

FLAT³: Feature Location & Textual Tracing Tool

Trevor Savage, Meghan Revelle, Denys Poshyvanyk

SEMERU Group @ William and Mary



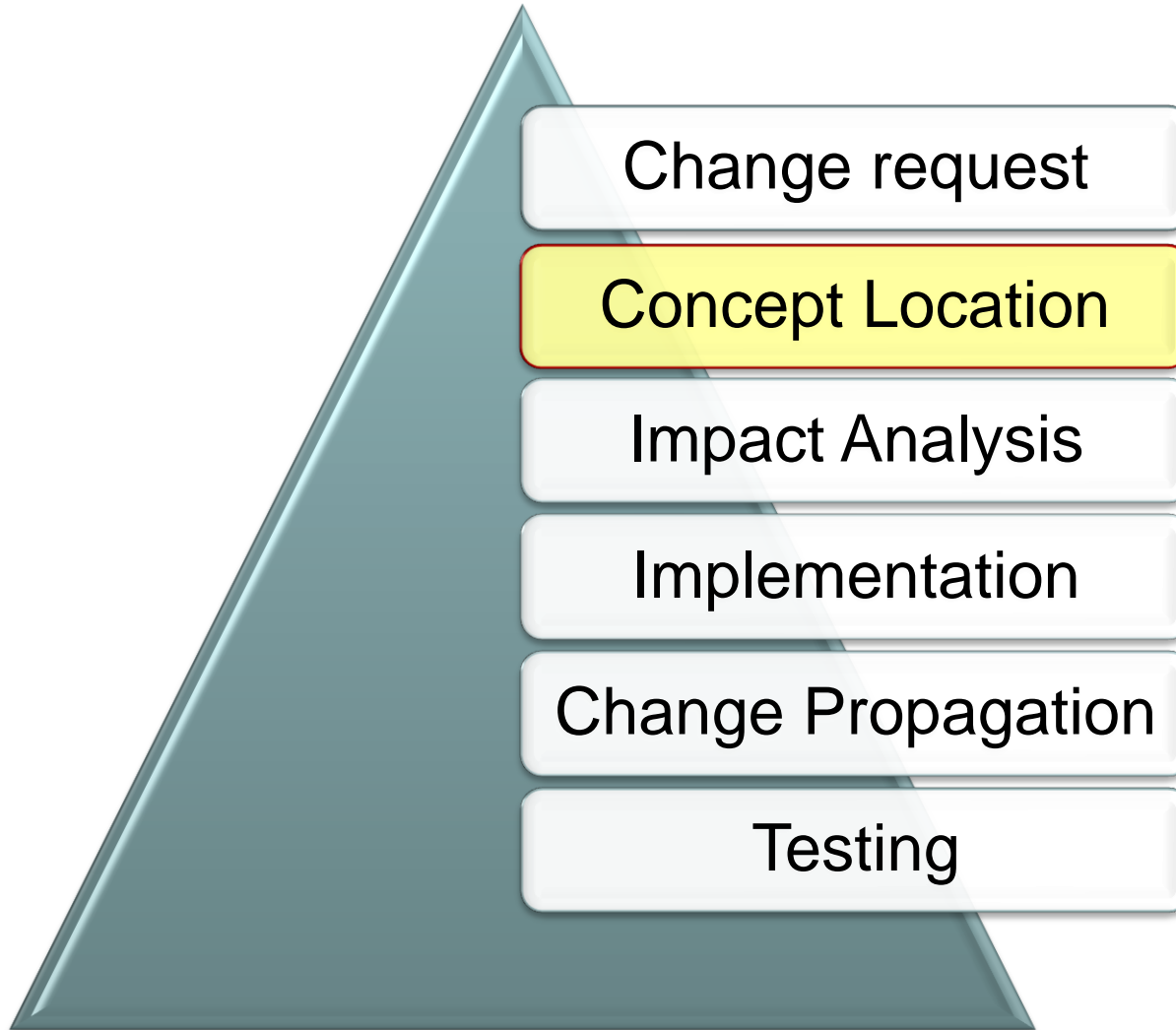
SEMERU



Addressed Problem

- The software developer has to maintain large software systems with:
 - Little or no domain knowledge
 - Absence of the original developer
 - Badly organized, missing, or out of date documentation

Concept Location in Source Code



“Finding a Needle in a Haystack”

```
...
** Search term type representation
* Date Representation
* 2023-04
*
#include <iostream>
#include <string>
#include <string>
#include <string>
#include <string>
using std::cin;
using std::cout;
using std::endl;
using std::string;

BTNode BTNode(BTNode left, BTNode right,
               BTNode parent, pair<KeyType, ValueType> item)
{
    left(left), right(right), parent(parent), item(item) {}
}

KeyType BTNode::getKey()
{
    return _item.first;
}

void BTNode::setKey(KeyType key)
{
    _item.first = key;
}

ValueType BTNode::getValue()
{
    return _item.second;
}

void BTNode::setValue(ValueType value)
{
    _item.second = value;
}

std::ostream operator<>(std::ostream os, const BTNode& n)
{
    os << "(" << n._item.first << ", " << n._item.second << ")";
    return os;
}

pair<BTNode*, bool> BST::find(KeyType key)
{
    Search term for key returns possibilities
    // If empty tree, false
    // If not found, node to parent where node should go, false
    // If found, node to node with key, true
}

...
** Search term type representation
* Date Representation
* 2023-04
*
#include <iostream>
#include <string>
#include <string>
#include <string>
#include <string>
using std::cin;
using std::cout;
using std::endl;
using std::string;

BTNode BTNode(BTNode left, BTNode right,
               BTNode parent, pair<KeyType, ValueType> item)
{
    left(left), right(right), parent(parent), item(item) {}
}

KeyType BTNode::getKey()
{
    return _item.first;
}

void BTNode::setKey(KeyType key)
{
    _item.first = key;
}

ValueType BTNode::getValue()
{
    return _item.second;
}

void BTNode::setValue(ValueType value)
{
    _item.second = value;
}

std::ostream operator<>(std::ostream os, const BTNode& n)
{
    os << "(" << n._item.first << ", " << n._item.second << ")";
    return os;
}

pair<BTNode*, bool> BST::find(KeyType key)
{
    Search term for key returns possibilities
    // If empty tree, false
    // If not found, node to parent where node should go, false
    // If found, node to node with key, true
}

...
** Search term type representation
* Date Representation
* 2023-04
*
#include <iostream>
#include <string>
#include <string>
#include <string>
#include <string>
using std::cin;
using std::cout;
using std::endl;
using std::string;

BTNode BTNode(BTNode left, BTNode right,
               BTNode parent, pair<KeyType, ValueType> item)
{
    left(left), right(right), parent(parent), item(item) {}
}

KeyType BTNode::getKey()
{
    return _item.first;
}

void BTNode::setKey(KeyType key)
{
    _item.first = key;
}

ValueType BTNode::getValue()
{
    return _item.second;
}

void BTNode::setValue(ValueType value)
{
    _item.second = value;
}

std::ostream operator<>(std::ostream os, const BTNode& n)
{
    os << "(" << n._item.first << ", " << n._item.second << ")";
    return os;
}

pair<BTNode*, bool> BST::find(KeyType key)
{
    Search term for key returns possibilities
    // If empty tree, false
    // If not found, node to parent where node should go, false
    // If found, node to node with key, true
}

...
** Search term type representation
* Date Representation
* 2023-04
*
#include <iostream>
#include <string>
#include <string>
#include <string>
#include <string>
using std::cin;
using std::cout;
using std::endl;
using std::string;

BTNode BTNode(BTNode left, BTNode right,
               BTNode parent, pair<KeyType, ValueType> item)
{
    left(left), right(right), parent(parent), item(item) {}
}

KeyType BTNode::getKey()
{
    return _item.first;
}

void BTNode::setKey(KeyType key)
{
    _item.first = key;
}

ValueType BTNode::getValue()
{
    return _item.second;
}

void BTNode::setValue(ValueType value)
{
    _item.second = value;
}

std::ostream operator<>(std::ostream os, const BTNode& n)
{
    os << "(" << n._item.first << ", " << n._item.second << ")";
    return os;
}

pair<BTNode*, bool> BST::find(KeyType key)
{
    Search term for key returns possibilities
    // If empty tree, false
    // If not found, node to parent where node should go, false
    // If found, node to node with key, true
}

...
** Search term type representation
* Date Representation
* 2023-04
*
#include <iostream>
#include <string>
#include <string>
#include <string>
#include <string>
using std::cin;
using std::cout;
using std::endl;
using std::string;

BTNode BTNode(BTNode left, BTNode right,
               BTNode parent, pair<KeyType, ValueType> item)
{
    left(left), right(right), parent(parent), item(item) {}
}

KeyType BTNode::getKey()
{
    return _item.first;
}

void BTNode::setKey(KeyType key)
{
    _item.first = key;
}

ValueType BTNode::getValue()
{
    return _item.second;
}

void BTNode::setValue(ValueType value)
{
    _item.second = value;
}

std::ostream operator<>(std::ostream os, const BTNode& n)
{
    os << "(" << n._item.first << ", " << n._item.second << ")";
    return os;
}

pair<BTNode*, bool> BST::find(KeyType key)
{
    Search term for key returns possibilities
    // If empty tree, false
    // If not found, node to parent where node should go, false
    // If found, node to node with key, true
}

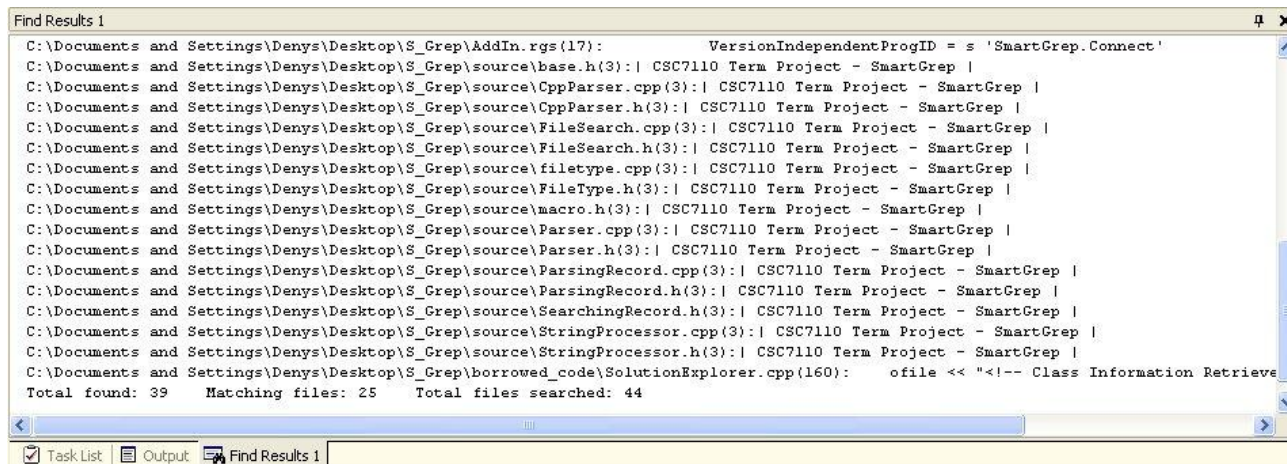
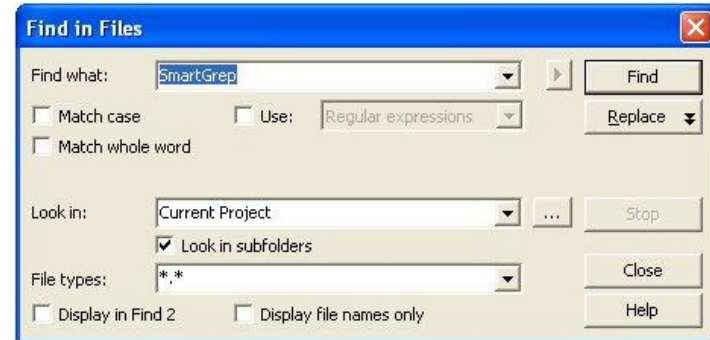
...

```

Concept 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 )
    {
        BSTNode* new_node = new BSTNode(0, 0, 0, item); // empty tree insert as root
        _root = new_node;
        return *_root;
    }
    pair<BSTNode*, bool> result = find(key); // search for key
    // location found
    if ( result.second )
    {
        return *(result.first); // true, node exists
    }
    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;
    }
}
```



Shortcomings of Static Concept Location

- Highly dependent on naming conventions and the developer's experience to write good queries
- Ignores other existing relationships between software components (such as, dependencies)
- May miss important parts of the source code

Feature Location with Software Reconnaissance - Dynamic Analysis

[Wilde'92][Antoniol'06]

Scenario NOT exercising the feature (trace 1)

readAndDispatch -- org.eclipse.swt.widgets.Display
checkDevice -- org.eclipse.swt.widgets.Display
isDisposed -- org.eclipse.swt.graphics.Device
drawMenuBar -- 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 (trace 2)

checkDevice -- org.eclipse.swt.widgets.Display
isDisposed -- org.eclipse.swt.graphics.Device
drawMenuBar -- 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
readAndDispatch -- org.eclipse.swt.widgets.Display
checkDevice -- org.eclipse.swt.widgets.Display
isDisposed -- org.eclipse.swt.graphics.Device
drawMenuBar -- org.eclipse.swt.widgets.Display
runPopups -- org.eclipse.swt.widgets.Display
runAsyncMessages -- org.eclipse.swt.widgets.Display
removeFirst -- org.eclipse.swt.widgets.Synchronizer

Shortcomings of Dynamic Concept Location

- Execution traces are large even for small systems
- Selecting (ir)relevant scenarios may be difficult
- Filtering the traces is equally problematic - best filtering methods still return hundreds of methods

Single Trace Information Retrieval (SITIR)

[ASE'07]

Source Code

Single Execution Trace

```
readAndDispatch -- org.eclipse.swt.widgets.Display
checkDevice -- org.eclipse.swt.widgets.Display
isDisposed -- org.eclipse.swt.graphics.Device
drawMenuBar -- 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
drawMenuBar -- 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
```

```
BTNode::BST insert(KeyType key, ValueType value)
// lookup key, create node/value if not present
// return pointer to node with key
{
    BTNode* n = getNode(key);
    n->time.second = value;
    return n;
}

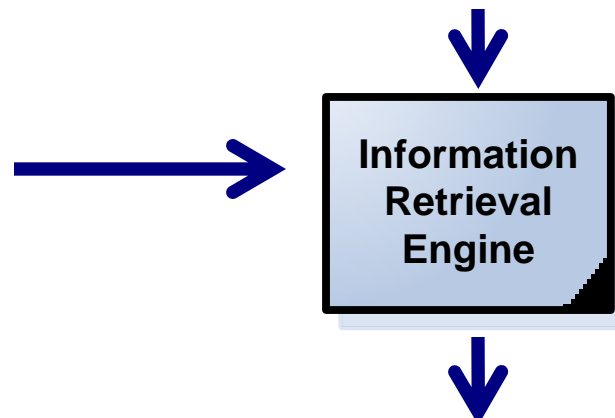
BTNode::BST getNode(KeyType key)
// lookup key, create key/value if not present
// return pointer to node with key
{
    pair(KeyType, ValueType) s(key, 0); // create item to be inserted
    if (!root)
    {
        // apply tree insert so root
        // exists = new node.
        return *root;
    }
    pair(BTNode*, bool) result = find(key); // search for key
    if (!result.second) // insertion failed
    {
        return *(result.first); // true, node exists
    }
    else // (also, create node
    {
        BTNode parent = result.first;
        BTNode* new_node = new BTNode(0, 0, parent, s);
        if (key < parent->key)
            parent->left = new_node;
        else
            parent->right = new_node;
        return *new_node;
    }
}

BTNode::BST insert(KeyType key, ValueType value)
// lookup key, create node/value if not present
// return pointer to node with key
{
    BTNode* n = getNode(key);
    n->time.second = value;
    return n;
}

BTNode::BST getNode(KeyType key)
// lookup key, create key/value if not present
// return pointer to node with key
{
    pair(KeyType, ValueType) s(key, 0); // create item to be inserted
    if (!root)
    {
        // apply tree insert so root
        // exists = new node.
        return *root;
    }
    pair(BTNode*, bool) result = find(key); // search for key
    if (!result.second) // insertion failed
    {
        return *(result.first); // true, node exists
    }
    else // (also, create node
    {
        BTNode parent = result.first;
        BTNode* new_node = new BTNode(0, 0, parent, s);
        if (key < parent->key)
            parent->left = new_node;
        else
            parent->right = new_node;
        return *new_node;
    }
}

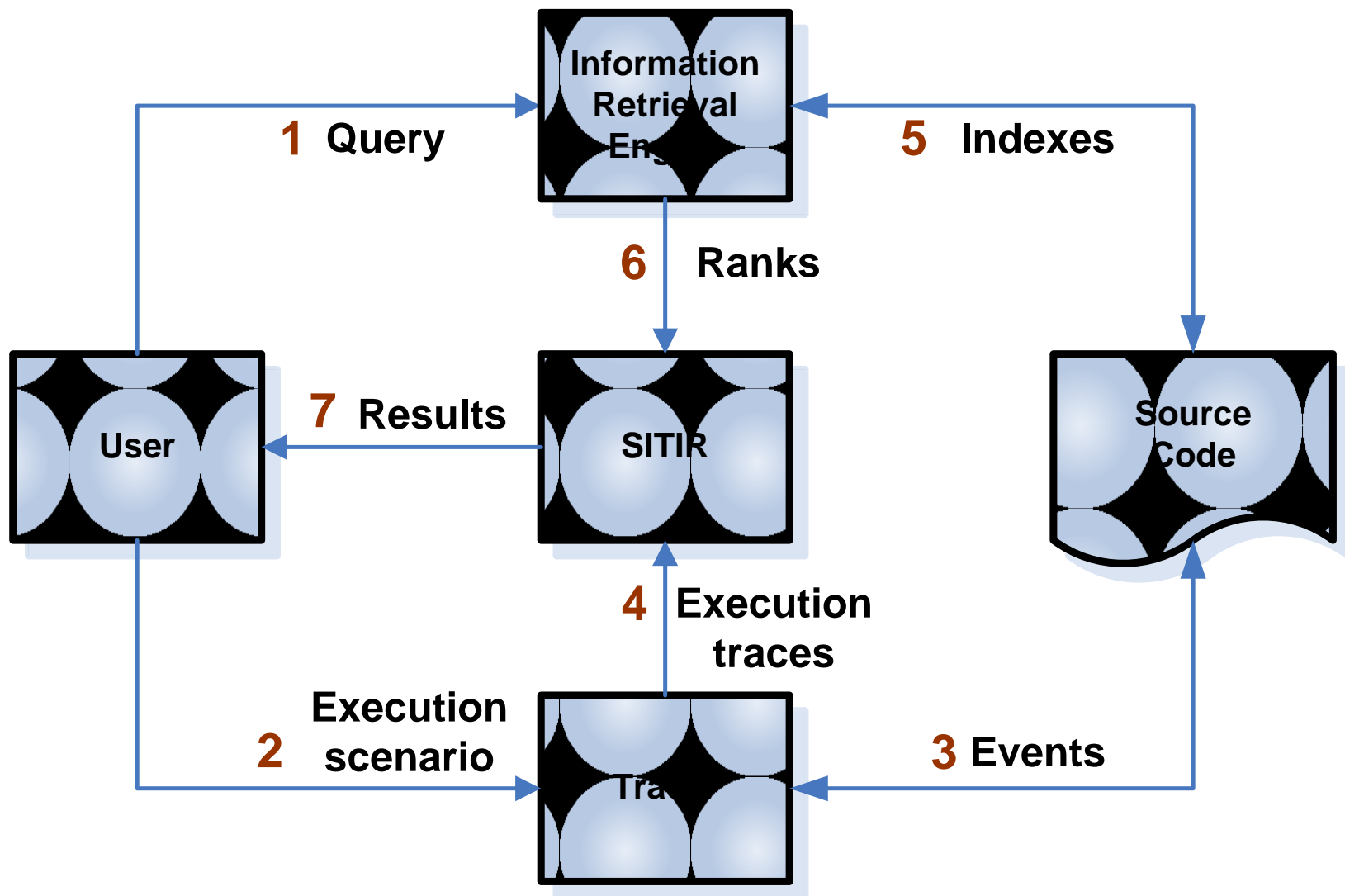
BTNode::BST insert(KeyType key, ValueType value)
// lookup key, create node/value if not present
// return pointer to node with key
{
    BTNode* n = getNode(key);
    n->time.second = value;
    return n;
}

BTNode::BST getNode(KeyType key)
// lookup key, create key/value if not present
// return pointer to node with key
{
    pair(KeyType, ValueType) s(key, 0); // create item to be inserted
    if (!root)
    {
        // apply tree insert so root
        // exists = new node.
        return *root;
    }
    pair(BTNode*, bool) result = find(key); // search for key
    if (!result.second) // insertion failed
    {
        return *(result.first); // true, node exists
    }
    else // (also, create node
    {
        BTNode parent = result.first;
        BTNode* new_node = new BTNode(0, 0, parent, s);
        if (key < parent->key)
            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

Feature Location with SITIR/FLAT³



Collecting Execution Traces in SITIR

- 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 Software with Apache Lucene

- Parsing source code and extracting documents
 - corpus is a collection of documents (e.g., methods)
- Removing non-literals and stop words
 - common words in English, programming language keywords
- Preprocessing: `split_identifiers` and `SplitIdentifiers`
- Indexing and retrieving semantic information with Lucene

Parsing Source Code and Extracting Documents

- Corpus is a collection of documents (e.g., methods, classes, files)

```
public void run(IProgressMonitor monitor)
    throws InvocationTargetException,
           InterruptedException(
    if ( m_iFlag == 0 )
        processCorpus (monitor, checkUpdate ());
    else if ( m_iFlag == 2 )
        processCorpus (monitor, UD_UPDATECORPUS);
    else
        processQueryString (monitor);

    if (monitor.isCanceled())
        throw new InterruptedException("The long running
)

public void run(IProgressMonitor monitor)
    throws InvocationTargetException,
           InterruptedException(
    if ( m_iFlag == 0 )
        processCorpus (monitor, checkUpdate ());
    else if ( m_iFlag == 2 )
        processCorpus (monitor, UD_UPDATECORPUS);
    else
        processQueryString (monitor);

    if (monitor.isCanceled())
        throw new InterruptedException("The long running
)

public void run(IProgressMonitor monitor)
    throws InvocationTargetException,
           InterruptedException(
    if ( m_iFlag == 0 )
        processCorpus (monitor, checkUpdate ());
    else if ( m_iFlag == 2 )
        processCorpus (monitor, UD_UPDATECORPUS);
    else
        processQueryString (monitor);

    if (monitor.isCanceled())
        throw new InterruptedException("The long running
)
```

```
public void run(IProgressMonitor monitor)
    throws InvocationTargetException,
           InterruptedException(
    if ( m_iFlag == 0 )
        processCorpus (monitor, checkUpdate ());
    else if ( m_iFlag == 2 )
        processCorpus (monitor, UD_UPDATECORPUS);
    else
        processQueryString (monitor);

    if (monitor.isCanceled())
        throw new InterruptedException("The long running
)

public void run(IProgressMonitor monitor)
    throws InvocationTargetException,
           InterruptedException(
    if ( m_iFlag == 0 )
        processCorpus (monitor, checkUpdate ());
    else if ( m_iFlag == 2 )
        processCorpus (monitor, UD_UPDATECORPUS);
    else
        processQueryString (monitor);

    if (monitor.isCanceled())
        throw new InterruptedException("The long running
)

public void run(IProgressMonitor monitor)
    throws InvocationTargetException,
           InterruptedException(
    if ( m_iFlag == 0 )
        processCorpus (monitor, checkUpdate ());
    else if ( m_iFlag == 2 )
        processCorpus (monitor, UD_UPDATECORPUS);
    else
        processQueryString (monitor);

    if (monitor.isCanceled())
        throw new InterruptedException("The long running
)
```

```
public void run(IProgressMonitor monitor)
    throws InvocationTargetException,
           InterruptedException(
    if ( m_iFlag == 0 )
        processCorpus (monitor, checkUpdate ());
    else if ( m_iFlag == 2 )
        processCorpus (monitor, UD_UPDATECORPUS);
    else
        processQueryString (monitor);

    if (monitor.isCanceled())
        throw new InterruptedException("The long running
)

public void run(IProgressMonitor monitor)
    throws InvocationTargetException,
           InterruptedException(
    if ( m_iFlag == 0 )
        processCorpus (monitor, checkUpdate ());
    else if ( m_iFlag == 2 )
        processCorpus (monitor, UD_UPDATECORPUS);
    else
        processQueryString (monitor);

    if (monitor.isCanceled())
        throw new InterruptedException("The long running
)

public void run(IProgressMonitor monitor)
    throws InvocationTargetException,
           InterruptedException(
    if ( m_iFlag == 0 )
        processCorpus (monitor, checkUpdate ());
    else if ( m_iFlag == 2 )
        processCorpus (monitor, UD_UPDATECORPUS);
    else
        processQueryString (monitor);

    if (monitor.isCanceled())
        throw new InterruptedException("The long running
)
```

Parsing Source Code and Extracting Documents

- Corpus is a collection of documents (e.g., methods, classes, files)

```
public void run(IProgressMonitor monitor)
    throws InvocationTargetException,
           InterruptedException{
    if ( m_iFlag == 0 )
        processCorpus (monitor, checkUpdate ());
    else if ( m_iFlag == 2 )
        processCorpus (monitor, UD_UPDATECORPUS);
    else
        processQueryString (monitor);

    if (monitor.isCanceled())
        throw new InterruptedException("The long running
```

```
public void run(IProgressMonitor monitor)
    throws InvocationTargetException,
           InterruptedException{
    if ( m_iFlag == 0 )
        processCorpus (monitor, checkUpdate ());
    else if ( m_iFlag == 2 )
        processCorpus (monitor, UD_UPDATECORPUS);
    else
        processQueryString (monitor);

    if (monitor.isCanceled())
        throw new InterruptedException("The long running
```

```
public void run(IProgressMonitor monitor)
    throws InvocationTargetException,
           InterruptedException{
    if ( m_iFlag == 0 )
        processCorpus (monitor, checkUpdate ());
    else if ( m_iFlag == 2 )
        processCorpus (monitor, UD_UPDATECORPUS);
    else
        processQueryString (monitor);

    if (monitor.isCanceled())
        throw new InterruptedException("The long running
```

```
public void run(IProgressMonitor monitor)
    throws InvocationTargetException,
           InterruptedException{
    if ( m_iFlag == 0 )
        processCorpus (monitor, checkUpdate ());
    else if ( m_iFlag == 2 )
        processCorpus (monitor, UD_UPDATECORPUS);
    else
        processQueryString (monitor);

    if (monitor.isCanceled())
        throw new InterruptedException("The long running
```

```
public void run(IProgressMonitor monitor)
    throws InvocationTargetException,
           InterruptedException{
    if ( m_iFlag == 0 )
        processCorpus (monitor, checkUpdate ());
    else if ( m_iFlag == 2 )
        processCorpus (monitor, UD_UPDATECORPUS);
    else
        processQueryString (monitor);

    if (monitor.isCanceled())
        throw new InterruptedException("The long running
```

```
public void run(IProgressMonitor monitor)
    throws InvocationTargetException,
           InterruptedException{
    if ( m_iFlag == 0 )
        processCorpus (monitor, checkUpdate ());
    else if ( m_iFlag == 2 )
        processCorpus (monitor, UD_UPDATECORPUS);
    else
        processQueryString (monitor);

    if (monitor.isCanceled())
```

```
public void run(IProgressMonitor monitor)
    throws InvocationTargetException,
           InterruptedException{
    if ( m_iFlag == 0 )
        processCorpus (monitor, checkUpdate ());
    else if ( m_iFlag == 2 )
        processCorpus (monitor, UD_UPDATECORPUS);
    else
        processQueryString (monitor);

    if (monitor.isCanceled())
        throw new InterruptedException("The long running
```

```
public void run(IProgressMonitor monitor)
    throws InvocationTargetException,
           InterruptedException{
    if ( m_iFlag == 0 )
        processCorpus (monitor, checkUpdate ());
    else if ( m_iFlag == 2 )
        processCorpus (monitor, UD_UPDATECORPUS);
    else
        processQueryString (monitor);

    if (monitor.isCanceled())
        throw new InterruptedException("The long running
```

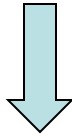
```
public void run(IProgressMonitor monitor)
    throws InvocationTargetException,
           InterruptedException{
    if ( m_iFlag == 0 )
        processCorpus (monitor, checkUpdate ());
    else if ( m_iFlag == 2 )
        processCorpus (monitor, UD_UPDATECORPUS);
    else
        processQueryString (monitor);

    if (monitor.isCanceled())
        throw new InterruptedException("The long running
```

Source Code is Text Too

```
public void run(IProgressMonitor monitor)
    throws InvocationTargetException,
        InterruptedException{
    if ( m_iFlag == 0 )
        processCorpus (monitor, checkUpdate ());
    else if ( m_iFlag == 2 )
        processCorpus (monitor, UD_UPDATECORPUS);
    else
        processQueryString (monitor);

    if (monitor.isCanceled())
        throw new InterruptedException("The long running
}
}
```



public void run IProgressMonitor monitor throws
InvocationTargetException InterruptedException if m_iFlag
processCorpus monitor checkUpdate else if m_iFlag
processCorpus monitor UD_UPDATECORPUS else
processQueryString monitor if monitor isCancelled throw
new InterruptedException the long running

Removing Stop Words

- Common words in English, programming language keywords

```
public void run IProgressMonitor monitor throws
InvocationTargetException InterruptedException if
m_iFlag the processCorpus monitor checkUpdate else
if m_iFlag processCorpus monitor UD_UPDATECORPUS
else a processQueryString monitor if monitor
isCancelled throw new InterruptedException the
long running
```


Splitting Identifiers

```
public void run IProgressMonitor monitor throws InvocationTargetException  
InterruptedException if m_iFlag the processCorpus monitor checkUpdate else if  
m_iFlag processCorpus monitor UD_UPDATECORPUS else a processQueryString  
monitor if monitor isCancelled throw new InterruptedException the long  
running
```



- IProgressMonitor = i progress monitor
- InvocationTargetException = invocation target exception
- m_iFlag = m i flag
- UD_UPDATECORPUS = ud updatecorpus

Indexing Source Code with Lucene

```
public void run(IProgressMonitor monitor)
    throws InvocationTargetException,
           InterruptedException{
    if ( m_iFlag == 0 )
        processCorpus(monitor, checkUpdate());
    else if ( m_iFlag == 2 )
        processCorpus(monitor, UD_UPDATECORPUS);
    else
        processQueryString(monitor);

    if (monitor.isCanceled())
        throw new InterruptedException("The long running
}
}
```



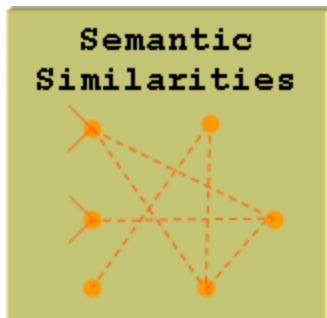
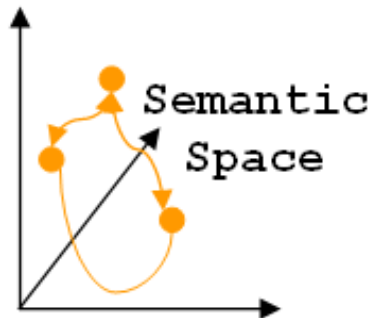
	process	flag	monitor
run	3	2	6
method1	x	x	x
method2	x	x	x
...	x	x	x



TF-IDF weighting

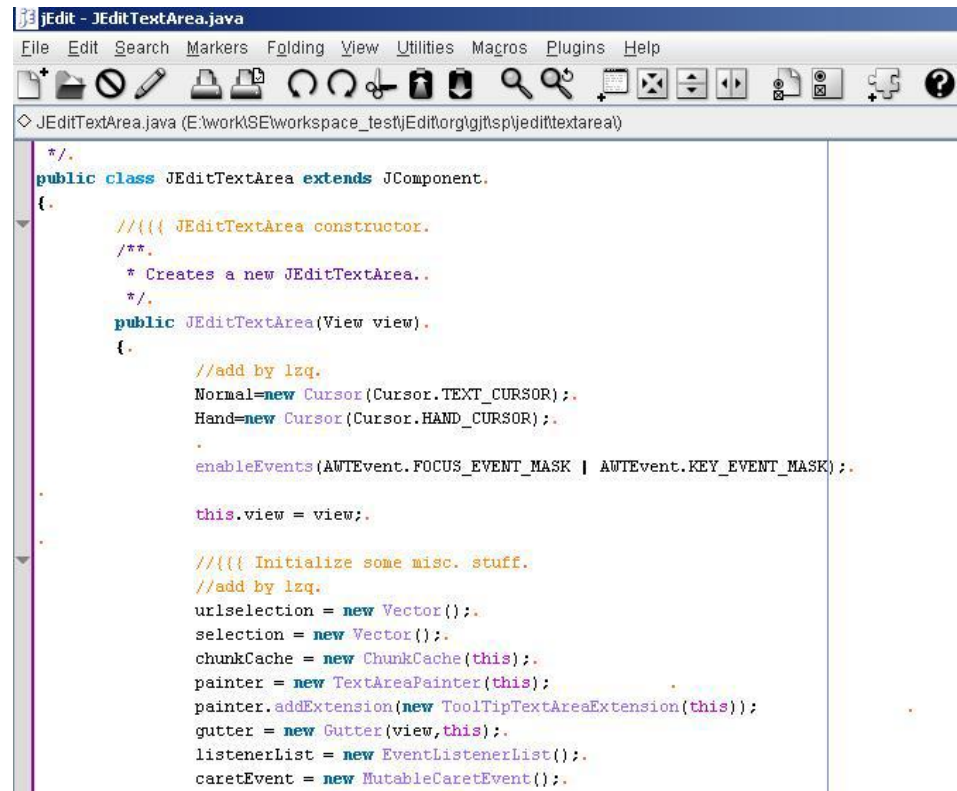


Similarity Measure:
Cosine of the contained angle
between the vectors



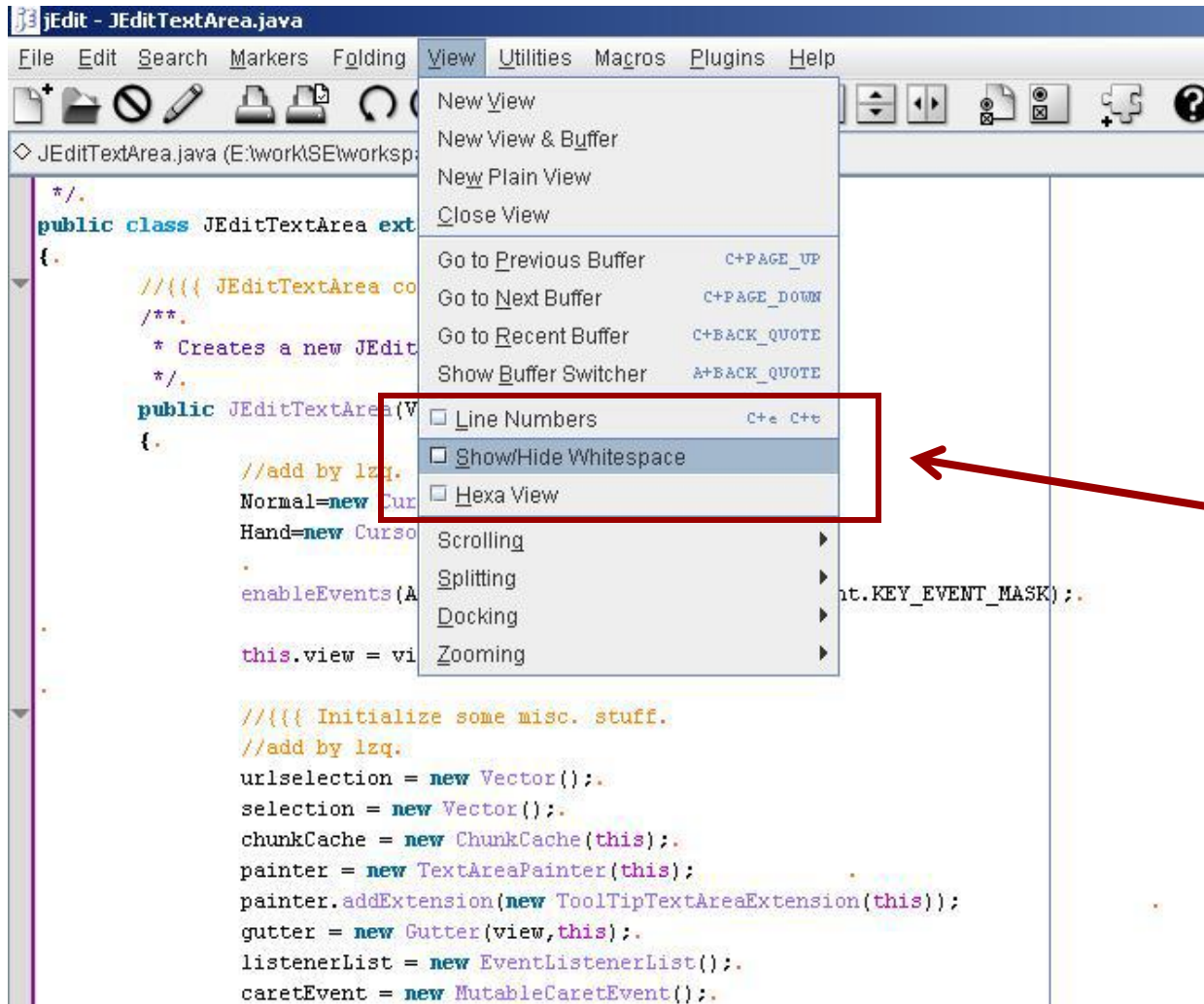
Example of using SITIR

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



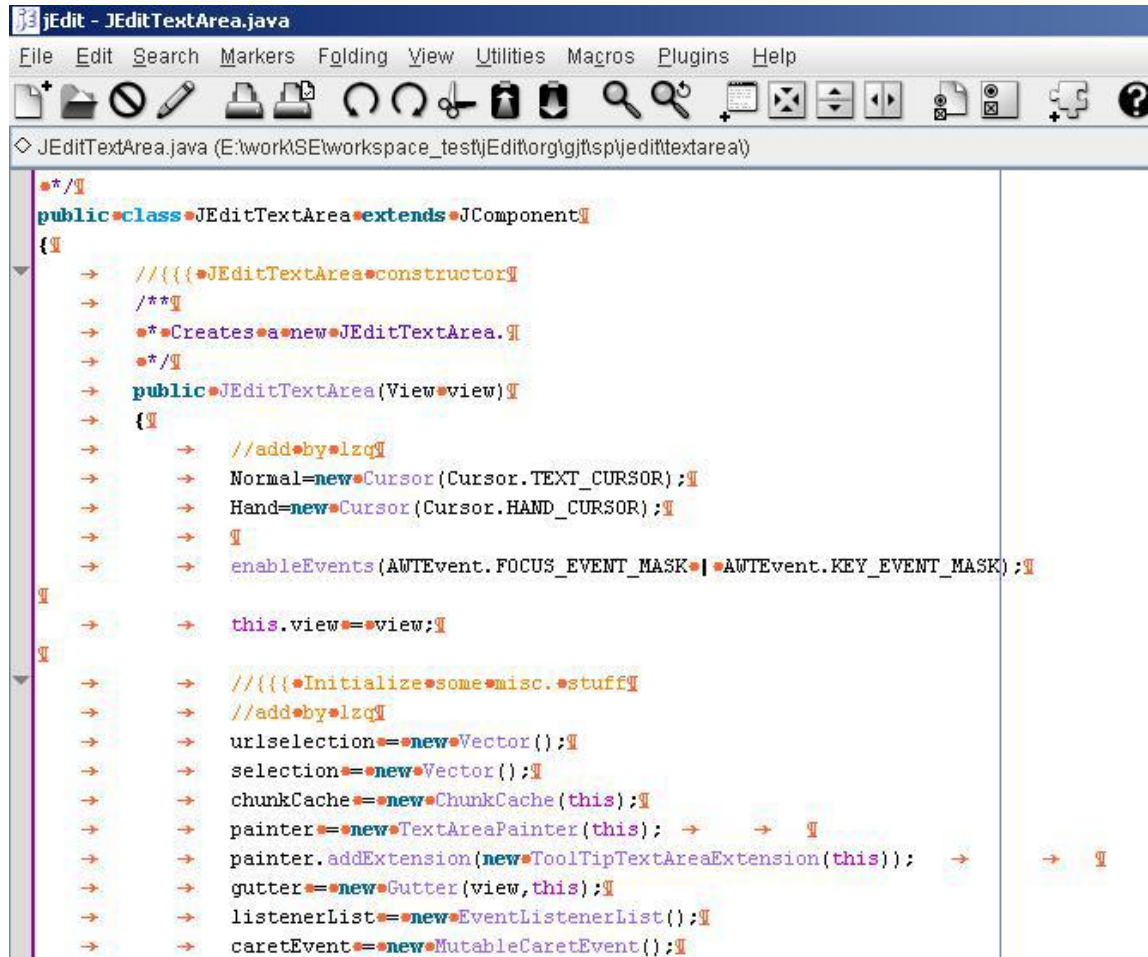
```
File Edit Search Markers Folding View Utilities Macros Plugins Help
JEdit - JEditTextArea.java
E:\work\SE\workspace_test\JEdit\org\gjtsp\jedit\textarea\
public class JEditTextArea extends JComponent.
{
    /**
     * {{{ JEditTextArea constructor.
     * Creates a new JEditTextArea..
     */
    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();.
    }
}
```

Scenario Exercising the Feature



**Start
Tracing**

Scenario Exercising the Feature



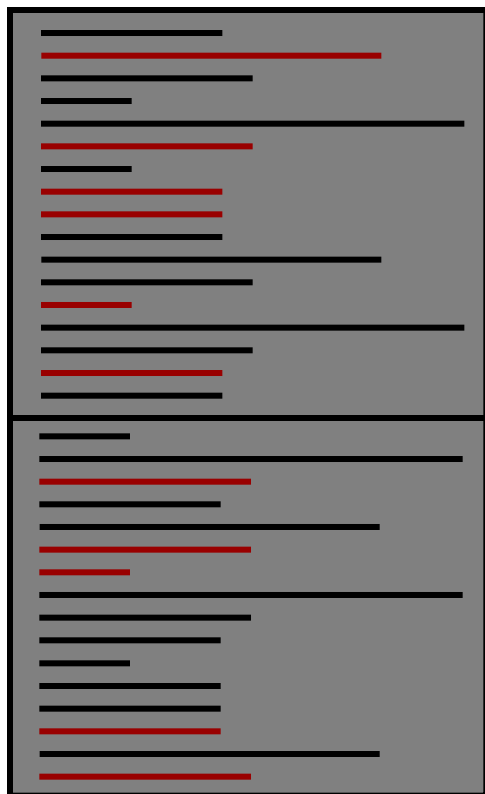
```
public class JEditTextArea extends JComponent
{
    /**
     * Creates a new JEditTextArea.
     */
    public JEditTextArea(View view)
    {
        //add by lzq
        Normal = new Cursor(Cursor.TEXT_CURSOR);
        Hand = new Cursor(Cursor.HAND_CURSOR);
        enableEvents(AWTEvent.FOCUS_EVENT_MASK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

        this.view = view;

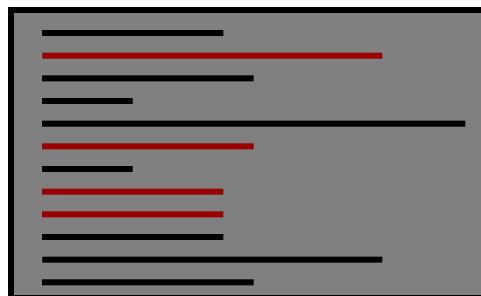
        //Initialize some misc. stuff
        //add by lzq
        urlselection = new Vector();
        selection = new Vector();
        chunkCache = new ChunkCache(this);
        painter = new TextAreaPainter(this);
        painter.addExtension(new ToolTipTextAreaExtension(this));
        gutter = new Gutter(view,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

**Stop
Tracing**

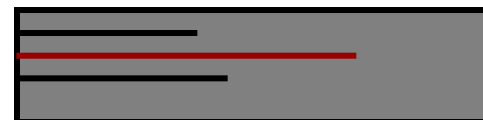
Example of using SITIR/FLAT³ - Results



Executed methods



**IR-based
rankings**



SITIR

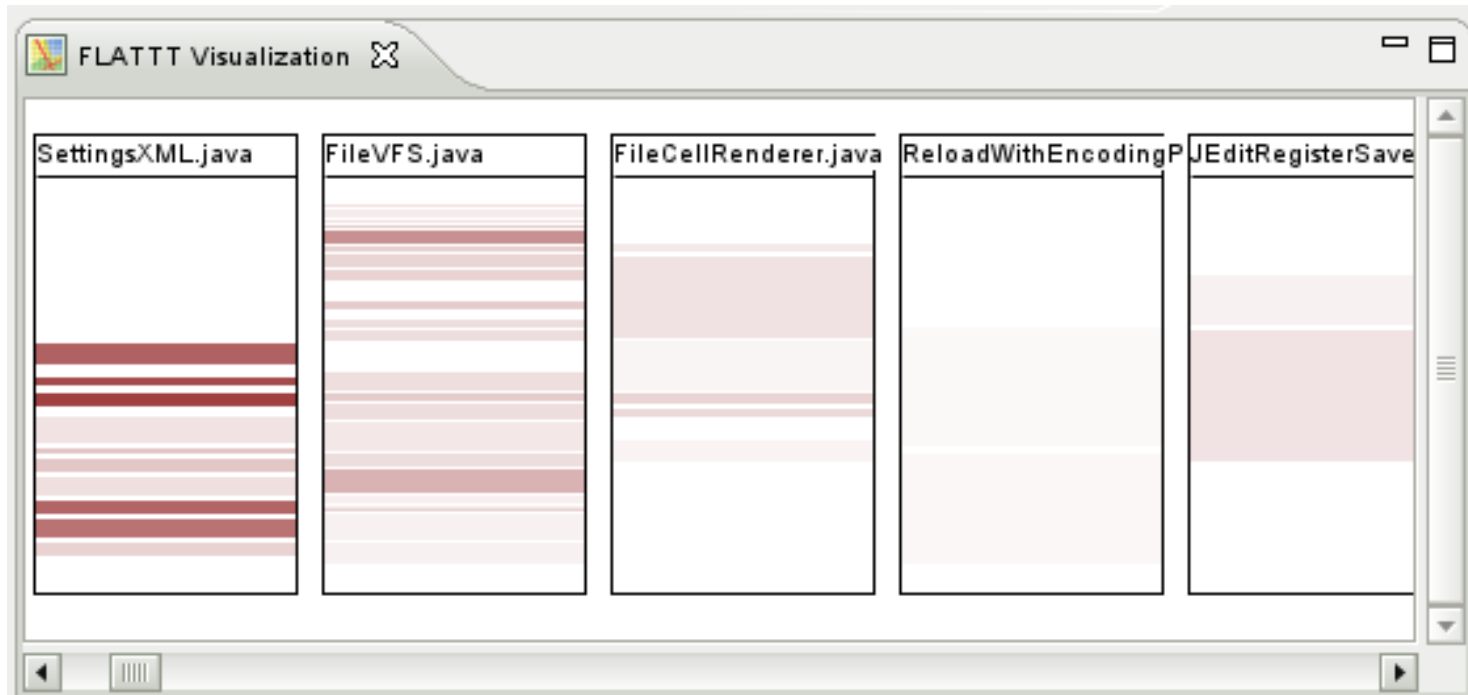
- Number of methods identified in the trace - **284**
- The position of the first relevant method according to IR ranking - **56**
- Position of the first relevant method according to SITIR - **7**

DEMO: Locating Features in JEdit using FLAT³

- JEdit
 - 105KLOC
 - 910 classes
 - 5,530 methods
- Feature: “Word Count”
- Feature: “Save as”

Other features

- Annotating features
- Visualizing results
- Saving/loading complex traces



Acknowledgements

- ConcernMapper
- ConcernTagger
- MuTT
- AspectBrowser

FLAT³

- Download the tool and complete source code at SEMERU web-site:
- <http://www.cs.wm.edu/semeru/flat3>



SEMERU