Feature Location via Information Retrieval based Filtering of a Single Scenario Execution Trace

Dapeng Liu, Andrian Marcus, *Denys Poshyvanyk*, Václav Rajlich

SEVERE Group @

WAYNE STATE UNIVERSITY

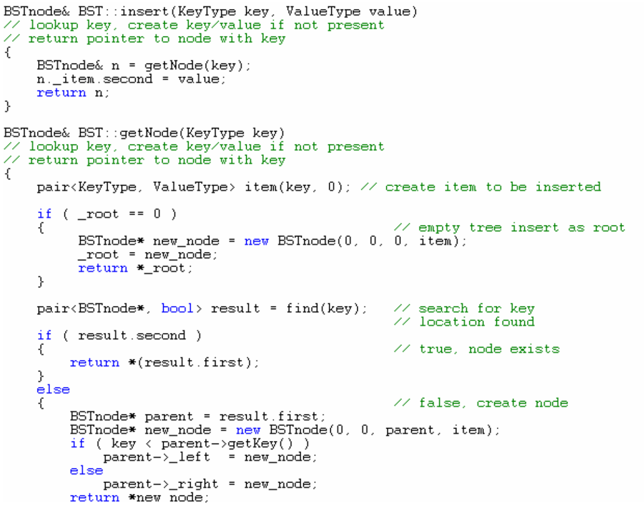
Incremental Change of Software

- Change request
- Feature location
- Impact analysis
- Implementation
- Change propagation
- Testing

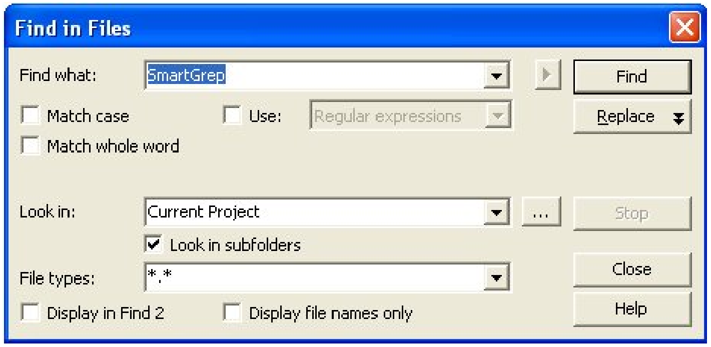
Incremental Change of Software

- Change request
- **Feature location**
- Impact analysis
- Implementation
- Change propagation
- Testing

Feature Location with Regular Expressions



```
BSTnode& BST::insert(KeyType key, ValueType value)
// lookup key, create key/value if not present
// return pointer to node with key
{
    BSTnode& n = getNode(key);
    n._item.second = value;
    return n;
}

BSTnode& BST::getNode(KeyType key)
// lookup key, create key/value if not present
// return pointer to node with key
{
    pair<KeyType, ValueType> item(key, 0); // create item to be inserted

    if ( _root == 0 )
    {                                             // empty tree insert as root
         BSTnode* new_node = new BSTnode(0, 0, 0, item);
         _root = new_node;
         return *_root;
    }

    pair<BSTnode*, bool> result = find(key);    // search for key
                                                // location found
    if ( result.second )
    {                                           // true, node exists
        return *(result.first);
    }
    else
    {                                           // false, create node
        BSTnode* parent = result.first;
        BSTnode* new_node = new BSTnode(0, 0, parent, item);
        if ( key < parent->getKey() )
            parent->_left  = new_node;
        else
            parent->_right = new_node;
        return *new node;
```



Find what: SmartGrep

☐ Match case   ☐ Use: Regular expressions

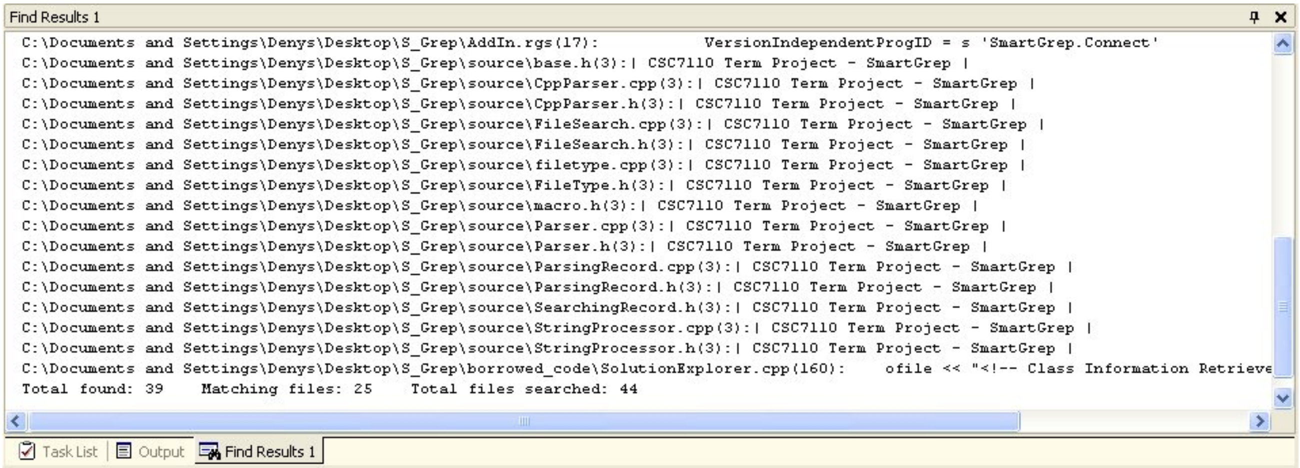
☐ Match whole word

Look in: Current Project

☑ Look in subfolders

File types: *.*

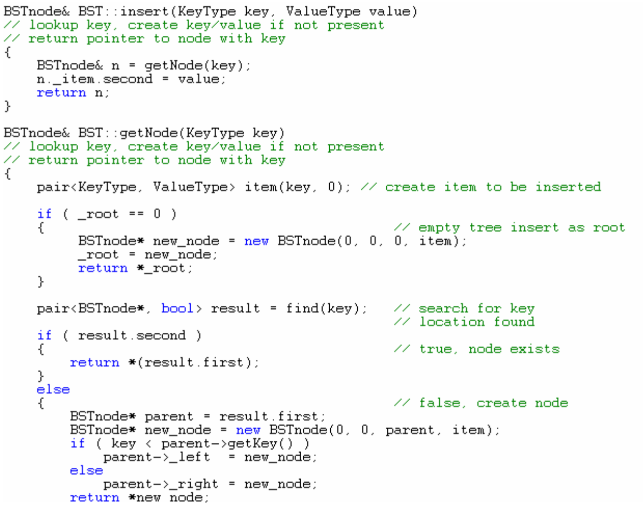
☐ Display in Find 2   ☐ Display file names only



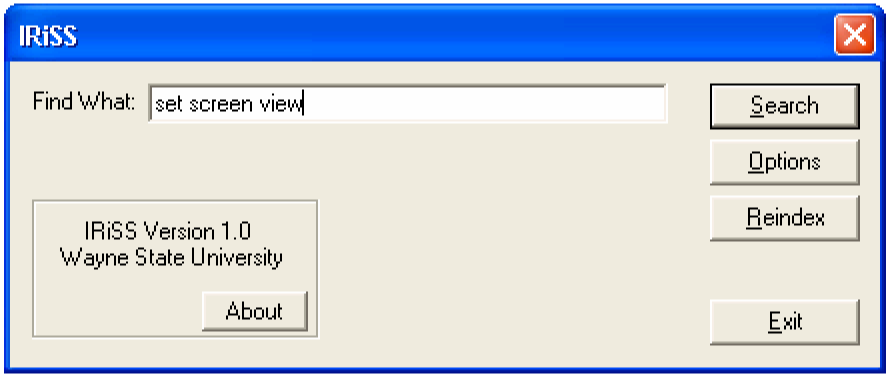
```
C:\Documents and Settings\Denys\Desktop\S_Grep\AddIn.rgs(17):           VersionIndependentProgID = s 'SmartGrep.Connect'
C:\Documents and Settings\Denys\Desktop\S_Grep\source\base.h(3):| CSC7110 Term Project - SmartGrep |
C:\Documents and Settings\Denys\Desktop\S_Grep\source\CppParser.cpp(3):| CSC7110 Term Project - SmartGrep |
C:\Documents and Settings\Denys\Desktop\S_Grep\source\CppParser.h(3):| CSC7110 Term Project - SmartGrep |
C:\Documents and Settings\Denys\Desktop\S_Grep\source\FileSearch.cpp(3):| CSC7110 Term Project - SmartGrep |
C:\Documents and Settings\Denys\Desktop\S_Grep\source\FileSearch.h(3):| CSC7110 Term Project - SmartGrep |
C:\Documents and Settings\Denys\Desktop\S_Grep\source\filetype.cpp(3):| CSC7110 Term Project - SmartGrep |
C:\Documents and Settings\Denys\Desktop\S_Grep\source\FileType.h(3):| CSC7110 Term Project - SmartGrep |
C:\Documents and Settings\Denys\Desktop\S_Grep\source\macro.h(3):| CSC7110 Term Project - SmartGrep |
C:\Documents and Settings\Denys\Desktop\S_Grep\source\Parser.cpp(3):| CSC7110 Term Project - SmartGrep |
C:\Documents and Settings\Denys\Desktop\S_Grep\source\Parser.h(3):| CSC7110 Term Project - SmartGrep |
C:\Documents and Settings\Denys\Desktop\S_Grep\source\ParsingRecord.cpp(3):| CSC7110 Term Project - SmartGrep |
C:\Documents and Settings\Denys\Desktop\S_Grep\source\ParsingRecord.h(3):| CSC7110 Term Project - SmartGrep |
C:\Documents and Settings\Denys\Desktop\S_Grep\source\SearchingRecord.h(3):| CSC7110 Term Project - SmartGrep |
C:\Documents and Settings\Denys\Desktop\S_Grep\source\StringProcessor.cpp(3):| CSC7110 Term Project - SmartGrep |
C:\Documents and Settings\Denys\Desktop\S_Grep\source\StringProcessor.h(3):| CSC7110 Term Project - SmartGrep |
C:\Documents and Settings\Denys\Desktop\S_Grep\borrowed_code\SolutionExplorer.cpp(160):    ofile << "<!-- Class Information Retrieve
Total found: 39    Matching files: 25    Total files searched: 44
```

WAYNE STATE UNIVERSITY

Feature Location with Information Retrieval [Marcus'04]

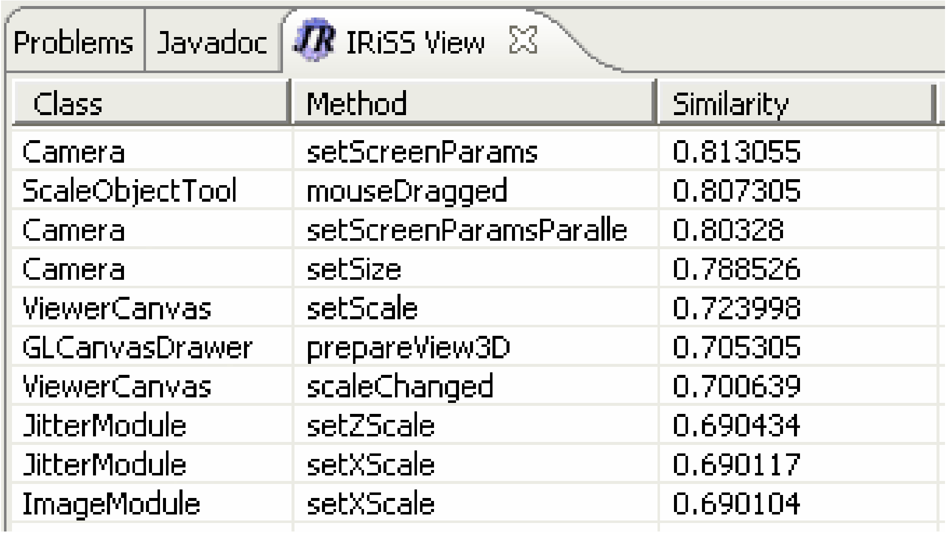


```
BSTnode& BST::insert(KeyType key, ValueType value)
// lookup key, create key/value if not present
// return pointer to node with key
{
    BSTnode& n = getNode(key);
    n._item.second = value;
    return n;
}

BSTnode& BST::getNode(KeyType key)
// lookup key, create key/value if not present
// return pointer to node with key
{
    pair<KeyType, ValueType> item(key, 0); // create item to be inserted

    if ( _root == 0 )
    {                                           // empty tree insert as root
        BSTnode* new_node = new BSTnode(0, 0, 0, item);
        _root = new_node;
        return *_root;
    }

    pair<BSTnode*, bool> result = find(key);    // search for key
                                                // location found
    if ( result.second )
    {                                           // true, node exists
       return *(result.first);
    }
    else
    {                                           // false, create node
       BSTnode* parent = result.first;
       BSTnode* new_node = new BSTnode(0, 0, parent, item);
       if ( key < parent->getKey() )
          parent->_left  = new_node;
       else
          parent->_right = new_node;
       return *new node;
```



Find What: set screen view

IRiSS Version 1.0
Wayne State University

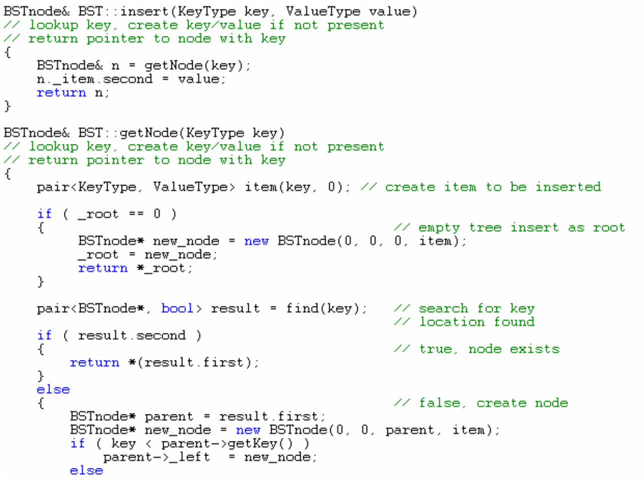


Class	Method	Similarity
Camera	setScreenParams	0.813055
ScaleObjectTool	mouseDragged	0.807305
Camera	setScreenParamsParalle	0.80328
Camera	setSize	0.788526
ViewerCanvas	setScale	0.723998
GLCanvasDrawer	prepareView3D	0.705305
ViewerCanvas	scaleChanged	0.700639
JitterModule	setZScale	0.690434
JitterModule	setXScale	0.690117
ImageModule	setXScale	0.690104

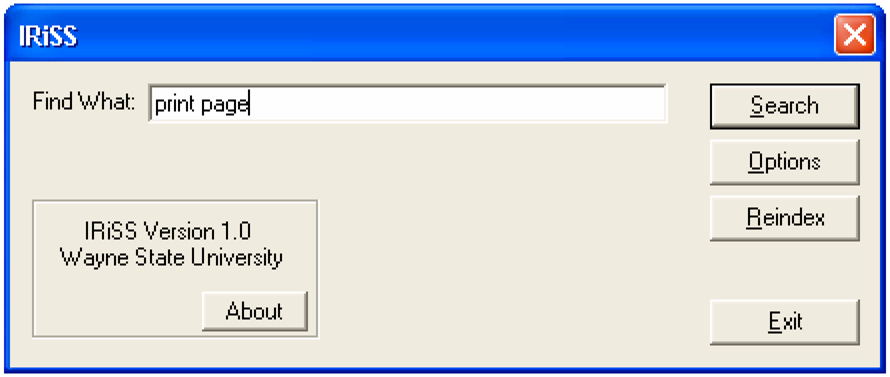
SEVERE GROUP
SoftwarE Visualization and Evolution REsearch GROUP

Feature Location with Information Retrieval and Formal Concept Analysis

[Poshyvanyk'07]

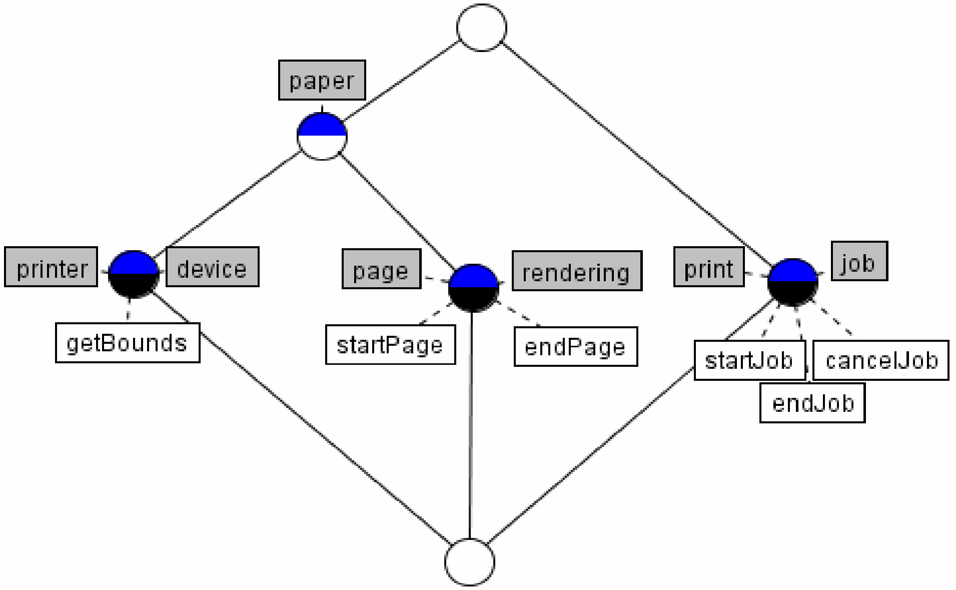


```
BSTnode& BST::insert(KeyType key, ValueType value)
// lookup key, create key/value if not present
// return pointer to node with key
{
    BSTnode& n = getNode(key);
    n._item.second = value;
    return n;
}

BSTnode& BST::getNode(KeyType key)
// lookup key, create key/value if not present
// return pointer to node with key
{
    pair<KeyType, ValueType> item(key, 0); // create item to be inserted

    if ( _root == 0 )
    {                                           // empty tree insert as root
         BSTnode* new_node = new BSTnode(0, 0, 0, item);
         _root = new_node;
         return *_root;
    }

    pair<BSTnode*, bool> result = find(key);    // search for key
                                                // location found
    if ( result.second )
    {                                           // true, node exists
        return *(result.first);
    }
    else
    {                                           // false, create node
        BSTnode* parent = result.first;
        BSTnode* new_node = new BSTnode(0, 0, parent, item);
        if ( key < parent->getKey() )
            parent->_left  = new_node;
        else
            parent->_right = new_
        return *new node;
```



Find What: print page

IRiSS Version 1.0
Wayne State University



Feature Location with Software Reconnaissance [Wilde'92]

Scenario NOT exercising the feature

```
readAndDispatch  --  org.eclipse.swt.widgets.Display
checkDevice  --  org.eclipse.swt.widgets.Display
isDisposed  --  org.eclipse.swt.graphics.Device
drawMenuBars  --  org.eclipse.swt.widgets.Display
runPopups  --  org.eclipse.swt.widgets.Display
filterMessage  --  org.eclipse.swt.widgets.Display
windowProc  --  org.eclipse.swt.widgets.Display
windowProc  --  org.eclipse.swt.widgets.Control
WM_TIMER  --  org.eclipse.swt.widgets.Control
windowProc  --  org.eclipse.swt.widgets.Display
windowProc  --  org.eclipse.swt.widgets.Control
WM_TIMER  --  org.eclipse.swt.widgets.Control
windowProc  --  org.eclipse.swt.widgets.Display
windowProc  --  org.eclipse.swt.widgets.Control
```

Scenario exercising the feature

```
checkDevice  --  org.eclipse.swt.widgets.Display
isDisposed  --  org.eclipse.swt.graphics.Device
drawMenuBars  --  org.eclipse.swt.widgets.Display
runPopups  --  org.eclipse.swt.widgets.Display
filterMessage  --  org.eclipse.swt.widgets.Display
windowProc  --  org.eclipse.swt.widgets.Display
windowProc  --  org.eclipse.swt.widgets.Control
WM_TIMER  --  org.eclipse.swt.widgets.ProgressBar
WM_TIMER  --  org.eclipse.swt.widgets.Control
windowProc  --  org.eclipse.swt.widgets.Display
windowProc  --  org.eclipse.swt.widgets.Control
WM_TIMER  --  org.eclipse.swt.widgets.Control
windowProc  --  org.eclipse.swt.widgets.Display
windowProc  --  org.eclipse.swt.widgets.Control
```

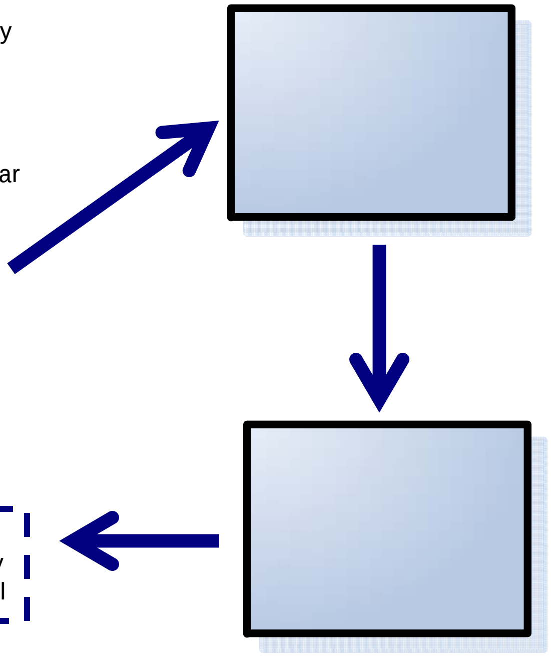
Feature Location with Scenario-based Probabilistic Ranking [Antoniol'06]

Scenario NOT exercising the feature

readAndDispatch -- org.eclipse.swt.widgets.Display
checkDevice -- org.eclipse.swt.widgets.Display
isDisposed -- org.eclipse.swt.graphics.Device
drawMenuBars -- org.eclipse.swt.widgets.Display
runPopups -- org.eclipse.swt.widgets.Display
filterMessage -- org.eclipse.swt.widgets.Display
windowProc -- org.eclipse.swt.widgets.Display
windowProc -- org.eclipse.swt.widgets.Control
WM_TIMER -- org.eclipse.swt.widgets.Control
windowProc -- org.eclipse.swt.widgets.Display
windowProc -- org.eclipse.swt.widgets.Control
WM_TIMER -- org.eclipse.swt.widgets.Control
windowProc -- org.eclipse.swt.widgets.Display
windowProc -- org.eclipse.swt.widgets.Control

Scenario exercising the feature

checkDevice -- org.eclipse.swt.widgets.Display
isDisposed -- org.eclipse.swt.graphics.Device
drawMenuBars -- org.eclipse.swt.widgets.Display
runPopups -- org.eclipse.swt.widgets.Display
filterMessage -- org.eclipse.swt.widgets.Display
windowProc -- org.eclipse.swt.widgets.Display
windowProc -- org.eclipse.swt.widgets.Control
WM_TIMER -- org.eclipse.swt.widgets.ProgressBar
WM_TIMER -- org.eclipse.swt.widgets.Control
windowProc -- org.eclipse.swt.widgets.Display
windowProc -- org.eclipse.swt.widgets.Control
WM_TIMER -- org.eclipse.swt.widgets.Control
windowProc -- org.eclipse.swt.widgets.Display
windowProc -- org.eclipse.swt.widgets.Control



WM_TIMER -- org.eclipse.swt.widgets.Control
windowProc -- org.eclipse.swt.widgets.Display
windowProc -- org.eclipse.swt.widgets.Control

Feature Location with Probabilistic Ranking Of MEthodS and Information Retrieval

[Poshyvanyk'07]

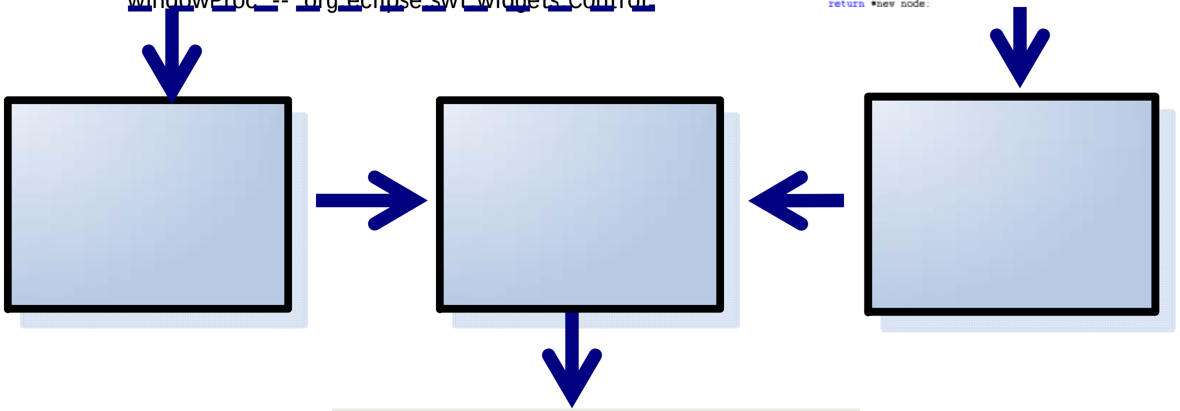
Scenario NOT exercising the feature

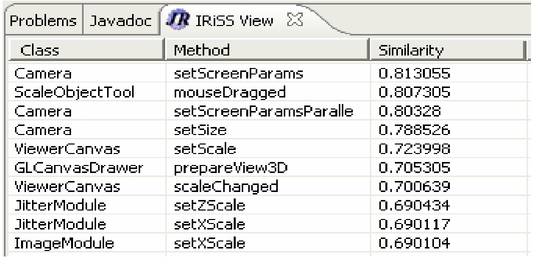
readAndDispatch -- org.eclipse.swt.widgets.Display
checkDevice -- org.eclipse.swt.widgets.Display
isDisposed -- org.eclipse.swt.graphics.Device
drawMenuBars -- org.eclipse.swt.widgets.Display
runPopups -- org.eclipse.swt.widgets.Display
filterMessage -- org.eclipse.swt.widgets.Display
windowProc -- org.eclipse.swt.widgets.Display
windowProc -- org.eclipse.swt.widgets.Control
WM_TIMER -- org.eclipse.swt.widgets.Control

Scenario exercising the feature

checkDevice -- org.eclipse.swt.widgets.Display
isDisposed -- org.eclipse.swt.graphics.Device
drawMenuBars -- org.eclipse.swt.widgets.Display
runPopups -- org.eclipse.swt.widgets.Display
checkDevice -- org.eclipse.swt.widgets.Display
isDisposed -- org.eclipse.swt.graphics.Device
drawMenuBars -- org.eclipse.swt.widgets.Display
runPopups -- org.eclipse.swt.widgets.Display
filterMessage -- org.eclipse.swt.widgets.Display
windowProc -- org.eclipse.swt.widgets.Display
windowProc -- org.eclipse.swt.widgets.Control

```
BSTnode& BST::insert(KeyType key, ValueType value)
// lookup key, create key/value if not present
// return pointer to node with key
{
    BSTnode& n = getNode(key);
    n._item.second = value;
    return n;
}

BSTnode& BST::getNode(KeyType key)
// lookup key, create key/value if not present
// return pointer to node with key
{
    pair<KeyType, ValueType> item(key, 0); // create item to be inserted

    if ( _root == 0 )
    {                                          // empty tree insert as root
        BSTnode* new_node = new BSTnode(0, 0, 0, item);
        _root = new_node;
        return *_root;
    }

    pair<BSTnode*, bool> result = find(key);   // search for key
                                               // location found
    if ( result.second )
    {                                          // true, node exists
       return *(result.first);
    }
    else
    {                                          // false, create node
       BSTnode* parent = result.first;
       BSTnode* new_node = new BSTnode(0, 0, parent, item);
       if ( key < parent->getKey() )
           parent->_left  = new_node;
       else
           parent->_right = new_node;
       return *new node;
```



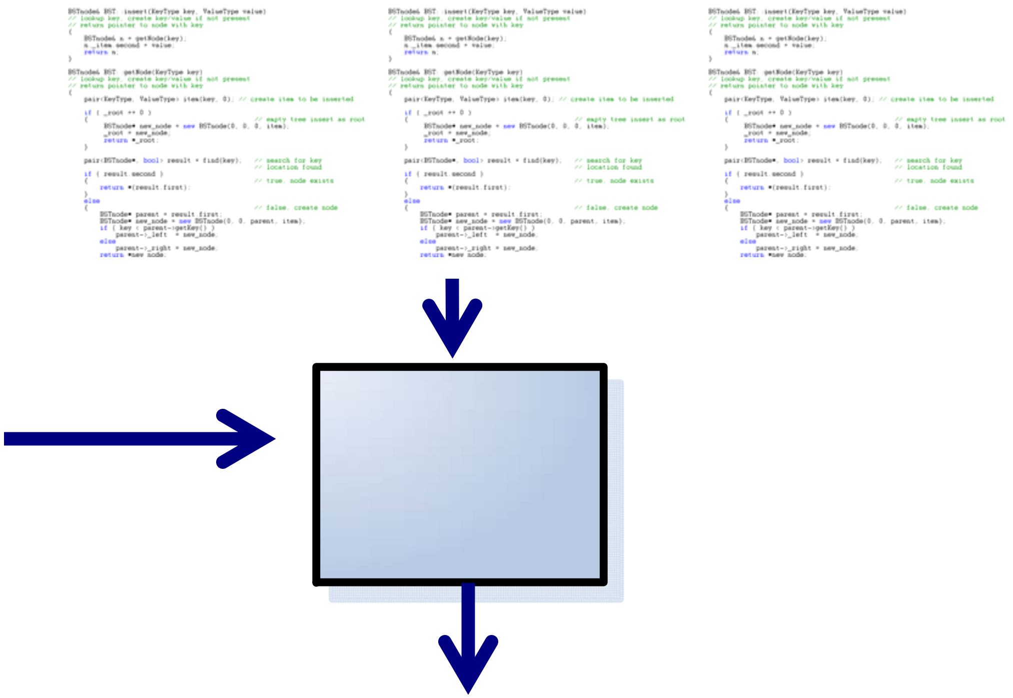


Class	Method	Similarity
Camera	setScreenParams	0.813055
ScaleObjectTool	mouseDragged	0.807305
Camera	setScreenParamsParalle	0.80328
Camera	setSize	0.788526
ViewerCanvas	setScale	0.723998
GLCanvasDrawer	prepareView3D	0.705305
ViewerCanvas	scaleChanged	0.700639
JitterModule	setZScale	0.690434
JitterModule	setXScale	0.690117
ImageModule	setXScale	0.690104

SITIR = SIngle Trace + Information Retrieval (SITIR)

Single Execution Trace

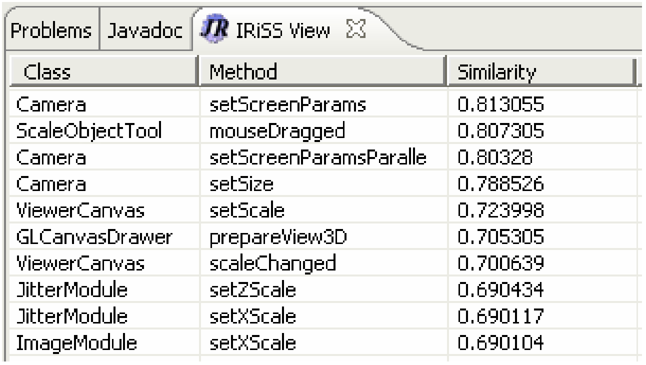
readAndDispatch -- org.eclipse.swt.widgets.Display
checkDevice -- org.eclipse.swt.widgets.Display
isDisposed -- org.eclipse.swt.graphics.Device
drawMenuBars -- org.eclipse.swt.widgets.Display
runPopups -- org.eclipse.swt.widgets.Display
filterMessage -- org.eclipse.swt.widgets.Display
windowProc -- org.eclipse.swt.widgets.Display
windowProc -- org.eclipse.swt.widgets.Control
WM_TIMER -- org.eclipse.swt.widgets.Control
windowProc -- org.eclipse.swt.widgets.Display
windowProc -- org.eclipse.swt.widgets.Control
WM_TIMER -- org.eclipse.swt.widgets.Control
windowProc -- org.eclipse.swt.widgets.Display
windowProc -- org.eclipse.swt.widgets.Control
readAndDispatch -- org.eclipse.swt.widgets.Display
checkDevice -- org.eclipse.swt.widgets.Display
isDisposed -- org.eclipse.swt.graphics.Device
drawMenuBars -- org.eclipse.swt.widgets.Display
runPopups -- org.eclipse.swt.widgets.Display
filterMessage -- org.eclipse.swt.widgets.Display
windowProc -- org.eclipse.swt.widgets.Display
windowProc -- org.eclipse.swt.widgets.Control
WM_TIMER -- org.eclipse.swt.widgets.Control
windowProc -- org.eclipse.swt.widgets.Display
windowProc -- org.eclipse.swt.widgets.Control
WM_TIMER -- org.eclipse.swt.widgets.Control
windowProc -- org.eclipse.swt.widgets.Display
windowProc -- org.eclipse.swt.widgets.Control



```
BSTnode& BST::insert(KeyType key, ValueType value)
// lookup key, create key/value if not present
// return pointer to node with key
{
    BSTnode& n = getNode(key);
    n._item.second = value;
    return n;
}

BSTnode& BST::getNode(KeyType key)
// lookup key, create key/value if not present
// return pointer to node with key
{
    pair<KeyType, ValueType> item(key, 0); // create item to be inserted

    if ( _root == 0 )
    {                                          // empty tree insert as root
        BSTnode* new_node = new BSTnode(0, 0, 0, item);
        _root = new_node;
        return *_root;
    }

    pair<BSTnode*, bool> result = find(key);   // search for key
                                               // location found
    if ( result.second )
    {                                          // true, node exists
       return *(result.first);
    }
    else
    {                                          // false, create node
       BSTnode* parent = result.first;
       BSTnode* new_node = new BSTnode(0, 0, parent, item);
       if ( key < parent->getKey() )
           parent->_left  = new_node;
       else
           parent->_right = new_node;
       return *new_node;
```

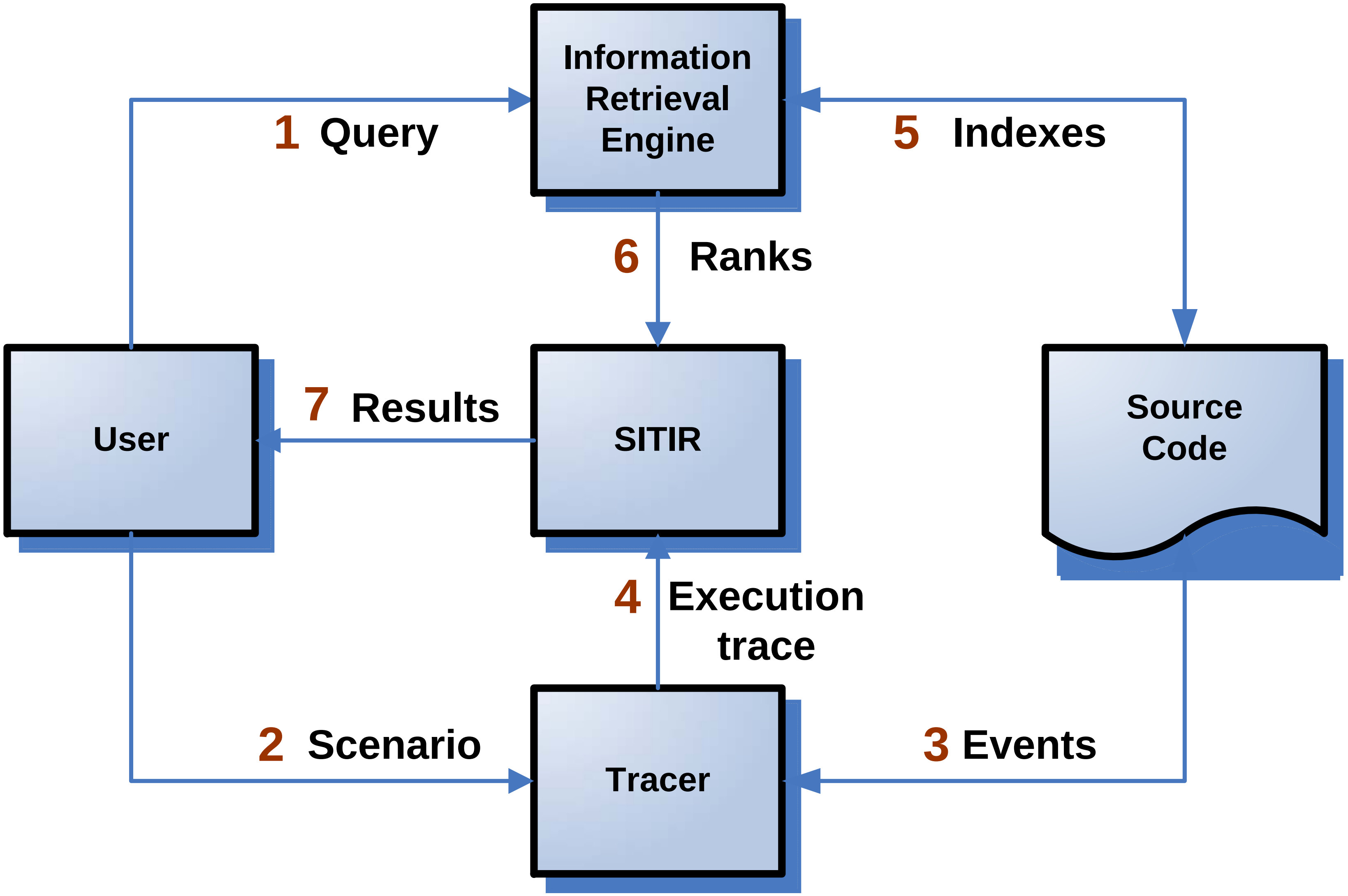
```
BSTnode& BST::insert(KeyType key, ValueType value)
// lookup key, create key/value if not present
// return pointer to node with key
{
    BSTnode& n = getNode(key);
    n._item.second = value;
    return n;
}

BSTnode& BST::getNode(KeyType key)
// lookup key, create key/value if not present
// return pointer to node with key
{
    pair<KeyType, ValueType> item(key, 0); // create item to be inserted

    if ( _root == 0 )
    {                                          // empty tree insert as root
        BSTnode* new_node = new BSTnode(0, 0, 0, item);
        _root = new_node;
        return *_root;
    }

    pair<BSTnode*, bool> result = find(key);   // search for key
                                               // location found
    if ( result.second )
    {                                          // true, node exists
       return *(result.first);
    }
    else
    {                                          // false, create node
       BSTnode* parent = result.first;
       BSTnode* new_node = new BSTnode(0, 0, parent, item);
       if ( key < parent->getKey() )
           parent->_left  = new_node;
       else
           parent->_right = new_node;
       return *new_node;
```

```
BSTnode& BST::insert(KeyType key, ValueType value)
// lookup key, create key/value if not present
// return pointer to node with key
{
    BSTnode& n = getNode(key);
    n._item.second = value;
    return n;
}

BSTnode& BST::getNode(KeyType key)
// lookup key, create key/value if not present
// return pointer to node with key
{
    pair<KeyType, ValueType> item(key, 0); // create item to be inserted

    if ( _root == 0 )
    {                                          // empty tree insert as root
        BSTnode* new_node = new BSTnode(0, 0, 0, item);
        _root = new_node;
        return *_root;
    }

    pair<BSTnode*, bool> result = find(key);   // search for key
                                               // location found
    if ( result.second )
    {                                          // true, node exists
       return *(result.first);
    }
    else
    {                                          // false, create node
       BSTnode* parent = result.first;
       BSTnode* new_node = new BSTnode(0, 0, parent, item);
       if ( key < parent->getKey() )
           parent->_left  = new_node;
       else
           parent->_right = new_node;
       return *new_node;
```



Class	Method	Similarity
Camera	setScreenParams	0.813055
ScaleObjectTool	mouseDragged	0.807305
Camera	setScreenParamsParalle	0.80328
Camera	setSize	0.788526
ViewerCanvas	setScale	0.723998
GLCanvasDrawer	prepareView3D	0.705305
ViewerCanvas	scaleChanged	0.700639
JitterModule	setZScale	0.690434
JitterModule	setXScale	0.690117
ImageModule	setXScale	0.690104

Filtering Single Execution Trace with Information Retrieval



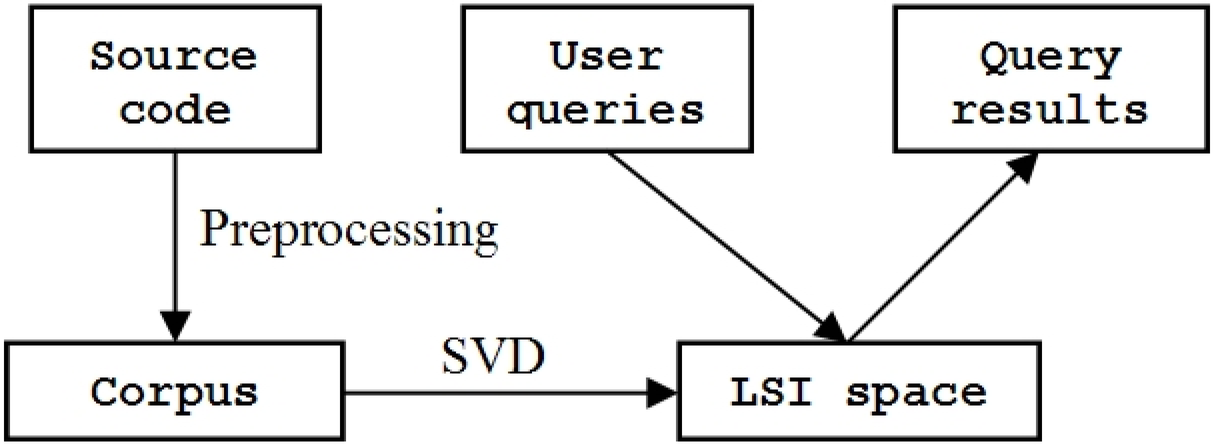
Collecting Execution Traces

- Java Platform Debugger Architecture (JPDA)[1]
  - Infrastructure to build end-user debugging applications for Java platform
- JPDA highlights:
  - Debugger works on a separate virtual machine
  - Minimal interference of a tracing tool with a subject program
  - Separate thread-based traces
  - Marked traces (start/stop recording)

[1] http://java.sun.com/javase/technologies/core/toolsapis/jpda/

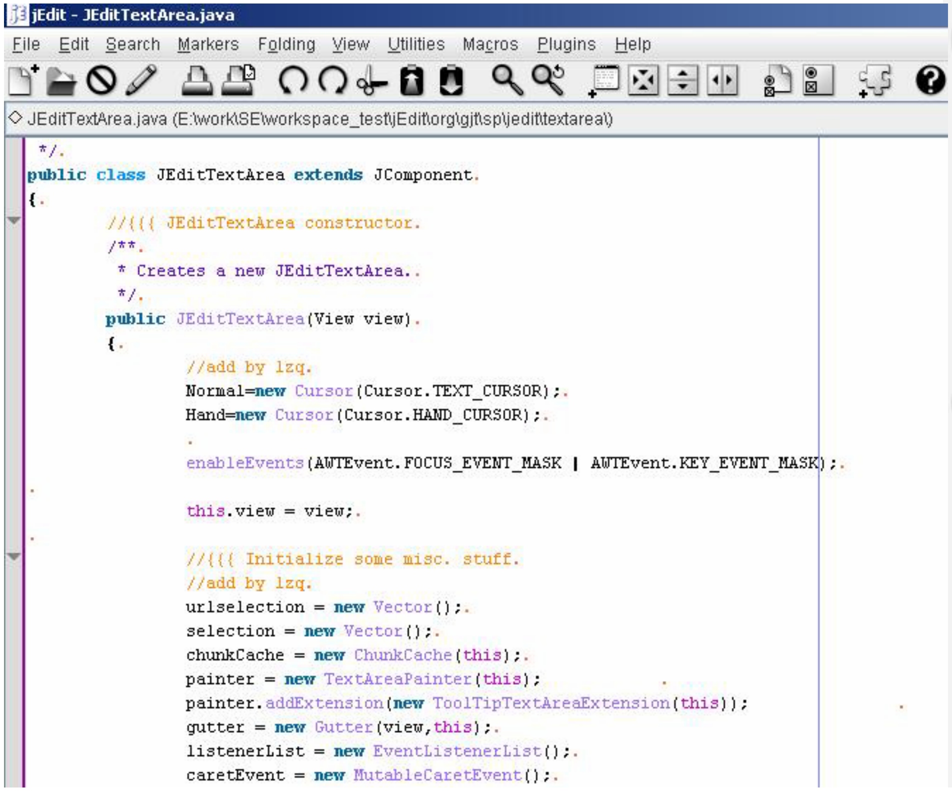
Indexing Source Code with Latent Semantic Indexing

- Parsing to extract semantic information (i.e., comments and identifiers)
- Pre-processing: split_identifiers & SplitIdentifiers
- Representing software systems as a term-document matrix



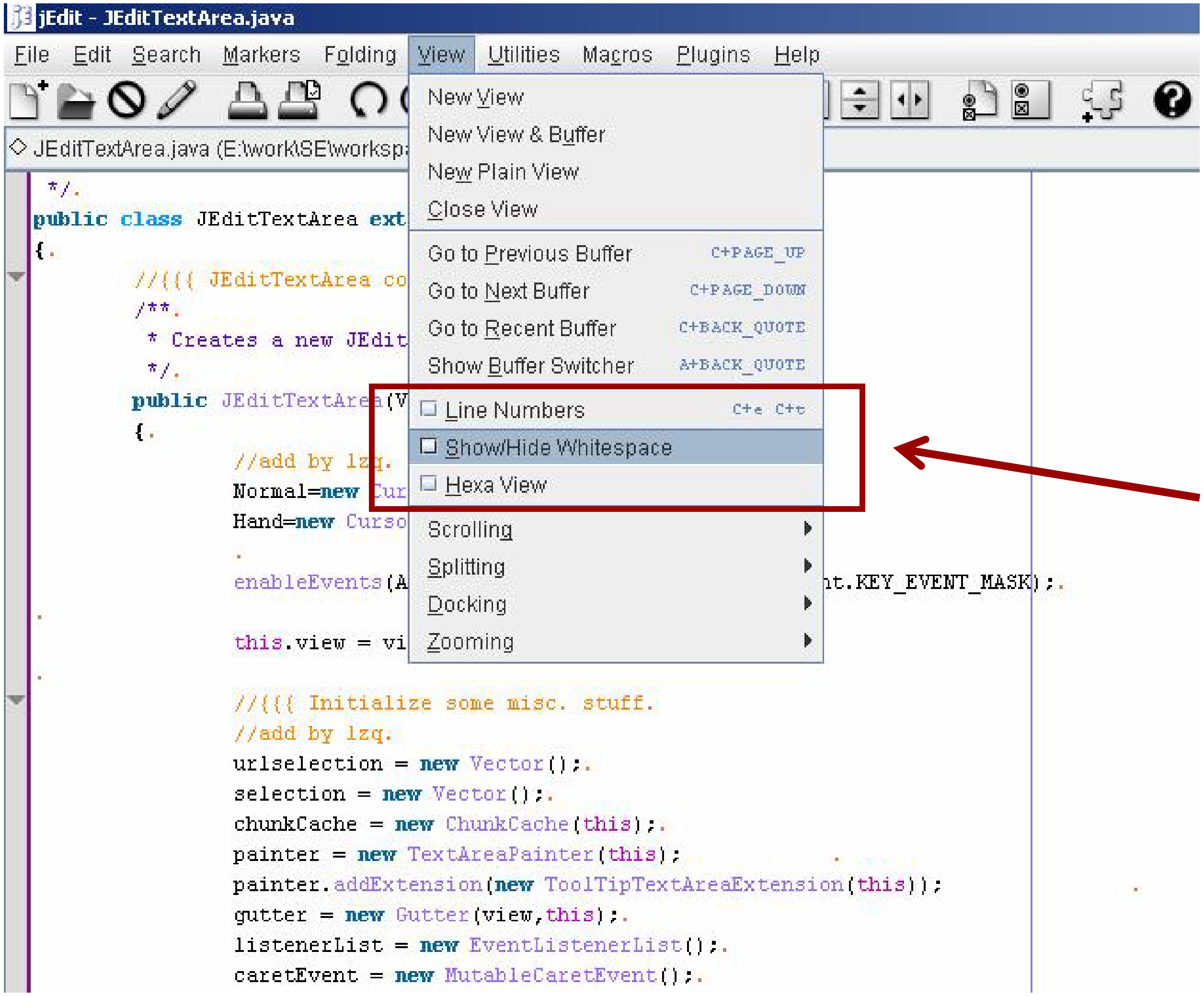
Example of using SITIR

- Locating a feature in JEdit
- Feature: “showing **white-space** as a **visible** symbol in the **text area**”
- Steps:
  - Run scenario
  - Run query
  - Explore results


jEdit - JEditTextArea.java
File
Edit
Search
Markers
Folding
View
Utilities
Macros
Plugins
Help
JEditTextArea.java (E:\work\SE\workspace_test\jEdit\org\gjt\sp\jedit\textarea\)

```
 */.
public class JEditTextArea extends JComponent.
{.
        //{{{ JEditTextArea constructor.
        /**.
         * Creates a new JEditTextArea..
         */.
        public JEditTextArea(View view).
        {.
                //add by lzq.
                Normal=new Cursor(Cursor.TEXT_CURSOR);.
                Hand=new Cursor(Cursor.HAND_CURSOR);.
                .
                enableEvents(AWTEvent.FOCUS_EVENT_MASK | AWTEvent.KEY_EVENT_MASK);.
.
                this.view = view;.
.
                //{{{ Initialize some misc. stuff.
                //add by lzq.
                urlselection = new Vector();.
                selection = new Vector();.
                chunkCache = new ChunkCache(this);.
                painter = new TextAreaPainter(this);                .
                painter.addExtension(new ToolTipTextAreaExtension(this));          .
                gutter = new Gutter(view,this);.
                listenerList = new EventListenerList();.
                caretEvent = new MutableCaretEvent();.
```

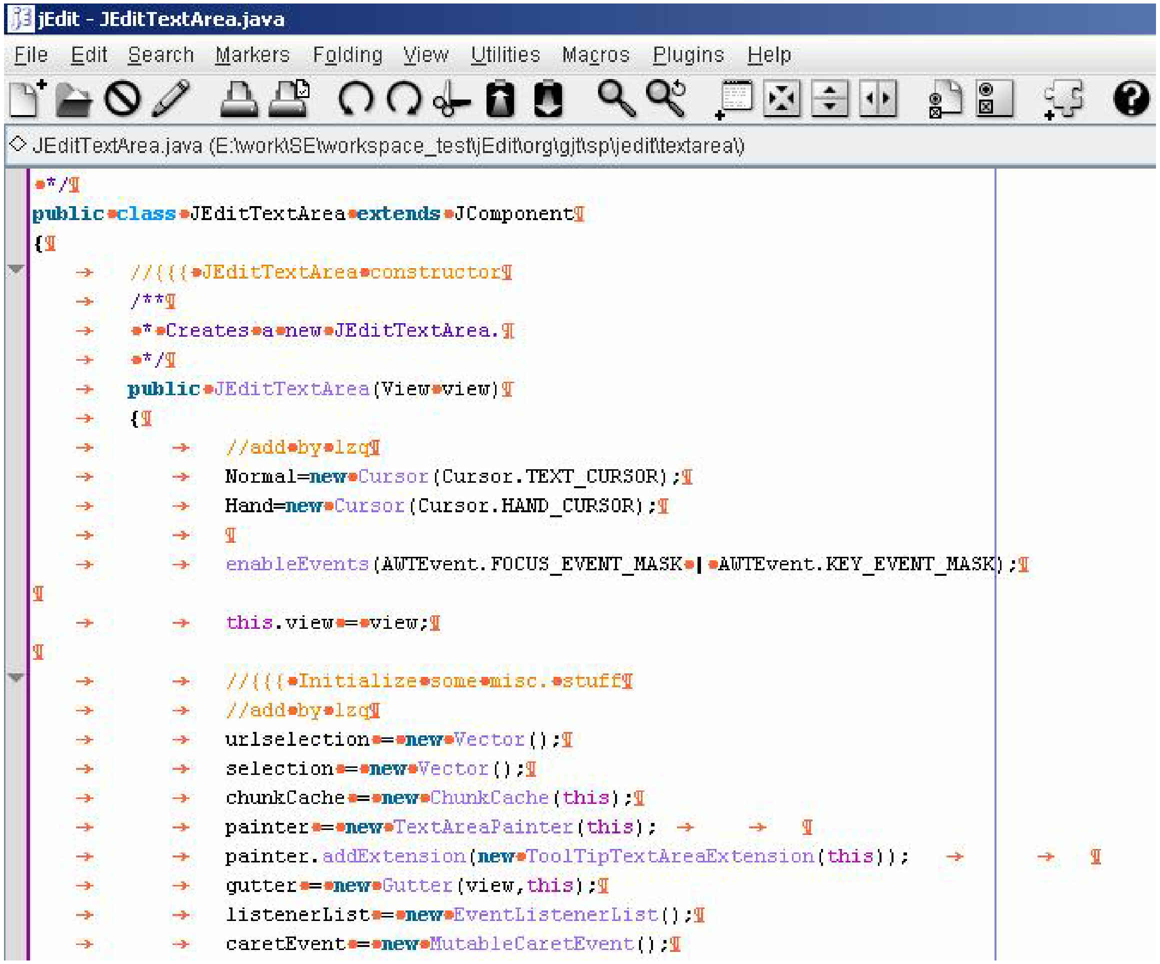
Example of using SITIR – Locating a Feature in JEdit


jEdit - JEditTextArea.java
File Edit Search Markers Folding View Utilities Macros Plugins Help
JEditTextArea.java (E:\work\SE\worksp
New View
New View & Buffer
New Plain View
Close View
Go to Previous Buffer C+PAGE_UP
Go to Next Buffer C+PAGE_DOWN
Go to Recent Buffer C+BACK_QUOTE
Show Buffer Switcher A+BACK_QUOTE
□ Line Numbers C+e C+t
□ Show/Hide Whitespace
□ Hexa View
Scrolling
Splitting
Docking
Zooming

```
 */
public class JEditTextArea ext
{
	//{{{ JEditTextArea co
	/**
	 * Creates a new JEdit
	 */
	public JEditTextArea(V
	{
		//add by lzq
		Normal=new Cur
		Hand=new Curso

		enableEvents(A                          nt.KEY_EVENT_MASK);

		this.view = vi

		//{{{ Initialize some misc. stuff
		//add by lzq
		urlselection = new Vector();
		selection = new Vector();
		chunkCache = new ChunkCache(this);
		painter = new TextAreaPainter(this);
		painter.addExtension(new ToolTipTextAreaExtension(this));
		gutter = new Gutter(view,this);
		listenerList = new EventListenerList();
		caretEvent = new MutableCaretEvent();
```

Start Tracing

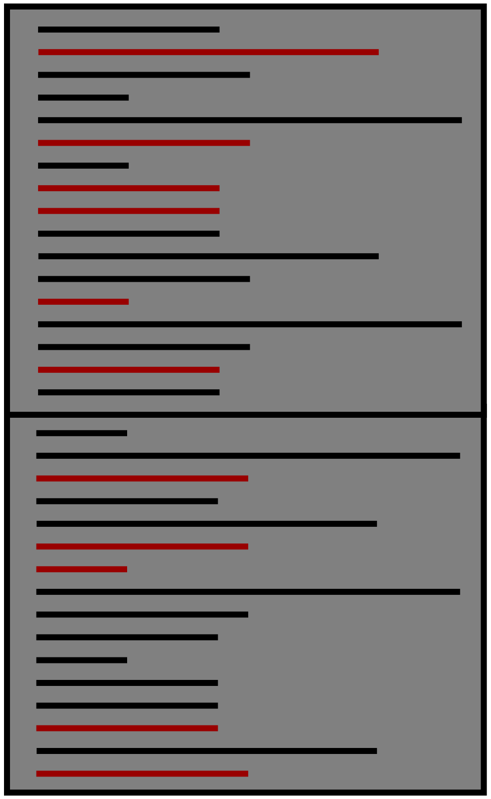
Example of using SITIR – Locating a Feature in JEdit


jEdit - JEditTextArea.java
File Edit Search Markers Folding View Utilities Macros Plugins Help
JEditTextArea.java (E:\work\SE\workspace_test\jEdit\org\gjt\sp\jedit\textarea\)

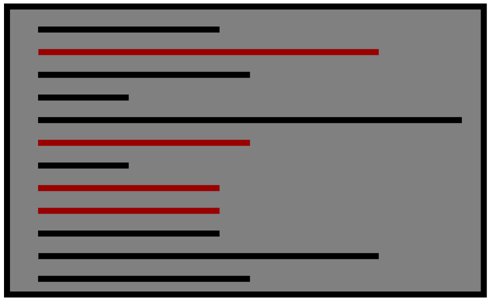
```
 */
public class JEditTextArea extends JComponent
{
	//{{{ JEditTextArea constructor
	/**
	 * Creates a new JEditTextArea.
	 */
	public JEditTextArea(View view)
	{
		//add by lzq
		Normal=new Cursor(Cursor.TEXT_CURSOR);
		Hand=new Cursor(Cursor.HAND_CURSOR);
		
		enableEvents(AWTEvent.FOCUS_EVENT_MASK | AWTEvent.KEY_EVENT_MASK);

		this.view = view;

		//{{{ Initialize some misc. stuff
		//add by lzq
		urlselection = new Vector();
		selection = new Vector();
		chunkCache = new ChunkCache(this);
		painter = new TextAreaPainter(this);		
		painter.addExtension(new ToolTipTextAreaExtension(this));		
		gutter = new Gutter(view,this);
		listenerList = new EventListenerList();
		caretEvent = new MutableCaretEvent();
```

Stop Tracing

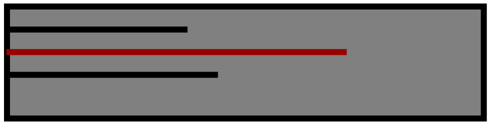
Example of using SITIR – Results



Single Trace



IR-based ranking



SITIR

- Number of methods in single trace – **284**
- The position of the first relevant method according to IR ranking – **56**
- Position of the first relevant method according to SITIR - **7**

Case Studies

- Locating features in JEdit
- Locating features associated with bugs in Eclipse

- Case study objectives:
  - Compare SITIR with other approaches
  - Study the impact of full and marked traces on the results
  - Study the impact of user queries on the results

Locating Features in JEdit

- Evaluation – the position of the first relevant method

Feature	MARKED TRACE	LSI	SITIR
Search	202	59 (36)[1]	14 (9)
Show whitespace	284	56 (43)	7 (5)
Add marker	304	5	1

[1] Refined queries are in parentheses

Different Users and Queries

Feature	#	Query
Search	1	search find phrase word text
	2	search final all forward backward case sensitive
	3	find search locate match indexof findnext
	4	searchdialog find findbtn searchselection save searchfileset searchandreplace
Show white space	1	red dot newline whitespace view show display tab
	2	show hide whitespace blank space display
	3	symbol replace changecolor setvisible addlayer whitespace loadsymbol
	4	userinput textareapainter paint whitespace newline pnt
Add marker	1	marker select word display text
	2	add remove marker markers
	3	select highlight mark change background
	4	buffer addmarker marker selection

Results for Different Users and Queries

Feature	Dev	Full Trace	Marked Trace	LSI	SITIR full	SITIR marked
Search	1	1477	202	61	11	6
	2	1477	202	243	57	20
	3	1477	202	32	13	6
	4	1477	202	189	36	11
Show white space	1	1462	284	956	152	30
	2	1462	284	626	130	48
	3	1462	284	497	104	16
	4	1462	284	78	23	8
Add marker	1	1478	304	26	5	1
	2	1478	304	1	1	1
	3	1478	304	3242	662	160
	4	1478	304	20	5	4

Full versus Marked Traces in SITIR

- Re-formulating queries on full traces brings no visible improvement to the results

- Using marked traces vice full with the same query always produces noticeable improvement to the results

- Observed improvement: reformulating a query and marking the trace at the same time!

Locating Features in Eclipse

- Comparing SITIR with PROMESIR
- Eclipse 2.1.3
  - 7K classes; 89K methods; 2.3MLOC; 57K unique words
- Features associated with bugs:
  - Well-known, documented and reproducible
  - Approved patches applied in recent releases
  - We locate the part of the feature associated with the bug

Locating Features in Eclipse - Results

Feature	SINGLE TRACE	SPR	PROMESIR	SITIR
Select	721	268	1	2
Add files	740	170	1	2
Search	771	456	3	2

Discussion

- Uses only single scenario
- Less sensitive to scenario selection
- Unobtrusive tracing mechanism
- IR-based indexing of source code can be easily extended to other languages
- SITIR results are comparable or better than other feature location methods