An Ethnographic Study of Copy and Paste Programming Practices in OOPL

Miryung Kim[1], Lawrence Bergman[2], Tessa Lau[2], and David Notkin[1]

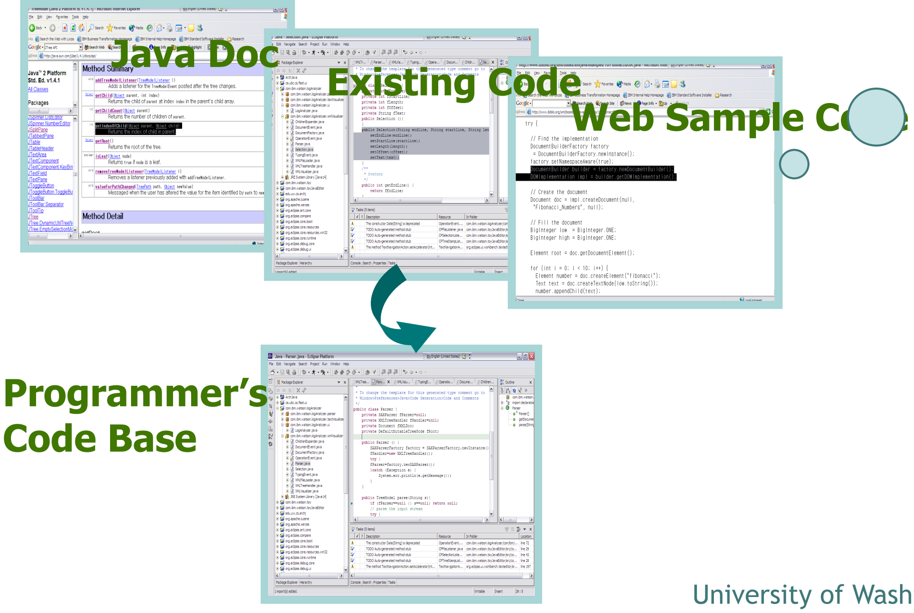
Department of Computer Science and Engineering

University of Washington[1],

IBM T.J. Watson Research Center[2]

Conventional Wisdom

Common but
Bad
Programming Practice



Java Doc

Existing Code

Web Sample Code

```
try {

  // Find the implementation
  DocumentBuilderFactory factory
   = DocumentBuilderFactory.newInstance();
  factory.setNamespaceAware(true);
  DocumentBuilder builder = factory.newDocumentBuilder();
  DOMImplementation impl = builder.getDOMImplementation();

  // Create the document
  Document doc = impl.createDocument(null,
   "Fibonacci_Numbers", null);

  // Fill the document
  BigInteger low  = BigInteger.ONE;
  BigInteger high = BigInteger.ONE;

  Element root = doc.getDocumentElement();

  for (int i = 0; i < 10; i++) {
    Element number = doc.createElement("fibonacci");
    Text text = doc.createTextNode(low.toString());
    number.appendChild(text);
```

Programmer's Code Base

```
 * To change the template for this generated type comment go to
 * Window>Preferences>Java>Code Generation>Code and Comments
 */
public class Parser {
    private SAXParser fParser=null;
    private XMLTreeHandler fHandler=null;
    private Document fXMLDoc;
    private DefaultMutableTreeNode fRoot;

    public Parser () {
        SAXParserFactory factory = SAXParserFactory.newInstance()
        fHandler=new XMLTreeHandler();
        try {
        fParser=factory.newSAXParser();
        }catch (Exception e) {
            System.err.println(e.getMessage());
        }
    }

    public TreeModel parse(String s){
        if (fParser==null || s==null) return null;
        // parse the input stream
        try {
```

	✓	!	Description	Resource	In Folder	Location
⚠			The constructor Date(String) is deprecated	OperationEvent....	com.ibm.watson.logAnalyzer/com/ibm/...	line 72
✓			TODO Auto-generated method stub	CPFileListener.java	com.ibm.watson.toyJavaEditor/src/co...	line 29
✓			TODO Auto-generated method stub	CPSelectionListe...	com.ibm.watson.toyJavaEditor/src/co...	line 42
✓			TODO Auto-generated method stub	CPTimeStampList...	com.ibm.watson.toyJavaEditor/src/co...	line 28
⚠			The method TextNavigationAction.setAccelerator(int...	TextNavigationA...	org.eclipse.ui.workbench.texteditor/sr...	line 197

University of Washington
IBM T.J. Watson Research Center

Contribution

- We address implications of copy and paste (C&P) programming practices.
  - Not only about saving typing.
  - C&P capture design decisions.
  - Programmers actively employ C&P history.
  - With tool support, programmers' intent of C&P can be expressed in a safer and more efficient manner.

Research Questions

- What are C&P usage patterns?
- Why do people copy and paste code?
- What kind of tool support is needed for C&P usage patterns?

Outline

- Ethnographic Study: Observation and Analysis
- Taxonomy
- Insights and Tool Ideas

Observation

- preliminary approach
  - direct observation
  - questions asked during observation
  - easy to identify intentions
  - unnatural coding behavior

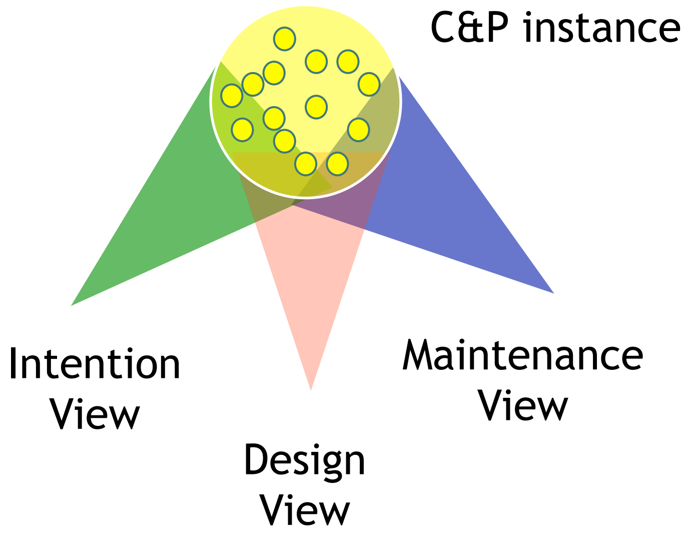
- final approach
  - logging editing operations with an instrumented text editor
  - replaying off-line
  - interviews
  - non-intrusive observation

Study Setting

	Direct Observation	Observation using a logger and a replayer
Subjects	researchers and summer students at IBM T.J. Watson	
No. of Subjects	4	5
Hours	about 10 hrs	about 50 hrs
Interviews	questions asked during observation	twice after analysis (30 mins – 1 hour/ each)
Programming Languages	Java, C++, and Jython	Java

Analysis

- contextual inquiry [Beyer98]
  - affinity process: developing hypotheses from data points
- data analysis from multiple perspectives



Outline

- Ethnographic Study: Observation and Analysis
- Taxonomy
- Insights and Tool Ideas

Programmers’ Intentions

- relocate/ regroup/ reorganize
- reorder
- refactoring
- reuse as a structural template
  - syntactic template
  - semantic template

Example – Syntactic Template

```
static {
        protectedClasses.add(“java.lang.Object”);
        protectedClasses.add(“java.lang.ref.Reference
    $ReferenceHandler”);
        protectedClasses.add(“java.lang.ref.Reference”);
        protectedClasses.add(“java.lang.ref.Reference$1”);
        protectedClasses.add(“java.lang.ref.Reference$Lock”);
        protectedMethods.add(“java.lang.Thread<init>”);
        protectedMethods.add(“java.lang.Object<init>”);

    protectedMethods.add(“java.lang.Thread.getThreadGroup”);
}
```

Semantic Template

- design patterns
- control structures
  - if – then – else
  - loop construct
- usage of a module
  - data structure access protocols

Example – Semantic Template: Usage of a Module

```
DOMNodeList *children = doc->getChildNodes();
int numChildren = children->getLength();

for (int i=0; i<numChildren; ++i)
    {
        DOMNode *child = (children->item(i));
        if (child->getNodeType() == DOMNode.ELEMENT_NODE)
        {
            DOMElement *element = (DOMElement*)child;
```

Code Snippets:
traverse over *Elements*
in a *Document*

Design

Design View

What are underlying design decisions that induce programmers to C&P in particular patterns?

- Why is text copied and pasted over and over in scattered places?
- Why are blocks of text copied together?
- What is the relationship between copied text and pasted text?

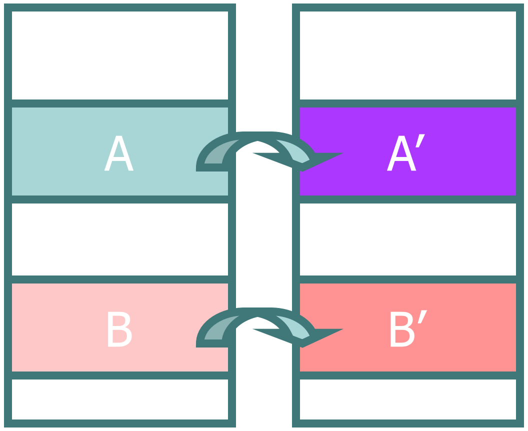
Why is text copied and pasted repeatedly?

- lack of modularity
  - crosscutting concerns
  - example – logging concern

```
if (logAllOperations) {
          try {
          PrintWriter w = getOutput();
          w.write(“$$$$$");
          ..
          } catch (IOException e) {
   }
}
```

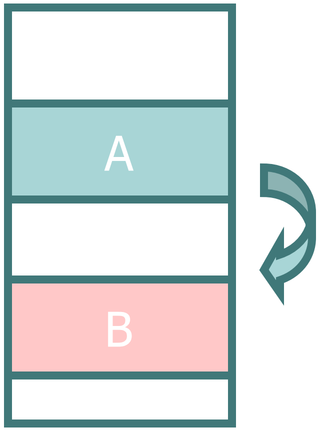
Design

Why are blocks of text copied together?



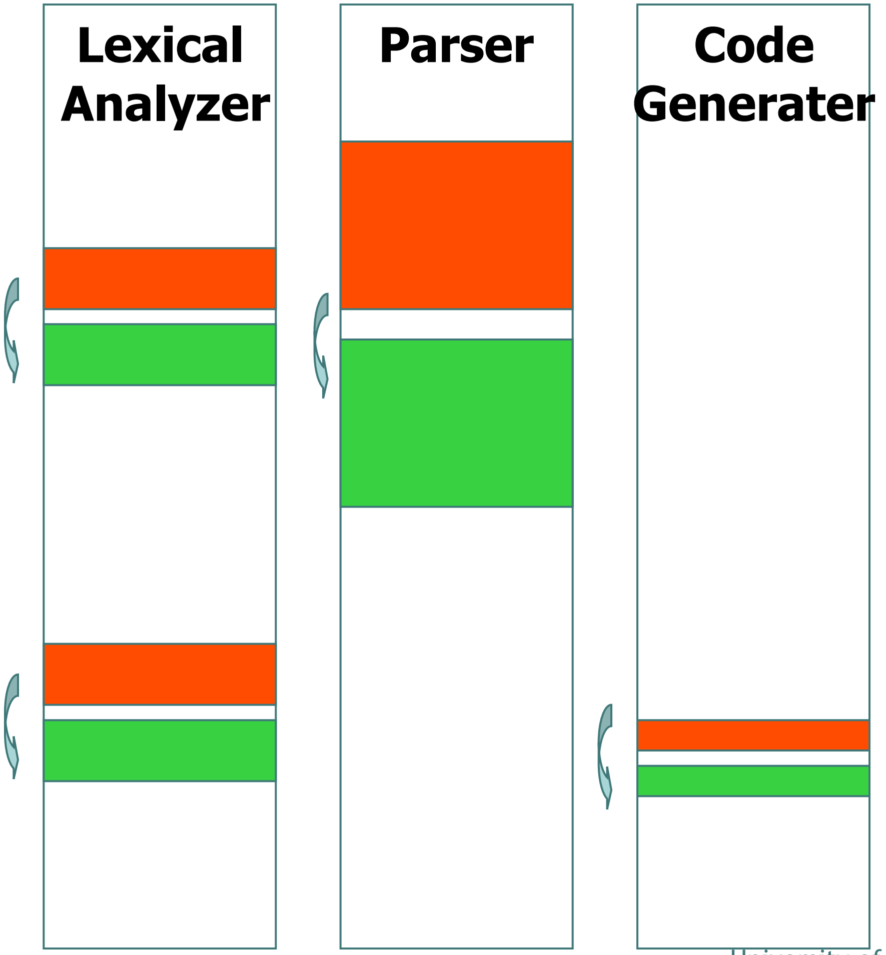
- comments
- references fields and constants
- caller method and callee method
- paired operations
  - openFile, closeFile, and writeToFile
  - enterCriticalSection, leaveCriticalSection

What is the relationship between copied and pasted text?



- type dependencies
- similar operations but different data structure
- parallel crosscutting concerns [Griswold01]

Example - Parallel Crosscutting Concern



- Parallel concerns are independent concerns but they crosscut a system in the similar way
- XML compiler
  - serialize
  - appendChildren

Maintenance Tasks

- short term
  - Programmers modify a pasted block to prevent naming conflicts.
  - Programmers remove code fragments irrelevant to the pasted context.
- long term
  - Programmers restructure code after frequent copy and paste of a large text.
  - Programmers tend to apply consistent changes to the code from the same origin.

Scope and Limitations

- programming languages
  - OOPL vs. functional PL
- development environment
  - Eclipse vs. other editors
- organization characteristics
  - team size, software lifecycle, etc
- duration of study
  - long term vs. short term

Outline

- Ethnographic Study: Observation and Analysis
- Taxonomy
- Insights and Tool Ideas

Insights

Limitations of particular programming languages produce unavoidable duplicates in a code base.

Insights

C&P dependencies are worth observing and maintaining

Tool requirements:

- visualize copied and pasted content
- explicitly maintain and represent C&P dependencies
- allow developers to communicate the intention behind C&P by annotation

Insights

Programmers copy an entire code snippet because it contains the structural template that they intend to reuse.

Tool requirements:

- learn a relevant structural template
- assist to modify the portion that is not part of the structural template

Insights

Programmers use their memory of C&P history to determine when to restructure code.

Tool requirements:

- monitor evolution patterns, frequency, and size of code duplicates
- suggest refactoring

Insights

Code snippets originating from the same source are likely to be changed in similar ways.

Tool requirements:

- monitor evolution of structural template within code duplicates
- warn programmers when they attempts to change inconsistently

Related Work

- study of code reuse [Lange89, Rosson93]
- information transparency [Griswold01]
- clone detection [Balazinska02, Baker92, Baxter98, Ducasse99, Kamiya02, Komondoor01, Krinke01]
- clone evolution patterns [Lague96, Antoniol02, Rysselberghe04, Godfrey04]

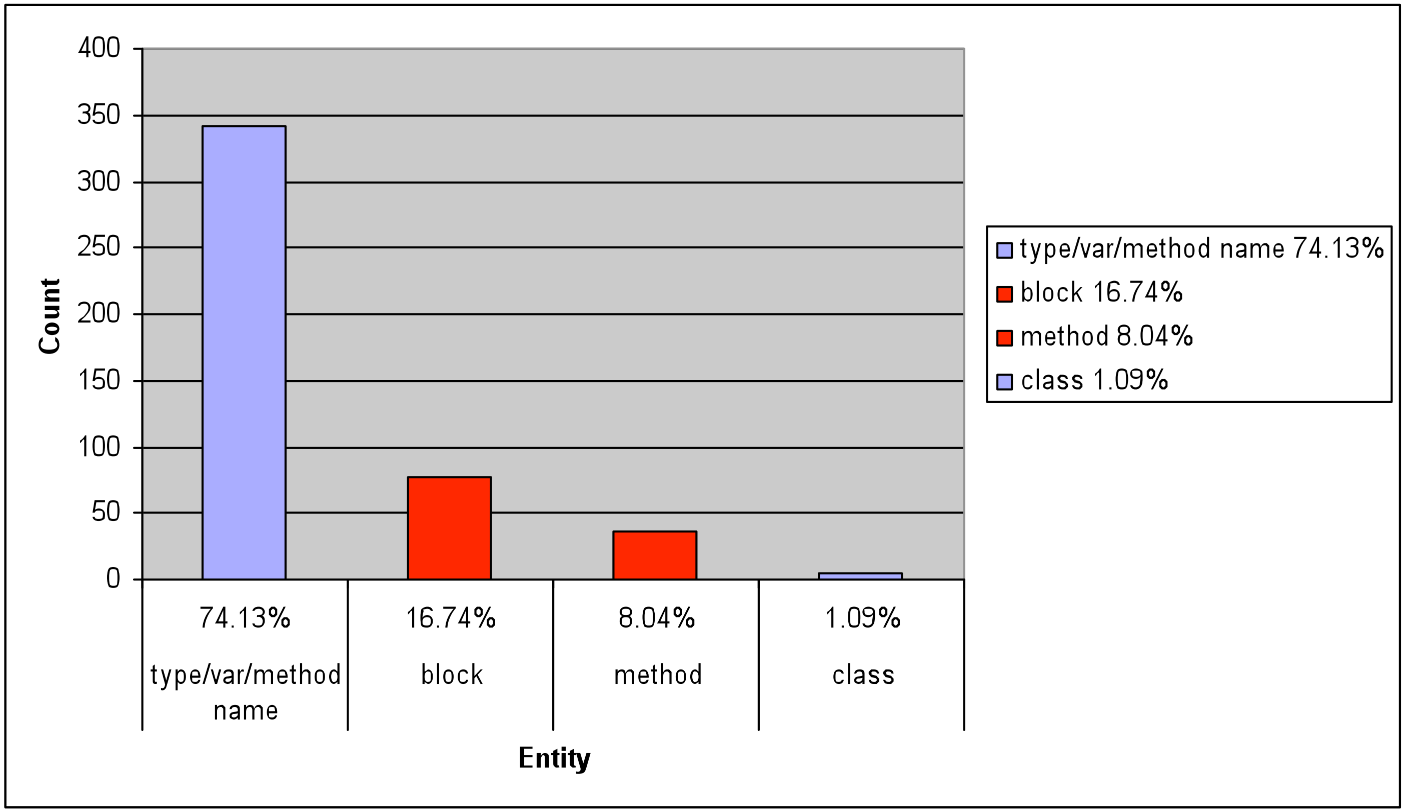
Conclusion

- development of the instrumented editor and the replayer
- study that systematically investigated C&P usage patterns and associated implications
- proposal of SE tools based on our insights

[redacted]

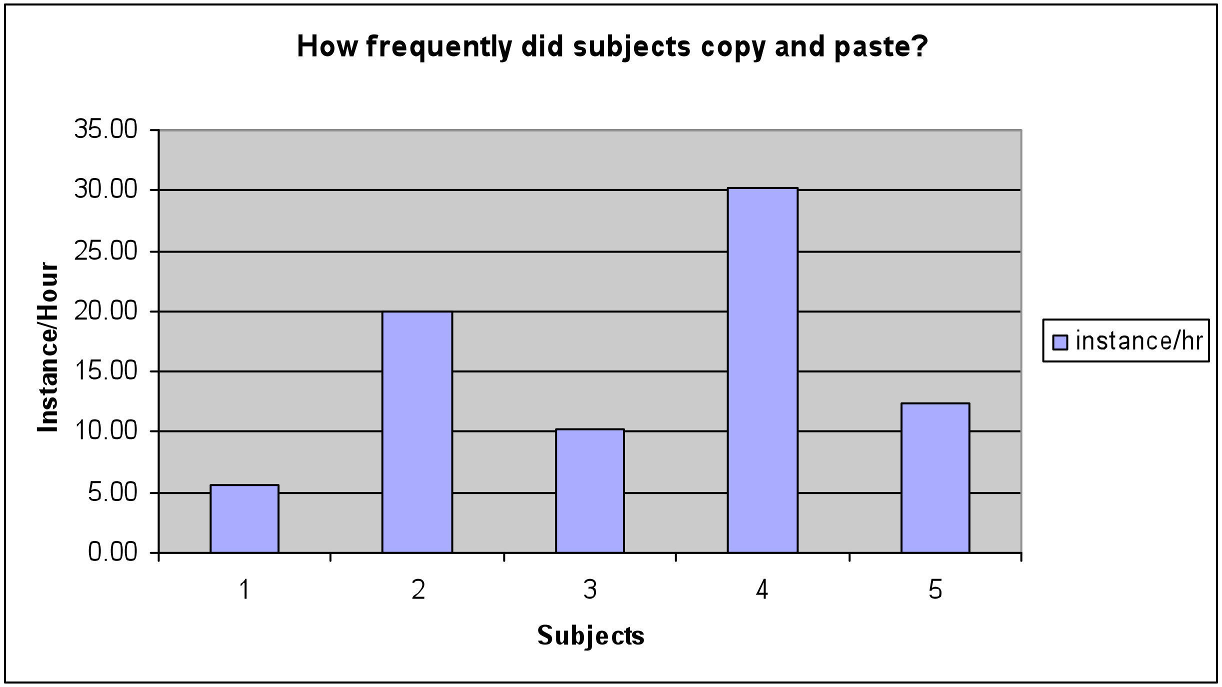
University of Washington
IBM T.J. Watson Research Center

What kind of code snippets do programmers copy and paste?

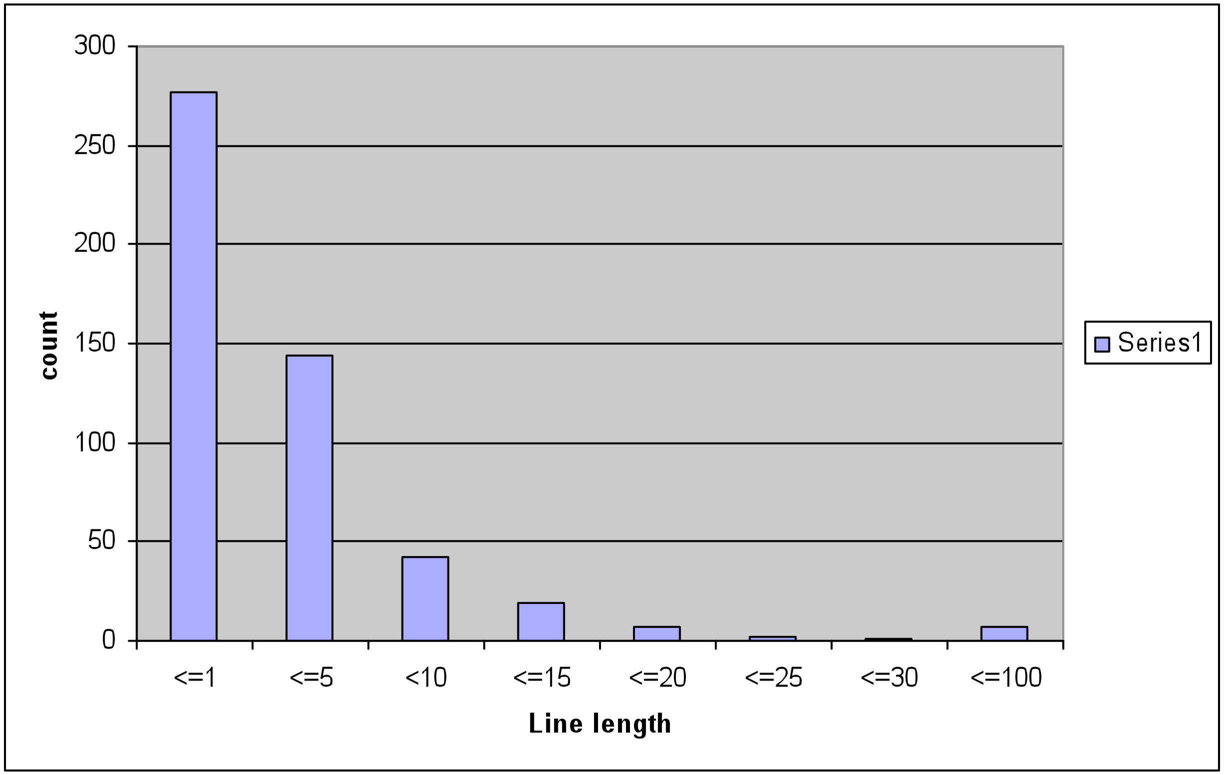


How frequently did subjects copy and paste?

- average:
about 16 inst/ hr
- median:
about 12 inst/ hr



How long is the code snippet involved in copy operations?


count
<=1
<=5
<10
<=15
<=20
<=25
<=30
<=100
Line length
Series1