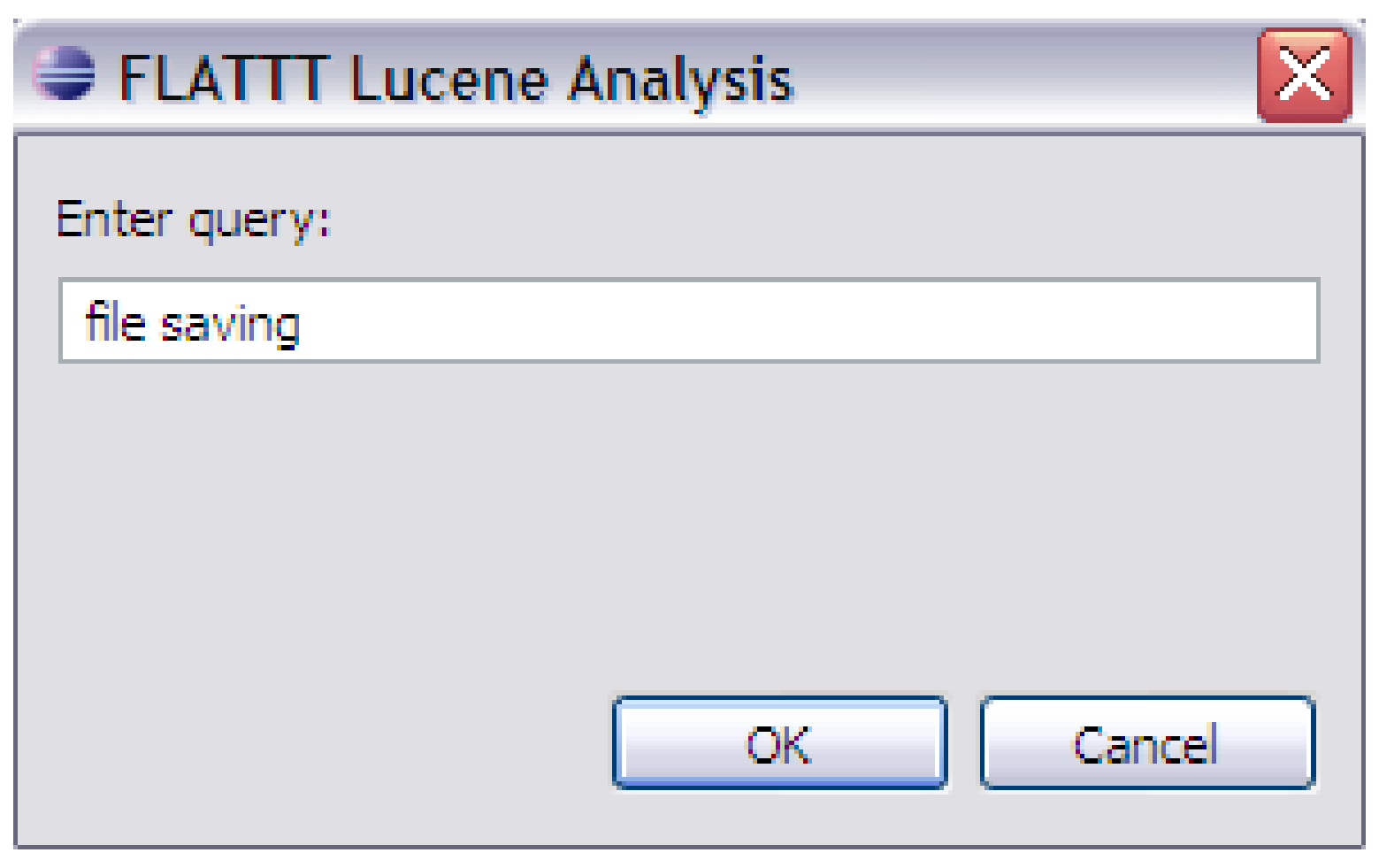
FLAT^3: Feature Location and Textual Tracing Tool

Trevor Savage, Meghan Revelle, and Denys Poshyvanyk
Department of Computer Science
The College of William and Mary
Williamsburg, Virginia

Abstract

Feature location is the process of finding the source code that implements a functional requirement of a software system. It plays an important role in software maintenance activities, but when it is performed manually, it can be challenging and time-consuming, especially for large, long-lived systems. This poster describes a tool called FLAT^3 that integrates textual and dynamic feature location techniques along with feature annotation capabilities and a useful visualization technique, providing a complete suite of tools that allows developers to quickly and easily locate the code that implements a feature and then save these annotations for future use.

Textual Feature Location

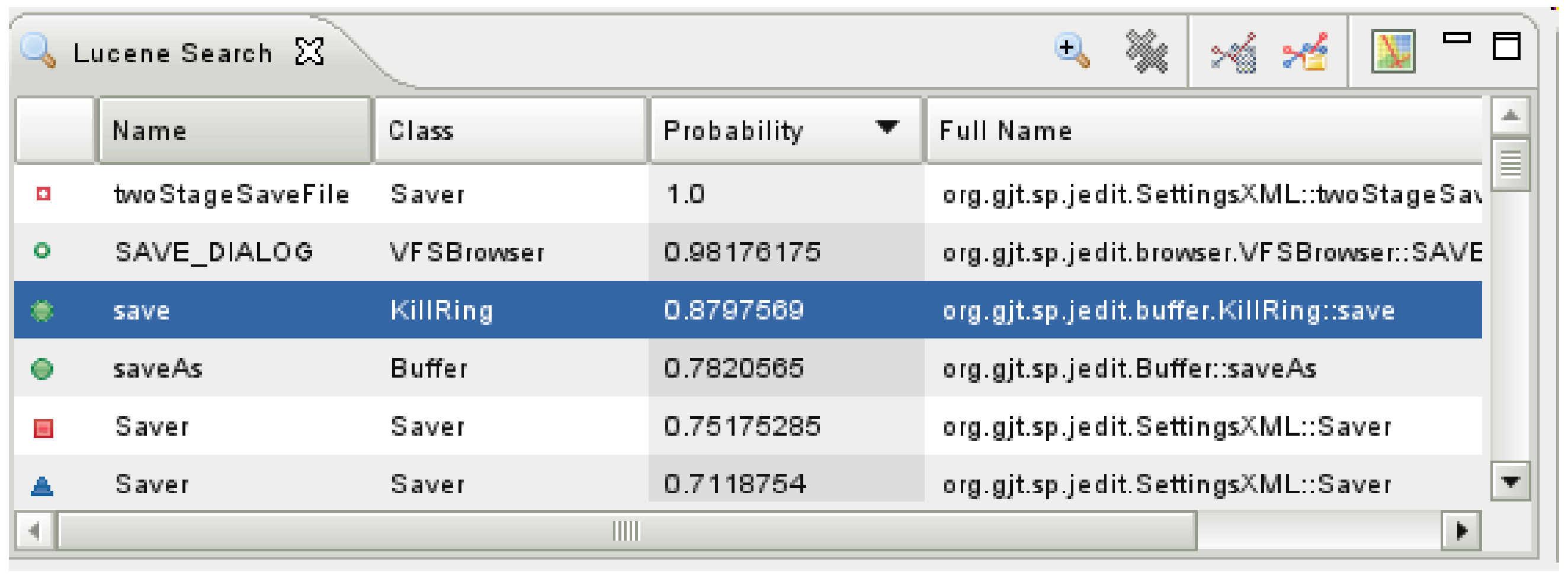

FLATTT Lucene Analysis

Enter query: file saving

OK | Cancel

The user enters a query describing the feature to be located.

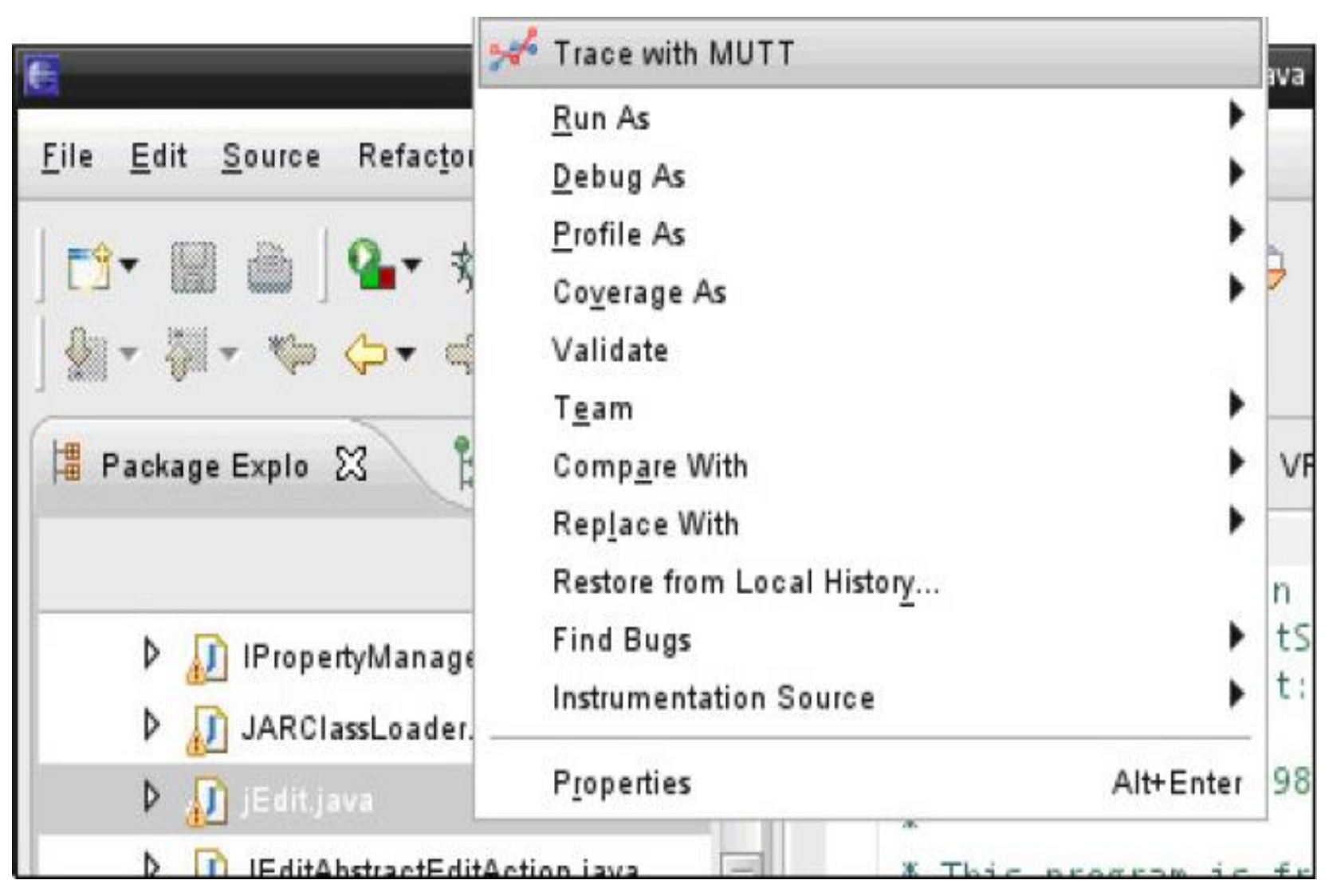
FLAT^3 indexes the source code in Eclipse's workspace and compares the query to all classes, methods, and fields, computing a similarity score for each.


Lucene Search

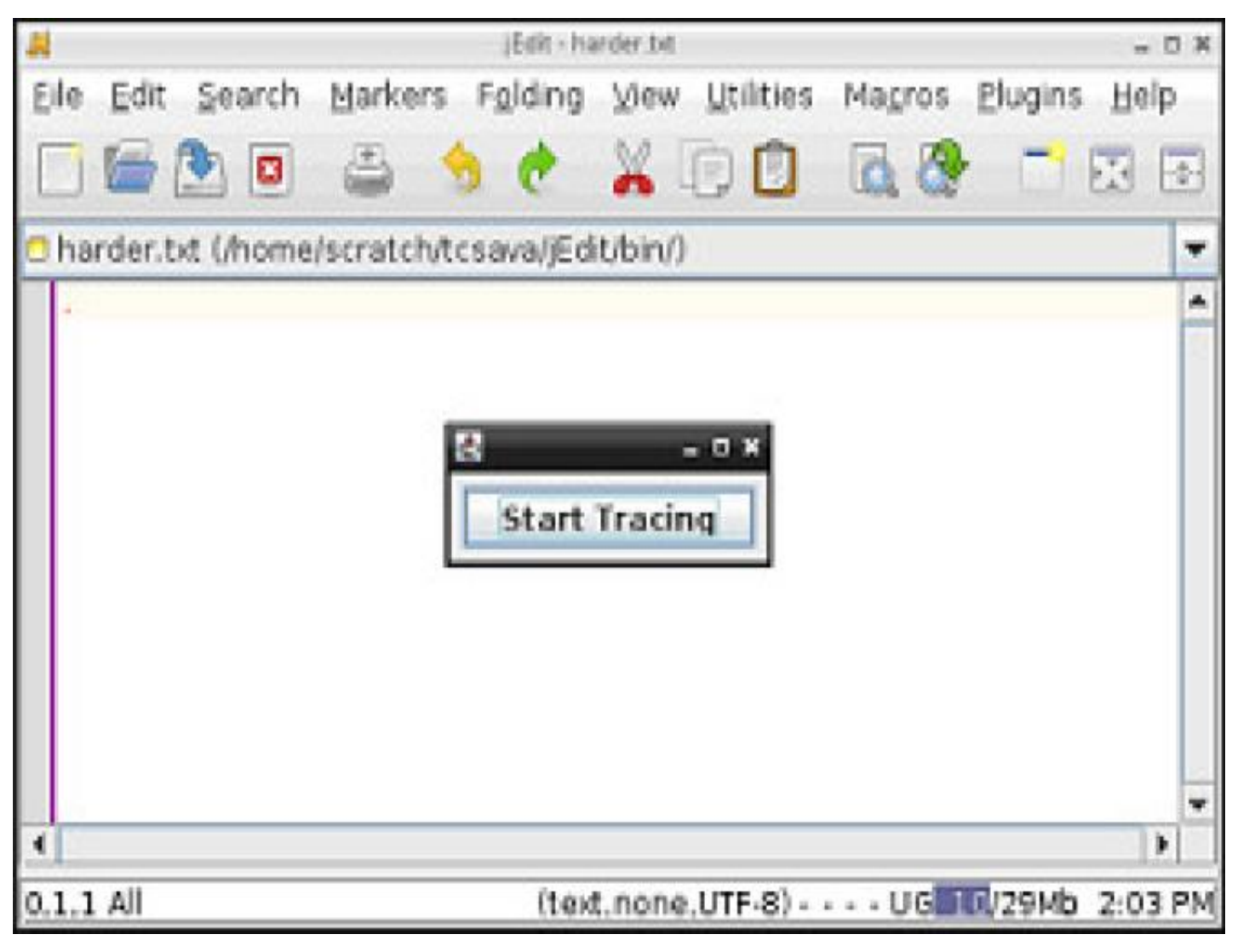
Name	Class	Probability	Full Name
twoStageSaveFile	Saver	1.0	org.gjt.sp.jedit.SettingsXML::twoStageSav
SAVE_DIALOG	VFSBrowser	0.98176175	org.gjt.sp.jedit.browser.VFSBrowser::SAVE
save	KillRing	0.8797569	org.gjt.sp.jedit.buffer.KillRing::save
saveAs	Buffer	0.7820565	org.gjt.sp.jedit.Buffer::saveAs
Saver	Saver	0.75175285	org.gjt.sp.jedit.SettingsXML::Saver
Saver	Saver	0.7118754	org.gjt.sp.jedit.SettingsXML::Saver

FLAT^3 returns a list of program elements ranked by their similarity to the query.

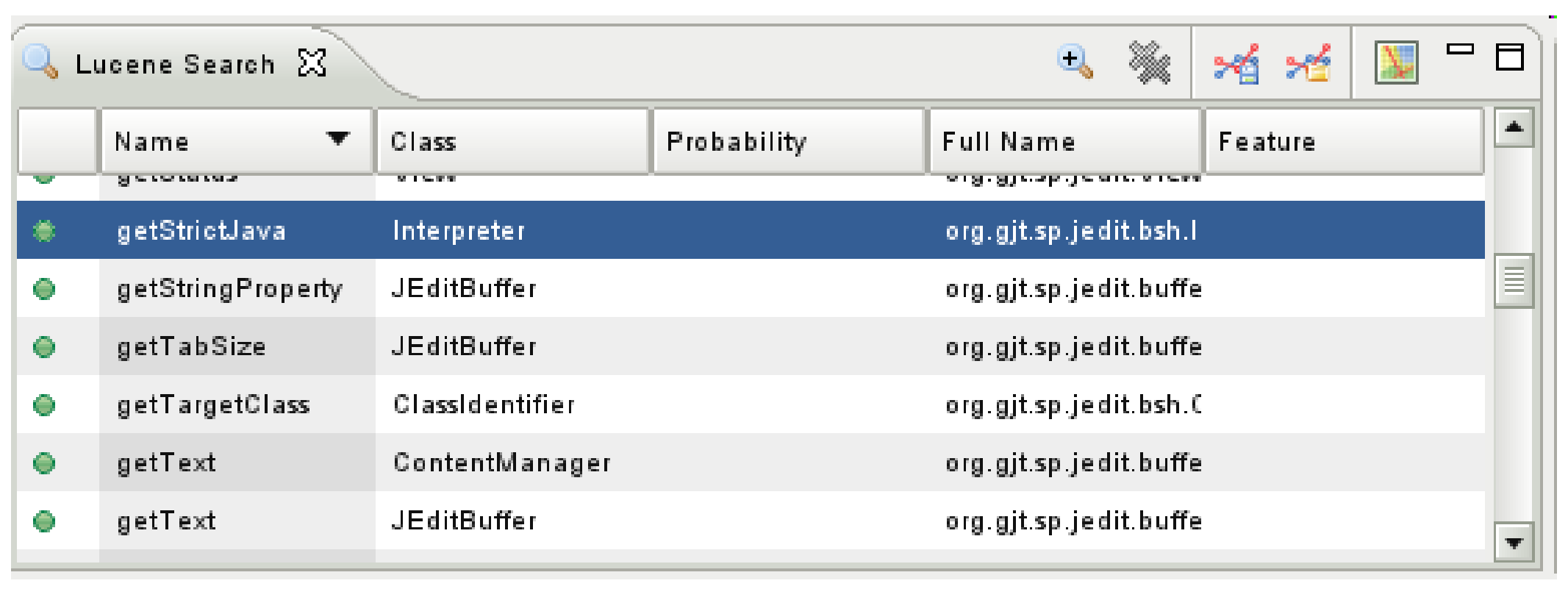
Dynamic Feature Location



The user starts MUTT, the tracing tool.


File Edit Search Markers Folding View Utilities Macros Plugins Help
harder.txt (/home/scratch/tcsava/jEdit/bin/)
Start Tracing

The user can turn tracing on and off while the application is executing.


Lucene Search

Name	Class	Probability	Full Name	Feature
getStrictJava	Interpreter		org.gjt.sp.jedit.bsh.I	
getStringProperty	JEditBuffer		org.gjt.sp.jedit.buffe	
getTabSize	JEditBuffer		org.gjt.sp.jedit.buffe	
getTargetClass	ClassIdentifier		org.gjt.sp.jedit.bsh.C	
getText	ContentManager		org.gjt.sp.jedit.buffe	
getText	JEditBuffer		org.gjt.sp.jedit.buffe	

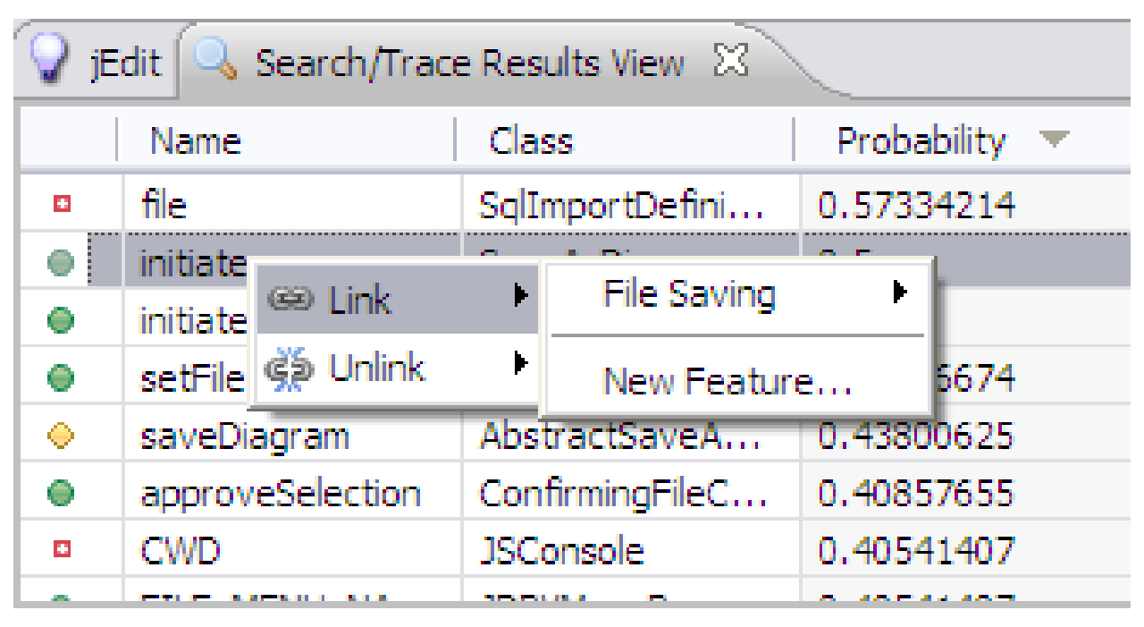
FLAT^3 returns a list of methods that were executed.

Integrated Feature Location

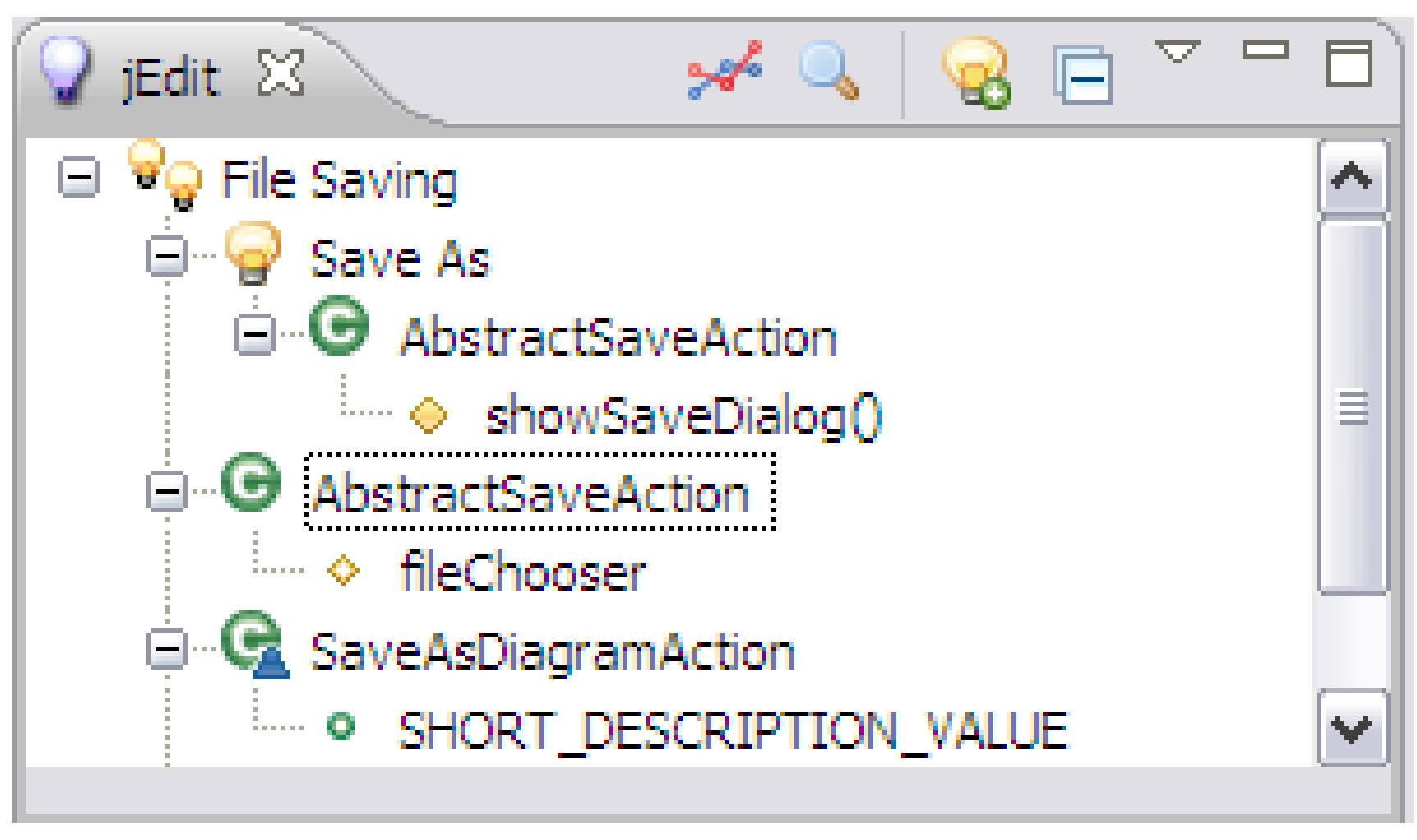
FLAT^3 supports the integration of textual and dynamic feature location following the SITIR approach, which stands for **SI**ngle Execution **T**race & **I**nformation **R**etrieval [1].

After collecting an execution trace, users may be faced with a long list of executed methods. They can refine their search by issuing a query that searches only the methods that were invoked, instead of the entire system. All the executed methods are ranked by their similarity to the query.

Annotations


jEdit | Search/Trace Results View

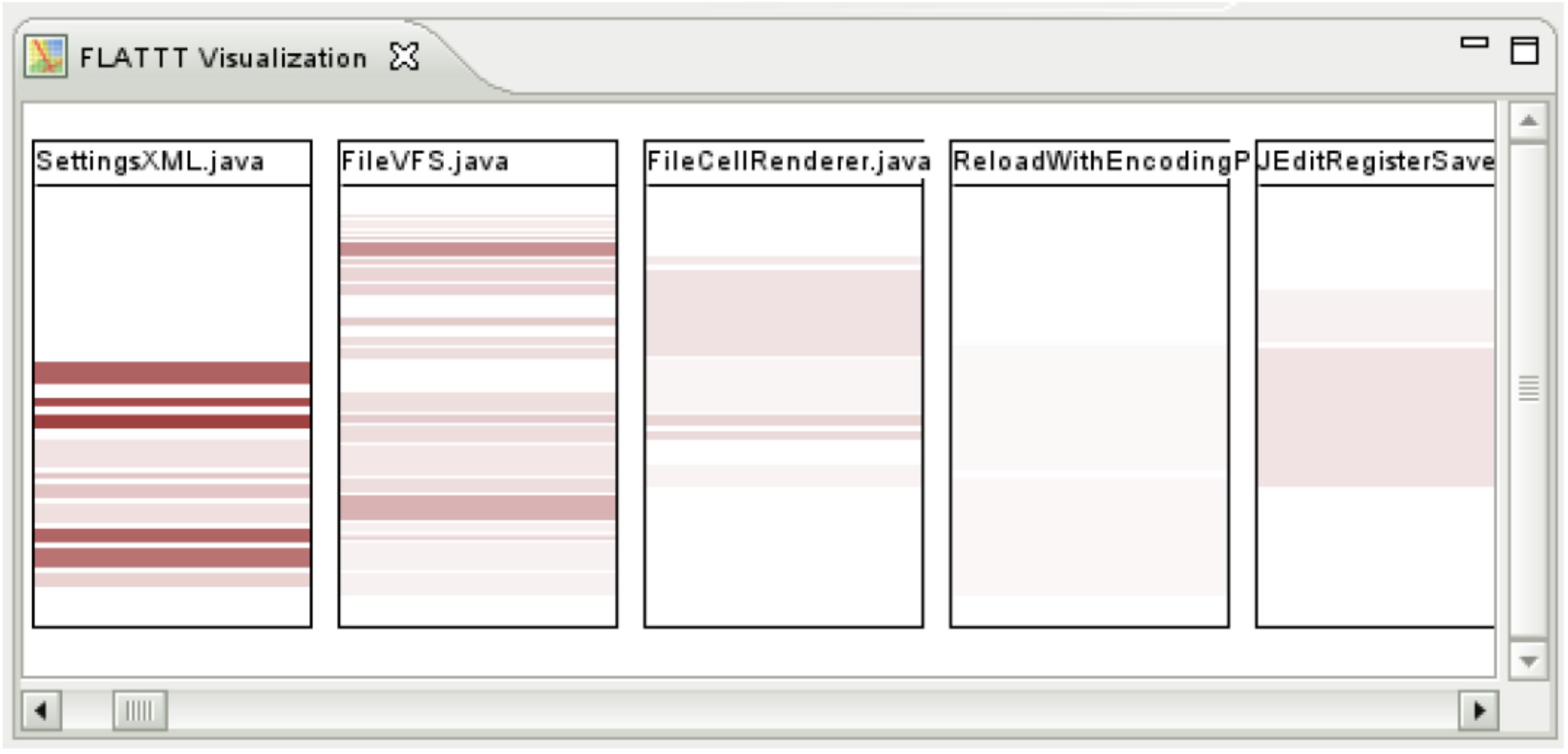
Name	Class	Probability
file	SqlImportDefini...	0.57334214
initiate		
initiate		
setFile		5674
saveDiagram	AbstractSaveA...	0.43800625
approveSelection	ConfirmingFileC...	0.40857655
CWD	JSConsole	0.40541407


jEdit

- File Saving
  - Save As
    - AbstractSaveAction
      - showSaveDialog()
  - AbstractSaveAction
    - fileChooser
  - SaveAsDiagramAction
    - SHORT_DESCRIPTION_VALUE

FLAT^3 allows the user to create and name features and then link source code to those features. These taggings or annotations are displayed in a special view. Features can be hierarchical and have sub-features. Above, *Save As* is a child of the *File Saving* feature.

Visualization



FLAT^3 can visualize the dispersion of a feature or search results throughout the classes of a system. The color of the highlighted rows indicates the degree of similarity of the code to the feature.

References

[1] Liu, D., Marcus, A., Poshyvanyk, D., and Rajlich, V.,"Feature Location via Information Retrieval based Filtering of a Single Scenario Execution Trace", in Proceedings of Int'l Conf. on Automated Software Engineering (ASE'07).
[2] http://www.cs.mcgill.ca/~martin/cm
[3] http://www.cs.columbia.edu/~eaddy/concerntagger
[4] http://lucene.apache.org/java/docs/index.html
[5] http://sourceforge.net/projects/muttracer
[6] http://cseweb.ucsd.edu/~wgg/Software/AB

Acknowledgments

FLAT^3 incorporates source code from the open source tools ConcernMapper [2], ConcernTagger [3], Lucene [4], and MUTT [5]. It also inherits visualization ideas from AspectBrowser [6]. Alison Smith and Scott Underwood contributed to an earlier version of the FLAT^3 tool. This work is supported in part by NSF CCF-0916260 and United States AFOSR FA9550-07-1-0030 grants. Any opinions, findings and conclusions expressed herein are the authors' and do not necessarily reflect those of the sponsors.

Further information

For further information, contact Denys Poshyvanyk, **denys@cs.wm.edu**

FLAT^3 can be downloaded from:
www.cs.wm.edu/semeru/flat3

Information on other projects from our research group can be found at:
www.cs.wm.edu/semeru