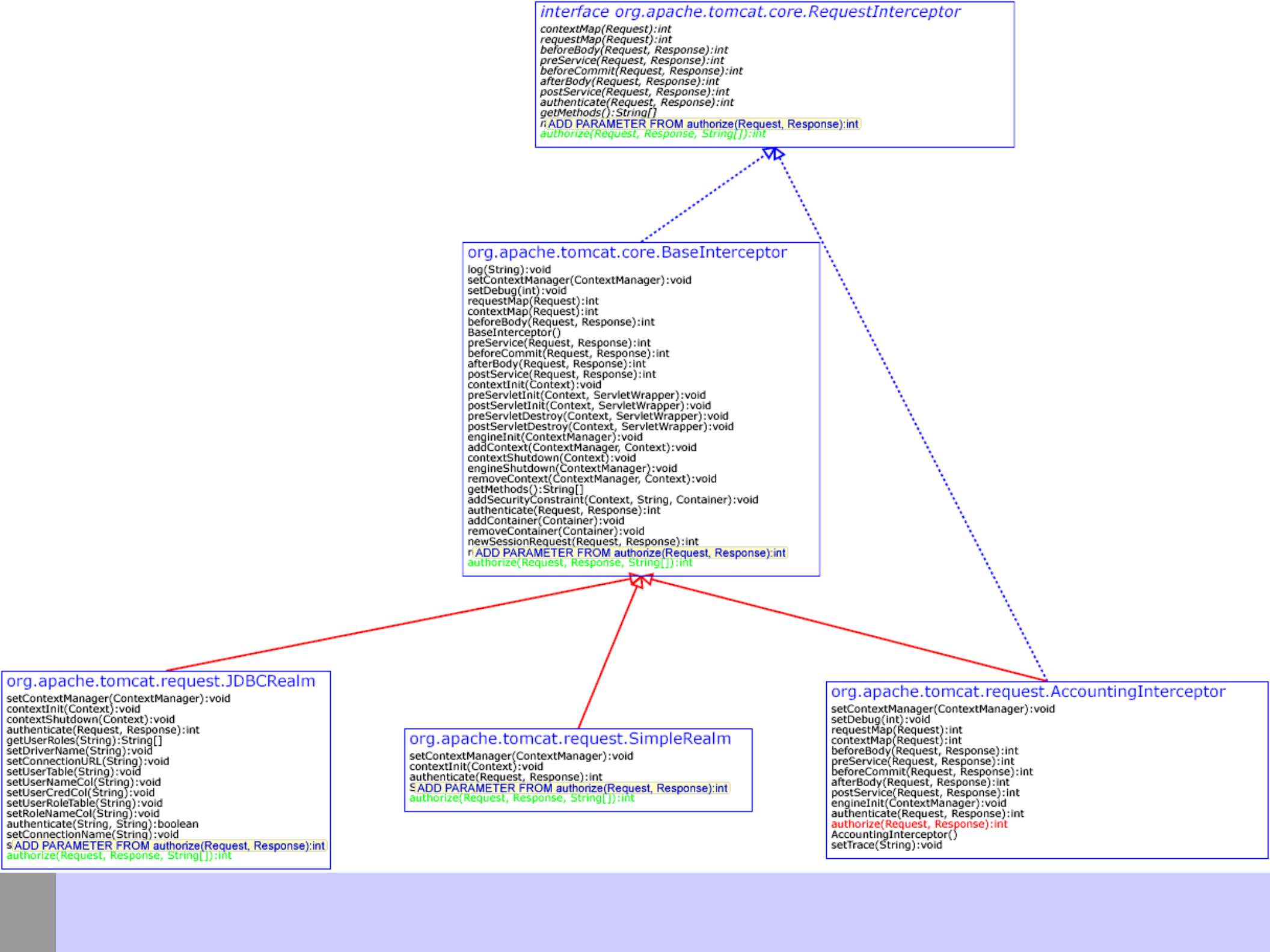
```
setContextManager(ContextManager):void  
contextInit(Context):void  
contextShutdown(Context):void  
authenticate(Request, Response):int  
getUserRoles(String):String[]  
setDriverName(String):void  
setConnectionURL(String):void  
setUserTable(String):void  
setUserNameCol(String):void  
setUserCredCol(String):void  
setUserRoleTable(String):void  
setRoleNameCol(String):void  
authenticate(String, String):boolean  
setConnectionName(String):void  
s ADD PARAMETER FROM authorize(Request, Response):int  
authorize(Request, Response, String[]):int
```

### org.apache.tomcat.request.SimpleRealm

```
setContextManager(ContextManager):void  
contextInit(Context):void  
authenticate(Request, Response):int  
s ADD PARAMETER FROM authorize(Request, Response):int  
authorize(Request, Response, String[]):int
```

### org.apache.tomcat.request.AccountingInterceptor

```
setContextManager(ContextManager):void  
setDebug(int):void  
requestMap(Request):int  
contextMap(Request):int  
beforeBody(Request, Response):int  
preService(Request, Response):int  
beforeCommit(Request, Response):int  
afterBody(Request, Response):int  
postService(Request, Response):int  
engineInit(ContextManager):void  
authenticate(Request, Response):int  
authorize(Request, Response):int  
AccountingInterceptor()  
setTrace(String):void
```



# *Our contribution*

- Approach to reconstruct refactorings from software archives
- Check consistency
- Case studies
  - jEdit: 5 candidates for erroneous transactions (3 subclasses, 2 sibling classes)
  - TomCat: 7 candidates for erroneous transactions (3 subclasses, 4 sibling classes)