

Using Data Fusion and Web Mining to Support Feature Location in Software



SEMERU



**WILLIAM
& MARY**

Bug 66914 - [typing] Error Message after undo copy/paste

Status: VERIFIED FIXED

Reported: 2004-06-14 08:16 EDT by Ralf Schmauder

Product: JDT

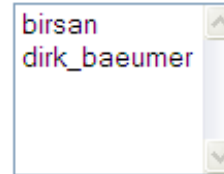
Modified: 2004-06-21 04:55 EDT ([History](#))

Component: Text

CC List: 2 users

Version: 3.0

Platform: PC All



Importance: P2 major ([vote](#))

Target Milestone: 3.0 RC3

Assigned To: Tom Hofmann

QA Contact:

[See Also:](#)

Attachments		
error log (5.43 KB, text/plain) 2004-06-14 08:17 EDT, Ralf Schmauder	<i>no flags</i>	Details
LinkedModeUI.diff (4.59 KB, patch) 2004-06-18 09:59 EDT, Tom Hofmann	<i>no flags</i>	Details Diff
Add an attachment (proposed patch, testcase, etc.)		View All

Ralf Schmauder 2004-06-14 08:16:24 EDT

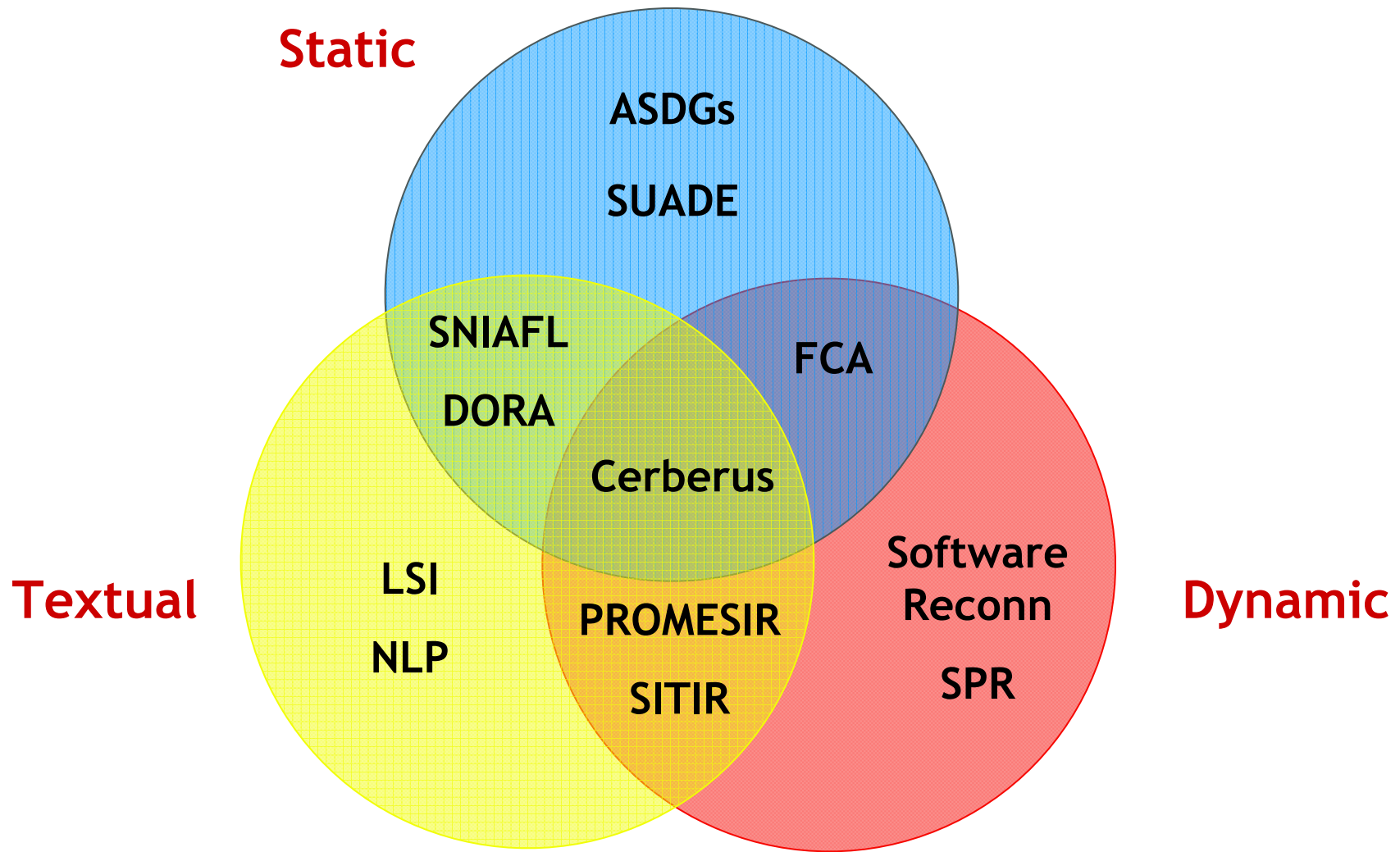
[Description](#)

```
-create a new Class and generate the main method
-type "sysout" and use the code completion
-type the double quote
-paste Hello World into the double quotes
-try to undo without saving using Ctrl+z
```

using undo in the menubar does work

Feature: a requirement that user can invoke and that has an observable behavior.

Existing Feature Location Work



Meghan Revelle and Denys Poshyvanyk. "Feature Location in Source Code: A Taxonomy and Survey." Submission to *Journal of Software Maintenance and Evolution: Research and Practice*.

Textual Feature Location

- **Information Retrieval (IR)**
 - Searching for documents or within docs for relevant information
- **First used for feature location by Marcus et al. in 2004***.
 - Latent Semantic Indexing** (LSI)
- **Utilized by many existing approaches: PROMESIR, SITIR, HIPIKAT etc.**

* Marcus, A., Sergeyev, A., Rajlich, V., and Maletic, J., "An Information Retrieval Approach to Concept Location in Source Code", in Proc. of Working Conference on Reverse Engineering, 2004, pp. 214-223.

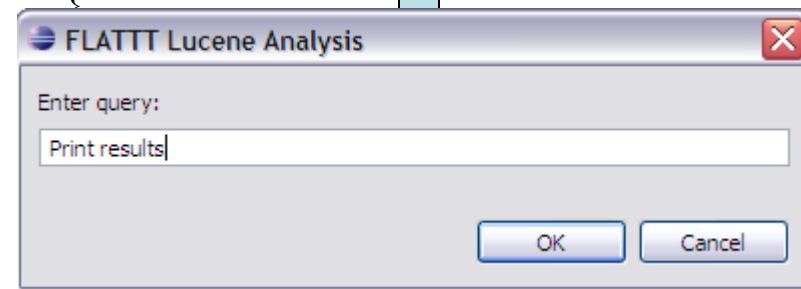
** Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R., "Indexing by Latent Semantic Analysis", *Journal of the American Society for Information Science*, vol. 41, no. 6, Jan. 1990, pp. 391-407.

Applying LSI to Source Code

- **Corpus creation**
 - Choose granularity
- **Preprocessing**
 - Stop word removal, splitting, stemming
- **Indexing**
 - Term-by-document matrix
 - Singular Value Decomposition
- **Querying**
 - User-formulated
- **Generate results**
 - Ranked list

syn		print	test	result	...	Result
resu						
p	m ₁	5	1	3	...	
p						
p	m ₂	

printFooter(result);

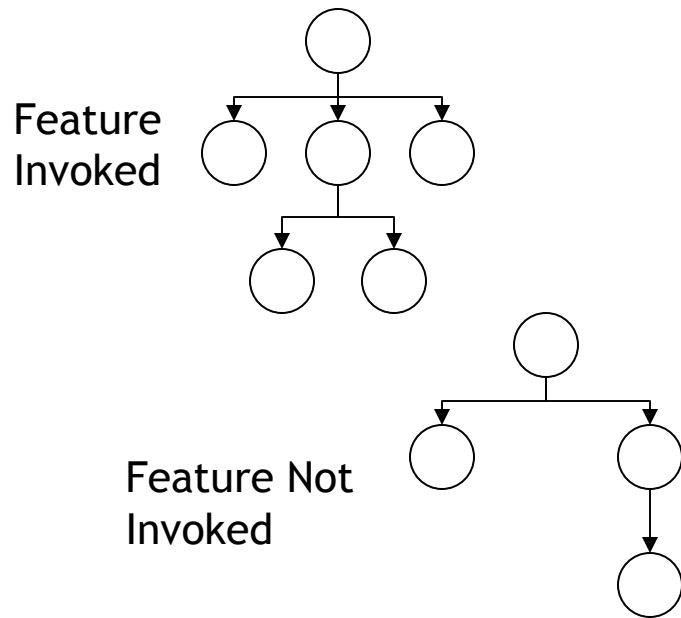


Name	Class	Probability	Full Name
nodeToString	DomProbe	1.0	com.ibatis.common.beans.DomProbe::nodeToString
PRINT_ACTION	JDBV	0.97933716	edu.uiuc.jdbv.JDBV::PRINT_ACTION
PrintPreview	PrintPreview	0.79962546	edu.uiuc.jdbv.util.PrintPreview::PrintPreview
NAME_VALUE	PrintPreviewAct...	0.79962546	edu.uiuc.jdbv.PrintPreviewAction::NAME_VALUE
NAME_VALUE	PrintAction	0.79962546	edu.uiuc.jdbv.PrintAction::NAME_VALUE
out	ConsoleTextArea	0.7915888	org.mozilla.javascript.tools.shell.ConsoleTextArea::...
err	ConsoleTextArea	0.7915888	org.mozilla.javascript.tools.shell.ConsoleTextArea::err

Dynamic Feature Location

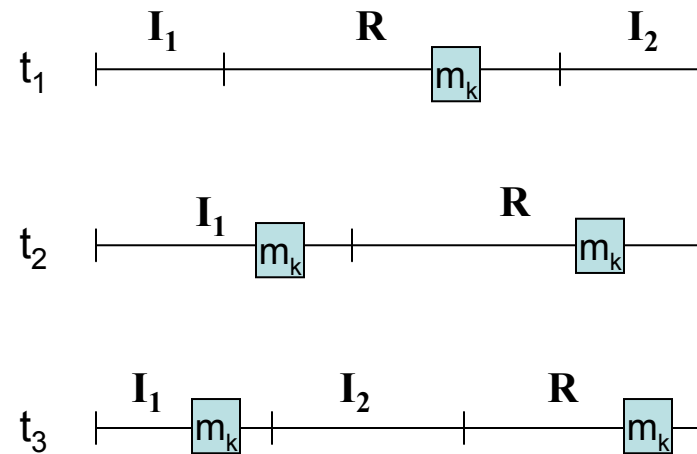
Software

Reconnaissance*



Scenario-based

Probabilistic Ranking (SPR)**



* Wilde, N. and Scully, M., "Software Reconnaissance: Mapping Program Features to Code", *Software Maintenance: Research and Practice*, vol. 7, no. 1, Jan.-Feb. 1995, pp. 49-62.

** Antoniol, G. and Guéhéneuc, Y. G., "Feature Identification: An Epidemiological Metaphor", *IEEE Trans. on Software Engineering*, vol. 32, no. 9, Sept. 2006, pp. 627-641.

Hybrid Feature Location

PROMESIR*

LSI score	SPR score	PROMESIR Score
m ₁₅ 0.91	m ₅₂ 0.80	m₆ 0.715
m ₁₆ 0.88	m ₄₇ 0.66	m ₄₇ 0.70
m ₂ 0.85	m₆ 0.64	m ₅₂ 0.70
m₆ 0.79	m ₂ 0.53	m ₂ 0.69
m ₄₇ 0.74	m ₁₅ 0.37	m ₁₅ 0.64
m ₅₂ 0.60	m ₁₆ 0.34	m ₁₆ 0.61
...

*Probabilistic Ranking of Methods Based on Execution Scenarios and Information Retrieval

Poshyvanyk, D., Guéhéneuc, Y. G., Marcus, A., Antonioli, G., and Rajlich, V., "Feature Location using Probabilistic Ranking of Methods based on Execution Scenarios and Information Retrieval", *IEEE Trans. on Software Engineering*, vol. 33, no. 6, June 2007, pp. 420-432.

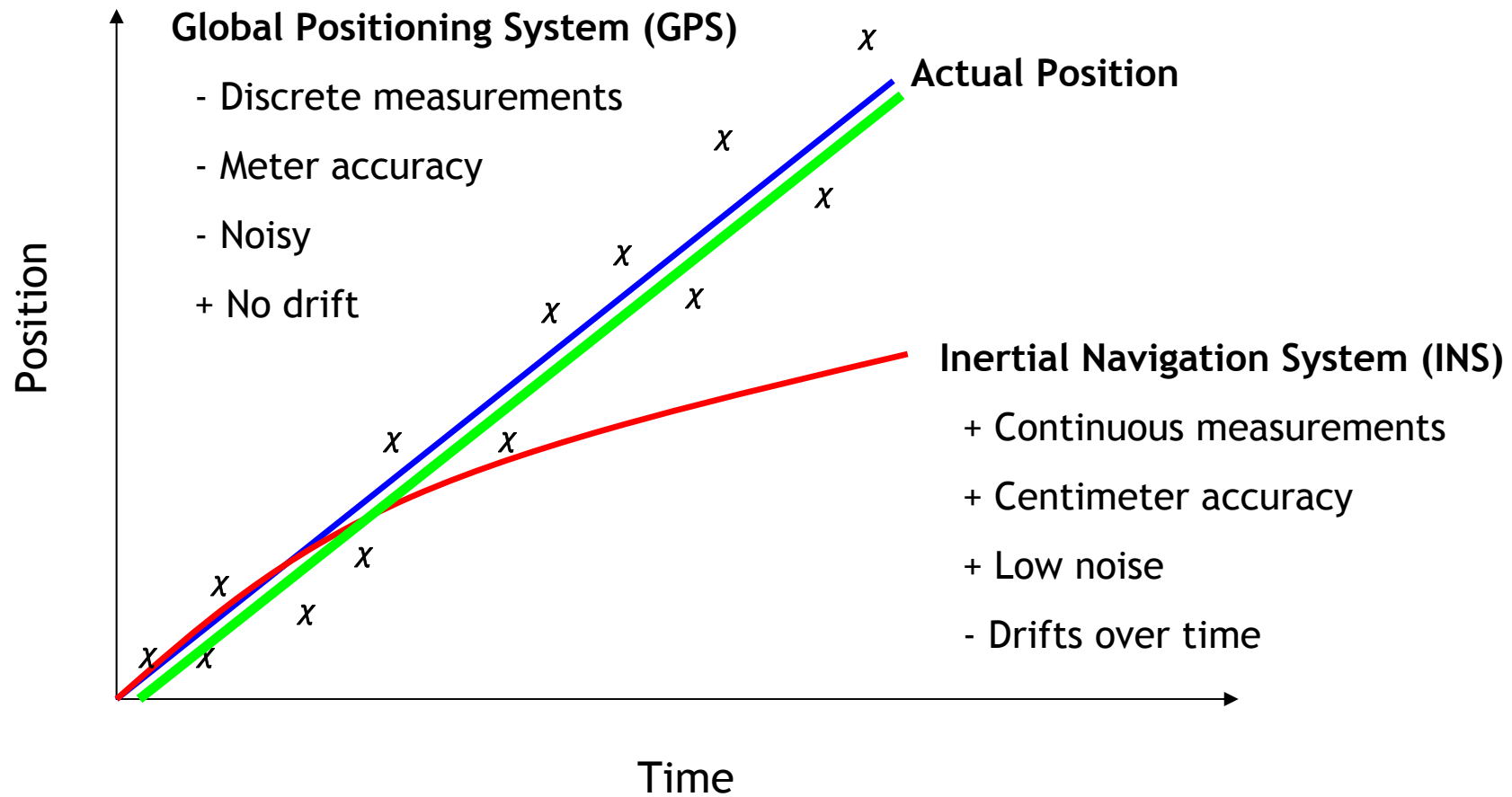
SITIR**

LSI score	Execution Trace
m ₁₅ 0.91	main
m₁₆ 0.88	m ₁
m ₂ 0.85	m ₂
m₆ 0.79	m ₆
m ₄₇ 0.74	m ₁₅
m₅₂ 0.60	m ₃
...	m ₄₇
...	...

**Single Trace and Information Retrieval

Liu, D., Marcus, A., Poshyvanyk, D., and Rajlich, V., "Feature Location via Information Retrieval based Filtering of a Single Scenario Execution Trace", in Proc. of International Conference on Automated Software Engineering, 2007, pp. 234-243.

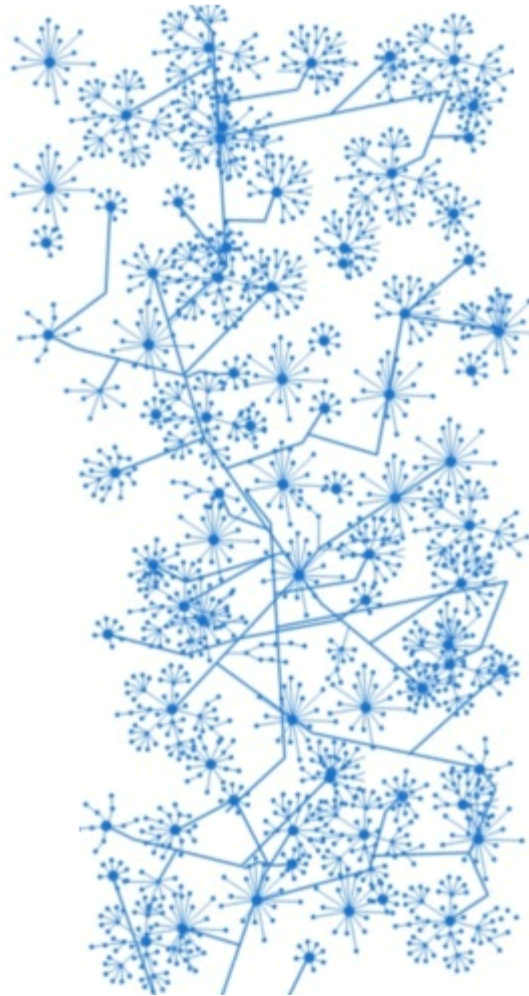
Data Fusion Example



Data Fusion for Feature Location

- Combining information from multiple sources will yield better results than if the data is used separately
 - Previous
 - Textual, Dynamic, and Static (i.e., Cerberus)
 - Current
 - Textual info from IR
 - Execution info from dynamic tracing
 - **Web mining**

Web Mining



mining

Results 1 - 10 of about 19,800,000 for web mining - (0.37 seconds)

[a, the free encyclopedia](#) [a](#) [m₂](#) [m₃](#) [m₄](#) [m₅](#) [m₆](#) [m₇](#) [m₈](#) [m₉](#) [m₁₀](#) [m₁₁](#) [m₁₂](#) [m₁₃](#) [m₁₄](#) [m₁₅](#) [m₁₆](#) [m₁₇](#) [m₁₈](#) [m₁₉](#) [m₂₀](#)

ition of data mining techniques to discover patterns from the targets, **web mining** can be divided into ...
[intent mining](#) - [Web structure mining](#)
[mining](#) - [Cached](#) - [Similar](#) - [↑](#) [×](#)

[ns and techniques - Google Books Result](#)
rputers - 427 pages
and related applications in a manner that encourages additional
duction of information overflow, which ...
bn=1591404142... - [↑](#) [×](#)

[arch: A Survey](#)
robot - [Quick View](#)
· [Related articles](#)
s a converging research area from several research communities,
mining and when comparing research in this ...
c/download.jsessionid...?doi=10.1.1... - [↑](#) [×](#)

upon in data mining terms, can be said to have three operations
ng natural groupings of users, ...
[web-mine/](#) - [Cached](#) - [Similar](#) - [↑](#) [×](#)

Web Mining Algorithms

PageRank

- Measure the relative importance of a web page
- Used by the Google search engine
- Link from X to Y means a vote by X for Y
- A node's PageRank depends on # incoming links and the PageRank of nodes that link to it

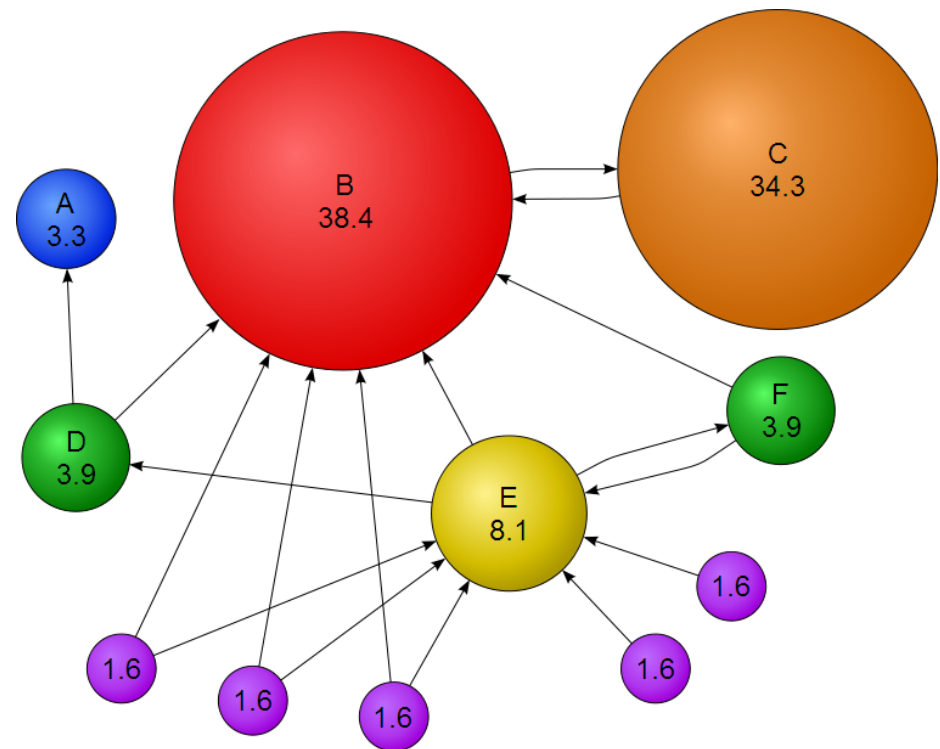
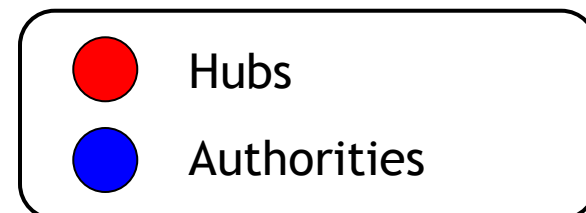
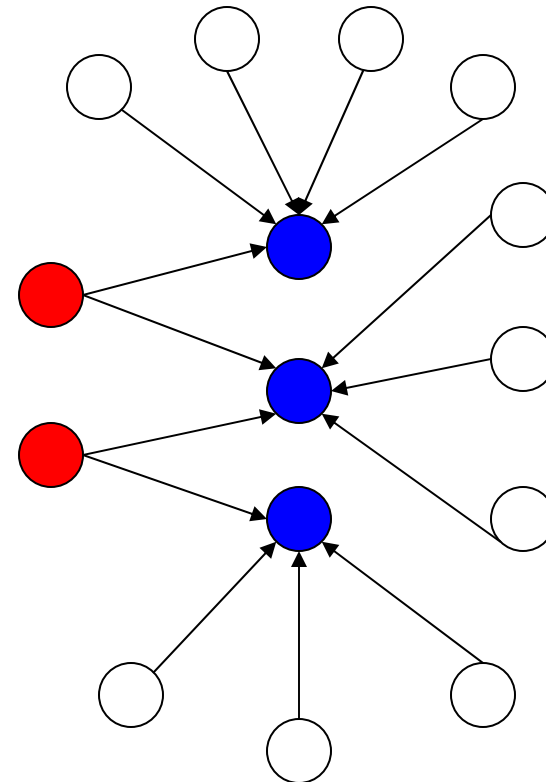


Image source: <http://en.wikipedia.org/wiki/Pagerank>

Web Mining Algorithms

HITS

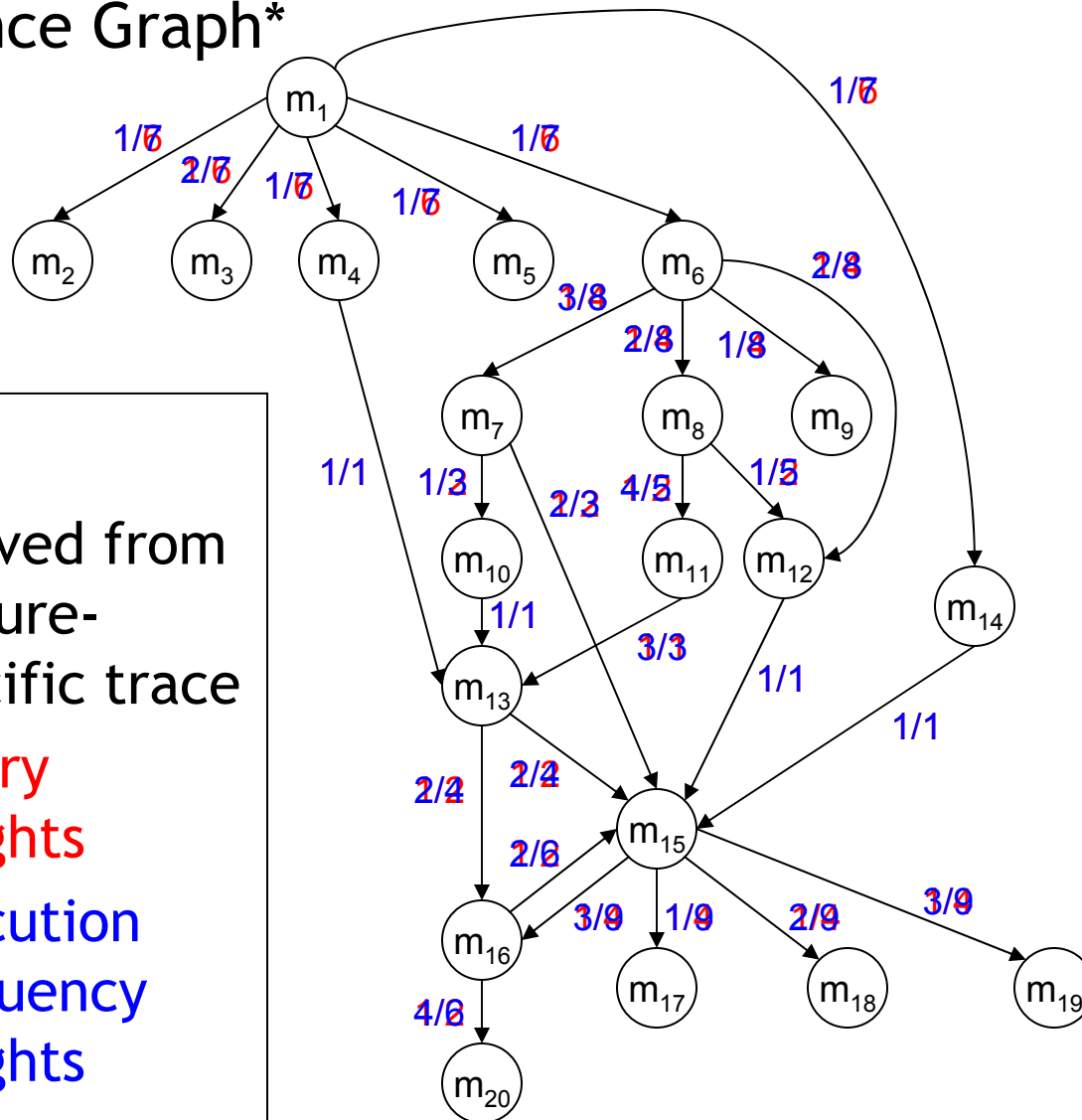
- Hyperlinked-Induced Topic Search
- Identifies hub and authority pages
- Hubs point to many good authorities
- Authorities are pointed to by many hubs



Probabilistic Program Dependence Graph*

PPDG

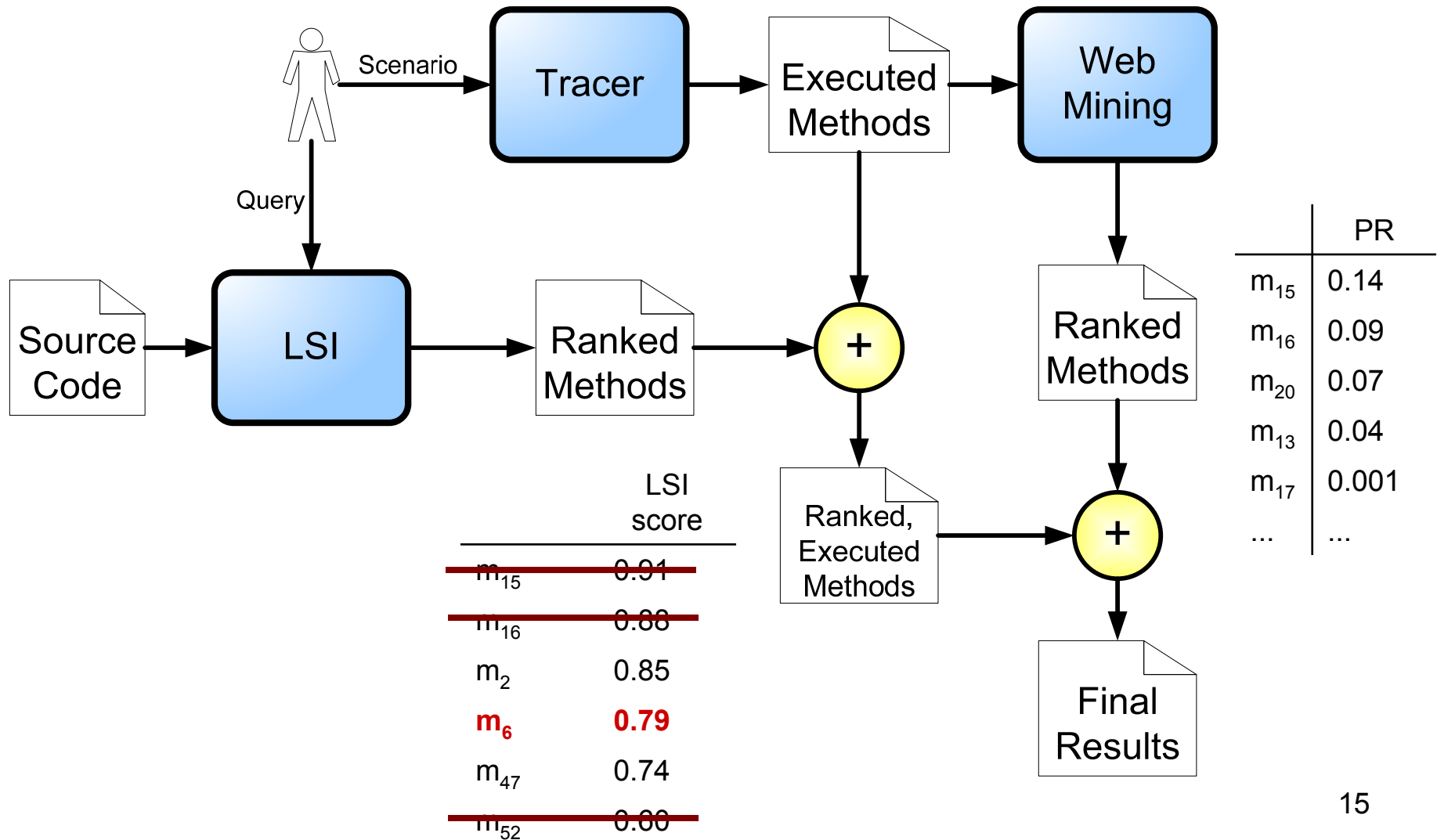
- Derived from feature-specific trace
- **Binary weights**
- **Execution frequency weights**



15
16
20
13
17
18
19
14
10
12
11
7
8
9
2
3
4
5
6
1

*Baah, G. K., Podgurski, A., and Harrold, M. J. 2008. The probabilistic program dependence graph and its application to fault diagnosis. In *Proceedings of the 2008 International Symposium on Software Testing and Analysis*, 2008.

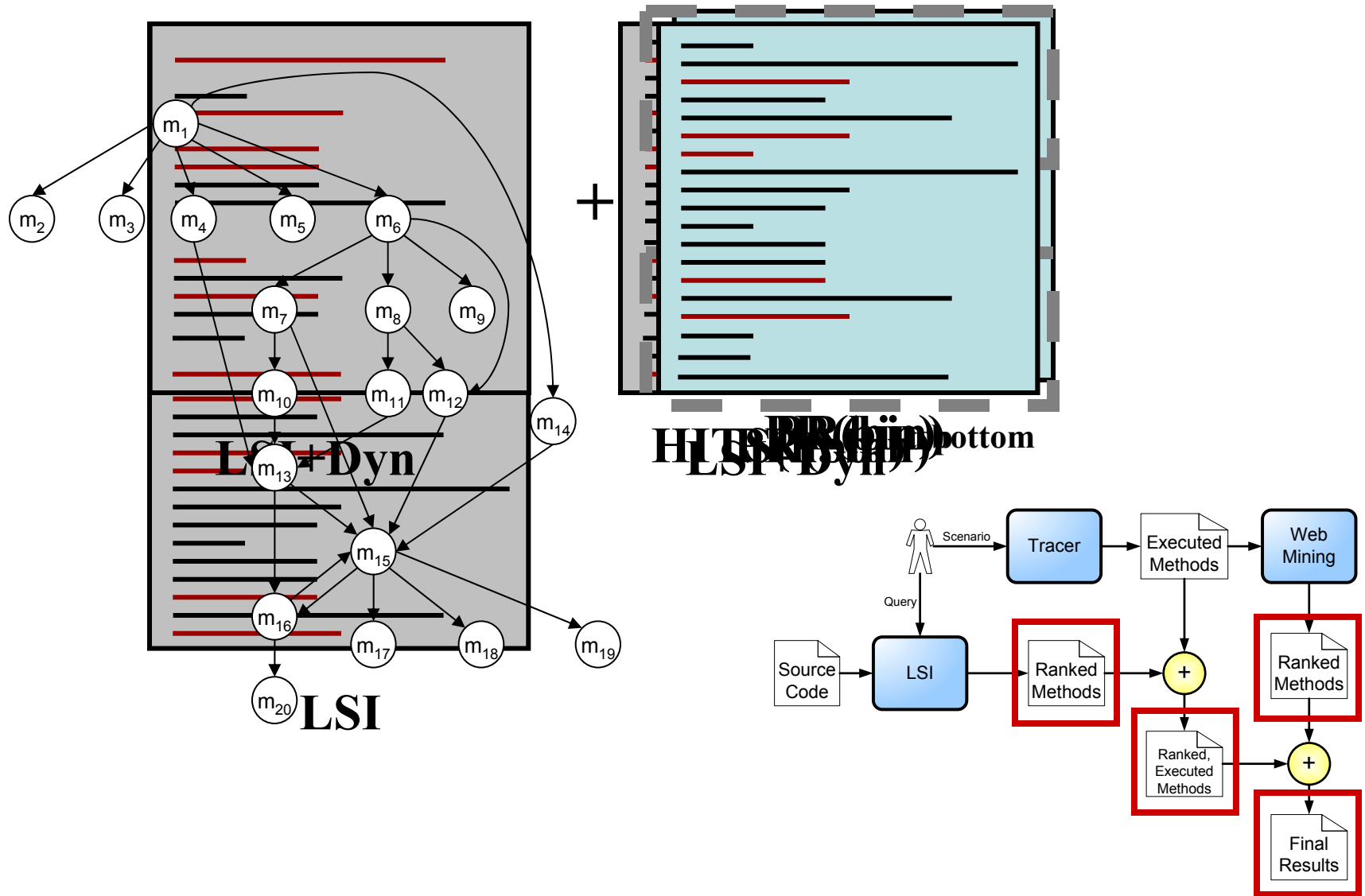
Incorporating Web Mining with Feature Location



Feature Location Techniques Evaluated

LSI & Dynamic Analysis	Web Mining	LSI, Dyn, & PageRank	LSI, Dyn, & HITS	
LSI	PR(bin)	LSI+Dyn+PR(bin) ^{top}	LSI+Dyn+HITS(h,bin) ^{top}	LSI+Dyn+HITS(h,bin) ^{bottom}
LSI+Dyn	PR(freq)	LSI+Dyn+PR(bin) ^{bottom}	LSI+Dyn+HITS(h,freq) ^{top}	LSI+Dyn+HITS(h,freq) ^{bottom}
(baseline)	HITS(h, bin)	LSI+Dyn+PR(freq) ^{top}	LSI+Dyn+HITS(a,bin) ^{top}	LSI+Dyn+HITS(a,bin) ^{bottom}
	HITS(h, freq)	LSI+Dyn+PR(freq) ^{bottom}	LSI+Dyn+HITS(a,freq) ^{top}	LSI+Dyn+HITS(a,freq) ^{bottom}
	HITS(a, bin)			
	HITS(a, freq)			
Use LSI to rank methods, prune unexecuted	Use web mining algorithm to rank methods.	Use LSI to rank methods. Prune unexecuted. Use web mining algorithm to also rank methods and prune top- or bottom- ranked methods from LSI+Dyn's results.		

Feature Location Techniques Explained



Subject Systems

- **Eclipse 3.0**
 - 10K classes, 120K methods, and 1.6 million LOC
 - 45 features
 - Gold set: methods modified to fix bug
 - Queries: short description from bug report
 - Traces: steps to reproduce bug



Bug 66914 - [typing] Error Message after undo copy/paste

Status: VERIFIED FIXED

Reported: 2004-06-14 08:16 EDT by Ralf Schmauder

Product: JDT

Modified: 2004-06-21 04:55 EDT ([History](#))

Component: Text

CC List: 2 users

Version: 3.0

Platform: PC All

birsan
dirk_baeumer

Importance: P2 major ([vote](#))

Target Milestone: 3.0 RC3

Assigned To: Tom Hofmann

QA Contact:

[See Also:](#)

Attachments

error log (5.43 KB, text/plain) 2004-06-14 08:17 EDT, Ralf Schmauder	<i>no flags</i>	Details
LinkedModeUI.diff (4.59 KB, patch) 2004-06-18 09:59 EDT, Tom Hofmann	<i>no flags</i>	Details Diff
Add an attachment (proposed patch, testcase, etc.)		View All

Ralf Schmauder 2004-06-14 08:16:24 EDT

[Description](#)

```
-create a new Class and generate the main method  
-type "sysout" and use the code completion  
-type the double quote  
-paste Hello World into the double quotes  
-try to undo without saving using Ctrl+z
```

```
using undo in the menubar does work
```

Subject Systems

- **Rhino 1.5**
 - 138 classes, 1,870 methods, and 32,134 LOC
 - 241 features
 - Gold set: Eaddy et al.'s dataset*
 - Queries: description in specification
 - Traces: test cases



* <http://www.cs.columbia.edu/~eaddy/concerntagger/>

Size of Traces

		Min	Max	25%	Med	75%	σ	μ
Eclipse	Methods	88K	1.5MM	312K	525K	1MM	666K	406K
	Unique Methods	1.9K	9.3K	3.9K	5K	6.3K	5.1K	2K
	Size-MB	9.5	290	55	98	202	124	83
	Threads	1	26	7	10	12	10	5
Rhino	Methods	160K	12MM	612K	909K	1.8MM	1.8MM	2.3MM
	Unique Methods	777	1.1K	870	917	943	912	54
	Size-MB	18	1,668	71	104	214	210	273
	Threads	1	1	1	1	1	1	0

Research Questions

- **RQ1**
 - Does combining web mining algorithms with an existing approach to feature location improve its effectiveness?
- **RQ2**
 - Which web-mining algorithms, HITS or PageRank, produces better results?

Data Collection & Testing

- **Effectiveness measure**

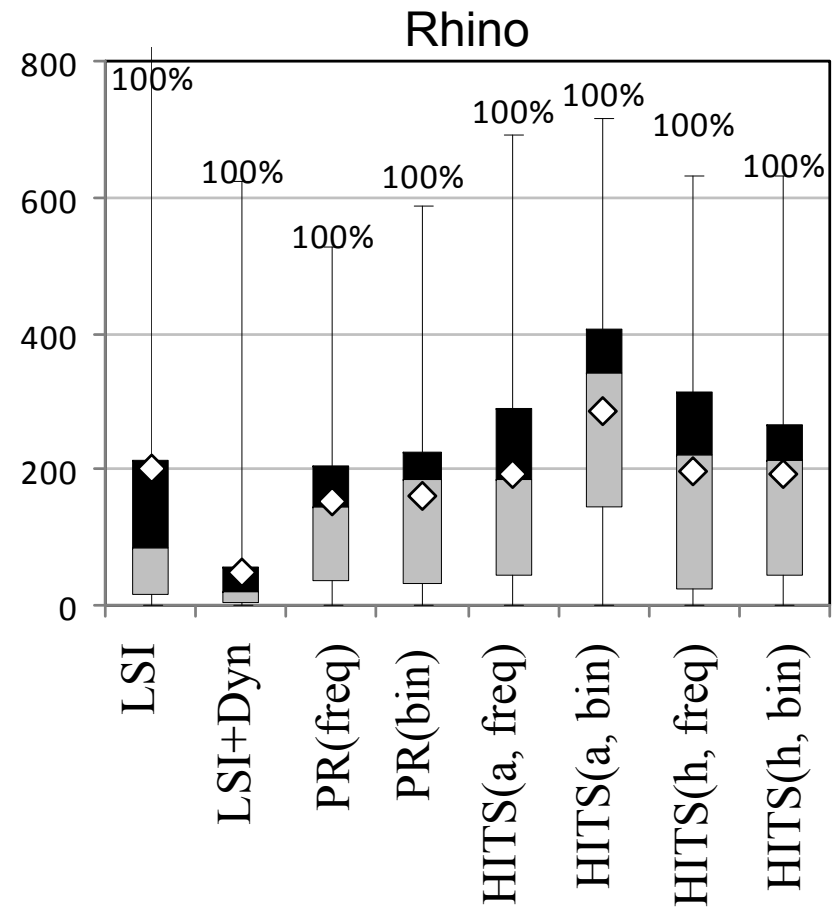
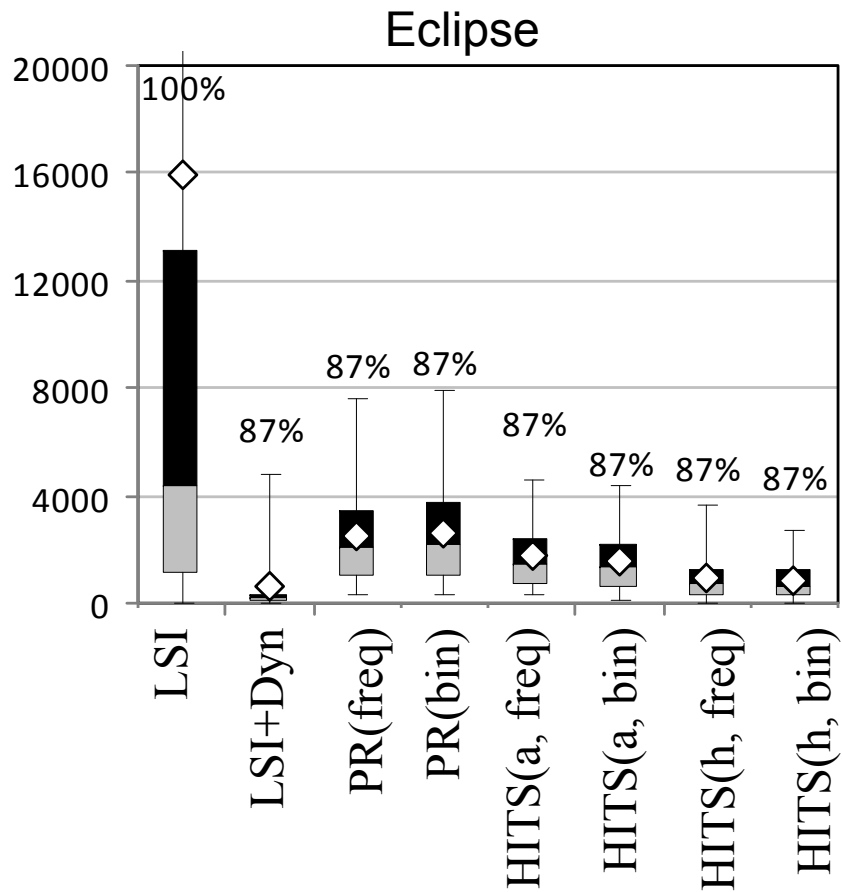
- Descriptive statistics
 - 45 Eclipse features
 - 241 Rhino features

	LSI score	
m_{15}	0.91	
m_{16}	0.88	
m_2	0.85	Effectiveness = 4
m_6	0.79	
m_{47}	0.74	
m_{52}	0.60	

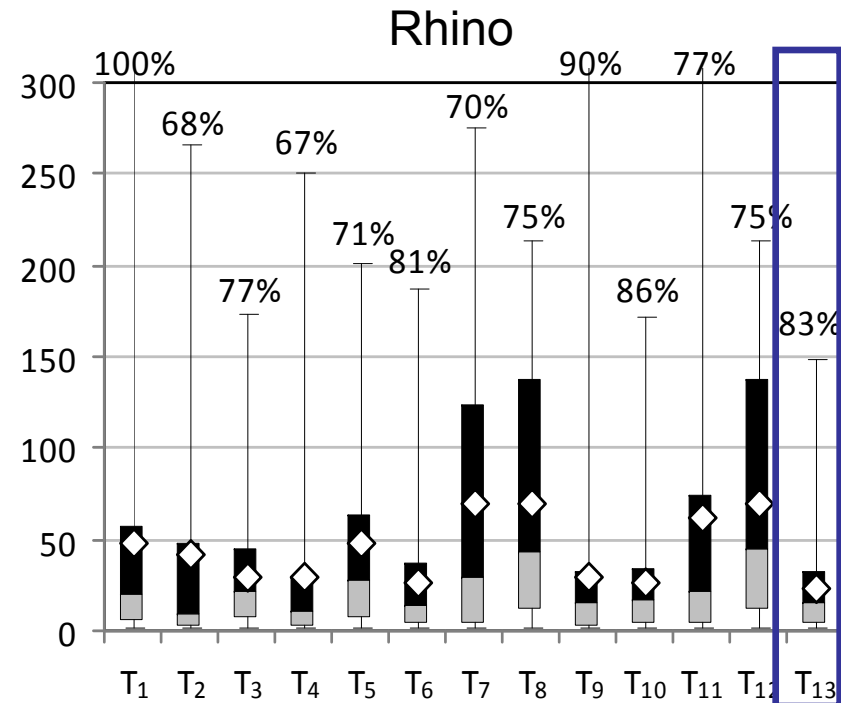
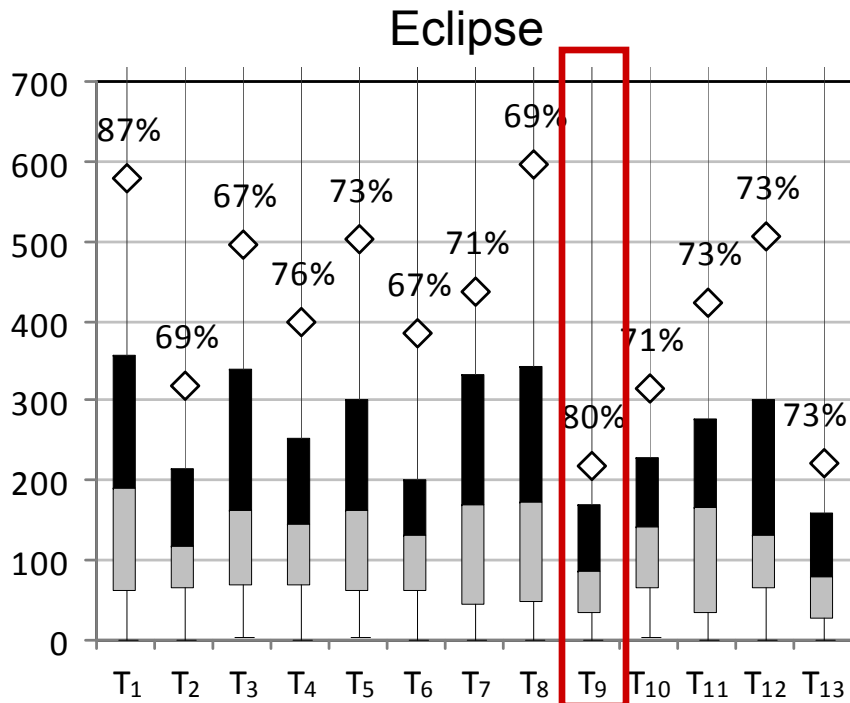
- **Statistical Testing**

- Wilcoxon rank sum test
- Null hypothesis
 - There is no significant difference between the effectiveness of X and the baseline (LSI+Dyn).
- Alternative hypothesis
 - The effectiveness of X is significantly better than the baseline (LSI+Dyn).

Results: Web Mining Techniques



Results: IR, Dyn, & Web Mining



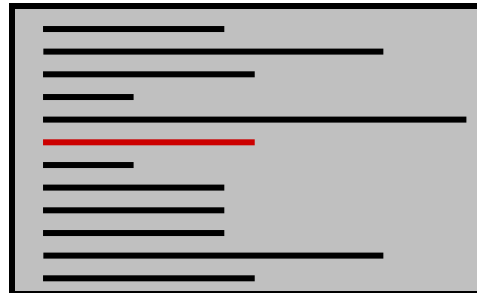
- T₁. LSI+Dyn
- T₂. LSI+Dyn+PR(freq)^{top} [40, 60]%
- T₃. LSI+Dyn+PR(freq)^{bot} [20, 70]%
- T₄. LSI+Dyn+PR(bin)^{top} [40, 60]%
- T₅. LSI+Dyn+PR(bin)^{bot} [10, 70]%
- T₆. LSI+Dyn+HITS(a, freq)^{top} [30, 70]%

- T₇. LSI+Dyn+HITS(a, freq)^{bot} [40, 60]%
- T₈. LSI+Dyn+HITS(h, freq)^{top} [10, 70]%
- T₉. LSI+Dyn+HITS(h, freq)^{bot} [60, 50]%
- T₁₀. LSI+Dyn+HITS(a, bin)^{top} [20, 70]%
- T₁₁. LSI+Dyn+HITS(a, bin)^{bot} [40, 40]%
- T₁₂. LSI+Dyn+HITS(h, bin)^{top} [10, 70]%
- T₁₃. LSI+Dyn+HITS(h, bin)^{bot} [70, 60]%

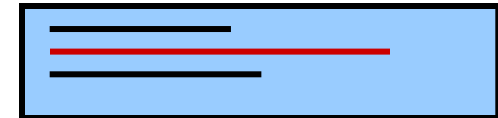
A Case in Point: Eclipse exclusion filter



LSI = 1,696



LSI+Dyn = 61



**LSI+Dyn+
HITS(h, bin)^{bottom}
= 24**

Results of the Wilcoxon Rank Sum test comparing these techniques to the baseline, LSI+Dyn.

$\alpha = 0.05$.

Null Hypothesis:
There is no significant difference between the effectiveness of X and the baseline, LSI+Dyn.

	Eclipse	Rhino	Null Hypothesis
PR(bin)	1	1	Not Rejected
PR(freq)	1	1	Not Rejected
HITS(h, bin)	1	1	Not Rejected
HITS(h, freq)	1	1	Not Rejected
HITS(a, bin)	1	1	Not Rejected
HITS(a, freq)	1	1	Not Rejected
LSI+Dyn+PR(bin) ^{top}	< 0.0001	< 0.0001	Rejected
LSI+Dyn+PR(bin) ^{bottom}	0.004	0	Rejected
LSI+Dyn+PR(freq) ^{top}	< 0.0001	< 0.0001	Rejected
LSI+Dyn+PR(freq) ^{bottom}	< 0.0001	0.74	Not Rejected
LSI+Dyn+HITS(a, freq) ^{top}	0	< 0.0001	Rejected
LSI+Dyn+HITS(a, freq) ^{bottom}	< 0.0001	0.99	Not Rejected
LSI+Dyn+HITS(h, freq) ^{top}	0	1	Not Rejected
LSI+Dyn+HITS(h, freq) ^{bottom}	< 0.0001	< 0.0001	Rejected
LSI+Dyn+HITS(a, bin) ^{top}	< 0.0001	< 0.0001	Rejected
LSI+Dyn+HITS(a, bin) ^{bottom}	< 0.0001	1	Not Rejected
LSI+Dyn+HITS(h, bin) ^{top}	0	1	Not Rejected
LSI+Dyn+HITS(h, bin) ^{bottom}	< 0.0001	< 0.0001	Rejected

Research Questions Revisited

- **RQ1**: Does combining web mining algorithms with an existing approach to feature location improve its effectiveness?
 - Yes
- **RQ2**: Which web-mining algorithms, HITS or PageRank, produces better results?
 - HITS

Best Techniques

- LSI+Dyn+HITS(h, freq)^{bottom}
- LSI+Dyn+HITS(h, bin)^{bottom}

- Methods with low HITS hub values are getters and setters

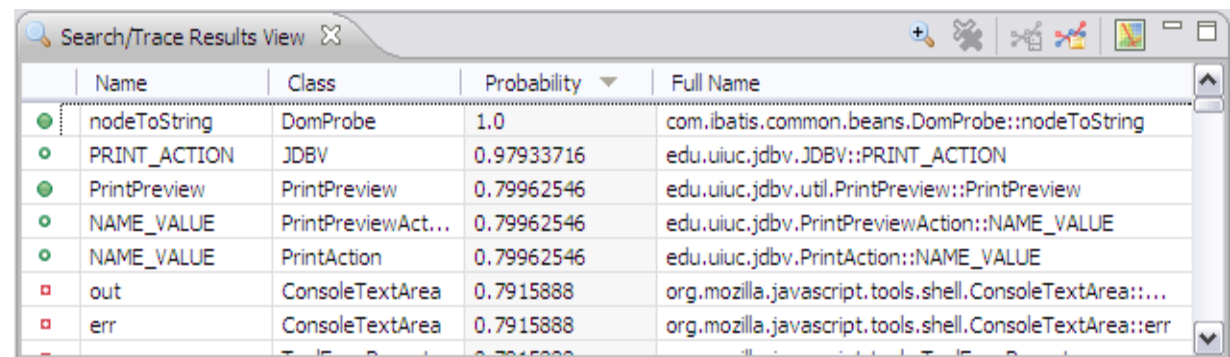
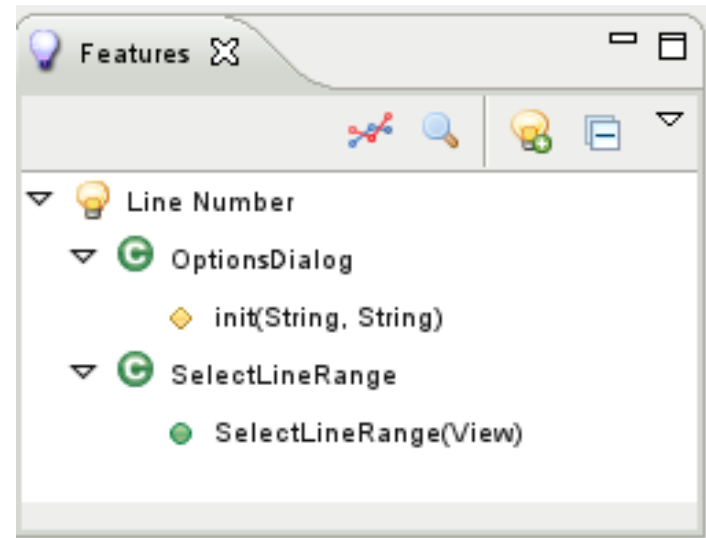
Current Work (not in the paper)

- HITS and PageRank on static vs. dynamic info
- Evaluation first relevant vs. all relevant methods
- Evaluation against fan-in and fan-out and heuristics based on setters and getters
- Impact of thresholds on the filtering power

Tool Support

FLAT³

- **FLAT³**
 - Eclipse Plug-in
 - Lucene-based IR
 - Execution tracing
 - Integration
 - Tagging
 - Metrics



	Name	Class	Probability	Full Name
●	nodeToString	DomProbe	1.0	com.ibatis.common.beans.DomProbe::nodeToString
●	PRINT_ACTION	JDBV	0.97933716	edu.uiuc.jdbv.JDBV::PRINT_ACTION
●	PrintPreview	PrintPreview	0.79962546	edu.uiuc.jdbv.util.PrintPreview::PrintPreview
●	NAME_VALUE	PrintPreviewAct...	0.79962546	edu.uiuc.jdbv.PrintPreviewAction::NAME_VALUE
●	NAME_VALUE	PrintAction	0.79962546	edu.uiuc.jdbv.PrintAction::NAME_VALUE
□	out	ConsoleTextArea	0.7915888	org.mozilla.javascript.tools.shell.ConsoleTextArea::...
□	err	ConsoleTextArea	0.7915888	org.mozilla.javascript.tools.shell.ConsoleTextArea::err

<http://www.cs.wm.edu/semeru/flat3/>

Trevor Savage, Meghan Reville, and Denys Poshvanyk. "FLAT3: Feature Location and Textual Tracing Tool." In the Proceedings of the 32nd International Conference on Software Engineering (ICSE'10), Formal Research Tool Demonstration, Cape Town, South Africa, May 2-8, 2010.

Summary

- Proposed and implemented **novel methods** for feature location based combinations of:
 - **Textual analysis, dynamic analysis and web mining**
- Evaluated proposed methods on **large**, open-source systems
- Developed **practical tools** for the proposed approaches
- Released **benchmarks** for feature location:
 - <http://www.cs.wm.edu/semeru/data/icpc10-data-fusion/>

Searching beyond a project ...



<http://www.xemplar.org/>

A Search Engine for Finding Highly-Relevant Applications

- Online repositories contain many millions of lines of code, but reusing them is a very difficult problem.
- The **high-level descriptions** of applications do not usually match its **low-level implementation** details.

```
public static long mystery(long a, long b)
{
    if (b==0)
        return a;
    else
        return mystery(b, a % b);
}
```



Compute the **Greatest Common Denominator**

- We present Exemplar, a source code search engine to bridge this mismatch by integrating API help documentation into the search process.

Example Programming Task

Write an application to record musical instrument data to a file in the MIDI file format



What programmers do

The programmer may check other applications for API calls from third-party packages used to read data from a MIDI device and then print to a file.

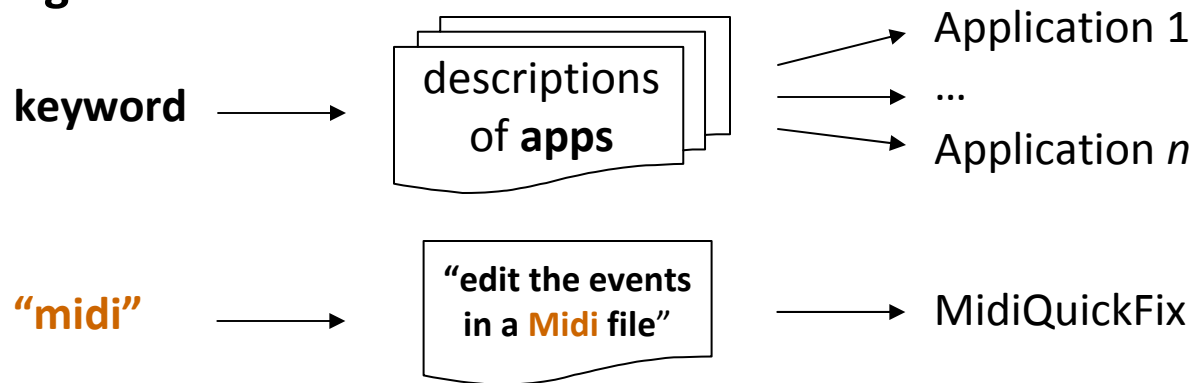
```
List allInfos = new ArrayList();
List providers = getMidiDeviceProviders();
...
MidiDevice.Info[] infosArray
    = (MidiDevice.Info[]) allInfos.toArray(new MidiDevice.Info[0]);

for(int i = 0; i < infosArray.size(); i++) {
    fileOutput.print(infosArray[i]);
    ...
}
```

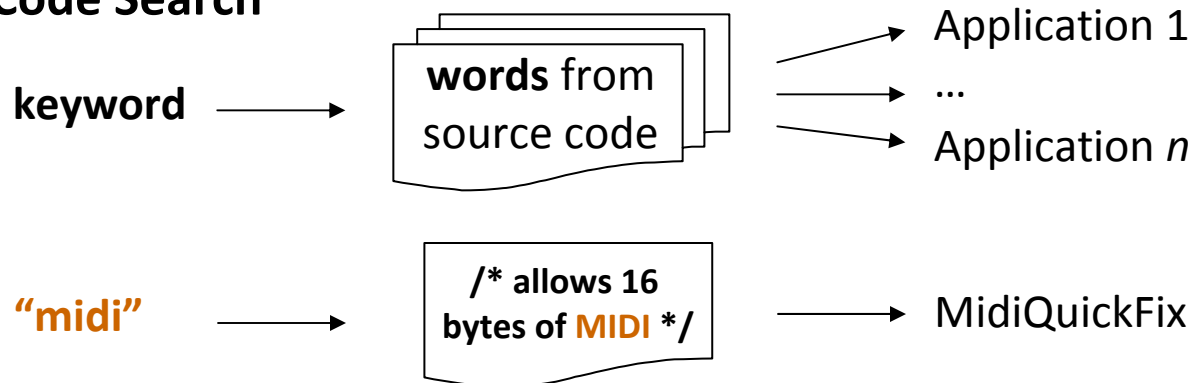
8,000 Java projects that we extracted from SourceForge make over 11,000,000 calls to the official Java API.

Many search engines rely on words from applications

SourceForge

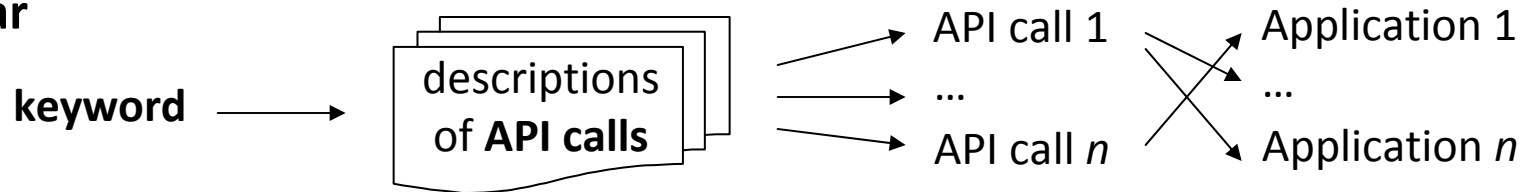


Google Code Search

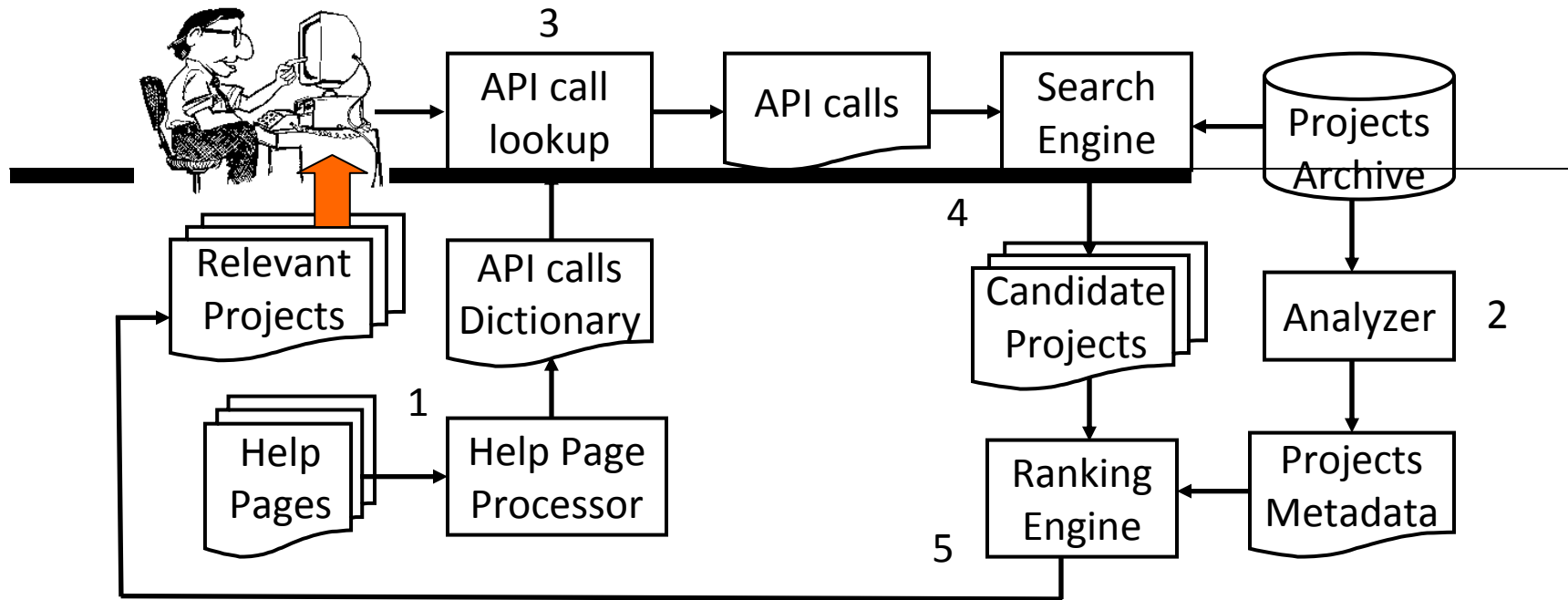


Our idea is to augment standard code search to include API documentation

Exemplar



"record midi file"



javax.sound.midi.MidiDevice.getReceiver()

... Obtains a MIDI IN receiver through which the MIDI device may receive MIDI data ...

javax.imageio.ImageWriter.write()

... Appends a complete image stream containing a single image ...

java.awt.geom.AffineTransform.getScaleY()

... scaling element (m11) of the 3x3 affine transformation matrix ...

AffineTransform.getScaleY()
AffineTransform.createInverse()

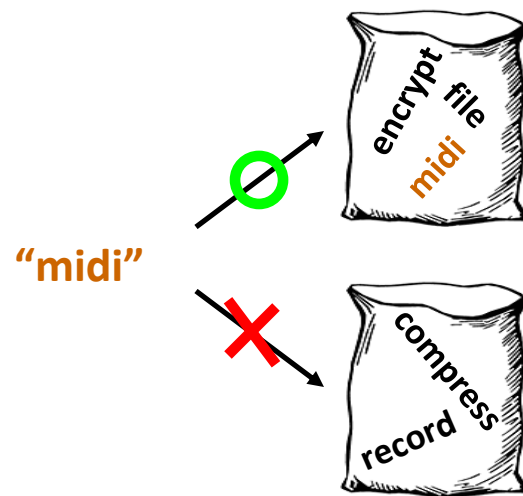
Jazilla

ShortMessage.ShortMessage()
MidiDevice.getReceiver()
MidiEvent.MidiEvent()

Tritonus

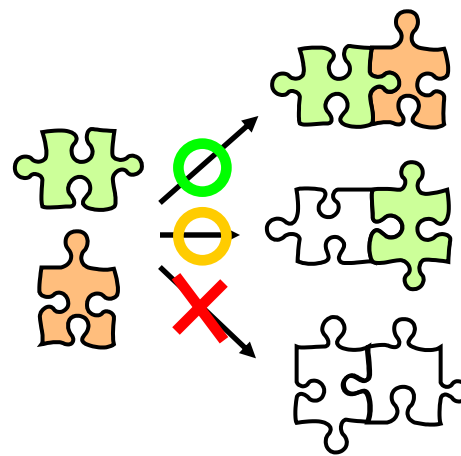
There are three components to compute scores in Exemplar's ranking system.

Word Occurrences (WOS)



Exemplar ranks applications higher when their descriptions contain keywords from the query

Relevant API Calls (RAS)



An application's RAS score is raised if it makes more calls to relevant methods in the API

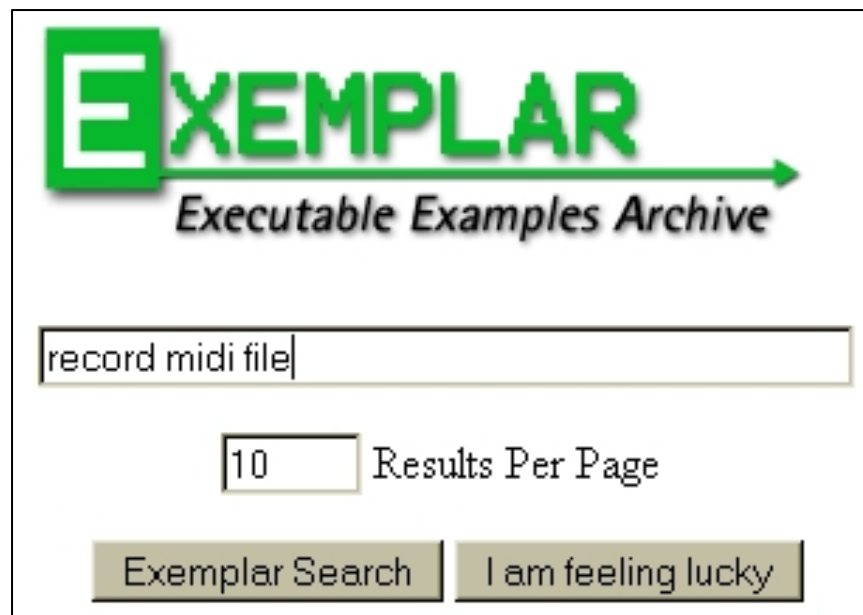
Dataflow Connections (DCS)

"record midi file"

```
String dev = getDevice();  
String buf[] = A.readMidi(msg);  
B.write(buf);
```

If two relevant API calls share data in an application, Exemplar ranks that application higher

The user enters a high-level query.



The image shows a screenshot of the Exemplar website's search interface. At the top left is the logo, which consists of a green square containing a white stylized 'E' followed by the word 'XEMPLAR' in green, with a green arrow pointing to the right underneath. Below the logo is the text 'Executable Examples Archive'. Below this is a search input field containing the text 'record midi file'. Underneath the search field is a dropdown menu showing '10' and the text 'Results Per Page'. At the bottom of the interface are two buttons: 'Exemplar Search' and 'I am feeling lucky'.

<http://www.exemplar.org/>

The search returns a list of projects, their descriptions, and their scores.

Project Name	Relevance Score	Description
MidiQuickFix	100%::45.59%	MidiQuickFix allows you to directly edit the events in a Midi file. It is intended to make it easy to find and fix problems, such as setting volume and pan values for a track, without the need for a complex Midi sequencing program.
Saiph	100%::30.71%	Java-based (multiplatform) tool for algorithmic musical composition. Saiph generates sequences made of tracks made of segments with musical events, currently notes and MIDI controllers. It supports MIDI and MusicXML file output.
PJLMidiParser	100%::0%	PJLMidiParser provides efficient parsers, written in Java, for MIDI files. It is like XML SAX parsers in that it is event-driven; the parsing is initiated and then triggers callback handlers in response to events in the MIDI file.
Tritonus	0%::100%	Tritonus is an independent implementation of the Java Sound API (http://www.javasoft.com/products/java-media/sound/index.html).
TuxGuitar	0%::82.8%	TuxGuitar is a multitrack guitar tablature editor and player written in Java-SWT, It can open GuitarPro, PowerTab and TablEdit files.

The programmer can view a list of API calls and their locations within projects.

File	Line No	API Used
0.3.0/tritonus-0.3.0.tar.gz/tritonus-0.3.0/src/javax/sound/midi/MetaMessage.java	93	javax:sound:midi:MetaMessage::MetaMessage
0.3.0/tritonus-0.3.0.tar.gz/tritonus-0.3.0/src/javax/sound/midi/Track.java	54	javax:sound:midi:MidiEvent::getTick
0.3.0/tritonus-0.3.0.tar.gz/tritonus-0.3.0/src/javax/sound/midi/Track.java	54	javax:sound:midi:MidiEvent::MidiEvent
0.3.0/tritonus-0.3.0.tar.gz/tritonus-0.3.0/src/javax/sound/midi/Track.java	98	javax:sound:midi:MidiEvent::getTick
0.3.0/tritonus-0.3.0.tar.gz/tritonus-0.3.0/src/javax/sound/midi/Sequence.java	79	javax:sound:midi:Sequence::createTrack
0.3.0/tritonus-0.3.0.tar.gz/tritonus-0.3.0/src/javax/sound/midi/MidiSystem.java	60	javax:sound:midi:MidiSystem::getMidiDeviceProviders

Exemplar: EXEcutable exaMPLes ARchive

For more details, see our technical paper:

M. Grechanik, C. Fu, Q. Xie, C. McMillan, D. Poshyvanyk, and C. Cumby, "A Search Engine For Finding Highly Relevant Applications," *Proc. of 32nd ACM/IEEE International Conference on Software Engineering*, p. 10, May 2-8 2010.

<http://www.exemplar.org/>



Other “Interesting” Tools and Engines

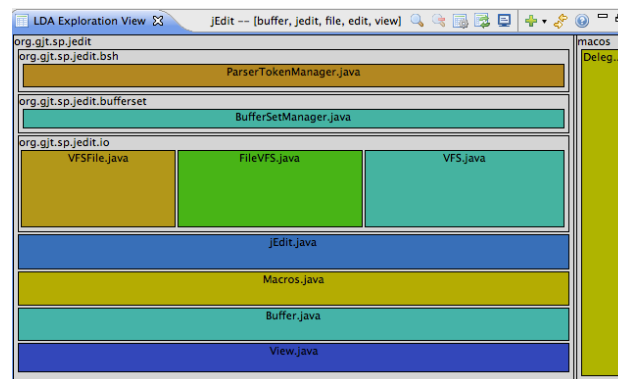
Portfolio



CLAN



TopicXP



SEMERU: Research Team @ W&M



Thank you. Questions?

SEMERU @ William and Mary

<http://www.cs.wm.edu/semeru/>

denys@cs.wm.edu



SEMERU



**WILLIAM
& MARY**