

Actively Comparing Clones Inside The Code Editor

Ferosh Jacob¹ Daqing Hou² Patricia Jablonski²

¹Department of Computer and Information Sciences
University of Alabama at Birmingham
Birmingham AL 35205

²Electrical and Computer Engineering
Clarkson University
Potsdam NY 13699

May 7, 2010

Outline

1 Introduction

- Why CSeR?
- Example scenario
- With CSeR

2 Design

- Requirements
- Implementation
- An example

3 Validation

- Robustness
- Comparison with existing tools
- Usefulness

4 Conclusion & future work

- Conclusion

What is missing in current copy-paste scenario?

Copy-Paste

Lack of tool support for copy-paste induced clones

- Implicit, hidden relationship between clones
- Clone differences not obvious
 - ⇒ Time consuming to recover detailed differences between clones
 - ⇒ Guide for conducting similar copy-modify operations in future

GUI comparison of SetFilterWizardPage & ExclusionInclusionDialog

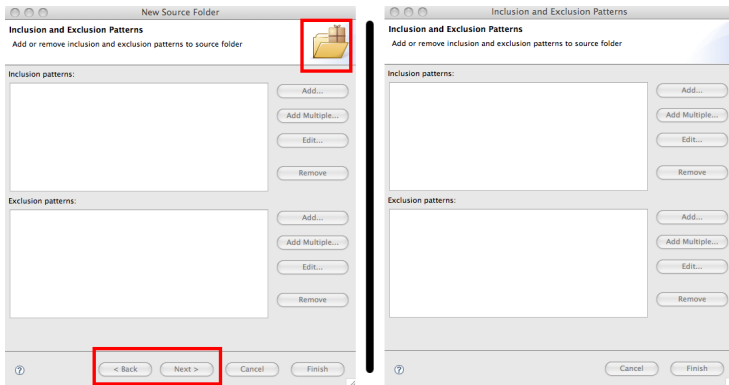
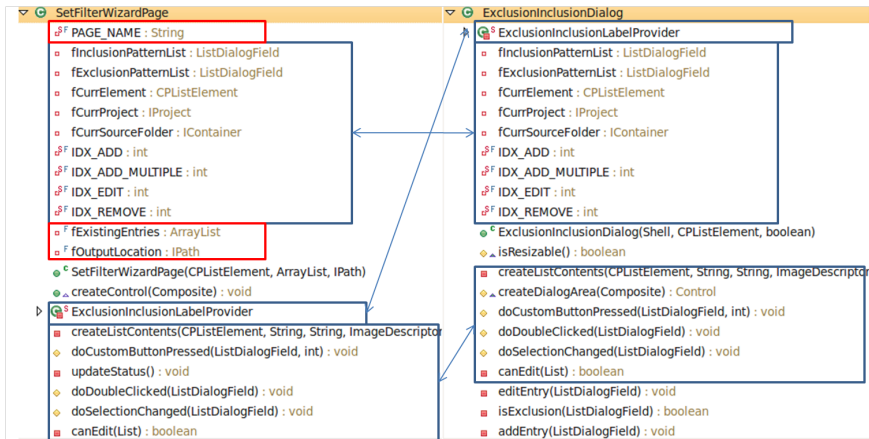


Figure: SetFilterWizardPage & ExclusionInclusionDialog

Detailed differences: SetFilterWizardPage & ExclusionInclusionDialog



Tracking and visualizing differences

```

58 public class ExclusionInclusionDialog extends StatusDialog { 1 {NewElementWizardPage->StatusDialog}
59
60     private static class ExclusionInclusionLabelProvider extends LabelProvider {
61
62         private Image fElementImage;
63
64         public ExclusionInclusionLabelProvider(ImageDescriptor descriptor) {
65             ImageDescriptorRegistry registry= JavaPlugin.getImageDescriptorRegistry();
66             fElementImage= registry.get(descriptor);
67         }
68
69         public Image getImage(Object element) {
70             return fElementImage;
71         }
72
73         public String getText(Object element) {
74             return BasicElementLabels.getFilePattern((String) element);
75         }
76     }
77
78
79
80     private ListDialogField fInclusionPatternList; 2 private static final String PAGE_NAME="SetFilterWizardPage";
81     private ListDialogField fExclusionPatternList;
82     private CListElement fCurrElement;
83     private IProject fCurrProject;
84
85     private IContainer fCurrSourceFolder;
86
87     private static final int IDX_ADD= 0;
88     private static final int IDX_ADD_MULTIPLE= 1;
89     private static final int IDX_EDIT= 2;
90     private static final int IDX_REMOVE= 3; 3
91
92
93     public ExclusionInclusionDialog(Shell parent, CListElement entryToEdit, boolean focusOnExcluded) { 4 5 6
94         super(parent); 7
95         fCurrElement= entryToEdit; 8
96         setTitle(NewWizardMessages.ExclusionInclusionDialog_title);

```

Actions and Goals in editing

- Actions
 - ⇒ Backspace, select and delete, refactoring tools
- Goals
 - ⇒ Renaming a variable, inserting a field

Use cases for CSeR

No	Use Case ID	Name
1	UC001	Consistent Tracking
2	UC002	Simple Name
3	UC003	Statements
4	UC004	Arguments
5	UC005	Parameters
6	UC006	Expressions
7	UC007	Comments
8	UC008	Keywords
9	UC009	Fields
10	UC010	Multiple Edit
11	UC011	Delete Operation
11	UC012	Conditional Statements

Implementation highlights

While copy and paste

- Build ASTs
- Find ranges of ASTs
- Keep correspondence between ranges

While editing

- Keep track of these ranges
- Build ASTs, calculate changes by comparing the AST *in the edited range*.
- Show changes in editor

Correspondence

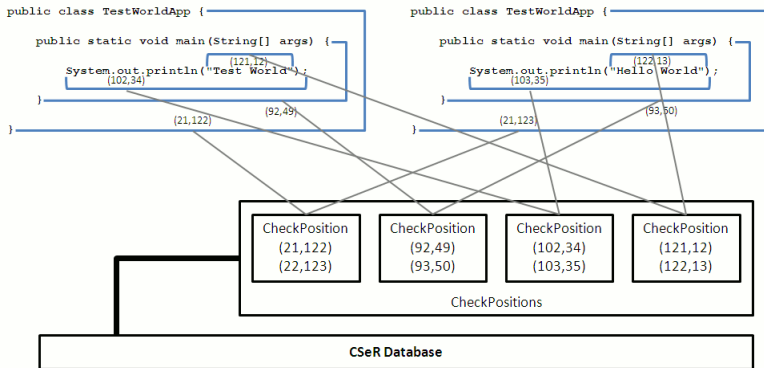


Figure: Correspondence between HelloWorldApp & TestWorldApp

Keeping correspondence while editing

Table: HelloWorldApp & TestWorldApp ranges

No	Range Description	Orginal	Current	Recent
1	Entire package name - package com. sample	0, 19	0, 19	0, 19
2	First part package - com	8, 3	8, 3	8, 3
3	Second part package - sample	12, 6	12, 6	12, 6
4	Class declaration - public class Hello..	21, 124	21, 123	21, 122
5	Class name - HelloWorldApp	34, 13	34, 12	34, 12
6	Method declaration - public static voi..	54, 89	53, 89	53, 88
7	Method name - main	73, 4	72, 4	72, 4
8	Parameters - String[] args	78, 13	77, 13	77, 13
9	Parameter type name - String	78, 6	77, 6	77, 6
10	Parameter name - args	87, 4	86, 4	86, 4
11	Method block - { System. out. println(..	93, 50	92, 50	92, 49
12	Expression statement - Syst..World");	103, 34	102, 34	102, 33
13	Method invocation - Syst..elloWorld")	103, 33	102, 33	102, 32
14	Simple name - System	103, 6	102, 6	102, 6
15	Simple name - out	110, 3	109, 3	109, 3
16	Simple name -println	114, 7	113, 7	113, 7
17	String literal - "Hello World"	122, 13	121, 13	121, 12

Correspondence

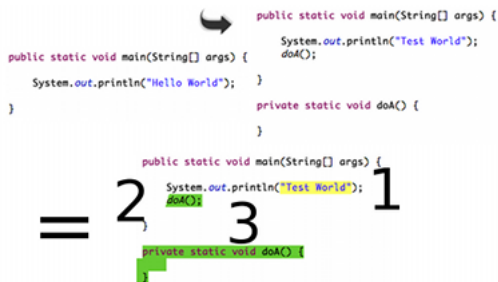


Figure: Three type of changes

Editing scenarios

No	Type	Goal Description	Action Description	Implemented
1	Names	Creating a name	Paste or Type	✓
2		Replacing part	Paste or Type	✓
3		Correcting typos	Backspace and Type	✓
4		Replacing name	Backspace, Type or Paste	✓
5		Removing name	Backspace, Delete or Type	✓
6		Splitting a name	Type in between	✓
7		Renaming	Using tools	×
8	Lists	Creating a new list	Type or Paste	✓
9		Inserting a new element	Type or Paste	✓
10		Removing an element	Delete, Type or Backspace	✓
11		Moving an element	Cut and Paste or Copy Paste and Delete	✓
12		Removing entire list	Backspace or Delete	✓
13		Flattening a list inside a list	Backspace or Delete	✓
14		Inserting a new expression	Type or Paste	✓
15	Expressions	Updating an expression	Type or Paste	✓
17		Moving an expression	Cut and Paste or Copy Paste and Delete	✓
16	Comments	Removing an expression	Delete, Type or Backspace	✓
18		Commenting code	Type Line or Block comment	✓
19		Creating an annotation		×
20		Commenting in expression	Type Block comments	✓
21	Keywords	Inserting a keyword		×
22		Removing a keyword		×
23		Updating a keyword		×

Text-based tools

Popular tools

- diff
- CompareEditor in eclipse
- Version editor

In comparison with CSeR

- Text-based
- Unable to distinguish code and comments
- Unable to find moved code
- Wrong correspondences

Text-based tools example

```

public class NewClassCreationWizardPage extends N
    protected final static String PAGE_NAME = "Ne
    protected static final int MAX_FIELD_CHARS =

    // field IDs
    protected static final int SOURCE_FOLDER_ID =
    protected static final int NAMESPACE_ID = 2;
    protected static final int CLASS_NAME_ID = 4;
    protected static final int BASE_CLASSES_ID =
    protected static final int METHOD_STUBS_ID =
    protected static final int HEADER_FILE_ID = 3
    protected static final int SOURCE_FILE_ID = 6
    protected static final int ALL_FIELDS = SOURC
        | CLASS_NAME_ID | BASE_CLASSES_ID | M
        | HEADER_FILE_ID | SOURCE_FILE_ID;
    protected int fLastFocusedField = 0;

    protected StringButtonDialogField fSourceFold
    protected SelectionButtonDialogField fNamespa
    protected StringButtonDialogField fNamespaced

import org.eclipse.jdt.core.IJavaElement;

import org.eclipse.jdt.internal.ui.IJavaHelpContextIds;
import org.eclipse.jdt.internal.ui.wizards.NewWizardMessages;

/**
 * Wizard page to create a new interface.
 * <p>
 * * Note: This class is not intended to be subclassed, but client
 * * To implement a different kind of a new interface wizard page,
 * * </p>
 * *
 * * @since 2.0
 * *
 * * @noextend This class is not intended to be subclassed by clie
 * */
public class NewInterfaceWizardPage extends NewTypeWizardPage {

    private final static String PAGE_NAME= "NewInterfaceWizardPa

/**

```

Text-based tools example

```

public class NewClassCreationWizardPage extends N
    protected final static String PAGE_NAME = "Ne
    protected static final int MAX_FIELD_CHARS =

    // field IDs
    protected static final int SOURCE_FOLDER_ID =
    protected static final int NAMESPACE_ID = 2;
    protected static final int CLASS_NAME_ID = 4;
    protected static final int BASE_CLASSES_ID =
    protected static final int METHOD_STUBS_ID =
    protected static final int HEADER_FILE_ID = 3
    protected static final int SOURCE_FILE_ID = 6
    protected static final int ALL_FIELDS = SOURC
        | CLASS_NAME_ID | BASE_CLASSES_ID | M
        | HEADER_FILE_ID | SOURCE_FILE_ID;
    protected int fLastFocusedField = 0;

    protected StringButtonDialogField fSourceFold
    protected SelectionButtonDialogField fNamespa
    protected StringButtonDialogField fNamespaced

import org.eclipse.jdt.core.IJavaElement;
import org.eclipse.jdt.internal.ui.IJavaHelpContextIds;
import org.eclipse.jdt.internal.ui.wizards.NewWizardMessages;

    Wizard page to create a new interface.
    <p>
    Note: This class is not intended to be subclassed, but client
    To implement a different kind of a new interface wizard page,
    </p>
    @since 2.0
    @noextend This class is not intended to be subclassed by clie
public class NewInterfaceWizardPage extends NewTypeWizardPage {
    private final static String PAGE_NAME= "NewInterfaceWizardPa

/**

```


AST-based Tools

Popular tools

- Breakaway
 - Changedistiller
-
- Level mismatch
 - Batch mode
 - Breakaway - For generalization tasks
 - ⇒ First match approach based on similarity threshold
 - ⇒ UnOrdered & ordered mismatches
 - Changedistiller - Comparing version of a Class
 - ⇒ Limited move operations

Experiment set up

Experiment Goal

Recreating the scenario through the following steps

- 1 Identify clones: say FileA, FileB are clones
- 2 Rename second File, FileB → OrginalFileB
- 3 Copy and paste first file with second file name, copy FileA & paste as FileB
- 4 Make changes in pasted file to make it second file, FileB=OrginalFileB

Experiments results overview

Eclipse (19), JavaLobby Community Platform (27), Literature (10)

No	Type Overall Distribution (517)	Description	Internal Distribution
1	Update (52 %, 268)	Variable Name (V_n)	50 %
2		Variable Type (V_t)	26 %
3		Literal (L)	9 %
4		Method (M)	13 %
5		Other (O)	<1 %
6	Delete (32 %, 165)	Statement (S)	40 %
7		Method Declaration (M)	30 %
8		Field Declaration (F)	20 %
9		Expression (E)	8 %
10		Parameter (P)	2 %
11		Class Declaration (C)	< 1 %
12	Insert (14, 72 %)	Statement (S)	37 %
16		Field Declaration (F)	37 %
15		Method Declaration (M)	16 %
13		Expression (E)	4 %
14		Parameter (P)	4 %
17		Class Declaration (C)	2 %
18	Move (2 %, 12)	Method Declaration (M)	58 %
19		Statement (S)	33 %
20		Class Declaration (C)	8 %

Conclusion & future work

Conclusion

- Code editor for tracking and visualizing changes continuously
- Different from existing tools
- Useful, robust and unique

Possible extensions in CSeR

- Clone groups
- Tracking code and identifying templates
- Version control integration

Thank You

Questions ?

Acknowledgements

I would like to thank my Ph.D. advisor, Dr. Jeff Gray (Associate Professor, University of Alabama), for sponsoring my trip to IWSC'10.