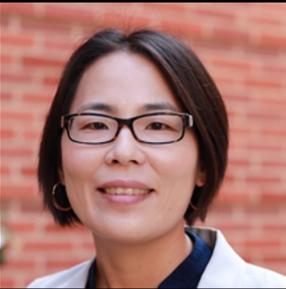


A Journey through Searching Similar Code

Miryung Kim

Professor and Vice Chair of Graduate Studies at UCLA
Amazon Scholar, Amazon Web Services



UCLA



Outline: *A Journey* through Searching Similar Code

What motivated us?

What were early attempts?

How serious is this problem?

How can we automate?

How can we examine variations at scale?

How to search with a human in the loop?

What ideas have motivated
searching similar code?



A Study of Copy and Paste Programming Practices [ISESE 2004]

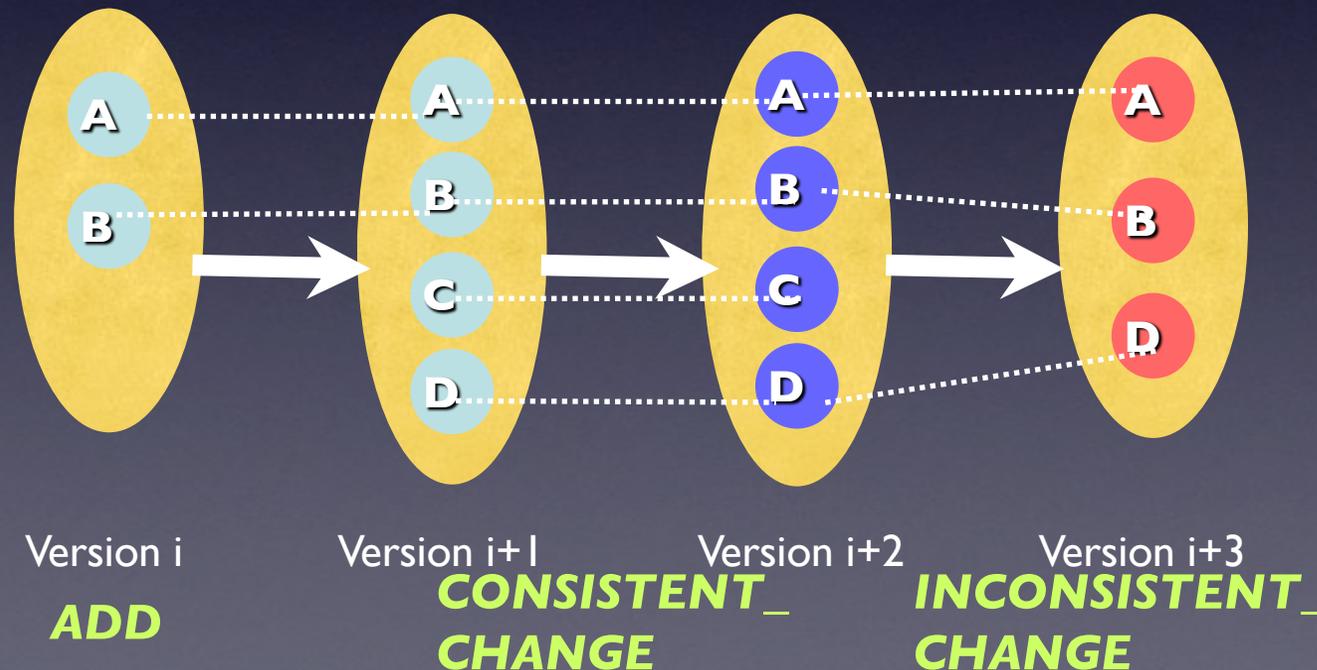
- To understand programmers' copy and paste coding behavior, we **built an Eclipse plug-in that records edits and replays the captured edits at IBM**
- Programmers often **create and manage** code clones with clear intent

An Empirical Study of Code Clone Genealogies [FSE 2015]

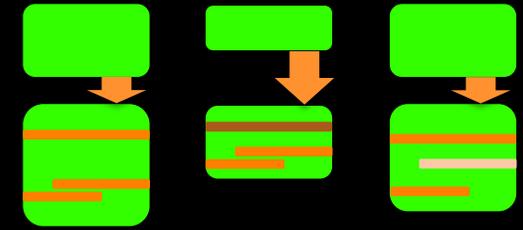
- We developed an approach that ***automatically reconstructs*** the history of code clones ***from a source code repository***
- We studied clone evolution in several Java open source projects.

Clone Genealogy

Clone genealogy is a representation that captures clone change patterns over a sequence of program versions



Systematic Changes (similar updates to similar code)



Consistent updates to clones

Managing multiple products, forked projects and versions

API evolution and ripple effects on client applications

Refactoring

What were early attempts to abstract *systematic changes*?

```
public class CmiRegistr
NameService {

    public void setPort
    ...
-   SQL.exec(query)
+   SafeSQL.exec(query)

}

}

...
```

```
public class JacORB implements NameService
{

    public void setPort (int p) {
-       if (TraceCarol.isDebugEnabled()) {
        ...
-       SQL.exec(query)
+       SafeSQL.exec(query)
    }

    ...
}
```

```
public class LmiRegistry extends
AbsRegistry implements NameService {

-   private int port = ...
-   private String host = null
    public void setPort (int p) {
        ...
-       SQL.exec(query)
+       SafeSQL.exec(query)
    }

    public int getPort() {
        return port;
    }

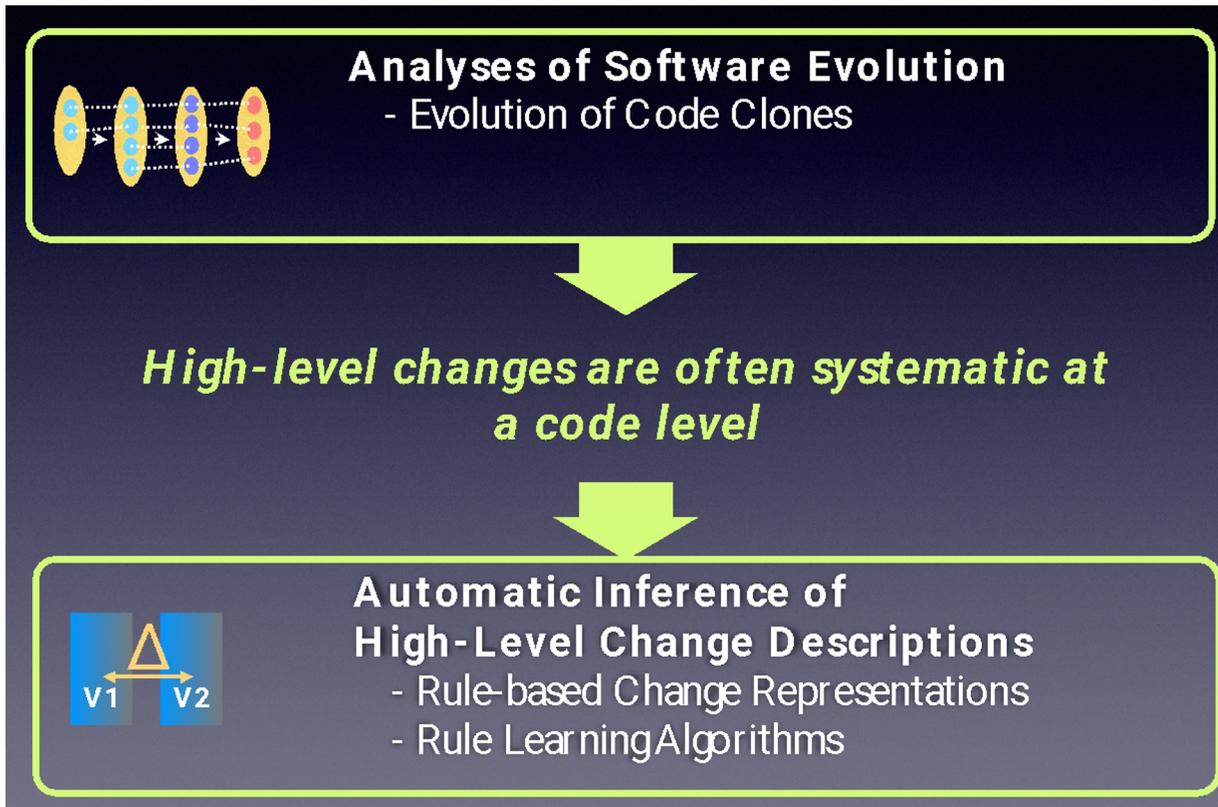
    public void setHost(String host)
```

Miryung's PhD @ U of Washington

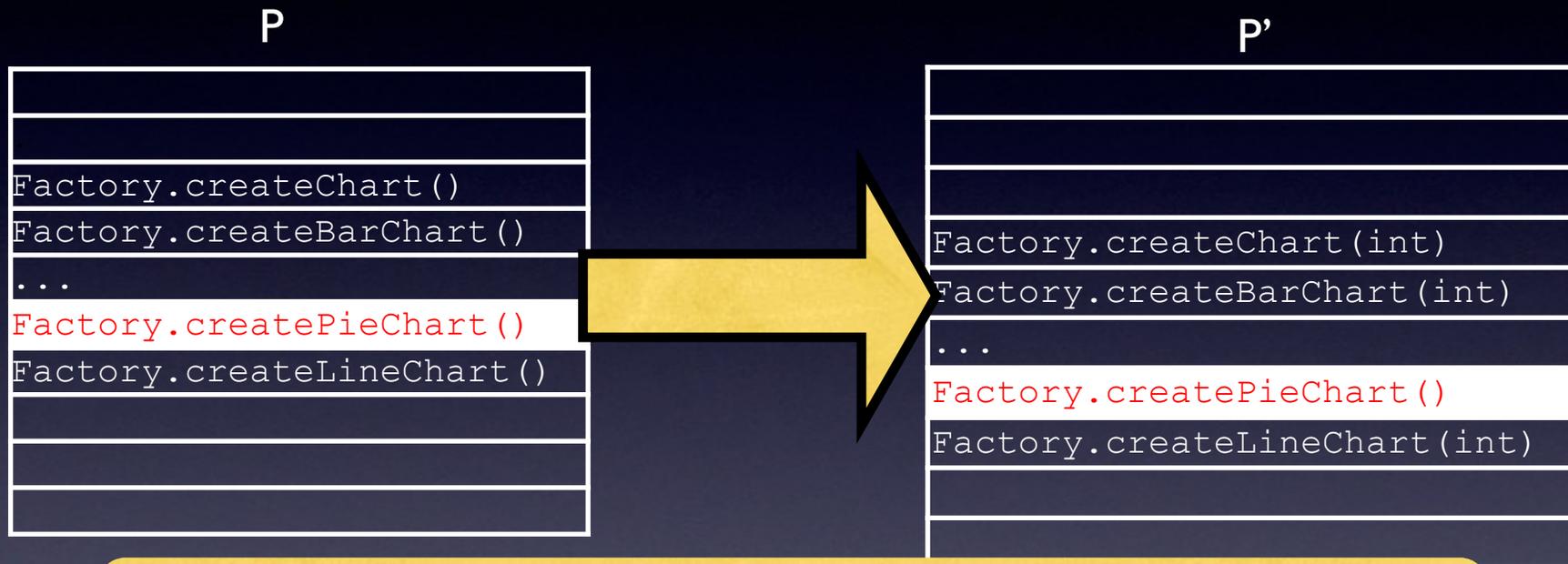
Automated Change Rule Inference



David Notkin
(1 Jan 1955 – 22 Apr 2013)



API Change Rule Inference [ICSE 2007]



```
FOR ALL x:method-header IN
  Factory.create*Chart(*)
    argAppend(x, [int])
  except {Factory.createPieChart() }
```

LSDiff Rule Inference [ICSE 2009]

- “Replace all calls to `SQL.exec` with `SafeSQL.exec`”

```
deleted_calls(m, "SQL.exec") =>  
added_calls(m, "SafeSQL.exec")
```

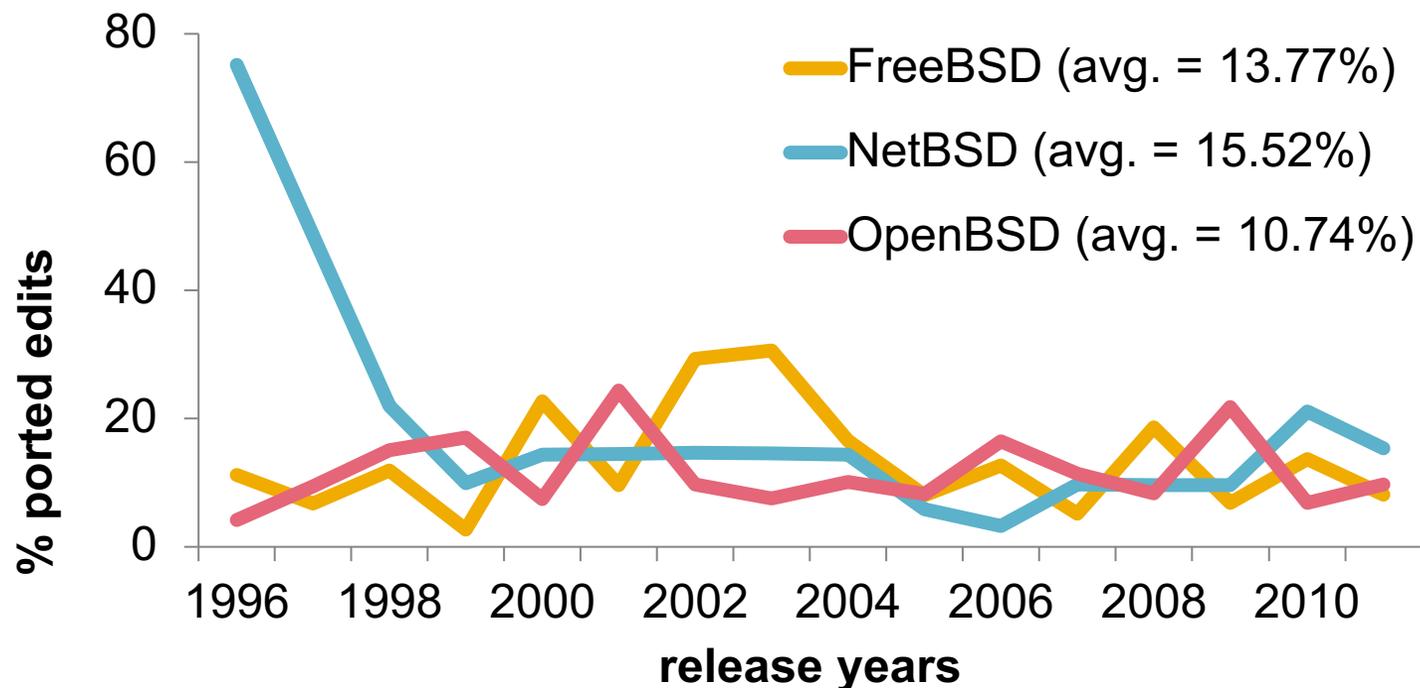
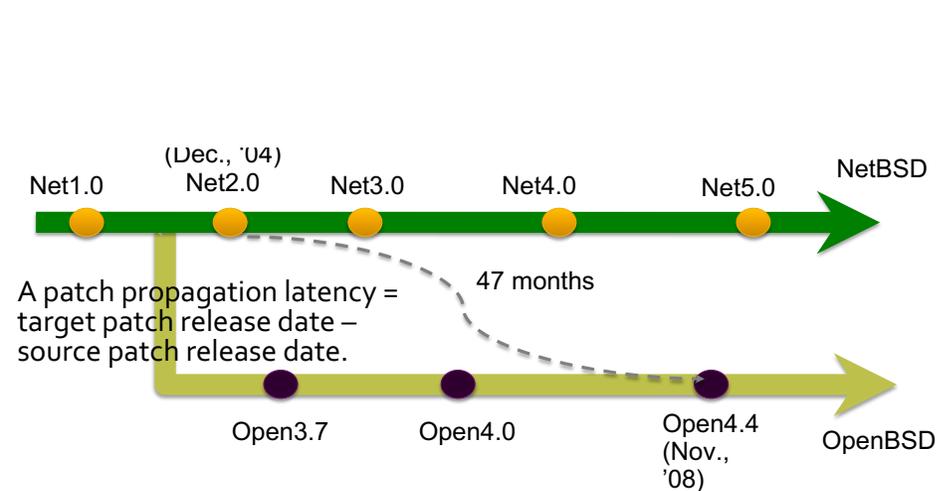
- “All `setHost` methods in `Service`'s subclasses in the old version deleted calls to `SQL.exec` except the `setHost` method in the `NameSvc` class.

```
past_subtype("Service", t)  $\wedge$  past_method(m,  
"setHost", t)  
 $\Rightarrow$  deleted_calls(m, "SQL.exec")  
except t="NameSvc"
```

How serious is this problem of searching similar code?

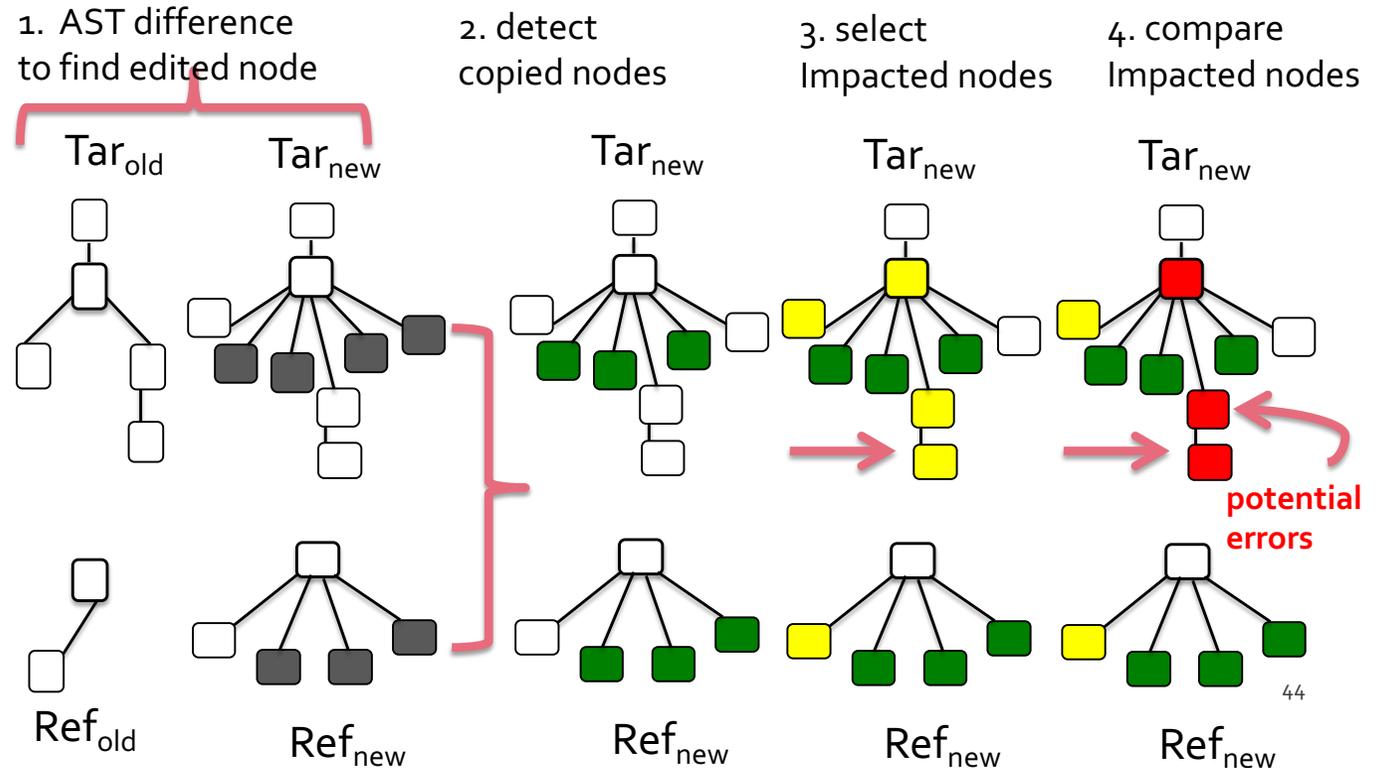
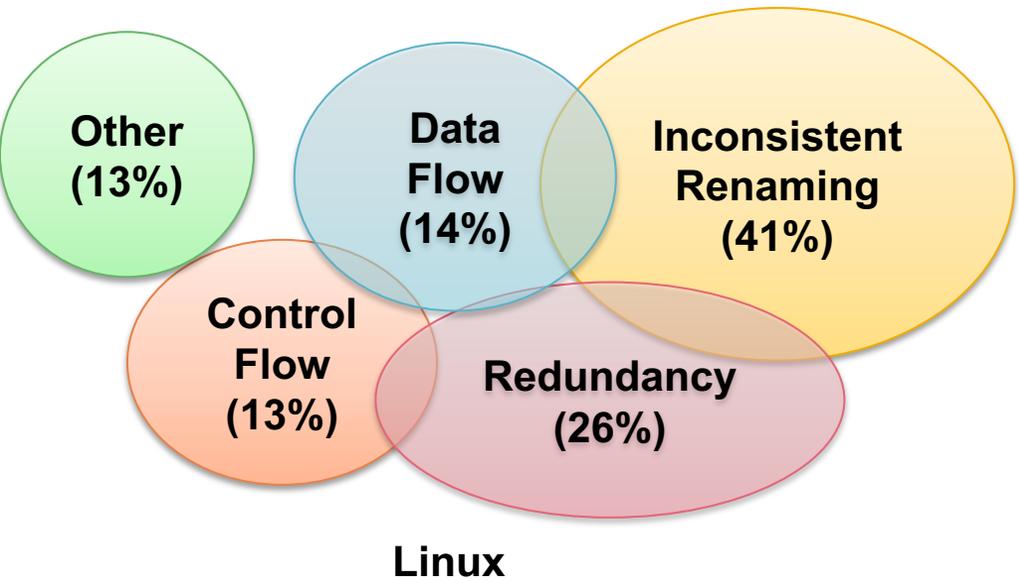


Repertoire: Cross-System Porting Analysis in Forked Projects [FSE 2012]



Porting consists of a significant portion of the BSD family evolution and a significant portion of active committers port changes

SPA: Detecting Semantic Inconsistencies in Ported Code in Linux [ASE 2013]

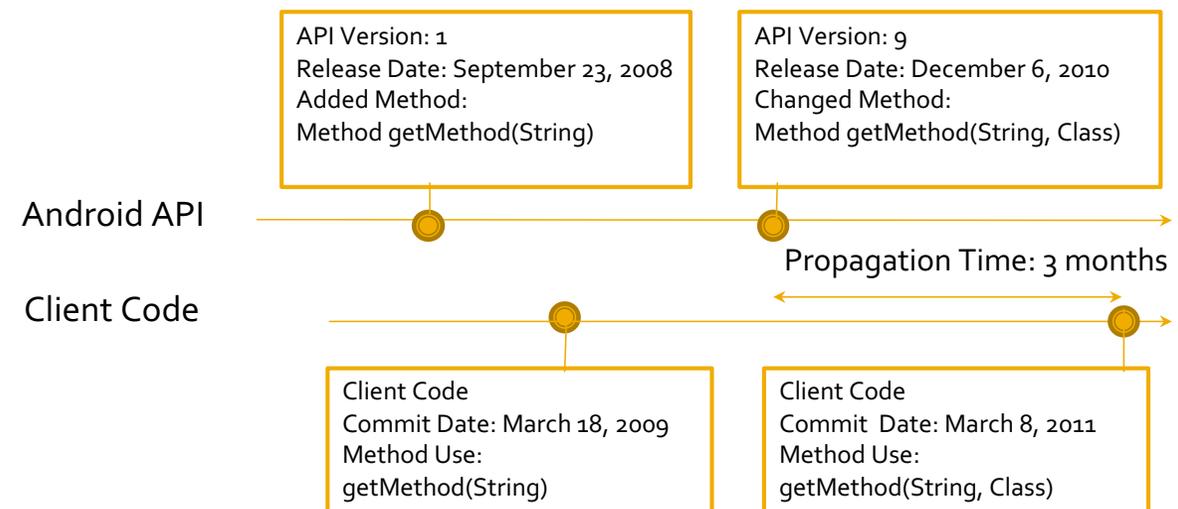
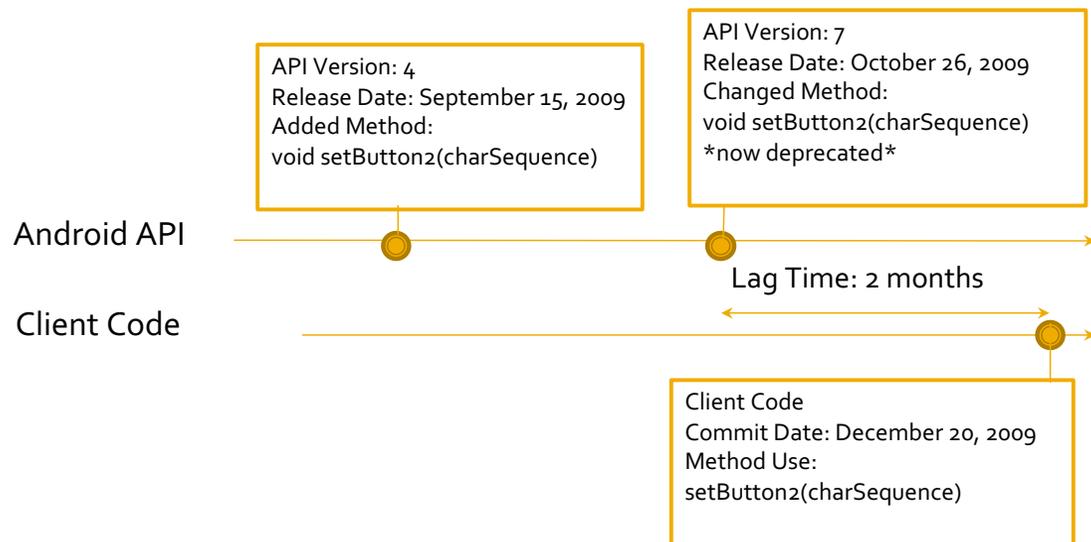


13% to 25% changes are reused in Linux and Microsoft projects

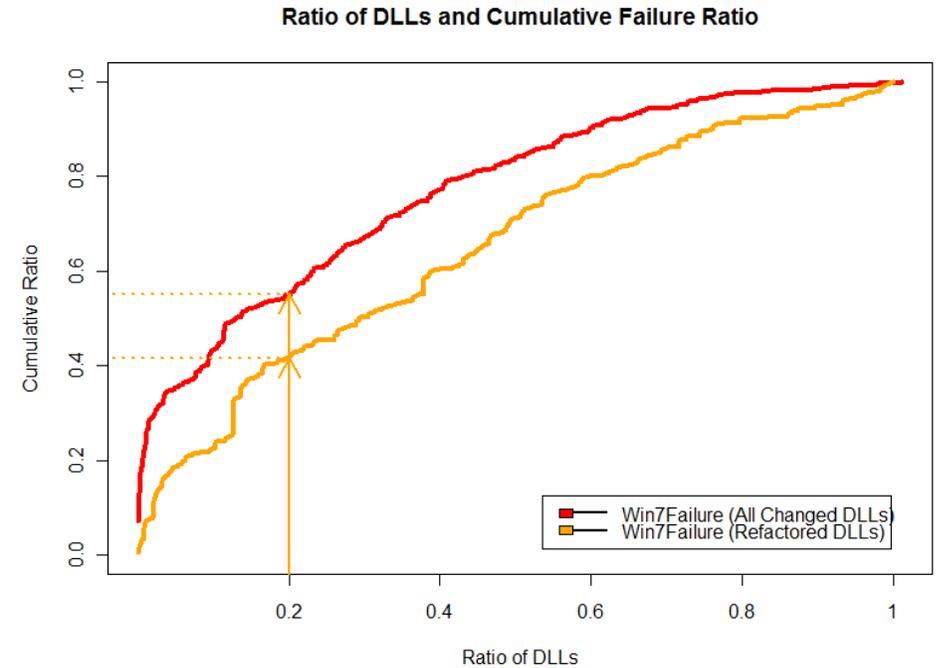
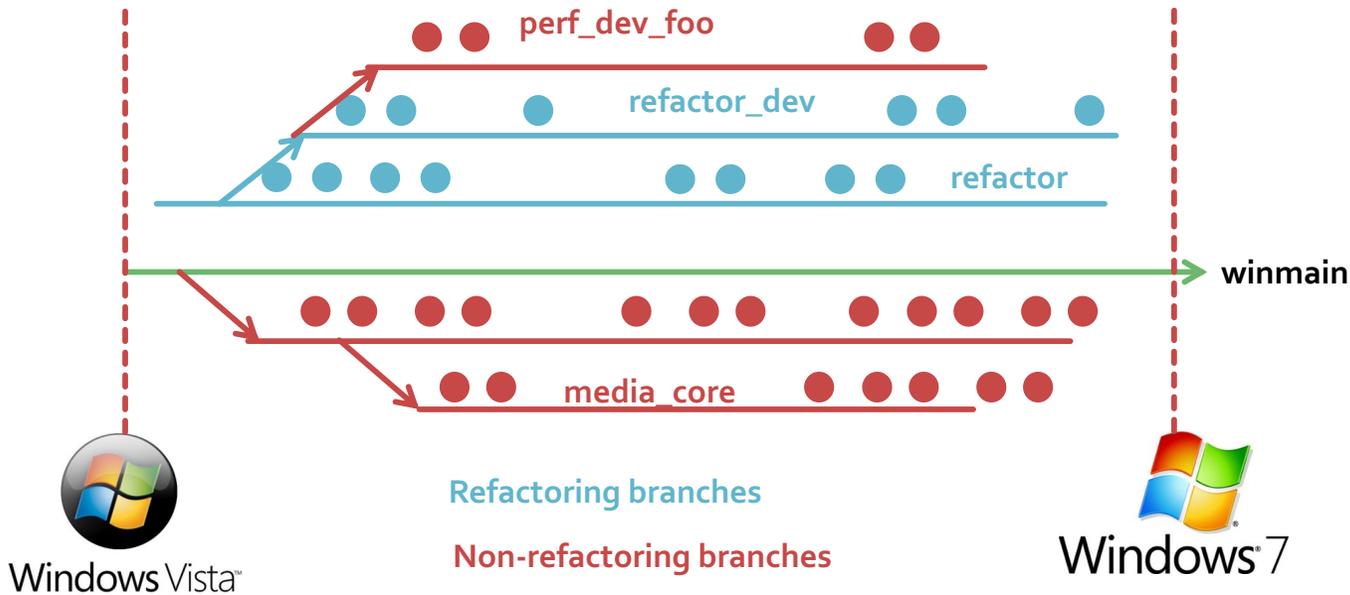
API Stability and Adoption in the Android Ecosystem [ICSME 2013]

Most Influential Paper Award from ICSME 2013

- Android is evolving fast at a rate of 115 API updates per month.
- 28% of API references in client apps are outdated with a median lagging time of 16 months.
- API usage adaptation code is **defect prone** than other code.



Microsoft: Quantifying Benefits of Windows Re-architecting [FSE '12, TSE '14]



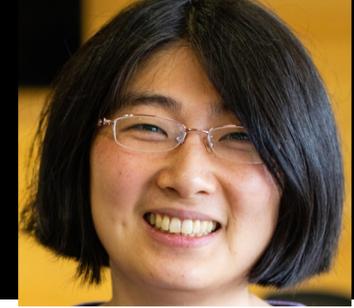
Refactoring churn is less defect-prone than regular churn.

How can we *automate*
systematic changes?



Na Meng's PhD @ UT Austin

Automating Systematic Changes

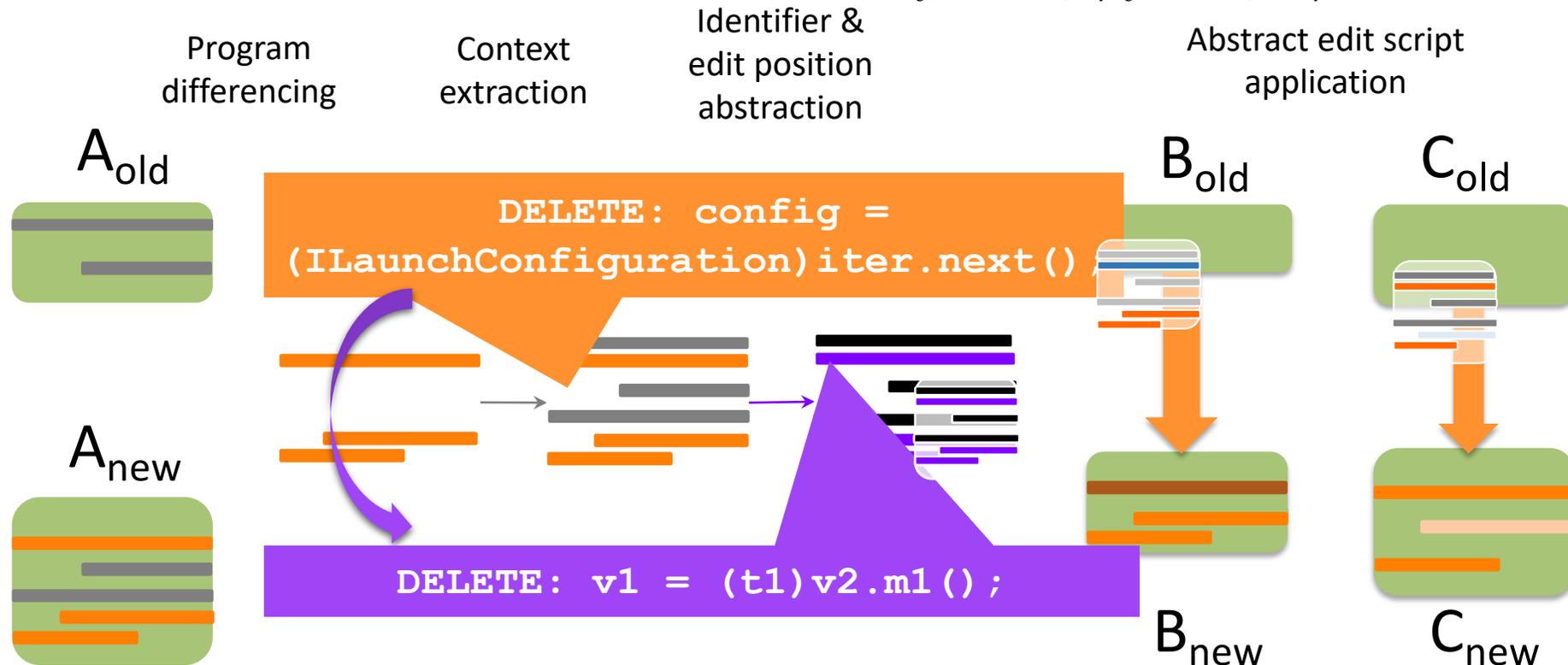


Systematic Editing: Generating Program Transformations from an Example

Na Meng Miryung Kim Kathryn S. McKinley
The University of Texas at Austin
mengna152173@gmail.com, miryung@ece.utexas.edu, mckinley@cs.utexas.edu

LASE: Locating and Applying Systematic Edits by Learning from Examples

Na Meng* Miryung Kim* Kathryn S. McKinley*†
The University of Texas at Austin* Microsoft Research†
mengna09@cs.utexas.edu, miryung@ece.utexas.edu, mckinley@microsoft.com



Systematic Editing: Generating Program Transformations from an Example [PLDI 2011]

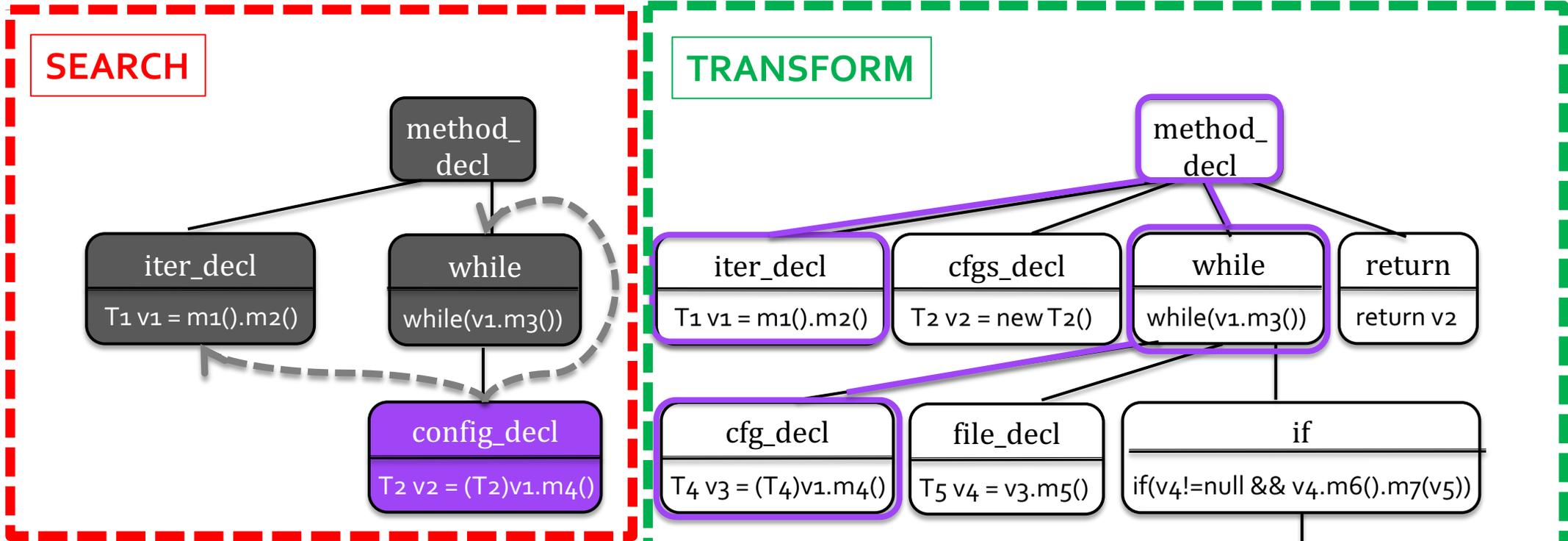
```
1. public ILaunchConfiguration[] getLaunchConfigurations
   (ILaunchConfigurationType type) throws CoreException {
2.     Iterator iter = getAllLaunchConfigurations().iterator();
3.     List configs = new ArrayList();
4.     + ILaunchConfiguration config = null;
5.     while (iter.hasNext()) {
6.         - ILaunchConfiguration config =
           (ILaunchConfiguration)iter.next();
7.         + config = (ILaunchConfiguration)iter.next();
8.         + if (!config.isValid()) {
9.             + config.reset();
10.        + }
11.        if (config.getType().equals(type)) {
12.            configs.add(config);
13.        }
14.    }
15.    return (ILaunchConfiguration[])configs.toArray
        (new ILaunchConfiguration[configs.size()]);
17.}
```

```
1. ... .. method_declaration(... ..){
2.     T1 v1 = m1().m2();
3.     ... ..
4.     while(v1.m3()){
5.     UPDATE: T2 v2 = (T2)v1.m4();
6.         TO: T2 v2 = null;
7.     INSERT: v2 = (T2)v1.m4();
8.     INSERT: if(!v2.m5()){
9.         INSERT: v2.m6();
10.    }
11.    ... ..
12.    }
13.    ... ..
14. }
```



Abstract edit script

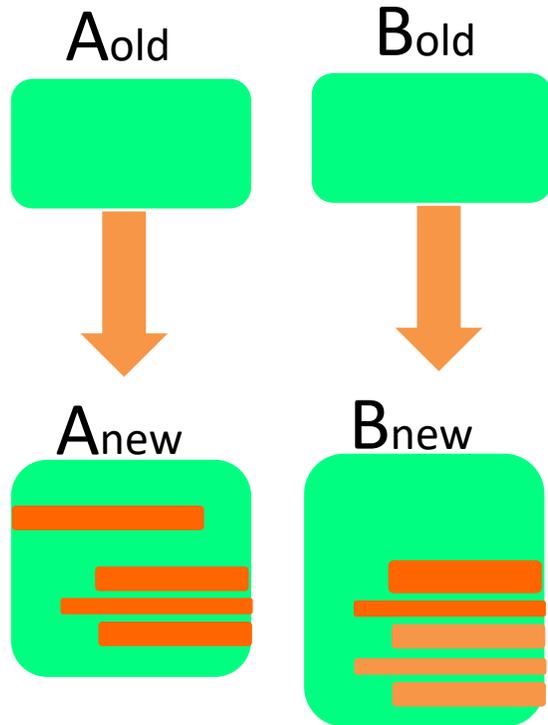
Abstract Edit Script



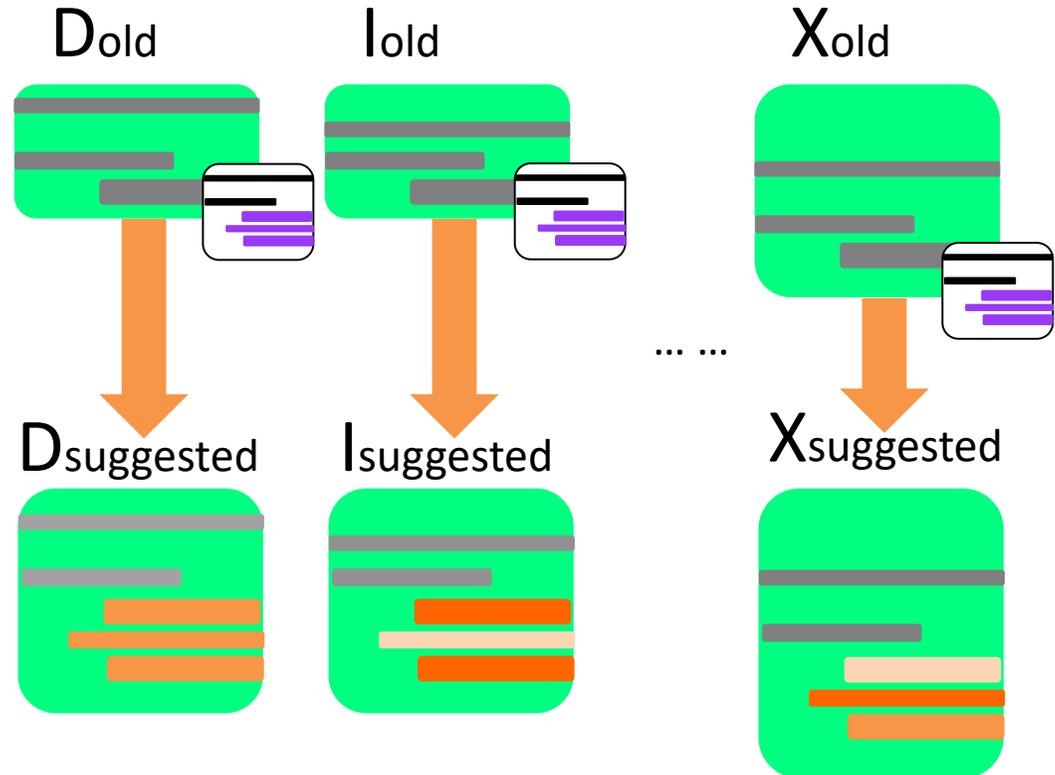
Variable Map	Method Map	Type Map
v1 <-> v1	m1 <-> m1	T1 <-> T1
v2 <-> v3	m2 <-> m2	T2 <-> T4
	m3 <-> m3	
	m4 <-> m4	

Lase: Locating and Applying Systematic Edits [ICSE 2013]

User selects examples



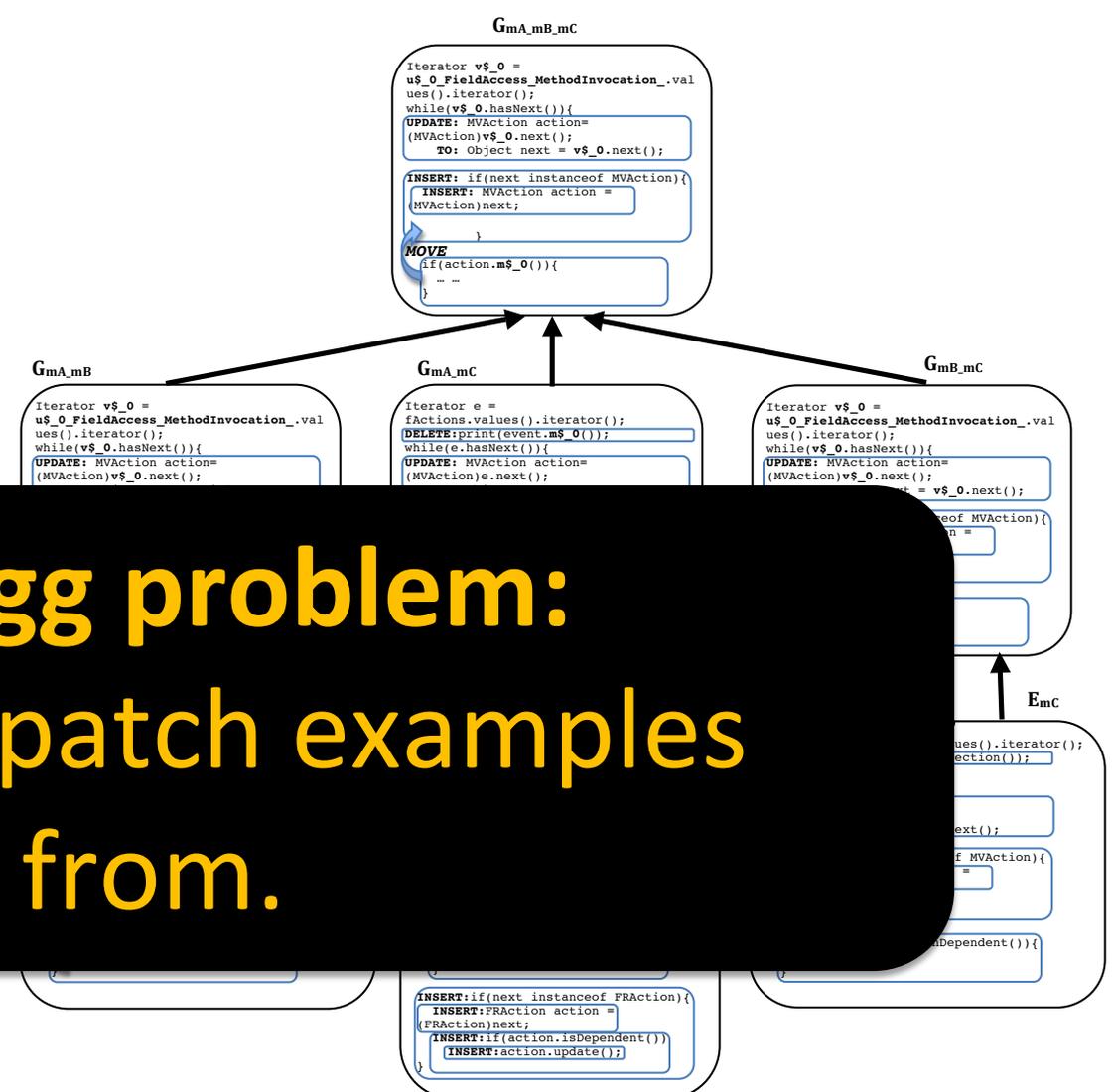
LASE selects methods & suggests edits



```
Comment[] getLeadingComments(ASTNode node){
-   if (this.leadingComments != null) {
+   if (this.leadingPts >= 0) {
-       int[] range = (int[]) this.leadingComments.get(node);
+       int[] range = null;
+       for (int i = 0; range == null && i <= this.leadingPtr; i++) {
+           if (this.leadingNodes[i] == node) range = this.leadingIndexes[i];
+       }
+       if (range != null) {
+           int length = range[1] - range[0] + 1;
+           Comment[] leadComments = new Comment[length];
+           System.arraycopy(this.comments, range[0], leadComments, 0,
```

```
Comment[] getTrailingComments(ASTNode node){
-   if (this.trailingComments != null) {
+   if (this.trailingPts >= 0) {
-       int[] range = (int[]) this.trailingComments.get(node);
+       int[] range = null;
+       for (int i = 0; range == null && i <= this.trailingPtr; i++) {
+           if (this.trailingNodes[i] == node) range = this.trailingIndexes[i];
+       }
+       if (range != null) {
+           int length = range[1] - range[0] + 1;
+           Comment[] trailComments = new Comment[length];
+           System.arraycopy(this.comments, range[0], trailComments, 0,
```

```
length);
public int getLeadingCommentsCount() {
    int end = this.leadingComments == null ? -1 : this.leadingComments.size();
    if (this.leadingComments != null) {
        if (this.leadingComments.get(0) != null) {
            int[] range = (int[]) this.leadingComments.get(0);
            int start = range[0];
            int end = range[1];
            for (int i = 0; range == null && i <= this.v$_1_; i++) {
                if (this.v$_2_[i] == node) range = this.v$_3_[i];
            }
            if (range[0] == -1 && range[1] == -1) {
                return 0;
            } else {
                return end - start + 1;
            }
        }
    }
    return end - 1;
}
```



**A chicken and egg problem:
Users needs to find patch examples
generalize from.**

Critics: *Interactive Code Search and Review [ICSE 2015]*

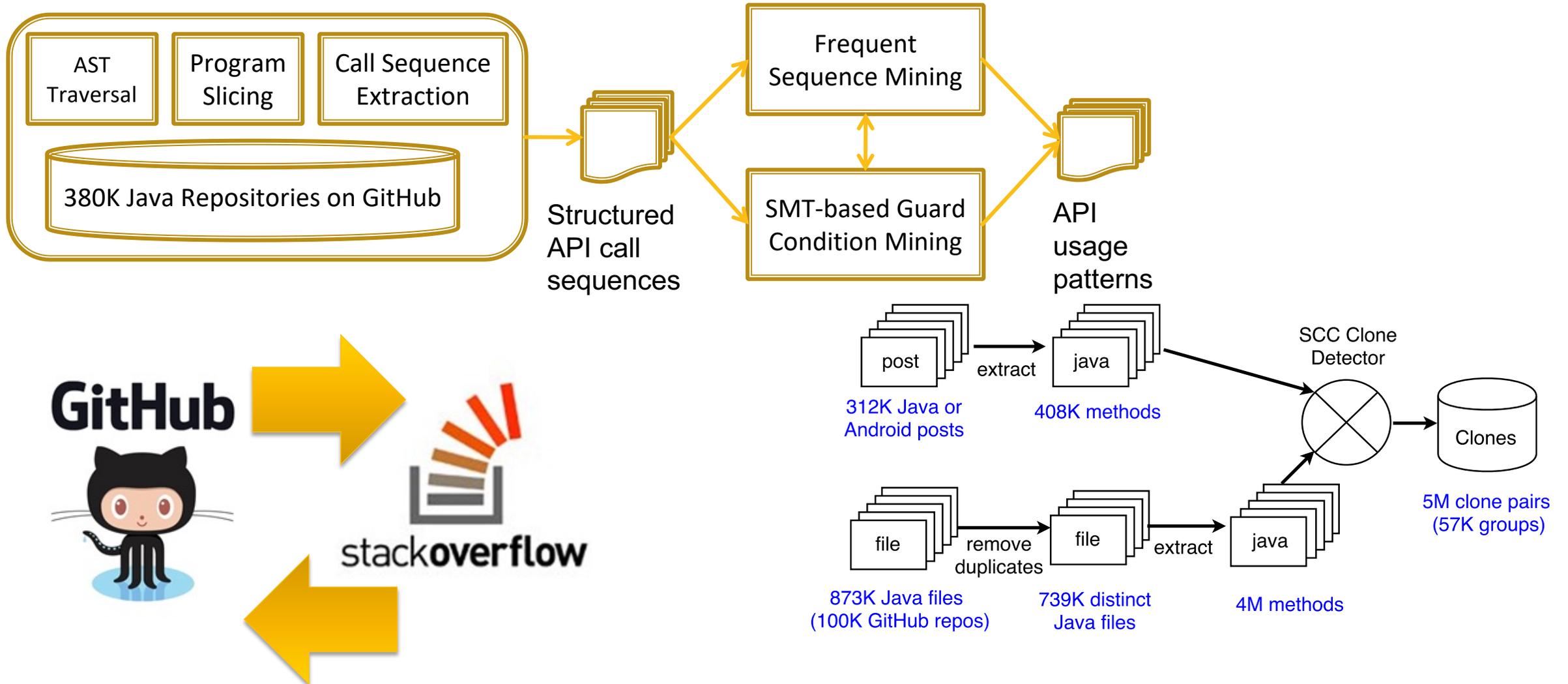
The screenshot shows the Eclipse IDE's 'Compare' window. The title bar reads 'Compare ('NEW_WIN3213516' - 'OLD_WIN3213515')'. Below the title bar, there are tabs for 'Structure Compare' and 'Java Structure Compare'. The main area is split into two panes: 'NEW_WIN3213516' on the left and 'OLD_WIN3213515' on the right. The left pane shows a code snippet: `case OS.VK_RETURN: Event event = new Event ();`. The right pane shows a corresponding code snippet: `Event event = new Event (); event.item = item; sendEvent (true, event);`. A red annotation at the top of the diff view reads: '① A user selects a sub-region of a diff patch, using Eclipse compare view.' Below the diff view, there are three panels: 'Matching Result', 'Diff Details', and 'Diff Template'. The 'Matching Result' panel shows a list of 'Matching Locations' with one entry selected: '[Eclipse SWT/win32/org/eclipse/swt/widgets][Widgets.java][keyPressedEvent]'. Below it, the 'Change Anomalies' panel shows two entries with red 'X' icons: '[Eclipse SWT/win32/org/eclipse/swt/widgets][Widgets.java][buttonUpEvent]' and '[Eclipse SWT/win32/org/eclipse/swt/widgets][Widgets.java][buttonReleaseEvent]'. The 'Diff Details' panel shows a code snippet with annotations: `int keyDownEvent(int wParam,int lParam) - ExpandItem item = items[focusIndex]; $EXCLUDED $EXCLUDED $T1 $V1 = new $T1(); - $V1.item = item; - sendEvent(true, $V1); + $V1.item = focusItem; + sendEvent((focusItem.expanded ? COL...`. The 'Diff Template' panel shows a similar code snippet with annotations: `int keyDownEvent(int wParam,int lParam) - ExpandItem item = items[focusIndex]; $EXCLUDED $EXCLUDED $T1 $V1 = new $T1(); - $V1.item = item; - sendEvent(true, $V1); + $V1.item = focusItem; + sendEvent((focusItem.expanded ? COL...`

**A chicken and egg problem:
Users need hints on what to generalize.**

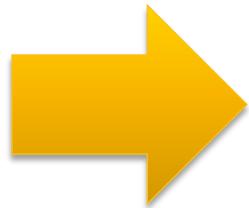
How can we mine and examine variations at scale?



Tianyi Zhang's PhD @ UCLA: Leveraging Commonalities and Variations at Scale



API Usage Mining from GitHub and API Misuse Detection in StackOverflow [ICSE '18]



stackoverflow Questions Developer Jobs

try this out, I did not test it, but from what i see in your code, alliances is not an a json array, also objects based on what i see in your json docume

```
JsonObject rootobj = root.getAsJsonObject();
JsonElement match_number = rootobj.get("match
JsonObject alliances = rootobj.getAsJsonObjec
JsonElement blue = alliances.getAsJsonObject)
JsonElement red = alliances.getAsJsonObject)
System.out.println(match_number.getAsString\
```

share improve this answer edited Apr

just to note, i just tested it and it works fine. – faljbour

add a comment

answer

Potential API Misuse

① Pop-up window

② API misuse description

You may want to check whether the receiver of `getAsString()` is not equal to null. You may also want to handle the potential Exception thrown by `getAsString()` by using a try-catch block here. 117 Github code examples also do this.

③ Fix suggestion

```
try {
    if (match_number!=null) {
        match_number.getAsString();
    }
} catch (Exception e) {
}
```

④ Like or dislike

👍 4 👎 2

See this in a GitHub example:

[ybonnel/gson](#) ⑤ Supporting GitHub examples
[Aleks-Ya/hh-java-api](#)
[ezterry/TTRSS_android_ezterry](#)

1 2 3 4

⑥ Pagination for multiple misuses

Analyzing and Supporting Adaptation of Online Code Examples [ICSE'19]



stackoverflow Questions Developer Jobs Tags Users Search...

Adding on the @Pso's comment, you can store all your Lat,Lng values in a JSON file and copy that file to the assets folder of your app.

1 So, let's say you save the data like this in `assets/locations.json` file,

```
{
  "data": [
    [-08.8123083,13.2249500],
    [-08.8265861,13.2274667],
    [-08.8328611,13.2182861],
    ....]
}
```

And then read it as, **① A Stack Overflow code example of interest**

```
public String getJSONFromAssets() {
    String json = null;
    try {
        InputStream inputData = getAssets().open("locations.json");
        int size = inputData.available();
        byte[] buffer = new byte[size];
        inputData.read(buffer);
        inputData.close();
        json = new String(buffer, "UTF-8");
    } catch (IOException ex) {
        ex.printStackTrace();
        return null;
    }
    return json;
}
```

Use the JSON data anywhere in your code like,

```
JSONObject obj = new JSONObject(getJSONFromAssets());
JSONArray arr = obj.getJSONArray("data");
if (arr != null)
    for (int i=0;i<arr.length();i++)
        locations.add(arr.get(i).toString());
```

share improve this answer edited May 17 '16 at 12:54 answered May 17 '16 at 10:44

Welcome to ExampleStack! **⑥ Copy the template**

Code Template **⑤ Undo the previous selection** Undo Selection Copy

```
String json = null;
try {
    InputStream is = getAssets().open("locations.json");
    int size = is.available();
    byte[] buffer = new byte[size];
    is.read(buffer);
    is.close();
    json = new String(buffer, "UTF-8");
} catch (IOException ex) {
    ex.printStackTrace();
    return null;
}
return json;
```

Adaptation Categories:

- Code Hardening
- Resolve Compilation Error
- Exception Handling
- Logic Customization
- Refactoring
- Miscellaneous

⑦ Colors of different adaptation categories

② Adaptation-aware code template

5 Similar GitHub Examples **③ A list of similar GitHub snippets**

GitHub link watch: 27 star: 197 contributor: 15 **④ GitHub snippet link & metrics**

```
public String loadJSONFromAsset(String jsonFileName) {
    String json = null;
    try {
        InputStream is = getAssets().open(jsonFileName);
        int size = is.available();
        byte[] buffer = new byte[size];
        is.read(buffer);
        is.close();
        json = new String(buffer, "UTF-8");
    } catch (IOException ex) {
        ex.printStackTrace();
        return null;
    }
    return json;
}
```

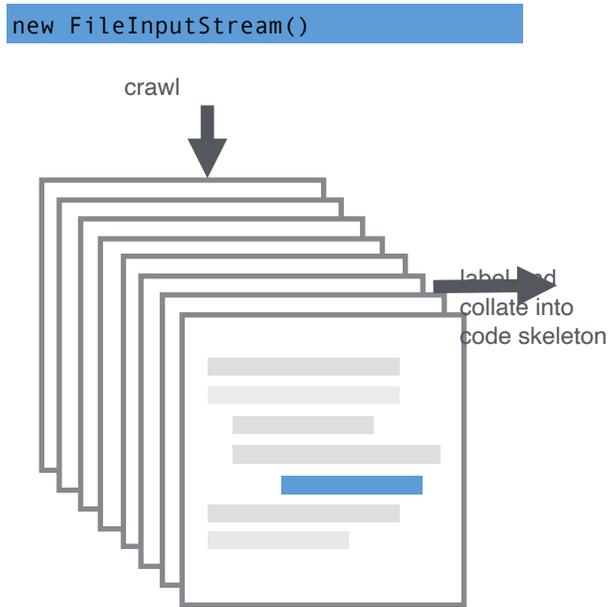
GitHub link watch: 2 star: 0 contributor: 1

```
private String loadJSONFromAsset() {
```

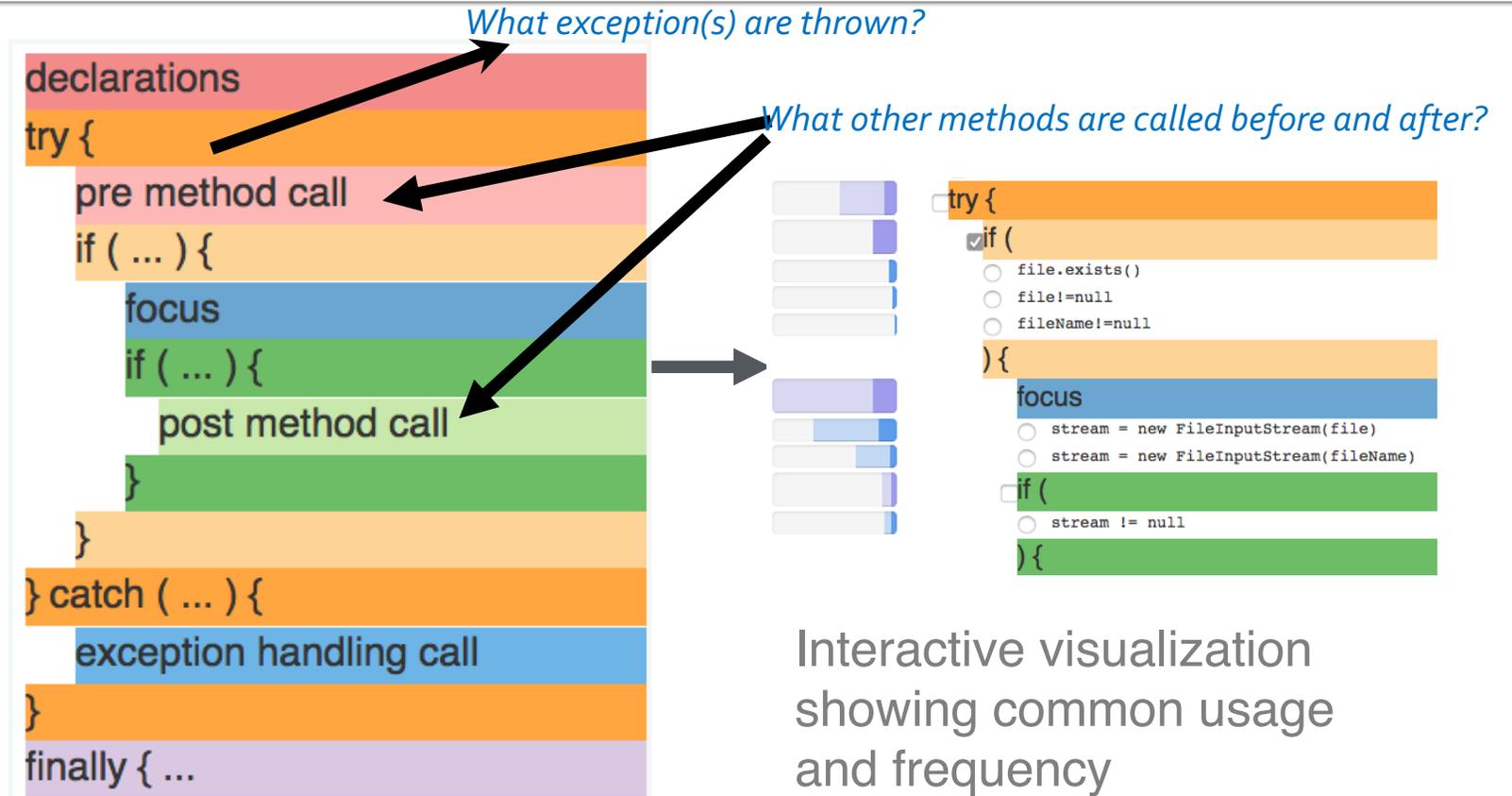
Visualizing API Usage Examples at Scale

[CHI'18]

Focal API



Many code examples using this call



Interactive visualization showing common usage and frequency

Cross-example counts for FileInputStream

Toggle Labels Fold Code Show Default Show Less Show More Show All

Counts

Blocks of options

- declarations
 - File file = new File(String)
 - File file = new File(*)
 - File file = new File(*,String)
- try {
 - pre method call
 - file.length()
 - file.getName()
 - file.getAbsolutePath()
 - if (
 - file.exists()
 - file!=null
 - fileName!=null
 -) {
 - focus
 - stream = new FileInputStream(file)
 - stream = new FileInputStream(fileName)
 - if (
 - stream != null
 - null != stream
 - stream.read(outputByte,0,4096) != -1
 -) {
 - post method call
 - stream.close()
 - Properties.load(stream)
 - new BufferedInputStream(stream)

100 concrete examples from GitHub

Reset Active Filters:

[Link to the GitHub source code](#)

```
@Override
public void readFromFile(String filename) throws IOException {
    in = new FileInputStream(filename);
    prop.load(in);
}
```

[Link to the GitHub source code](#)

```
private synchronized InputStream openStream() throws IOException {
    if (file != null) {
        return new FileInputStream(file);
    } else {
        return new ByteArrayInputStream(memory.getBuffer(), 0, memory.getCount());
    }
}
```

[Link to the GitHub source code](#)

```
public InputStream getResourceContents(String path) {
    File file = new File(_basePath + "/" + path);
    try {
        return new FileInputStream(file);
    } catch (FileNotFoundException e) {
        throw new IllegalArgumentException(e);
    }
}
```

[Link to the GitHub source code](#)

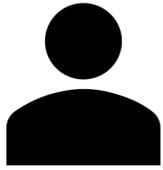
```
public InputStream getInputStream() throws MessagingException {
    try {
        return new BinaryTempFileBodyInputStream(new FileInputStream(mFile));
    } catch (IOException ioe) {
        throw new MessagingException("Unable to open body", ioe);
    }
}
```

[Link to the GitHub source code](#)

How can we construct a search pattern with a human in the loop?



ALICE: Active Inductive Logic Programming for Code Search [ICSE 2019]



Input: More instance labels

ALICE: Keep refining the query

Output: A smaller set of method locations that match the new query



Query Refinement via Active Learning

```
public void getUsername(String id) {  
    try {  
        ResultSet set = db.executeQuery(  
            "select name from users where id=" + id);  
        while (set.next()) { ... }  
    } catch (SQLException e) { ... }  
}
```



```
public void queryDatabase() {  
    try {  
        ResultSet result = s.executeQuery("select * from customers");  
        while (result.next()) { ... }  
    } catch (SQLException e) { ... }  
}
```



```
public List get() {  
    ResultSet set = stmt.executeQuery("select * from t");  
    List l = new List();  
    while (set.next()) { ... }  
    return l;  
}
```



Refined Query

```
methodDec (i0, m) ∧  
type (i1, ResultSet) ∧  
contains (i0, i1) ∧  
methodCall(i2, executeQuery) ∧  
contains (i0, i2) ∧  
looplike (i3, "*.next()") ∧  
contains (i0, i3) ∧  
exception (i4, SQLException),  
contains (i0, i4)
```

Query Refinement Optimization

$$\text{Specialize}(h_{i-1}, P, N) = \underset{h_i}{\operatorname{argmax}} \sum_{p \in P} [p \models h_i]$$

such that $h_i \models h_{i-1}$ and $\forall n \in N, n \not\models h_i$

The image shows a screenshot of the Eclipse IDE interface. The title bar reads "ALICE [Running]". The menu bar includes "File", "Edit", "Source", "Refactor", "Navigate", "Search", "Project", "Run", "Window", and "Help". The toolbar contains various icons for file operations and development tools. On the left, the Package Explorer shows a project structure with folders like "ALICE_FactExtractor Relational", "ALICE_Learner_Relational", "ALICE_Predicates", and "ALICE_UI", along with sub-packages "Arith", "biglambda", "NEW_JDT9801", and "NEW_MOTIF46738". The main editor window displays the file "DefaultCommentMapper.java" with the following code:

```
84     }
85     return index;
86 }
87
88
89 Comment[] getLeadingComments(ASTNode node) {
90     if (this.leadingPtr >= 0) {
91         int[] range = null;
92         for (int i=0; range==null && i<=this.leadingPtr; i++) {
93             if (this.leadingNodes[i] == node) range = this.leadingIndexes[i];
94         }
95
96         if (range != null) {
97             int length = range[1] - range[0] + 1;
```

A large black text box with yellow text is overlaid on the code, stating: "A chicken and egg problem: Users need hints on how to pick a discriminatory atom." Below the code, a list of files is visible, with the following entries highlighted in red and cyan:

- /NEW_JDT9801/search/org/eclipse/jdt/internal/core/search/matching/MethodMatcher.java
- /NEW_JDT9801/compiler/org/eclipse/jdt/internal/compiler/parser/JavadocParser.java
- /NEW_JDT9801/compiler/org/eclipse/jdt/internal/compiler/parser/Parser.java
- /NEW_JDT9801/dom/org/eclipse/jdt/core/dom/DefaultCommentMapper.java
- /NEW_JDT9801/dom/org/eclipse/jdt/core/dom/DefaultCommentMapper.java

The system tray at the bottom right shows the time "6:19 PM" and the text "Left 98".

Developer Tools for Big Data Systems & Heterogeneous Hardware



...



Amazon Scholar



Challenge: Too many possible choices to consider for pattern refinement

updateAAD(...)
Base64.decode(...)

updateAAD(...)
Base64.decode(...)

println(...)

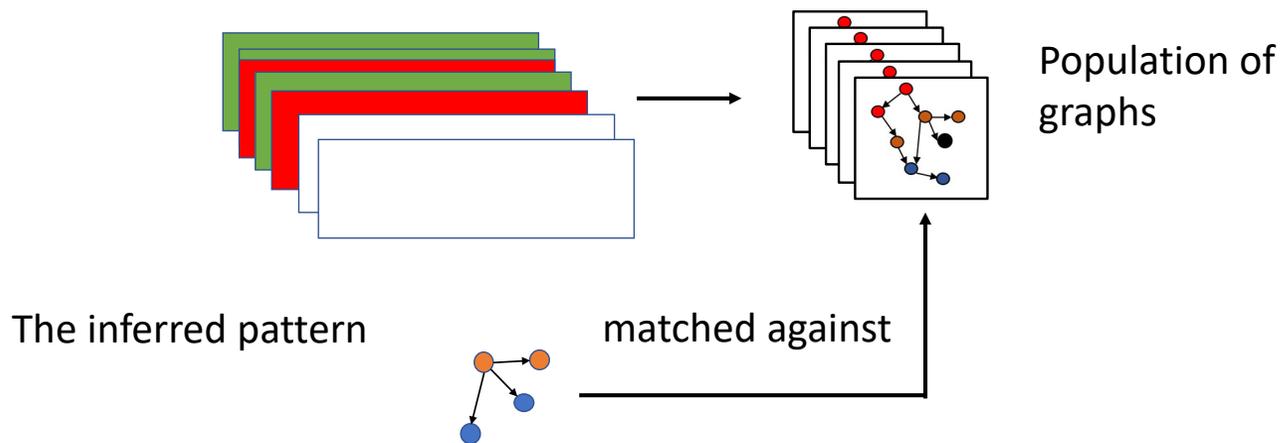


“Should I include **updateAAD** or **Base64.decode**?”

SURF: Scaling Code Pattern Inference with Interactive What-If Analysis [ICSE 2024]



- Matching on Program Dependence Graphs
- Simultaneous overlay
- Global distribution
- Impact Analysis, and
- What-if Analysis



3. Global distribution



Feature Choices

```

new SecretKeySpec(...);
new GCMParameterSpec(...);
new SecureRandom(...);

try {
    Cipher.getInstance(...);
    Cipher.getInstance(...);
    Cipher.doFinal(...);
    Cipher.init(...);
    Cipher.updateAAD(...);
} catch (
    NoSuchPaddingException
    NoSuchAlgorithmException
    InvalidKeyException
) {
    new IllegalArgumentException(...);
    new IllegalStateException(...);
    Log.getStackTraceString(...);
}
                
```

Matching

Including Cipher.updateAAD() would match 5 examples

(4 Positive + 0 Invalidated + 1 Negative)

Example 1000: Positive

```

public String decrypt(byte[] ciphertext, byte[] nonce, byte[] aad) {
    try {
        SecretKeySpec key = new SecretKeySpec("AES/GCM/NoPadding");
        Cipher cipher = Cipher.getInstance("AES/GCM/NoPadding");
        cipher.init(Cipher.DECRYPT_MODE, key, new SecretKeySpec("AES/GCM/NoPadding"), new GCMParameterSpec(128, nonce));
        if (aad != null) {
            cipher.updateAAD(aad);
        }
        return new String(cipher.doFinal(ciphertext));
    } catch (Exception e) {
        log.error("Error decrypting: " + e.getMessage());
    }
}
                
```

Not Matching

Including Cipher.updateAAD() does not match 5 examples

(0 Positive + 2 Invalidated + 0 Negative)

Example 1000: Invalidated

```

public String decrypt(String key, String ciphertext, String nonce, byte[] aad) throws IOException, NoSuchAlgorithmException, NoSuchPaddingException {
    String data = key; // init data
    //String key = "key";
    Cipher cipher = null;
    String pw = "";
    SecretKeySpec keySpec = new SecretKeySpec("AES/GCM/NoPadding");
    cipher = Cipher.getInstance("AES/GCM/NoPadding");
    cipher.init(Cipher.DECRYPT_MODE, keySpec, new SecretKeySpec("AES/GCM/NoPadding"), new GCMParameterSpec(128, nonce));
    cipher.doFinal(ciphertext);
    log.info("Invalid key exception");
    return cipher.doFinal(ciphertext);
}
                
```

5. What-If Analysis:

Matched by Cipher.doFinal() but not Cipher.updateAAD()

Example 1000: Positive

```

public static byte[] decrypt(MessageDigest md, byte[] ciphertext, byte[] nonce, byte[] aad) throws IOException, NoSuchAlgorithmException, NoSuchPaddingException {
    String pw = null;
    SecretFactory factory = SecretFactory.getInstance("PBKDF2WithHmacSHA1");
    KeySpec spec = new PBEKeySpec(password.toCharArray(), salt, iterations, keyLength);
    SecretKey key = factory.generateSecret(spec);
    SecretKeySpec keySpec = new SecretKeySpec(key.getEncoded(), "AES");
    Cipher cipher = Cipher.getInstance("AES/GCM/NoPadding");
    cipher.init(Cipher.DECRYPT_MODE, keySpec, new SecretKeySpec("AES/GCM/NoPadding"), new GCMParameterSpec(128, nonce));
    return cipher.doFinal(ciphertext);
}
                
```

Matched by Cipher.updateAAD() but not Cipher.doFinal()

Example 1000: Invalidated

```

public static byte[] decrypt(String key, GCMParameterSpec gcmSpec, byte[] ciphertext, byte[] nonce, byte[] aad) throws IOException, NoSuchAlgorithmException, NoSuchPaddingException {
    return cipher.doFinal(ciphertext);
}
                
```

2. Summary of code population overlaid on a skeleton

Design 1: Hint on Global Distribution

S	I	
	0.50	String.getBytes()
	0.00	Cipher.init()
	0.00	String.equals()

Support:

Number of times a statement appears, regardless of label

Information gain:

Improvement in entropy for separating positive from negative instances

$$\text{Entropy}(G_P) - \left(\frac{|G_m|}{|G_P|} \times \text{Entropy}(G_m) + \frac{|G_e|}{|G_P|} \times \text{Entropy}(G_e) \right)$$

Design 2: Impact Analysis

- Focuses attention to which instances will be included/excluded for each statement choice

R	I	Feature
0.00	0.00	new SecretKeySpec(...)
0.00	0.00	new GCMParameterSpec(...)
0.00	0.00	new SecureRandom(...)
0.00	0.00	try {
0.00	0.00	Cipher.getInstance(..."AES/GCM/NoPadding"...)
0.00	0.00	Cipher.getInstance(..."RSA/ECB/PKCS1PADDING"...)
0.00	0.00	Cipher.doFinal()
0.00	0.00	Cipher.updateAAD()
0.00	0.00	Cipher.updateAAD(associatedData);
0.43	0.43	NoSuchPaddingException
0.43	0.43	NoSuchAlgorithmException
0.43	0.43	InvalidKeyException
0.22	0.01	new IllegalArgumentException(...)
0.01	0.00	new IllegalStateException(...)
0.00	0.00	Log.getStackTraceString()

Matching

Including Cipher.updateAAD()
would match 5 examples

(4 positive + 0 unlabelled + 1 negative)

Example 1020: Positive Negative

```
/**
 * 解密并转换为字符串
 *
 * @param associatedData AAD, 额外的认证加密数据, 可以为空
 * @param nonce IV, 随机字符串初始化向量
 * @param ciphertext 密文
 * @return UTF-8编码的明文
 */
public String decrypt(byte[] associatedData, byte[] nonce,
byte[] ciphertext) {
    try {
        Cipher cipher = cipher.getInstance("AES/GCM/NoPadding");
        cipher.init(
            Cipher.DECRYPT_MODE,
            new SecretKeySpec(key, algorithm),
            new GCMParameterSpec(tagLengthBit, nonce));
        if (associatedData != null) {
            cipher.updateAAD(associatedData);
        }
        return new String(cipher.doFinal(ciphertext),
StandardCharsets.UTF_8);
    }
}
```

Not Matching

Including Cipher.updateAAD()
does not match 3 examples

(0 positive + 2 unlabelled + 0 negative)

Example 1006: Positive Negative

```
public String good(String sKey) throws
UnsupportedEncodingException, NoSuchProviderException
{
    String data = "key"; /* init data */

    //String sKey = "Skey";
    Cipher cipher = null;
    String pw = "";
    try {
        SecretKeySpec key = new
SecretKeySpec(sKey.getBytes(), "AES");
        cipher = cipher
.getInstance("AES/GCM/NoPadding");
        cipher.init(Cipher.DECRYPT_MODE, key);
    } catch (NoSuchAlgorithmException e) {
        log.info("error");
    } catch (InvalidKeyException e) {
        log.info("InvalidKeyException");
    } catch (NoSuchPaddingException e) {
        log.info("error");
    }
}
```

“If I include this specific code line, how many will I match?”

Design 3: What-If Analysis

Explore trade-offs involved in selecting one statement over another

R	I	Feature
<input type="checkbox"/>	0.00	new SecretKeySpec(...)
<input type="checkbox"/>	0.00	new GCMParameterSpec(...)
<input type="checkbox"/>	0.00	new SecureRandom(...)
<input type="checkbox"/>	0.00	try {
<input type="checkbox"/>	0.00	Cipher.getInstance(..."AES/GCM/NoPadding"...)
<input type="checkbox"/>	0.00	Cipher.getInstance(..."RSA/ECB/PKCS1PADDING"...)
<input type="checkbox"/>	0.00	Cipher.doFinal()
<input checked="" type="checkbox"/>	0.00	Cipher.init()
<input checked="" type="checkbox"/>	0.00	Cipher.updateAAD()
<input type="checkbox"/>	0.43	} catch (
<input type="checkbox"/>	0.43	NoSuchPaddingException
<input type="checkbox"/>	0.43	NoSuchAlgorithmException
<input type="checkbox"/>	0.43	InvalidKeyException
<input type="checkbox"/>	0.00) {
<input type="checkbox"/>	0.00	new IllegalArgumentException(...)
<input type="checkbox"/>	0.00	new IllegalStateException(...)
<input type="checkbox"/>	0.00	Log.getStackTraceString()
<input type="checkbox"/>	0.00	}

Why?

Comparing examples matching `new SecretKeySpec(...)` and `Cipher.updateAAD()`.

Matched by `new SecretKeySpec(...)` but not `Cipher.updateAAD()`

Matched by `Cipher.updateAAD()` but not `new SecretKeySpec(...)`

```
Example 1033:
public static String DecryptWithPassword(String password, byte[] iv, byte[] message, byte[] salt) {
    initIV();
    String new_message = null;
    try {
        SecretKeyFactory factory =
        SecretKeyFactory.getInstance("PBKDF2WithHmacSHA1");
        KeySpec spec = new PBEKeySpec(password.toCharArray(),
        salt, 65536, 256);
        SecretKey tmp = factory.generateSecret(spec);
        SecretKey secret_key = new
        (tmp.getEncoded(), "AES");
        Cipher cipher =
        cipher.getInstance("AES/GCM/NoPadding");
    }
}
```

```
Example 1000:
public static byte[] cipher(Key key, GCMParameterSpec gcpSpec, byte[] bytes,
byte[] aad, int mode) throws GeneralSecurityException {
    Cipher cipher = cipher.getInstance("AES/GCM/NoPadding");
    cipher.init(mode, key, gcpSpec);
    cipher.updateAAD(aad);
    return cipher.doFinal(bytes);
}
```

“Which code statement is better to include?”

Recap: *A Journey* through Searching Similar Code

- What motivated us? **Systematic changes**
- What were early attempts? **Rule-based change abstraction**
- How serious is this problem? **Pretty serious**
- How can we automate? **Generalized patch synthesis**

Several chicken and egg problems: Users need example patches, hints on what to generalize, and hints on how to pick a discriminatory atom.

- How can we examine variations at scale? **Simultaneous overlay**
- How to construct a search pattern with a human in the loop? **Hints on global distribution and interactive what-if analysis**

Discussion: Would've, could've, should've, etc.

- DSL
- Code embeddings
- LLM
- Information retrieval
- Search with multi-modality: text, video, etc.

Thank you!

Thanks to Baishakhi Ray, Na Meng, Tianyi Zhang, HongJin Kang, Myoungkyu Song, David Notkin, Elena Glassman, Dan Grossman, John Jacobellis, Björn Hartmann, Cristina Lopes, Tyler McDonnell, Nachiappan Nagappan, Mihir Mathur, Kathryn McKinley, Suzette Person, Joseph Pinedo, Hridesh Rajan, Anastasia Reinhardt, Neha Rungta, Aishwarya Sivaraman, Ganesha Upadhyaya, Guy Van den Broeck, Christopher Wiley, Gary Wilson Jr., Di Yang, Thomas Zimmermann



```

Original Version of Target
public Diff handleMouseInSides(Canvas
    canvas, MergeSourceViewer msv, int my){
    int lineHeight=msv.getTextWidget().getLineHeight();
    ChangeDiffs changeDiffs = getChangeDiffs();
    if (changeDiffs != null) {
        int shift=msv.getVerticalScrollOffset() + (2
        Iterator iter = changeDiffs.iterator();
        while (iter.hasNext()) {
            Diff diff= (Diff) iter.next();
            if (diff.isDeleted()) continue;
            if (fShowCurrentOnly2 && !isCurrentDiff(diff))
            Point region=new Point(0,0);
            msv.getLineRange(diff.getPosition(msv).region);
            int y=(region.x * lineHeight) + shift;
            int h=region.y * lineHeight;
            if (my >= y && my < y + h) return diff;
        }
    }
}

```

```

Suggested Version of Target
public Diff handleMouseInSides(Canvas canvas, MergeSourceViewer
    tp, int my){
    Merger merger = getMerger();
    int lineHeight= tp.getTextWidget().getLineHeight();
    if (merger != null) {
        Point region= new Point(0, 0);
        char leg = getLeg(tp);
        int shift= msv.getVerticalScrollOffset() + (2-LW);
        for(Iterator iterator = merger.changesIterator();
            iterator.hasNext();){
            Diff diff = (Diff)iterator.next();
            if (diff.isDeleted()) continue;
            if (fShowCurrentOnly2 && !isCurrentDiff(diff)) continue;
            msv.getLineRange(diff.getPosition(leg), region);
        }
    }
}

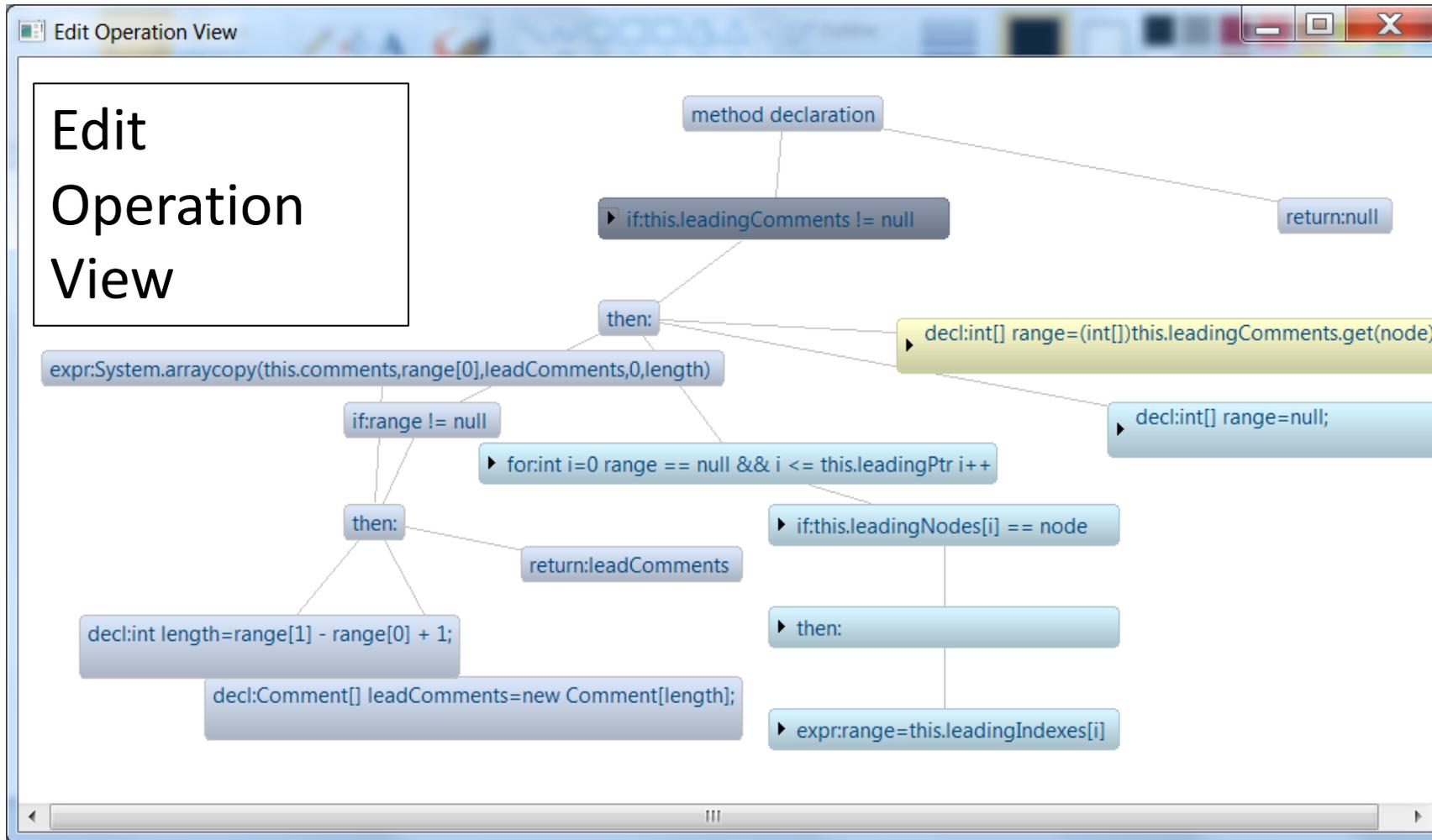
```

Console Rules View Sydit View

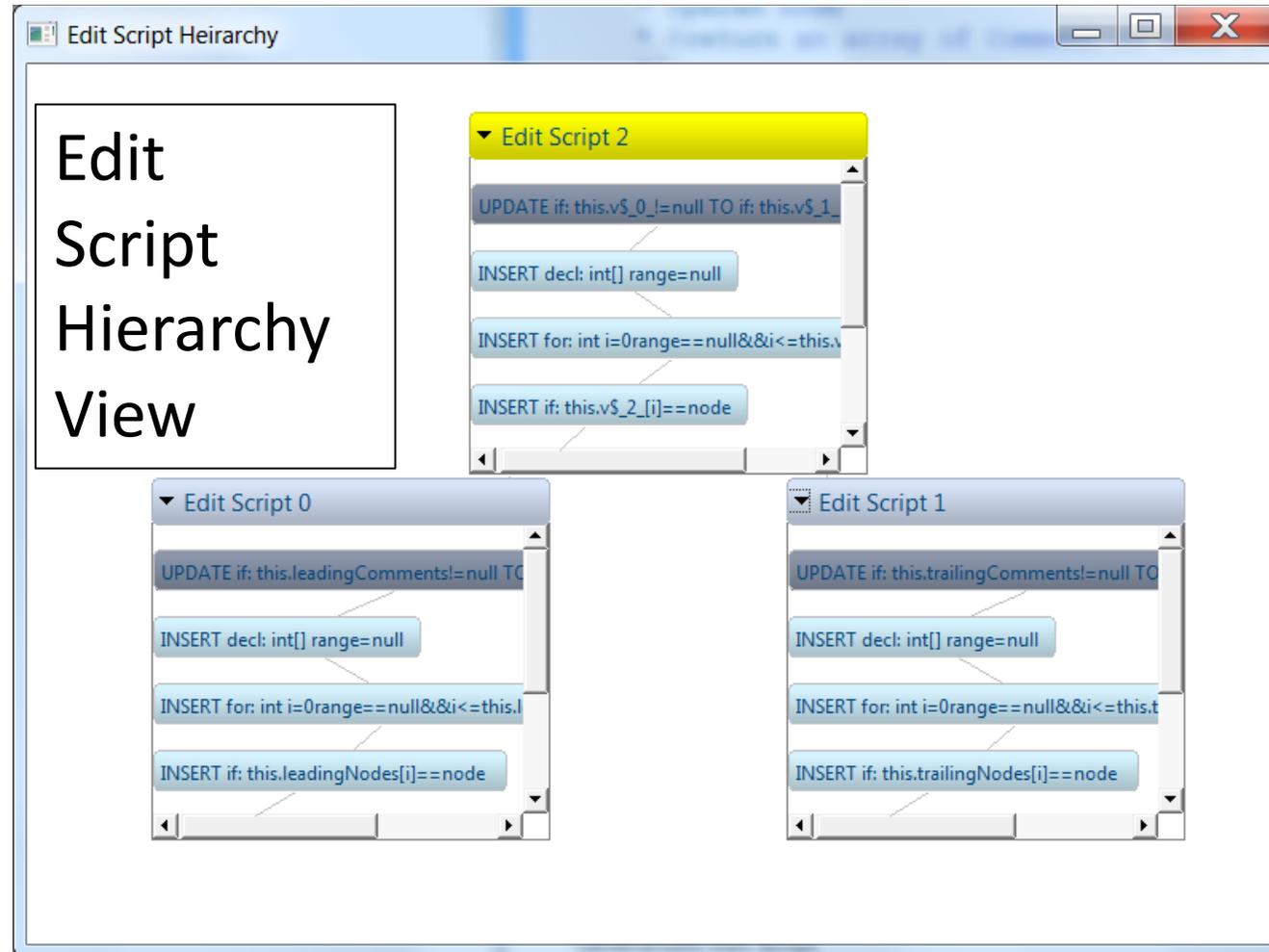
Id	Class Name	Method Name	File Path
▶ refactoring			
▶ add null check			
▼ reconstruct loop and			
▼ Example			
Old Method	SimplifiedTextMergeViewer	paintSides(GC, MergeSourceViewer, Canvas, boolean)	examples/src/SimplifiedTextMergeViewer.java
New Method	SimplifiedTextMergeViewer	paintSides(GC, MergeSourceViewer, Canvas, boolean)	examples2/src/SimplifiedTextMergeViewer.java
▼ Target			
Target Method	SimplifiedTextMergeViewer	handleMouseInSides(Canvas, MergeSourceViewer, int)	examples2/src/SimplifiedTextMergeViewer.java

View Suggested Version
Accept Suggested Version

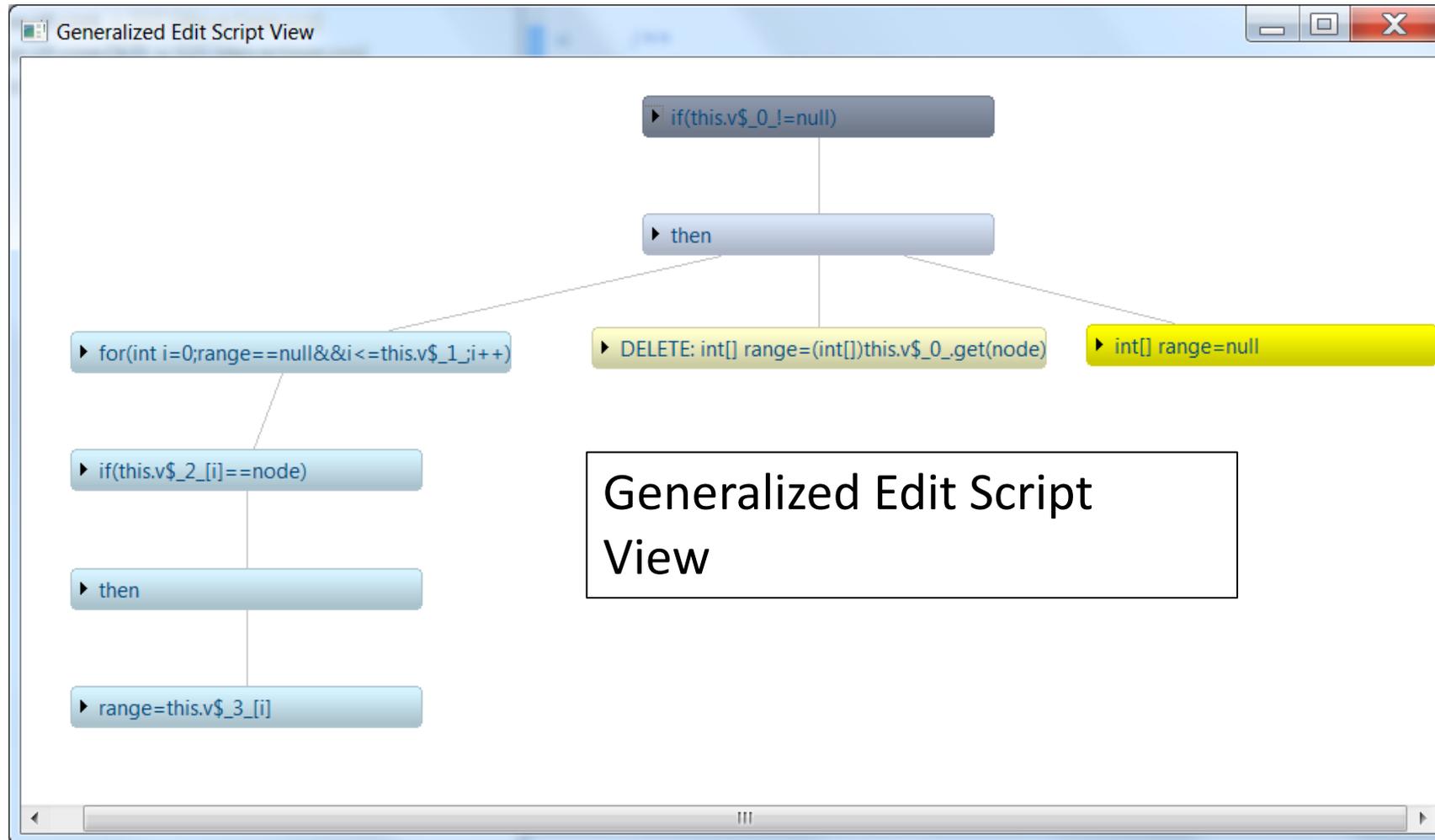
For each change example, a user can view AST edit operations



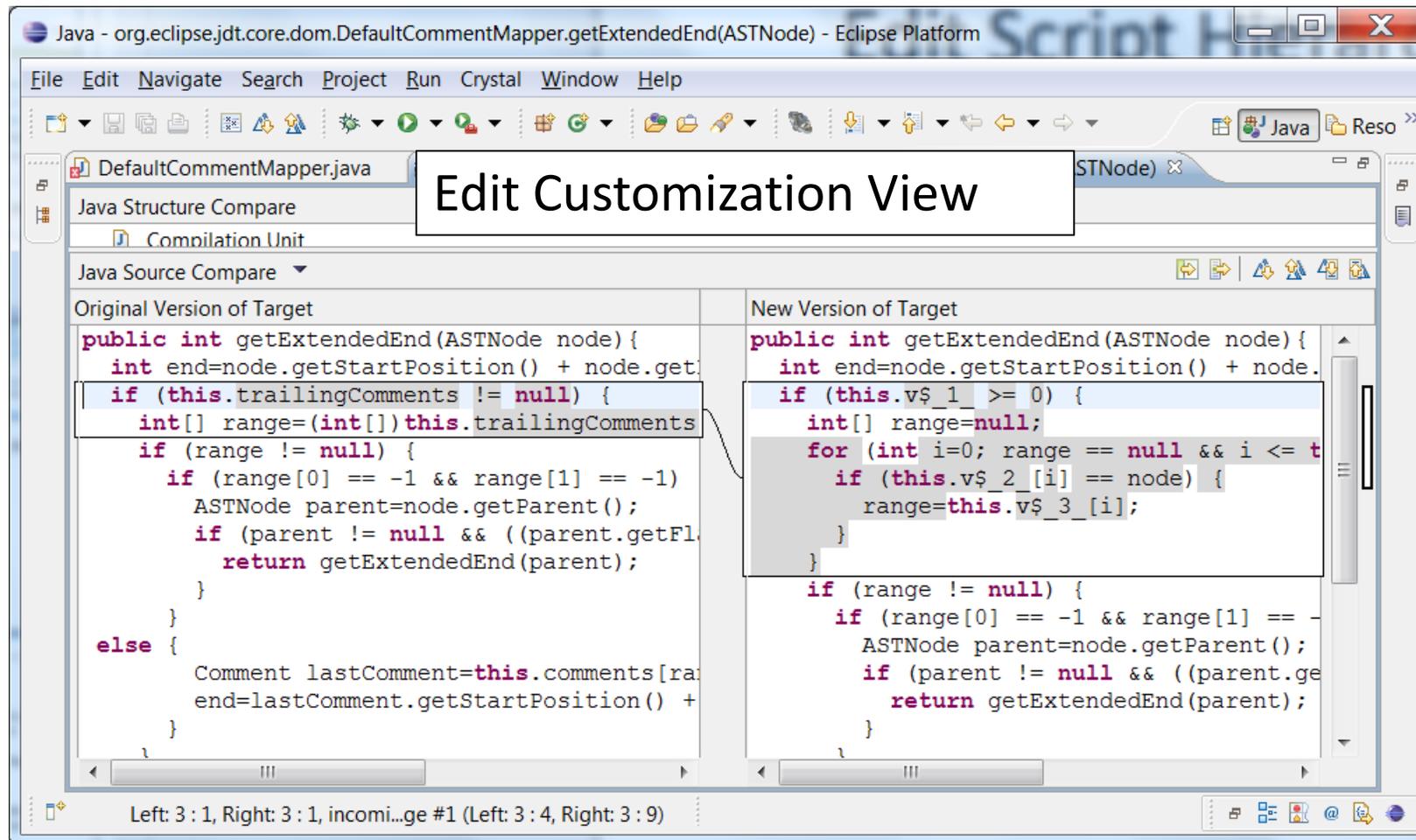
A user can view a hierarchy of edit scripts and select one of them



A User can inspect a generalized edit script



A user can correct suggested edits before applying the suggestion

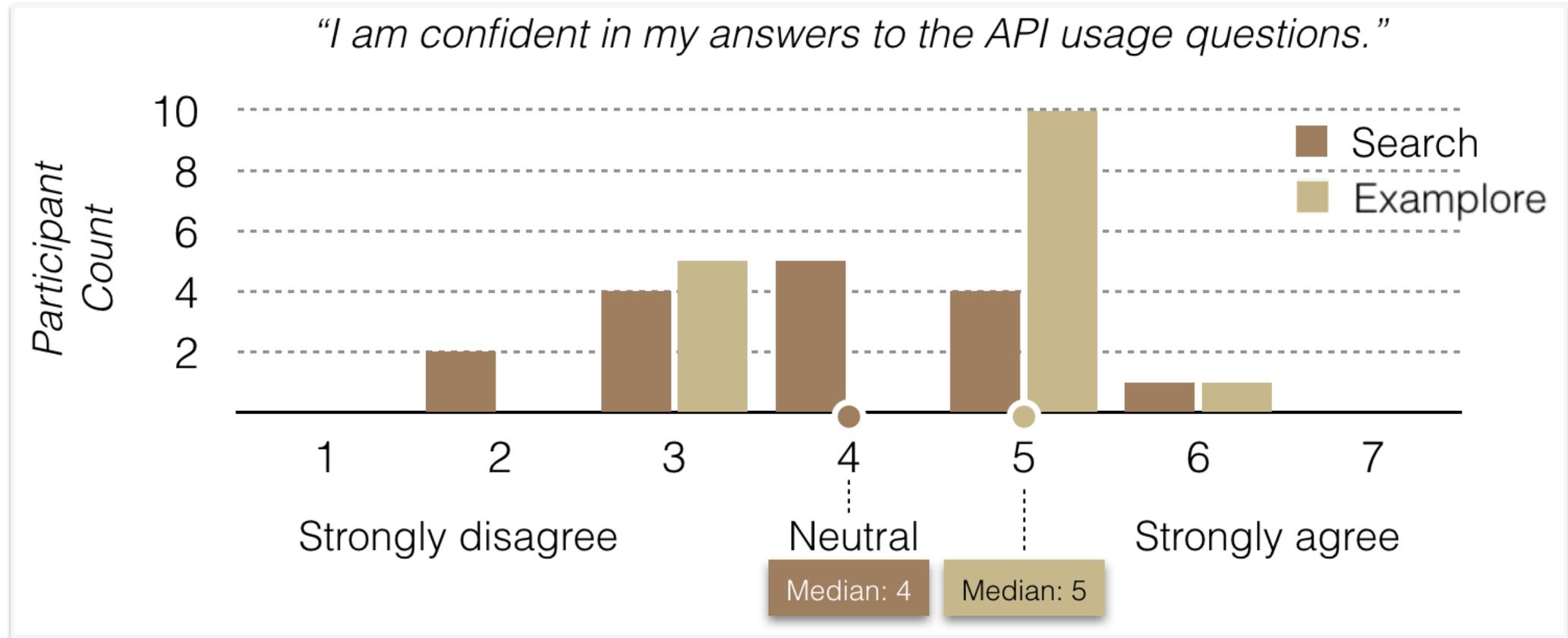


Index	Bug(patches)	m_i	Edit Location					Operations		
			Σ	✓	P%	R%	A%	E	C	$E_{A\%}$
2	82429(2)	16	13	12	92	75	81	9	9	100
4	139329(3)	6	2	2	100	33	74	6	3	50
7	103863(5)	7	7	7	100	100	100	34	34	100
8	129314(3)	3	4	4	100	100	100	2	2	100
16	95409(3)	7	9	9	100	100	78	4	4	100
24	98198(2)	9	15	15	100	100	95	3	3	100

On average, Lase finds edit locations with 99% precision, 89% recall, and applies edits with 91% accuracy.

For three bugs, Lase suggests in total 9 edits that developers missed and later confirmed.

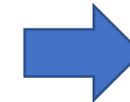
Lab Study Results



Represent Code as Logic Facts

Fact Predicate
if (ID, CONDITION)
loop (ID, CONDITION)
parent (ID, ID)
next (ID, ID)
methodCall (ID, NAME)
type (ID, NAME)
exception (ID, NAME)
methodDec (ID, NAME)

```
public void queryDB() {  
    try {  
        Connection con = DriverManager.getConnection(  
            "jdbc:mysql://localhost:3306/db","root","root");  
        Statement stmt = con.createStatement();  
        ResultSet rs = stmt.executeQuery("select * from emp");  
        while (rs.next()) {  
            System.out.println(rs.getInt(1));  
        }  
        con.close();  
    } catch (SQLException e) {  
        System.out.println(e);  
    }  
}
```



Extracted Logic Facts

```
methodDec (0, queryDB),  
type (1, Connection),  
parent (0, 1),  
methodCall (2, getConnection),  
parent (0, 2),  
next (2, 1),  
...  
loop (7, "rs.next()"),  
methodCall (8, getInt),  
parent (7, 8),  
...  
exception (10, SQLException),  
parent (0, 10),  
...
```

Formulate a Search Query

- A user selects a code example and annotate important features.

```
public void queryDB() {  
    try {  
        Connection con = DriverManager.getConnection(  
            "jdbc:mysql://localhost:3306/db","root","root");  
        Statement stmt = con.createStatement();  
        ResultSet rs = stmt.executeQuery("select * from emp");  
        while (rs.next()) {  
            System.out.println(rs.getInt(1));  
        }  
        con.close();  
    } catch (SQLException e){  
        System.out.println(e);  
    }  
}
```

A code example with user annotations



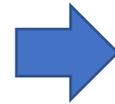
```
methodDec (i0, m) ∧  
type (i1, ResultSet) ∧  
contains (i0, i1) ∧  
methodCall(i2, executeQuery) ∧  
contains (i0, i2) ∧  
looplike (i3, "*.next()") ∧  
contains (i0, i3)
```

search query

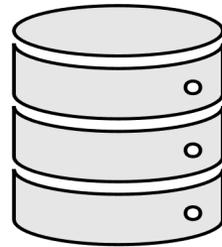
Logic-based Code Search

Search Query

```
methodDec (i0, m) ∧  
type (i1, ResultSet) ∧  
contains (i0, i1) ∧  
methodCall(i2, executeQuery) ∧  
contains (i0, i2) ∧  
looplike (i3, "*.next()") ∧  
contains (i0, i3)
```

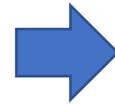


Fact Base



+

Fact Rules



Matched Code

```
public void getUsername(String id) {  
    try {  
        ResultSet set = db.executeQuery(  
            "select name from users where id=" + id);  
        while (set.next()) { ... }  
    } catch (SQLException e) { ... }  
}  
  
public void queryDatabase() {  
    try {  
        ResultSet result = s.executeQuery("select * from customers");  
        while (result.next()) { ... }  
    } catch (SQLException e) { ... }  
}  
  
public List get() {  
    ResultSet set = stmt.executeQuery("select * from t");  
    List l = new List();  
    while (set.next()) { ... }  
    return l;  
}
```

and 32 other matched locations

Align and aggregate structured call sequences into a single view

Toggle Labels Fold Code Show Default Show Less Show More

Counts Blocks of options

- declarations
 - File file = new File(String)
 - File file = new File(*)
 - File file = new File(*,String)
- try {
- pre method call
 - file.length()
 - file.getName()
 - file.getAbsolutePath()
- if (
 - file.exists()
 - file!=null
 - fileName!=null) {
- focus
 - stream = new FileInputStream(file)
 - stream = new FileInputStream(fileName)
- if (
 - stream != null
 - stream.read(outputByte,0,4096) != -1) {
- post method call

100 concrete examples from GitHub

Reset Active Filters:

[Link to the GitHub source code](#)

```
@Override
public void readFromFile(String filename) throws IOException {
    in = new FileInputStream(filename);
    prop.load(in);
}
```

[Link to the GitHub source code](#)

```
private synchronized InputStream openStream() throws IOException
if (file != null) {
    return new FileInputStream(file);
} else {
    return new ByteArrayInputStream(memory.getBuffer(), 0, memory
}
```

[Link to the GitHub source code](#)

```
public InputStream getResourceContents(String path) {
    File file = new File(_basePath + "/" + path);
    try {
        return new FileInputStream(file);
    } catch (FileNotFoundException e) {
        throw new IllegalAccessError("Resource not found: " + path);
    }
}
```



Glassman* and Zhang* et al. CHI 2018

Explore less frequent but critical API usage features

Toggle Labels Fold Code Show Default Show Less **Show More**

Counts

Blocks of options

- declarations
 - File file = new File(String)
 - File file = new File(*)
 - File file = new File(*,String)
- try {
- pre method call
 - file.length()
 - file.getName()
 - file.getAbsolutePath()
- if (
 - file.exists()
 - file!=null
 - fileName!=null) {
- focus
 - stream = new FileInputStream(file)
 - stream = new FileInputStream(fileName)
- if (
 - stream != null
 - stream.read(outputByte,0,4096) != -1

100 concrete examples from GitHub

Reset Active Filters:

[Link to the GitHub source code](#)

```
@Override
public void readFromFile(String filename) throws IOException {
    in = new FileInputStream(filename);
    prop.load(in);
}
```

[Link to the GitHub source code](#)

```
private synchronized InputStream openStream() throws IOException
if (file != null) {
    return new FileInputStream(file);
} else {
    return new ByteArrayInputStream(memory.getBuffer(), 0, memory
}
}
```

[Link to the GitHub source code](#)

```
public InputStream getResourceContents(String path) {
    File file = new File(_basePath + "/" + path);
    try {
        return new FileInputStream(file);
    } catch (FileNotFoundException e) {
        throw new IllegalArgumentException(e);
    }
}
```

Interactively building your own patterns

Toggle Labels Fold Code Show Default Show Less Show More Show All

Counts

Blocks of options

- declarations
- File file = new File(String)
- File file = new File(*)
- try {
- if (
 - file.exists()
 - file!=null) {
- focus**
- stream = new FileInputStream(file)
- if (
 - stream != null) {

Reset **Active Filters:** must have stream = new FileInputStream(file), some guard condition,

[Link to the GitHub source code](#)

```
private synchronized InputStream openStream() throws IOException {  
    if (file != null) {  
        return new FileInputStream(file);  
    } else {  
        return new ByteArrayInputStream(memory.getBuffer(), 0, memory.getCount());  
    }  
}
```

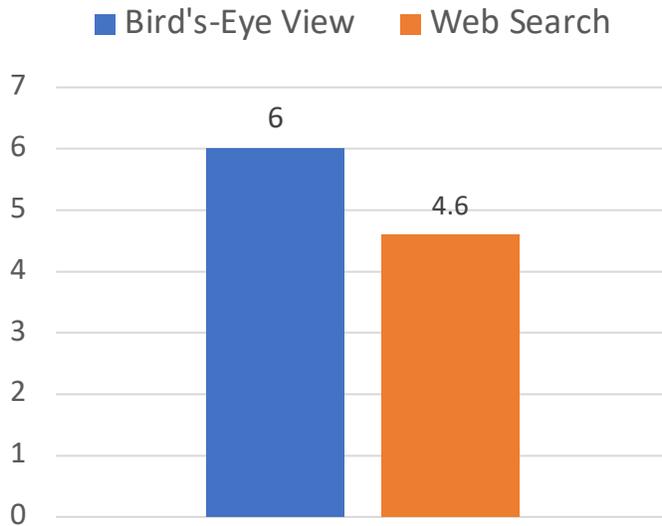
[Link to the GitHub source code](#)

```
public InputStream getInputStream() {  
    if (exists()) {  
        if (_file.isFile()) {  
            try {  
                return new FileInputStream(_file);  
            } catch (FileNotFoundException e) {  
                throw new RuntimeException(e);  
            }  
        } else if (_file.isDirectory()) {
```

A within-subject user study

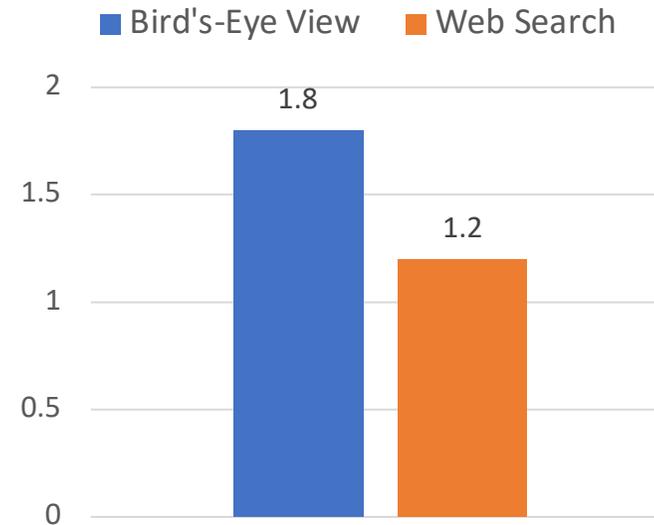
RQ1. Does the bird's-eye view help build robust API knowledge?

Number of questions that are answered correctly



(paired t-test: $t=3.02$, $df=15$, $p\text{-value}=0.0086$)

Number of correct answers per question

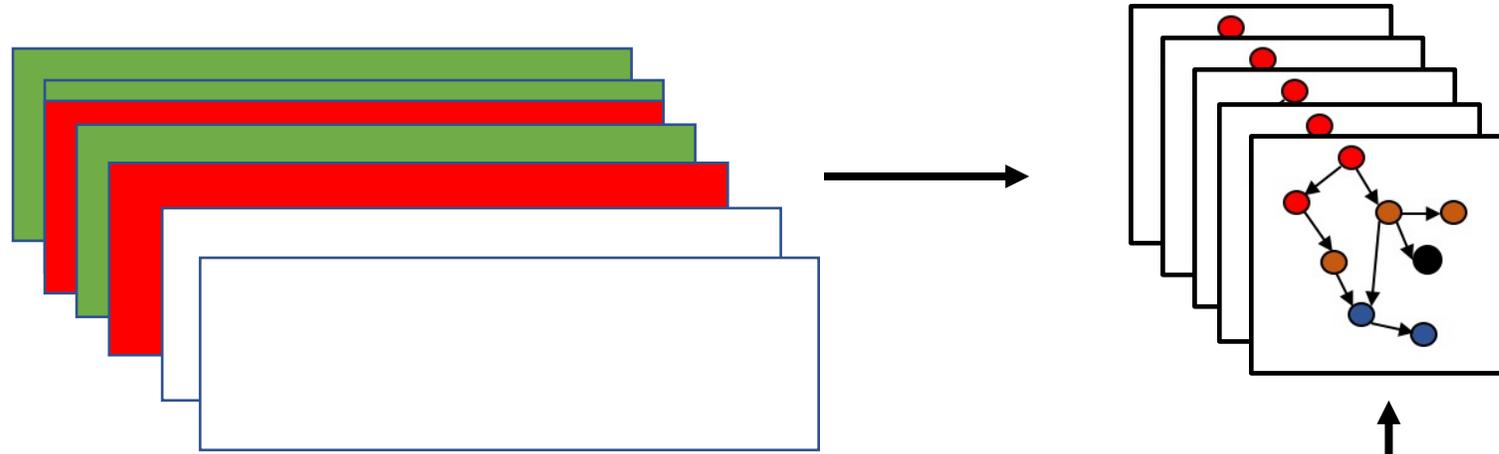


(paired t-test: $t=3.84$, $df=15$, $p\text{-value}=0.0016$)



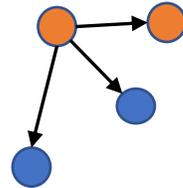
Key Result: Users with the bird's-eye view answered API usage questions more *correctly* and *comprehensively*.

Design 1: Infer Common PDG Subgraph



Population of graphs

The inferred pattern



matched against

Design 2: Simultaneous Overlay

“What constructors are
called?”

“What exceptions are
caught?”

```
// Declarations  
try {  
  
    // Method Calls  
  
} catch (  
  
    // Exceptions Caught  
  
) {  
  
    // Exception Handling Call  
  
}
```

Design 2: Simultaneous Overlay

Statement Choice 1 →
Statement Choice 2 →
Statement Choice 3 →

```
Feature  
new SecretKeySpec(...)  
new GCMParameterSpec(...)  
new SecureRandom(...)  
  
try {  
    Cipher.getInstance(..."AES/GCM/NoPadding"...)  
    Cipher.getInstance(..."RSA/ECB/PKCS1PADDING"...)  
  
    Cipher.doFinal()  
    Cipher.init()  
    Cipher.updateAAD()  
} catch (  
    NoSuchPaddingException  
    NoSuchAlgorithmException  
    InvalidKeyException  
  
){  
    new IllegalArgumentException(...)  
    new IllegalStateException(...)  
    Log.getStackTraceString()  
}
```

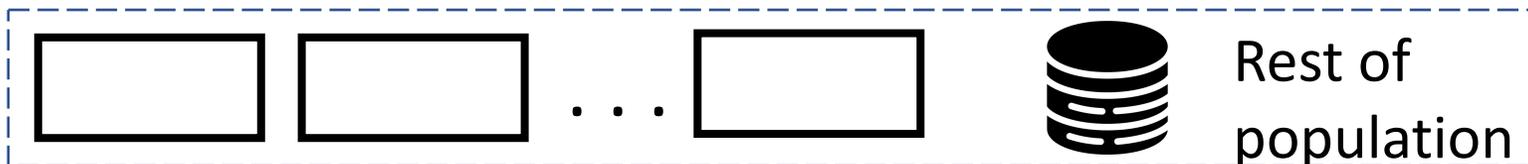
Challenge 1: Instance-level feedback provides too little information

```
Cipher.getInstance(AES)  
Cipher.init(...)  
System.println(...)
```

```
Cipher.getInstance(AES)  
Cipher.init(...)  
System.println(...)
```

```
Cipher.getInstance(DES)
```

```
Cipher.getInstance(..) ?  
  
Cipher.getInstance(AES) ?  
  
Cipher.getInstance(AES) ?  
System.println(...)
```



Results: Users can better understand the API usage distribution when given guidance

30%

more correct answers

20%

less time required

1.8X

more likely to construct the target pattern

Improvements are statistically significant ($p < 0.001$) following a mixed-effects linear model accounting for ordering, tool, and task.