

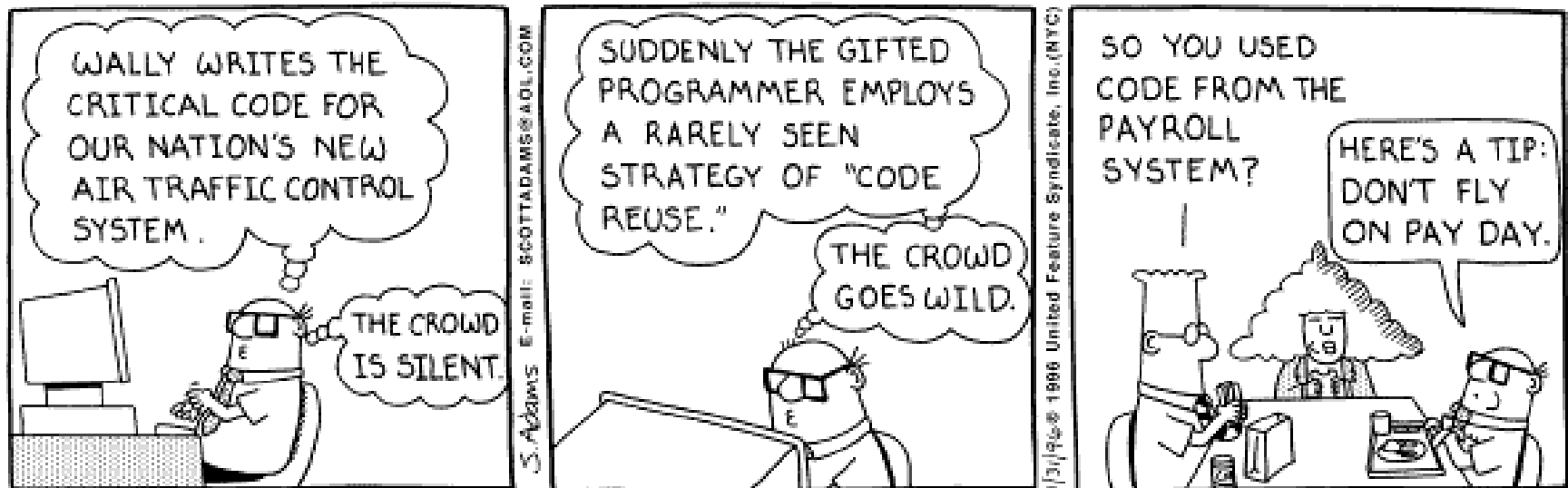


Exemplar: A Search Engine For Finding Highly Relevant Applications

**Mark Grechanik, Chen Fu, Qing Xie, Collin McMillan,
Denys Poshyvanyk and Chad Cumby**

Support: NSF CCF-0916139, NSF CCF-0916260, Accenture, and United States AFOSR grant number FA9550-07-1-0030.

Code Reuse Is Difficult



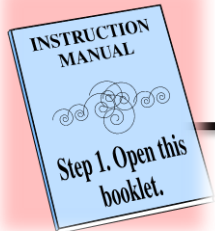
Copyright © 1996 United Feature Syndicate, Inc.
Redistribution in whole or in part prohibited

What do we look for when reusing code?

Problem And Solution Spaces

Problem Space

sweet, love,
harmony, ...



Requirements
Document



encrypt, send,
receive, XML, ...

Solution Space

Our Goal

EXEMPLAR
Executable Example Icons

high performance

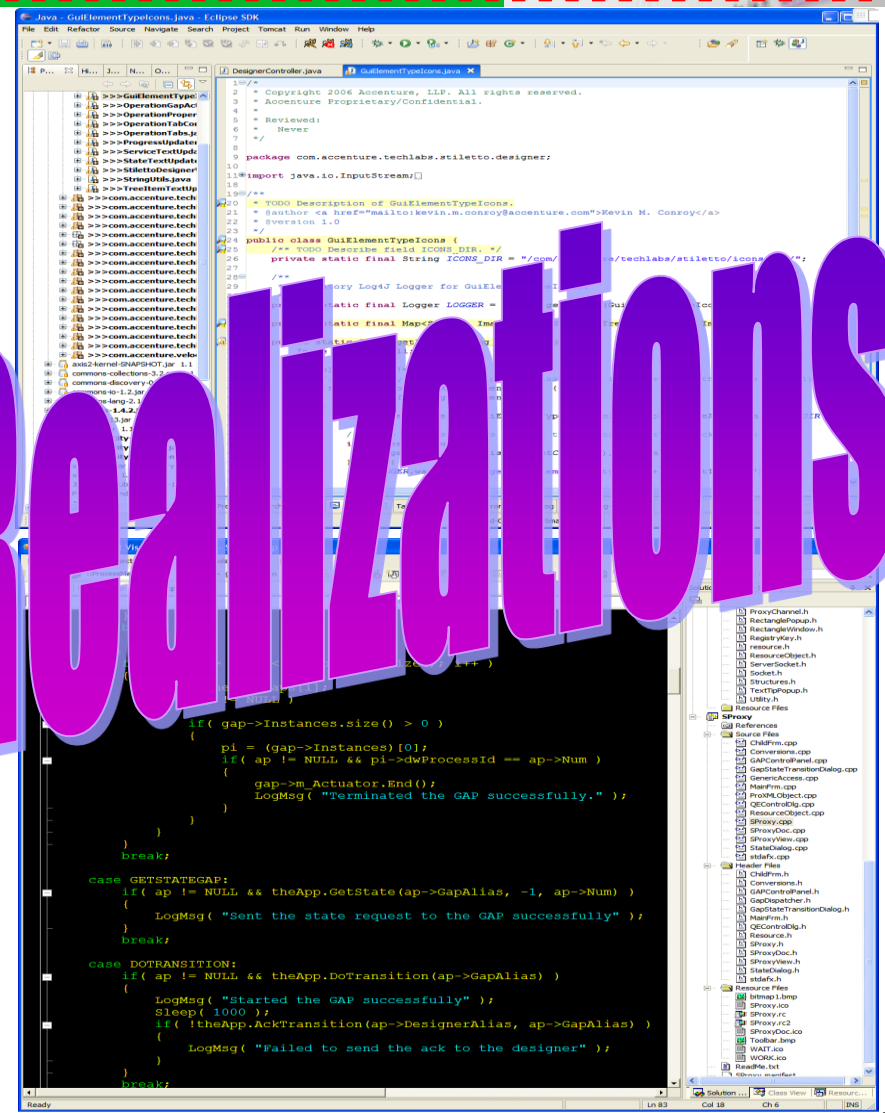
locate

Abstractions

secure

Arch

Realizations

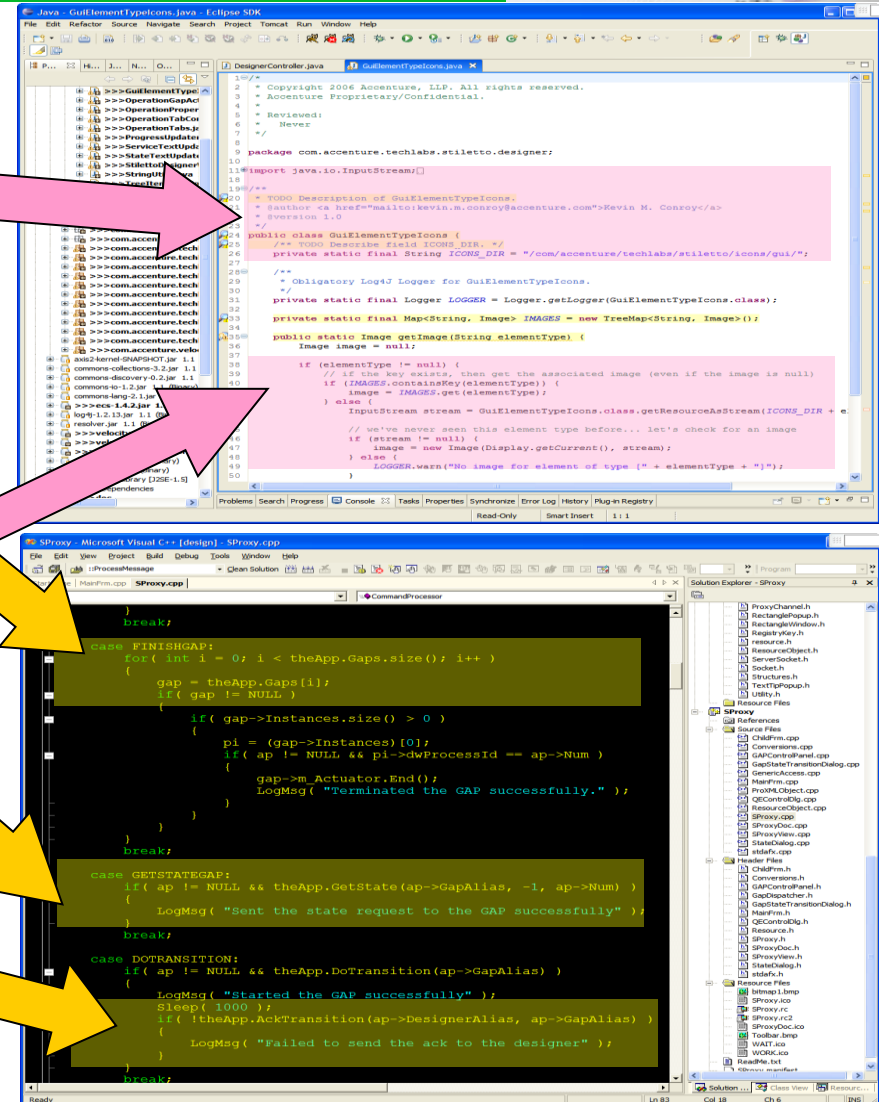


Our Goal

EXEMPLAR
Executable Example Icons

high performance

locate
service
send
XML
secure



Fundamental Problems

- Mismatch between the high-level intent reflected in the descriptions of applications and their low-level implementation details
- Concept assignment problem
 - to identify how high-level concepts are associated with their implementations in source code

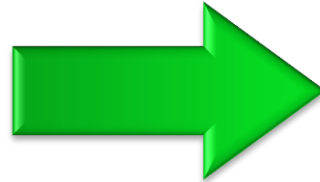
```
s = socket.socket(proto, socket.SOCK_DGRAM)
s.sendto(teststring, addr)
buf = data = receive(s, 100)
while data and '\n' not in buf:
    data = receive(s, 100)
    buf += data
```

Send
data

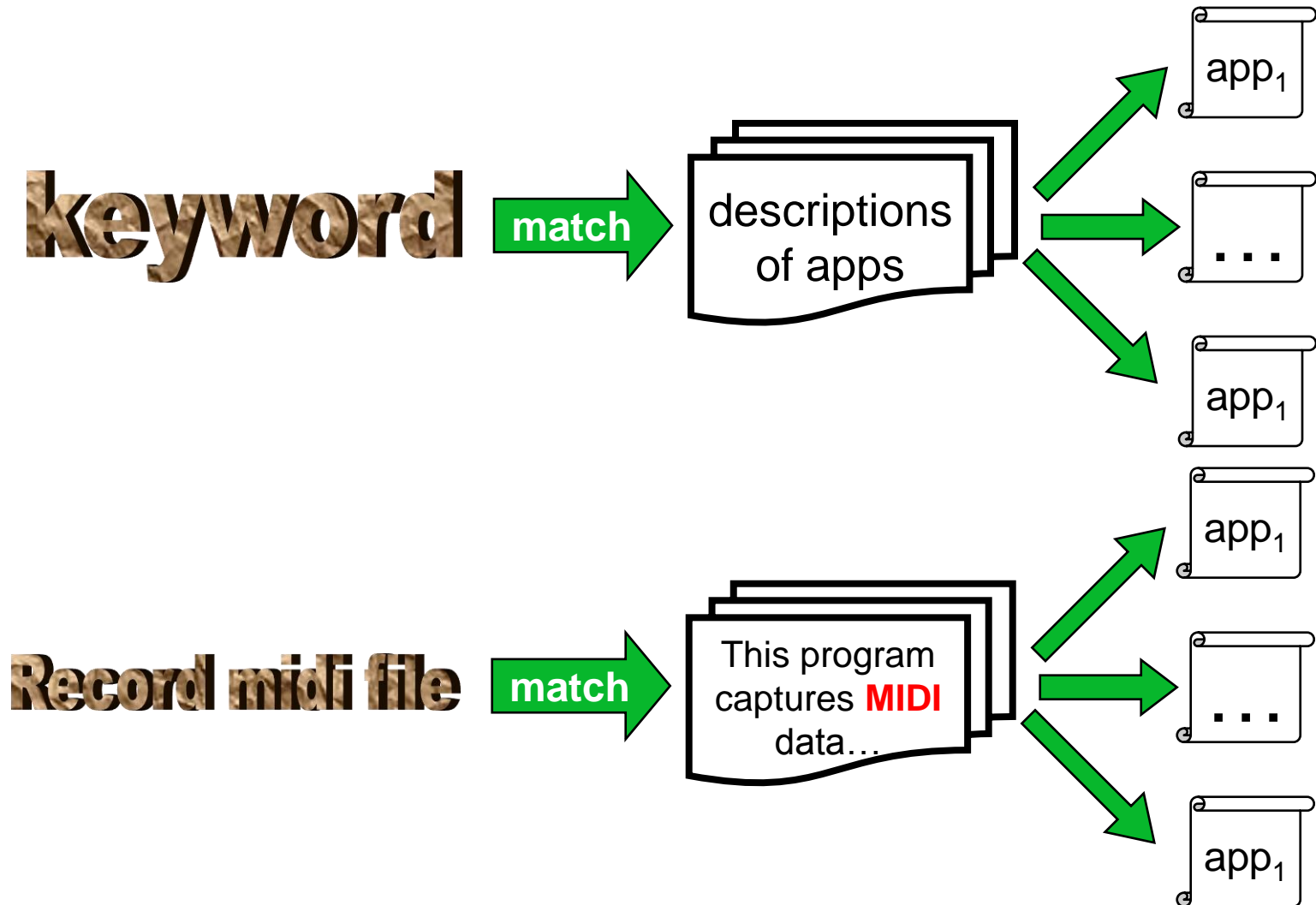


Example Programming Task

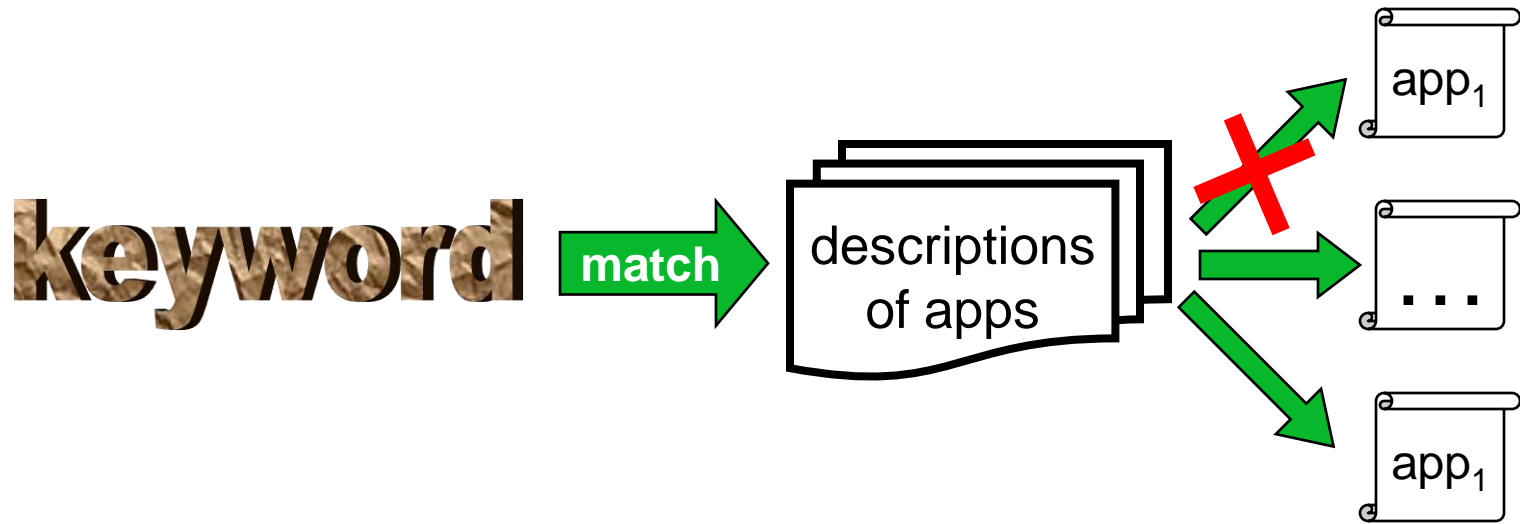
Write an application to record musical instrument data to a file in the MIDI file format.



What Search Engines Do

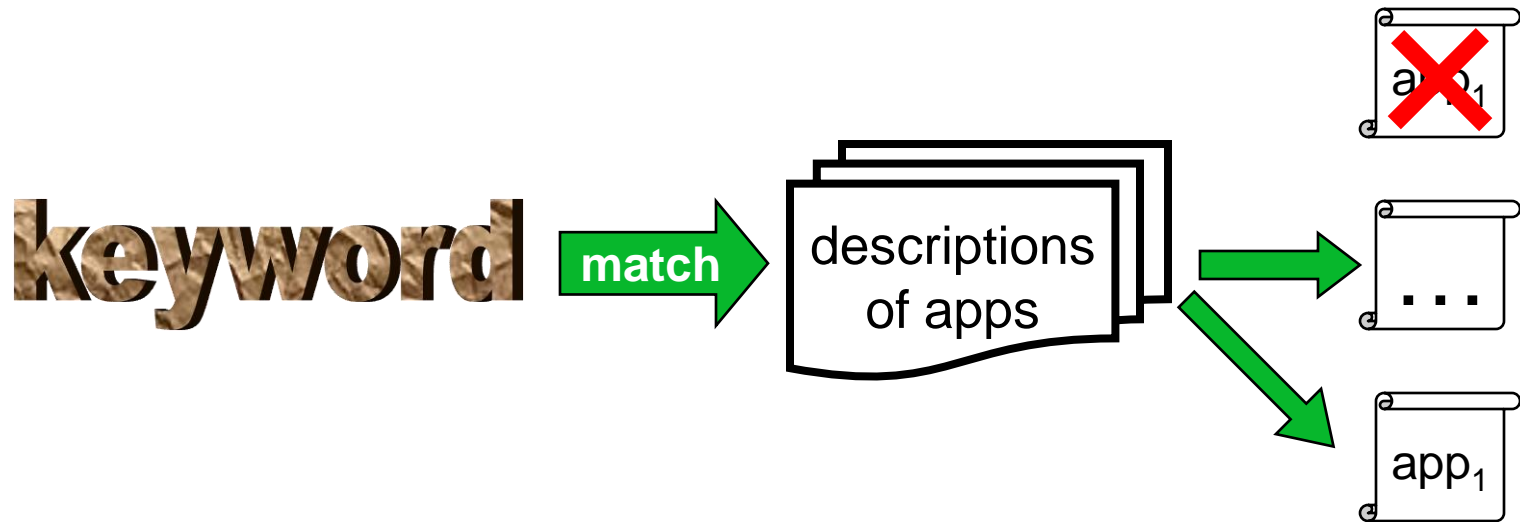


What Search Engines Do



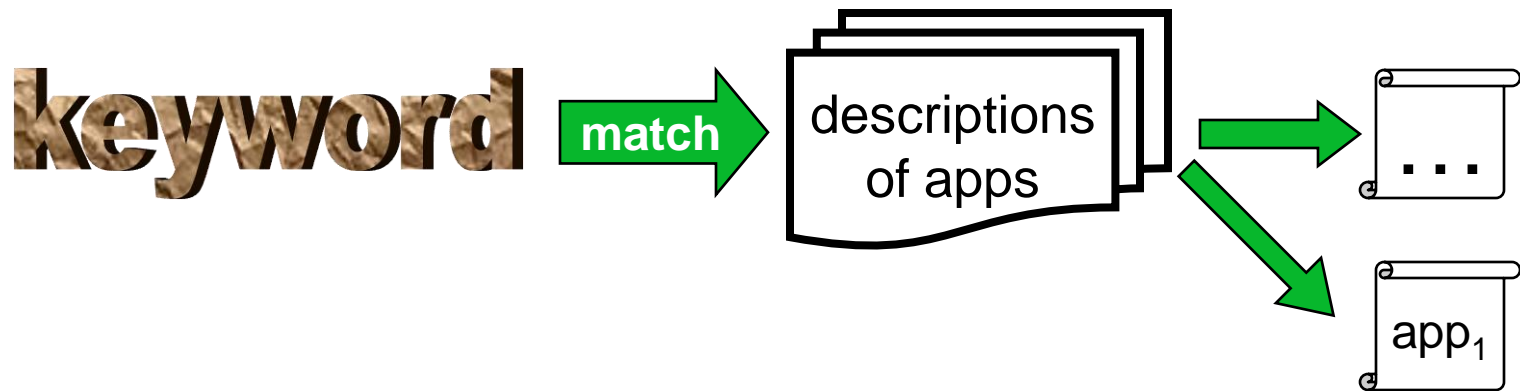
The Vocabulary Problem

What Search Engines Do



The Vocabulary Problem

What Search Engines Do



The Vocabulary Problem

Poorly Described Applications

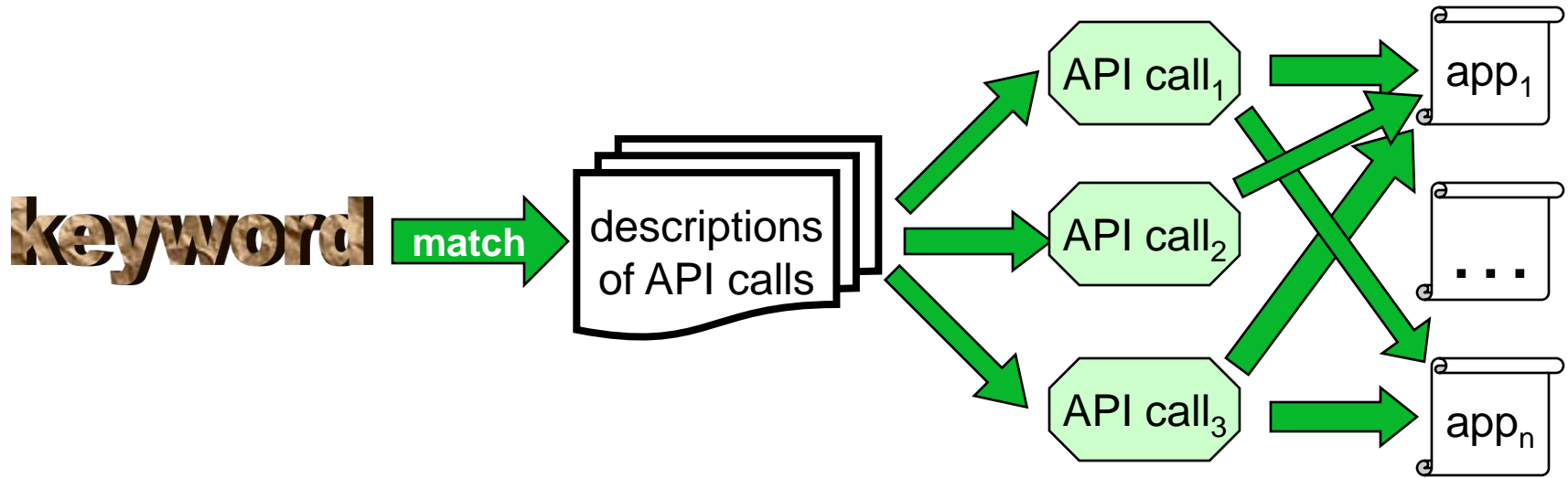


- Many application repositories are polluted with poorly functioning projects.
- Matches between keywords from the queries with words in the descriptions of the applications do not guarantee that these applications are relevant.

How Does It Work Now?

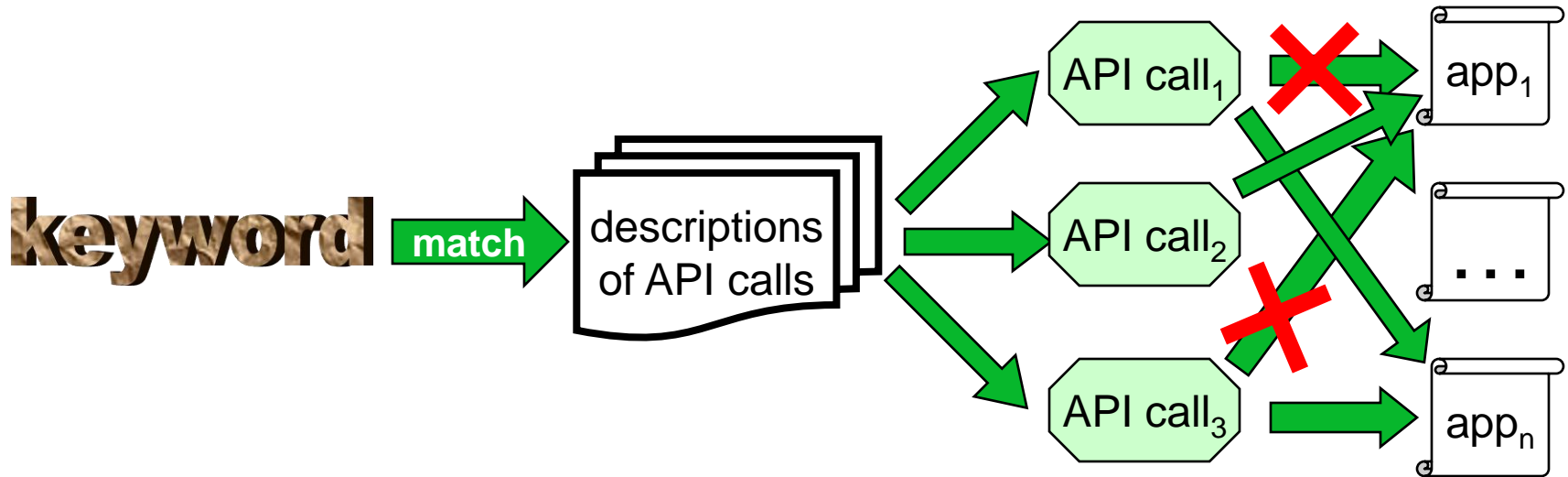
- ➡ Download application.
- ➡ Locate and examine fragments of the code that implement the desired features.
- ➡ Observe the runtime behavior of this application to ensure that this behavior matches requirements.
- ➡ This process is manual since programmers:
 - ➡ study the source code of the retrieved applications
 - ➡ locate various **API calls**
 - ➡ read information about these calls in help documents
- ➡ Still, it is difficult for programmers to link high-level concepts from requirements to their implementations in source code.

How Does Exemplar Work?



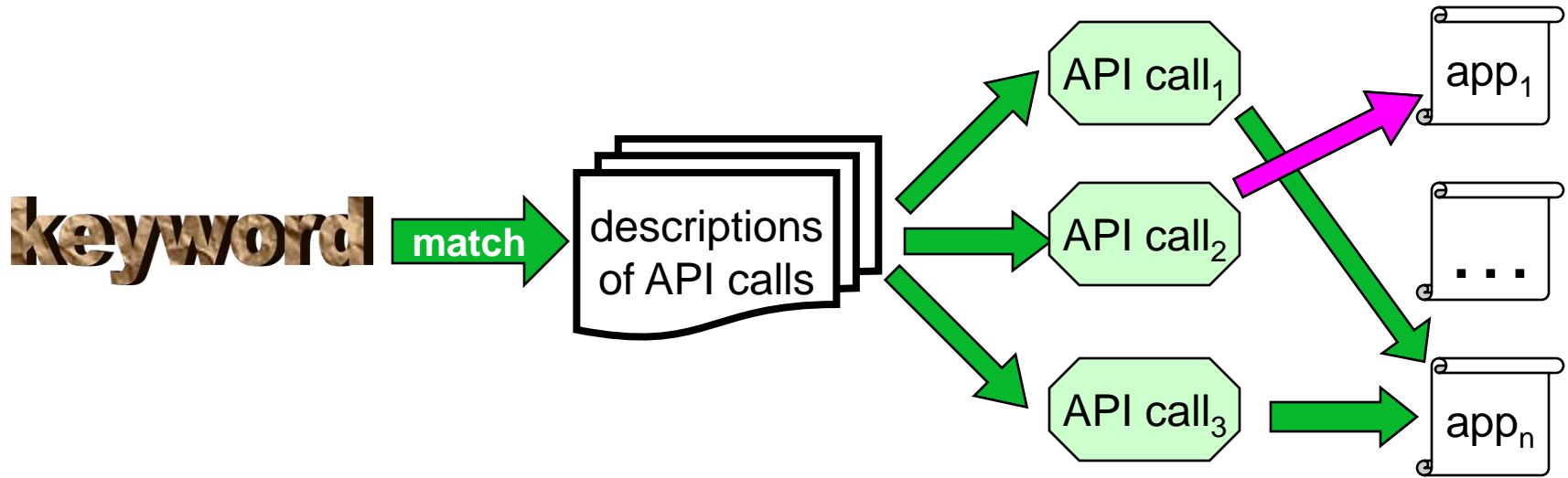
Exemplar uses help documents to produce the names of the API calls in return to user queries thereby expanding these queries. The richness of these vocabularies makes it more likely to find matches, and produce different API calls. If some help document does not contain a desired match, some other document may yield a match.

How Exemplar Works



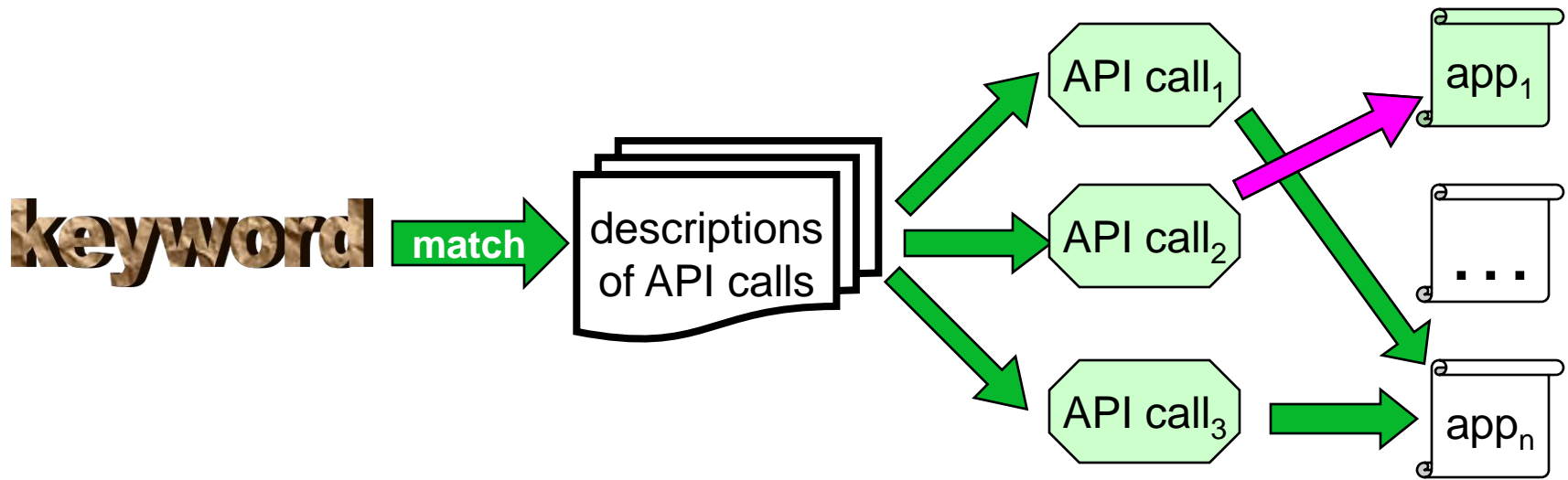
Exemplar uses help documents to produce the names of the API calls in return to user queries thereby expanding these queries. The richness of these vocabularies makes it more likely to find matches, and produce different API calls. If some help document does not contain a desired match, some other document may yield a match.

How Exemplar Works



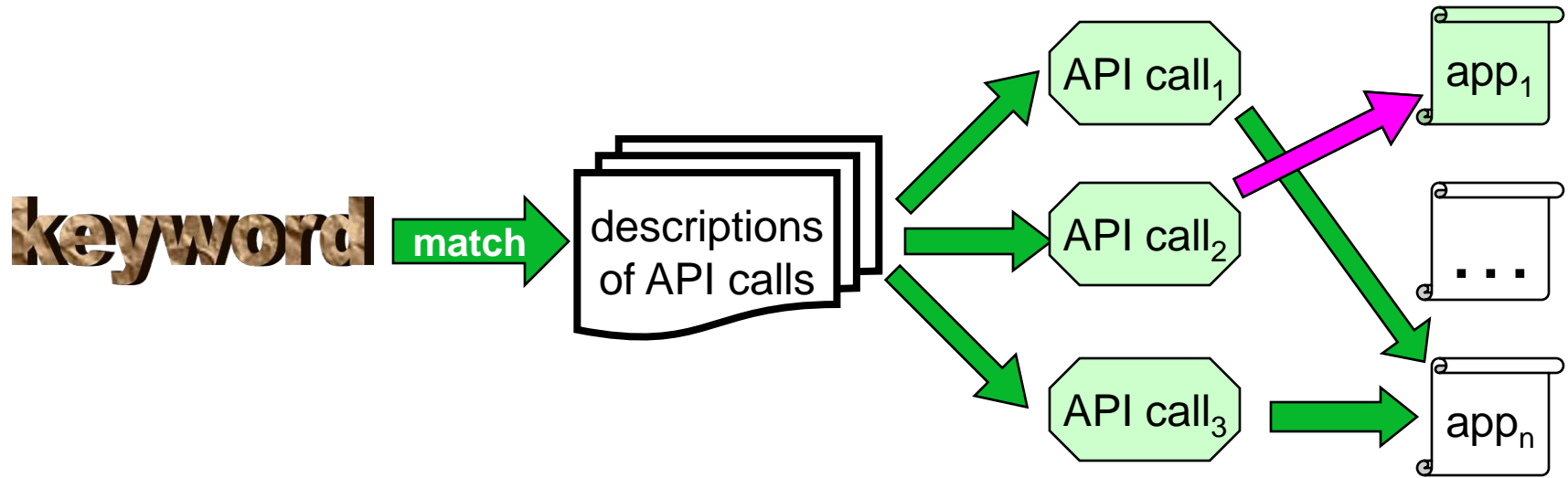
Exemplar uses help documents to produce the names of the API calls in return to user queries thereby expanding these queries. The richness of these vocabularies makes it more likely to find matches, and produce different API calls. If some help document does not contain a desired match, some other document may yield a match.

How Exemplar Works

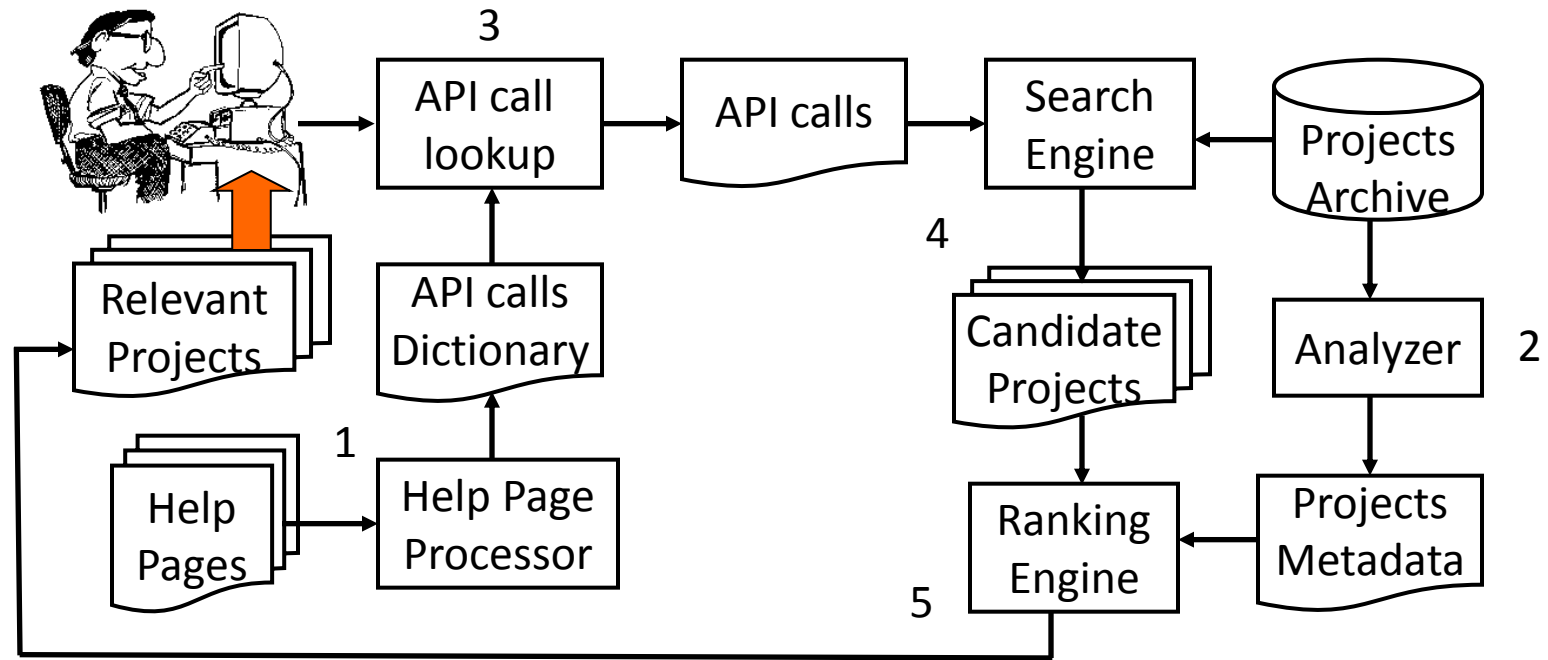


Search widely used library API documents. These documents contain rich vocabularies -> more likely to find right match

How Exemplar Works



"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

Query Expansion



high performance

- Reduce this query/document mismatch by expanding the query with keywords that have a similar meaning to the set of relevant documents
- New keywords come from help documents
- Initial query is expanded to include the names of the API calls whose semantics unequivocally reflects specific behavior of the matched applications

Solving An Instance of the Concept Assignment Problem



- API calls from help documents are linked to their locations in the applications source code.
- Programmers can navigate directly to these locations and see how high-level concepts from queries are implemented in the source code.

Intuition For Ranking

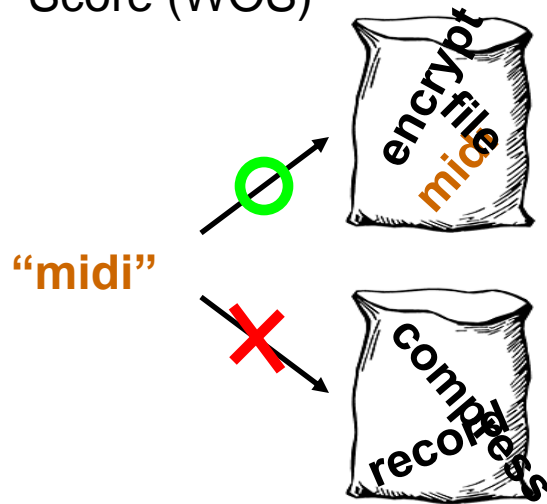


high performance

- More directly matched words -> higher ranking
- More API calls used -> higher ranking
 - Since API calls implement high-level concepts, more implemented concepts mean that the application is more relevant
- If API calls are connected using a dataflow -> higher ranking

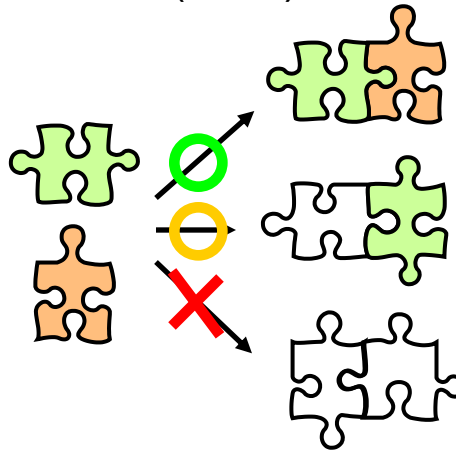
Three Ranking Scores

Word Occurrences
Score (WOS)



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

Relevant API Calls
Score (RAS)



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

Dataflow Connections
Score (DCS)

“record midi file”

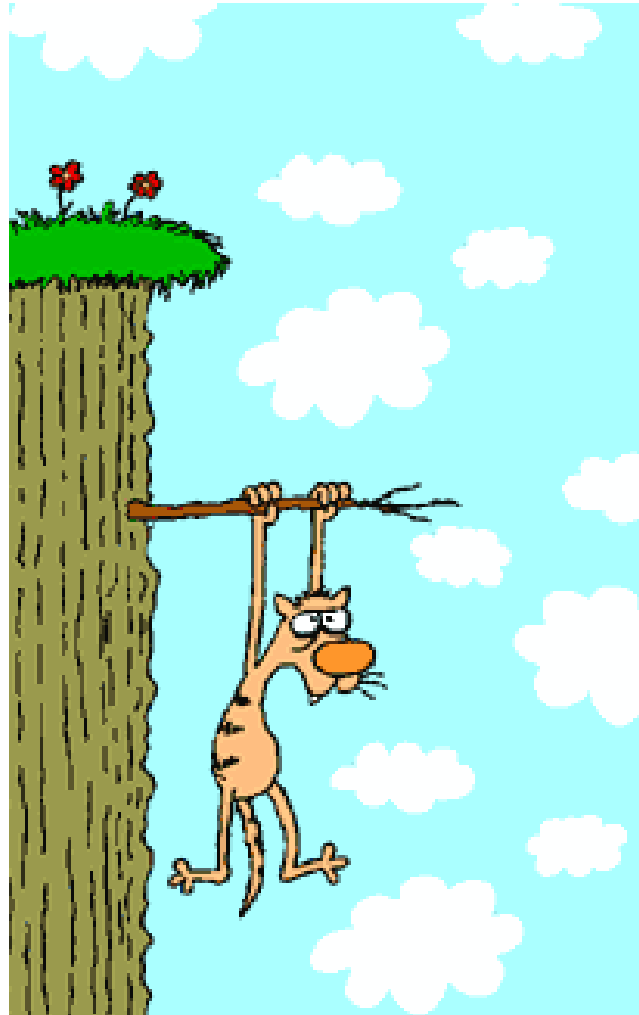
```
String dev = getDevice();
String buf[] =
A.readMidi(msg);
B.write(buf);
```

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

Hang In There, A Demo Is Coming

EXEMPLAR
Executable Examples - Review

high performance



To compare Exemplar and Sourceforge

- We need input from participants, there is no way to do it automatically

We follow a standard IR strategy for evaluation of search engine

- We use search engines that use equivalent large-scale code repositories

Structure of The Experiment

Participants were given tasks

- A short description of an application or some feature

Participants choose keywords that describe this task best

- Selecting keywords is **their** choice

Using search engine participants find and evaluate applications and rank them using their judgments

- Their evaluations are based on their confidence that they obtain by evaluating the source code of retrieved applications

Ranking



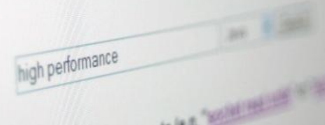
high performance

1. Completely irrelevant – there is absolutely nothing that you can use from this retrieved project, nothing in it is related to your keywords. The project may not even be uploaded to Sourceforge, only its description exists
2. Mostly irrelevant – only few remotely relevant code snippets or API calls in the project
3. Mostly relevant – a somewhat large number of relevant code snippets or API calls in the project
4. Highly relevant – you are confident that you can reuse code snippets or API calls in the project

Experimental Design and Results

Experiment	Group	Search Engine
1	Magenta	Exemplar with connectivity
	Green	Sourceforge
	Yellow	Exemplar with API calls, no connectivity
2	Magenta	Exemplar with API calls, no connectivity
	Green	Exemplar with connectivity
	Yellow	Sourceforge
3	Magenta	Sourceforge
	Green	Exemplar with API calls, no connectivity
	Yellow	Exemplar with connectivity

Thirty Nine Participants



- 26 participants are Accenture employees who work on consulting engagements as professional Java programmers for different client companies
- Remaining 13 participants are graduate students from the University of Illinois at Chicago who have at least six months of Java experience.
- 17 had programming experience with Java ranging from 1 to 3 years
- 22 participants have more than 3 years of Java experience
- 11 participants reported prior experience with Sourceforge
- 18 participants reported prior experience with other search engines
- 11 said that they never used code search engines
- 26 participants have bachelor degrees and 13 have master degrees in different technical disciplines.

Interesting Fact – The Cost of This Study



high performance

- Professional experienced programmers are very expensive, they charge more than \$50 per hour
- Accenture rate is \$150 per hour
 - $26 * 150 * 8 = \$31,200$
- Additional costs run for close to \$10K
 - Renting laptops with preinstalled images
 - Conference room with internet access
 - Various expenses
- Total cost is around \$40,000

Rejected Null Hypothesis

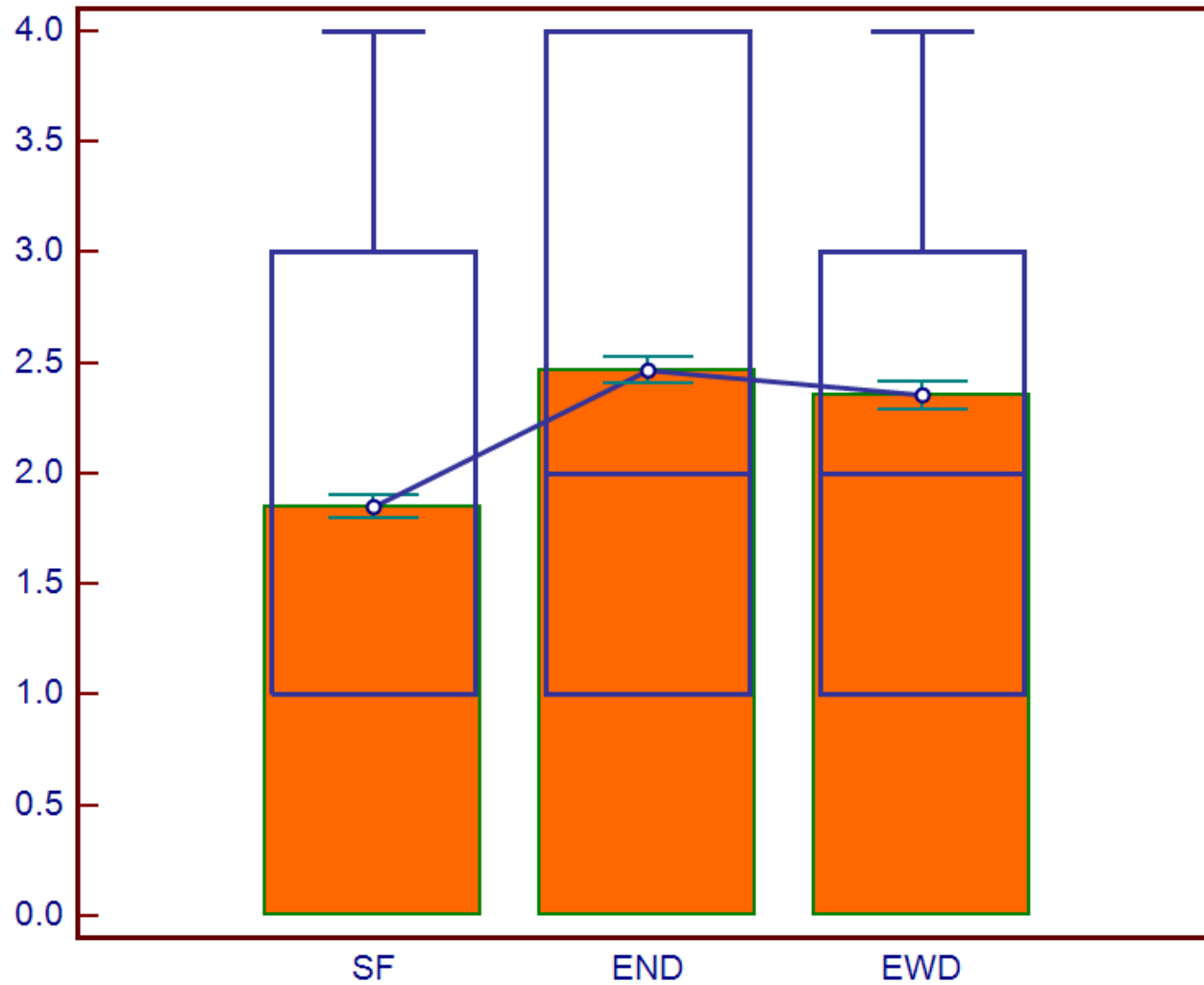
H_0

- The primary null hypothesis is that there is no difference in the numbers of Cs and Ps between participants who ranked results for Sourceforge versus Exemplar search engines.

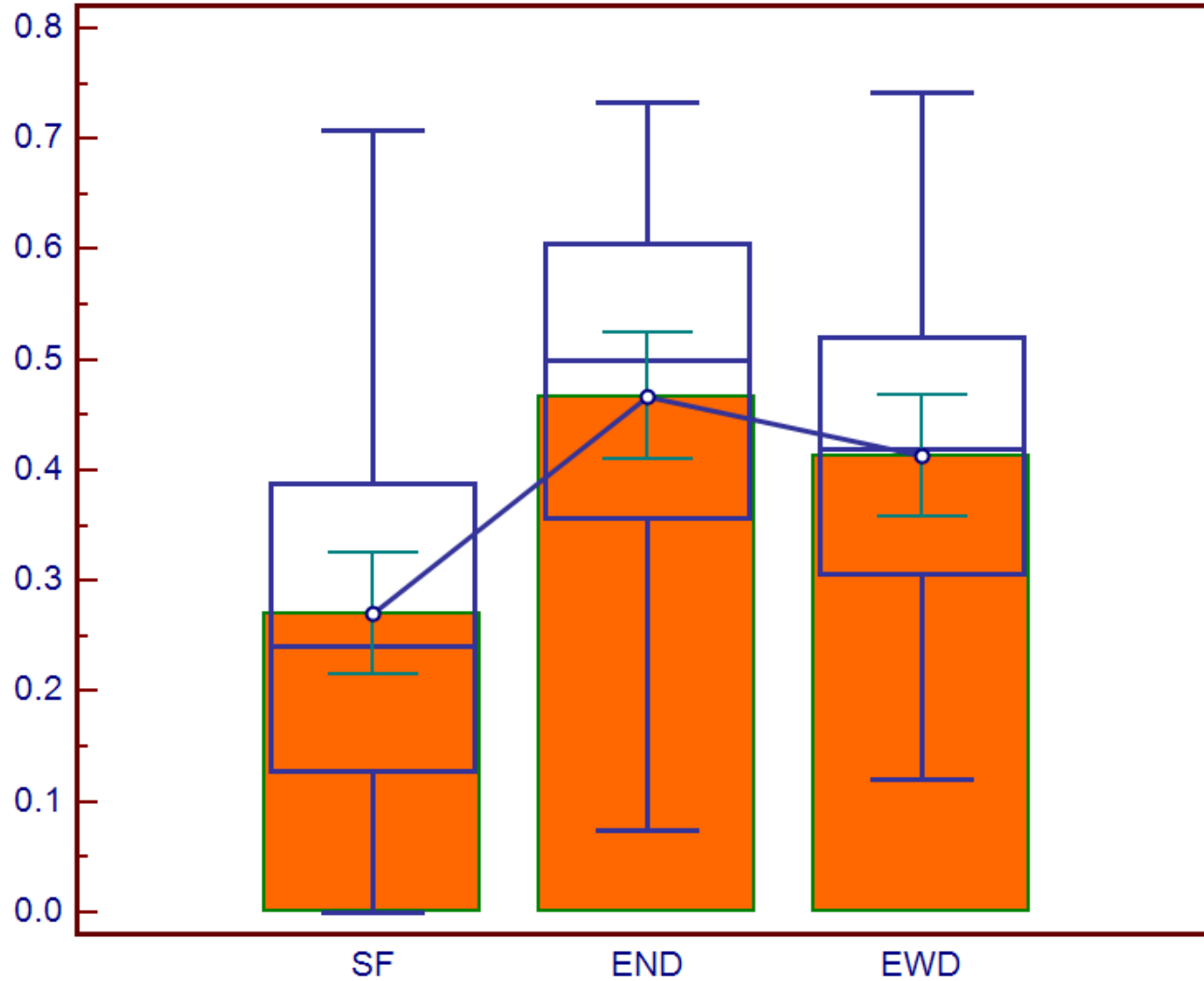
H_1

- An alternative hypothesis to H_0 is that there is statistically significant difference in the numbers of Cs and Ps between participants who ranked results for Sourceforge versus Exemplar search engines.

Rankings



Precision



Conclusions



high performance

- Exemplar is effective in the solution domain where it helps developers to find applications that contain relevant code fragments with API calls.
- Exemplar is available at **www.xemplar.org**
- Exemplar is currently used by different programmers from all over the world.

[Message All Members](#)[Promote Group with an Ad](#)[Edit Group Settings](#)[Edit Members](#)[Invite People to Join](#)[Create Group Event](#)[Leave Group](#)

Write something about Exemplar Users and Developers.

Information

Category:

Internet & Technology - Software

Description:

This is a group of professionals who use and work on Exemplar search engine. It is publicly and freely available at www.xemplar.org.

Privacy Type:

Open: All content is public.

Admins

■ [Mark Grechanik](#) (creator)

Officers

[Mark Grechanik](#)

Project Lead [remove]

[Qing Xie](#)

Project Lead

[Denys Poshyanyk](#) (William & Mary)

Project Lead

[Collin McMillan](#) (William & Mary)

A key contributor

[Chen Fu](#)

Project Lead

Members

6 of 36 members

[See All](#)

Exemplar Users and Developers

[Wall](#)[Info](#)[Discussions](#)[+](#)

Write something...

Attach:

[Share](#)

Mark Grechanik

[Can businesses be built around code search engines?](#)

Options
Remove

View Members

[Members](#)[Admins](#)[Not Yet Replied](#)[Blocked](#)[Requested](#)

Kevin Sullivan

[Make Admin](#)[×](#)

Alessio Di Stasio

Daniels Trading

[Make Admin](#)[×](#)

Vijay Dheeraj Reddy Mandadi

[Make Admin](#)[×](#)

Tao Xie

N.C. State

[Make Admin](#)[×](#)

Shyam Sunder Santoshi

[Make Admin](#)[×](#)

Palak Jain

[Make Admin](#)[×](#)[Close](#)

EXEMPLAR
Executable Examples Archive

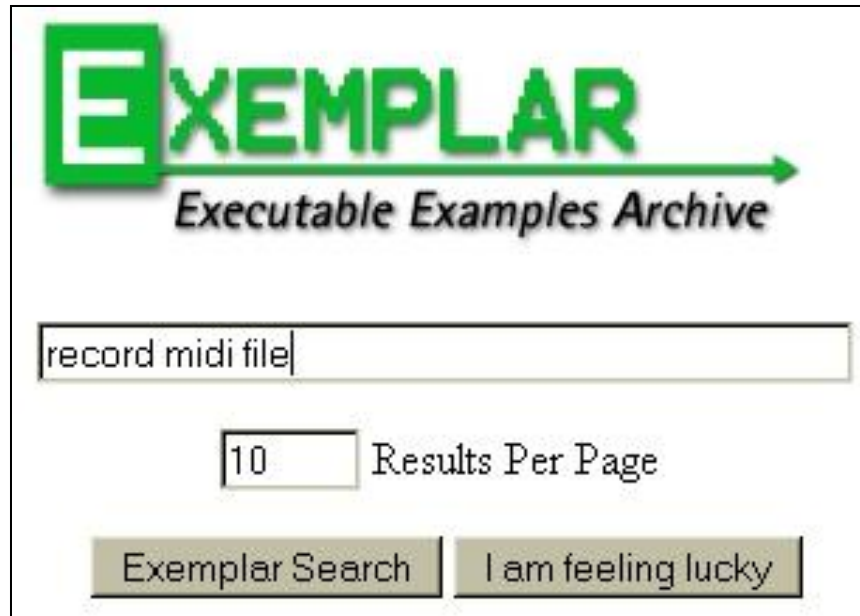
high performance



Thank you!

Questions?

The user enters a high-level query.



The image shows a web interface for the Exemplar Executable Examples Archive. At the top, there is a logo consisting of a green square with a white 'E' inside, followed by the word 'EXEMPLAR' in green capital letters. Below this, the text 'Executable Examples Archive' is written in a smaller, italicized font. Underneath the logo is a search input field containing the text 'record midi file'. Below the search field is a dropdown menu showing '10' and the text 'Results Per Page'. At the bottom, there are two buttons: 'Exemplar Search' and 'I am feeling lucky'.

EXEMPLAR
Executable Examples Archive

record midi file

10 Results Per Page

Exemplar Search 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). \r\n
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/javaz/sound/midi/MetaMessage.java	93	javax:sound:midi:MetaMessage:MetaMessage
0.3.0/tritonus-0.3.0.tar.gz/tritonus-0.3.0/src/javaz/sound/midi/Track.java	54	javax:sound:midi:MidiEvent:getTick
0.3.0/tritonus-0.3.0.tar.gz/tritonus-0.3.0/src/javaz/sound/midi/Track.java	54	javax:sound:midi:MidiEvent:MidiEvent
0.3.0/tritonus-0.3.0.tar.gz/tritonus-0.3.0/src/javaz/sound/midi/Track.java	98	javax:sound:midi:MidiEvent:getTick
0.3.0/tritonus-0.3.0.tar.gz/tritonus-0.3.0/src/javaz/sound/midi/Sequence.java	79	javax:sound:midi:Sequence:createTrack
0.3.0/tritonus-0.3.0.tar.gz/tritonus-0.3.0/src/javaz/sound/midi/MidiSystem.java	60	javax:sound:midi:MidiSystem:getMidiDeviceProviders