Creating and Evolving Software by Searching, Selecting and Synthesizing (S^3) Relevant Source Code

Denys Poshyvanyk, William and Mary

Mark Grechanik, Accenture Technology Labs &
University of Illinois, Chicago

How Many Open Source Applications Are There?

- Sourceforge.net reports that they host 180,000 projects as of August 1, 2008.
- There are dozens of other open source repositories containing tens of thousands of different applications.
- Companies have internal source control management systems containing hundreds of thousands of applications.

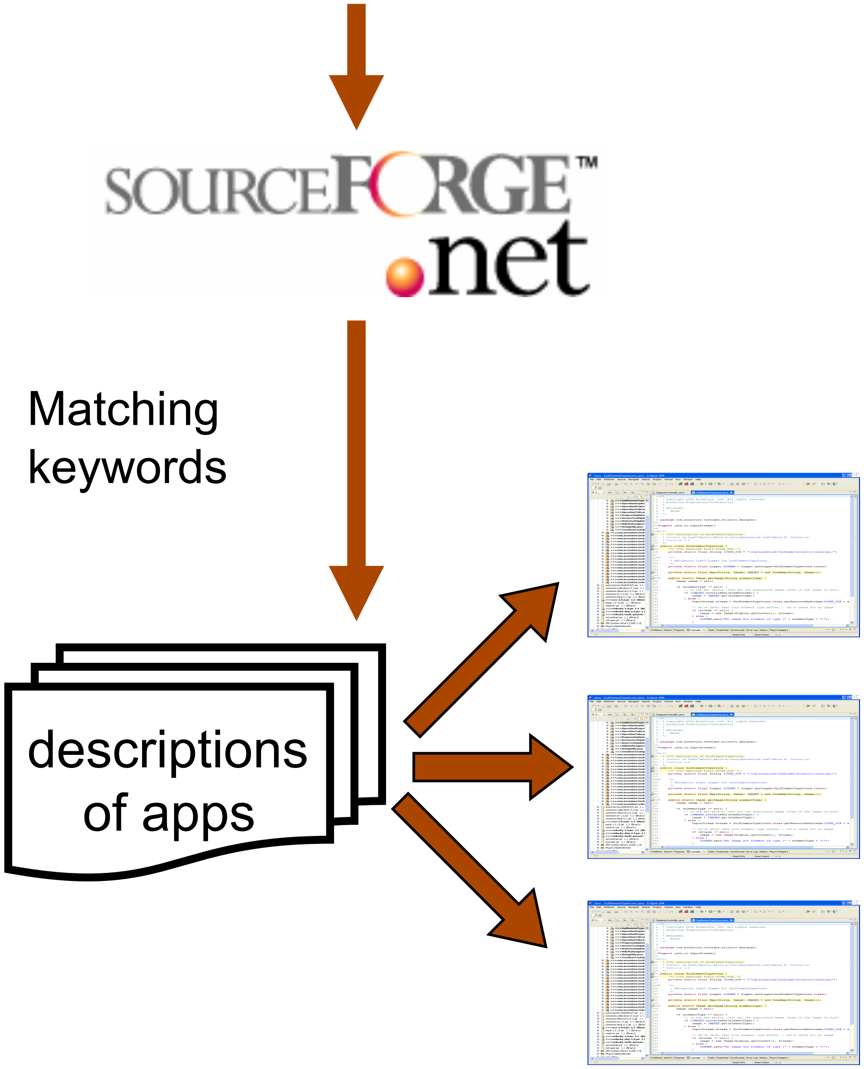
Problem

- Finding/checking existing software matching high-level user requirements
  - Would reduce the cost of many software projects
  - Would provide users with examples of different implementations

- Challenges:
  - Finding relevant applications is <u>difficult</u>
  - Evaluating retrieved applications is <u>difficult</u>

What Search Engines Do

“encrypt compress XML data”

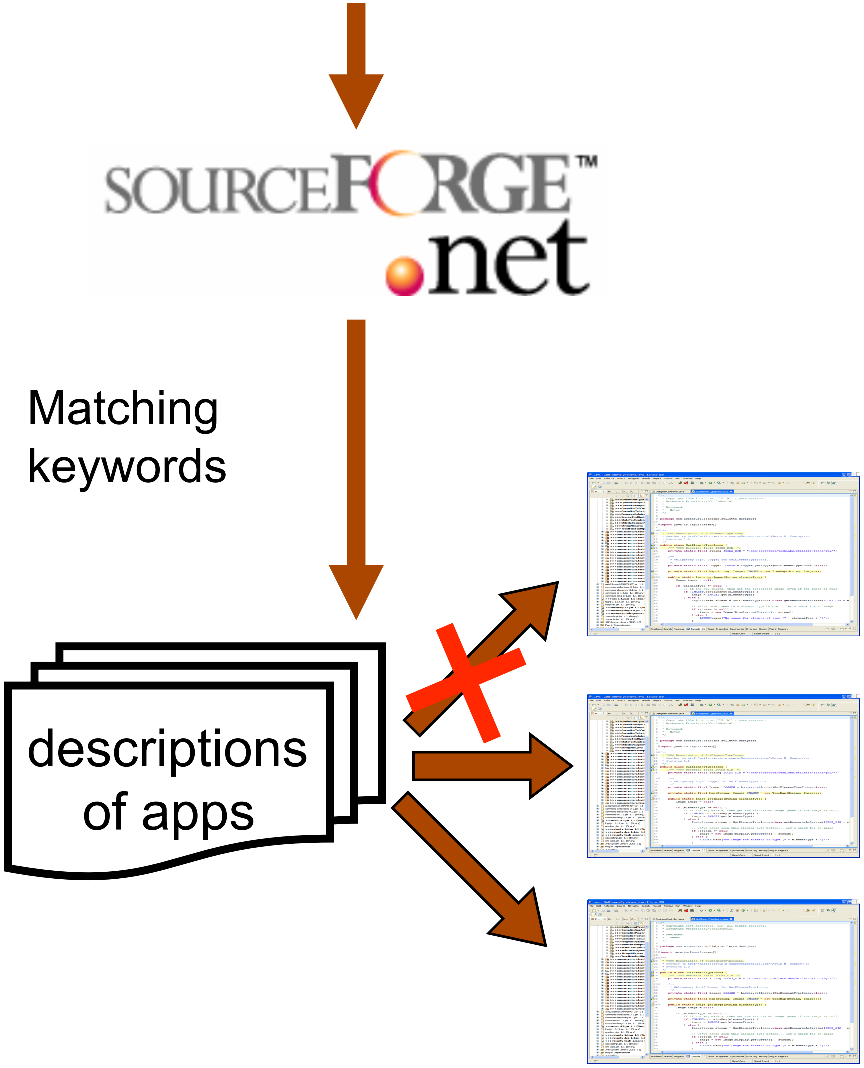


Matching keywords

descriptions of apps

What Search Engines Do

“encrypt compress XML data”



Matching keywords

descriptions of apps

Fundamental Problems

- Vocabulary problem
  - Mismatch between the high-level intent reflected in the descriptions of applications and their low-level implementation details
- Concept assignment problem

High level concept

“Send data”

Code snippet implementing “Send data”

```
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
```

- Many application repositories are polluted with poorly functioning projects

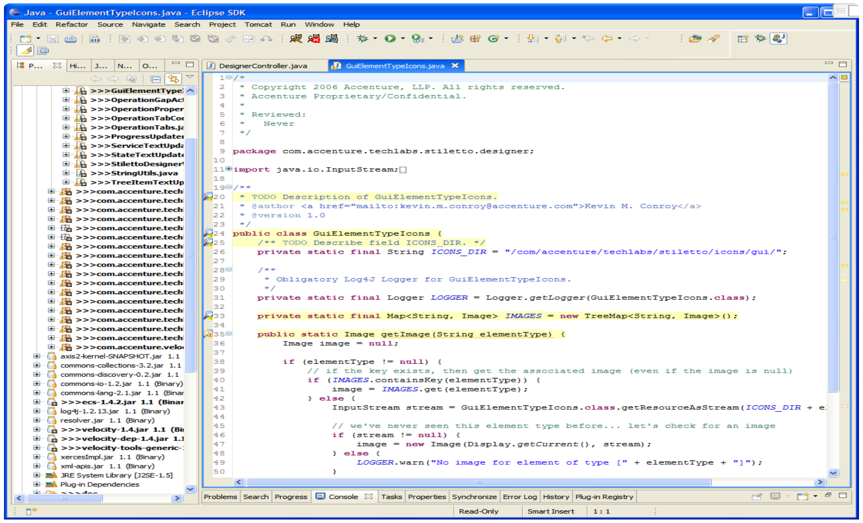
Working without a Tool

- Find relevant application(s)
- 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

Our Goal

"encrypt compress XML data"

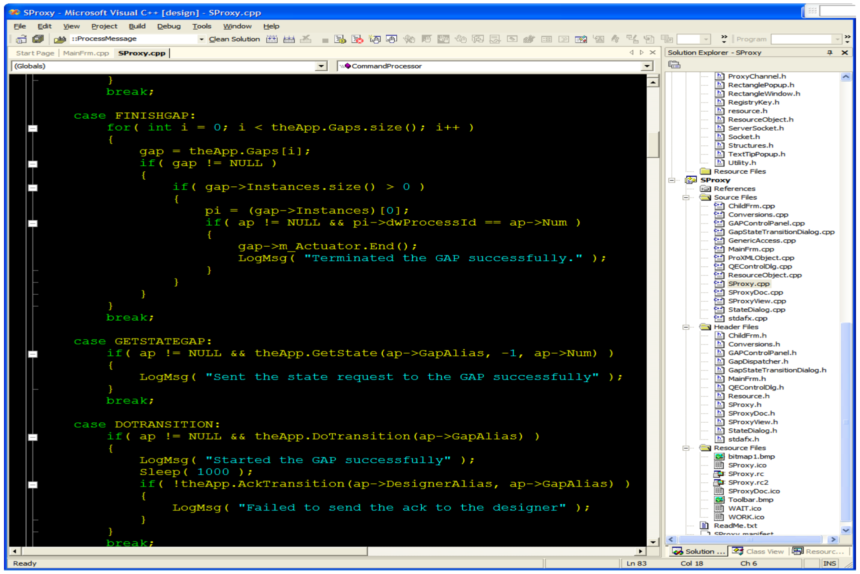
search



Java - GuiElementTypeIcons.java - Eclipse SDK
File Edit Refactor Source Navigate Search Project Tomcat Run Window Help
DesignerController.java
GuiElementTypeIcons.java

```java
/*
 * Copyright 2006 Accenture, LLP. All rights reserved.
 * Accenture Proprietary/Confidential.
 *
 * Reviewed:
 *   Never
 */

package com.accenture.techlabs.stiletto.designer;

import java.io.InputStream;

/**
 * TODO Description of GuiElementTypeIcons.
 * @author <a href="mailto:kevin.m.conroy@accenture.com">Kevin M. Conroy</a>
 * @version 1.0
 */
public class GuiElementTypeIcons {
    /** TODO Describe field ICONS_DIR. */
    private static final String ICONS_DIR = "/com/accenture/techlabs/stiletto/icons/gui/";

    /**
     * Obligatory Log4J Logger for GuiElementTypeIcons.
     */
    private static final Logger LOGGER = Logger.getLogger(GuiElementTypeIcons.class);

    private static final Map<String, Image> IMAGES = new TreeMap<String, Image>();

    public static Image getImage(String elementType) {
        Image image = null;

        if (elementType != null) {
            // if the key exists, then get the associated image (even if the image is null)
            if (IMAGES.containsKey(elementType)) {
                image = IMAGES.get(elementType);
            } else {
                InputStream stream = GuiElementTypeIcons.class.getResourceAsStream(ICONS_DIR + e

                // we've never seen this element type before... let's check for an image
                if (stream != null) {
                    image = new Image(Display.getCurrent(), stream);
                } else {
                    LOGGER.warn("No image for element of type [" + elementType + "]");
                }
```

Problems Search Progress Console Tasks Properties Synchronize Error Log History Plug-in Registry
Read-Only
Smart Insert
1 : 1

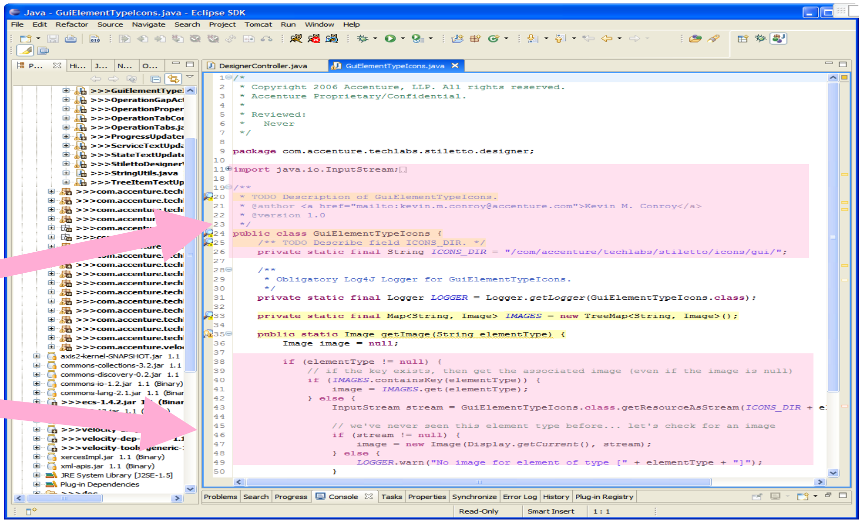


SProxy - Microsoft Visual C++ [design] - SProxy.cpp
File Edit View Project Build Debug Tools Window Help
::ProcessMessage
Clean Solution
Start Page
MainFrm.cpp
SProxy.cpp
(Globals)
CommandProcessor
Solution Explorer - SProxy

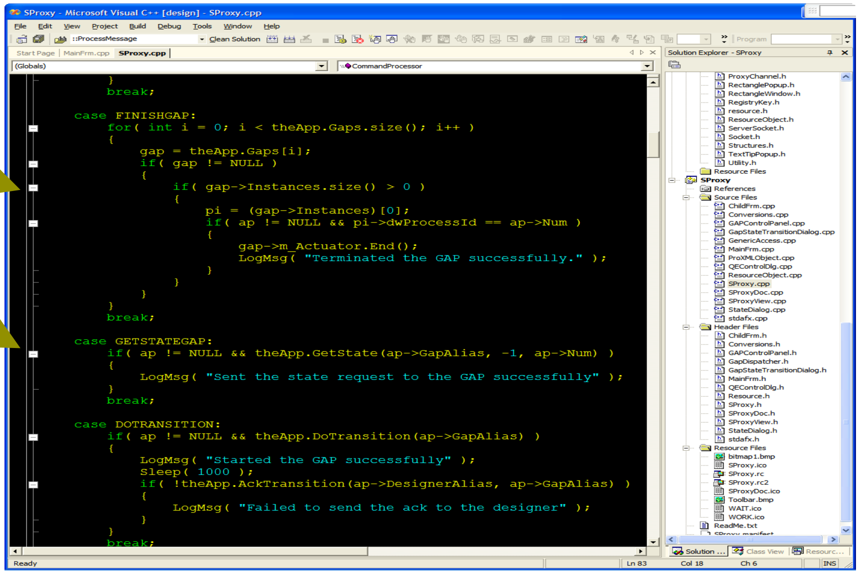
```cpp
        }
        break;

    case FINISHGAP:
        for( int i = 0; i < theApp.Gaps.size(); i++ )
        {
            gap = theApp.Gaps[i];
            if( gap != NULL )
            {
                if( gap->Instances.size() > 0 )
                {
                    pi = (gap->Instances)[0];
                    if( ap != NULL && pi->dwProcessId == ap->Num )
                    {
                        gap->m_Actuator.End();
                        LogMsg( "Terminated the GAP successfully." );
                    }
                }
            }
        }
        break;

    case GETSTATEGAP:
        if( ap != NULL && theApp.GetState(ap->GapAlias, -1, ap->Num) )
        {
            LogMsg( "Sent the state request to the GAP successfully" );
        }
        break;

    case DOTRANSITION:
        if( ap != NULL && theApp.DoTransition(ap->GapAlias) )
        {
            LogMsg( "Started the GAP successfully" );
            Sleep( 1000 );
            if( !theApp.AckTransition(ap->DesignerAlias, ap->GapAlias) )
            {
                LogMsg( "Failed to send the ack to the designer" );
            }
        }
        break;
```

Our Goal

“encrypt
compress
XML data”


Java - GuiElementTypeIcons.java - Eclipse SDK
File Edit Refactor Source Navigate Search Project Tomcat Run Window Help
DesignerController.java
GuiElementTypeIcons.java
Problems Search Progress Console Tasks Properties Synchronize Error Log History Plug-in Registry
Read-Only Smart Insert 1 : 1

```
/*
 * Copyright 2006 Accenture, LLP. All rights reserved.
 * Accenture Proprietary/Confidential.
 *
 * Reviewed:
 *   Never
 */

package com.accenture.techlabs.stiletto.designer;

import java.io.InputStream;

/**
 * TODO Description of GuiElementTypeIcons.
 * @author <a href="mailto:kevin.m.conroy@accenture.com">Kevin M. Conroy</a>
 * @version 1.0
 */
public class GuiElementTypeIcons {
    /** TODO Describe field ICONS_DIR. */
    private static final String ICONS_DIR = "/com/accenture/techlabs/stiletto/icons/gui/";

    /**
     * Obligatory Log4J Logger for GuiElementTypeIcons.
     */
    private static final Logger LOGGER = Logger.getLogger(GuiElementTypeIcons.class);

    private static final Map<String, Image> IMAGES = new TreeMap<String, Image>();

    public static Image getImage(String elementType) {
        Image image = null;

        if (elementType != null) {
            // if the key exists, then get the associated image (even if the image is null)
            if (IMAGES.containsKey(elementType)) {
                image = IMAGES.get(elementType);
            } else {
                InputStream stream = GuiElementTypeIcons.class.getResourceAsStream(ICONS_DIR + e

                // we've never seen this element type before... let's check for an image
                if (stream != null) {
                    image = new Image(Display.getCurrent(), stream);
                } else {
                    LOGGER.warn("No image for element of type [" + elementType + "]");
                }
```

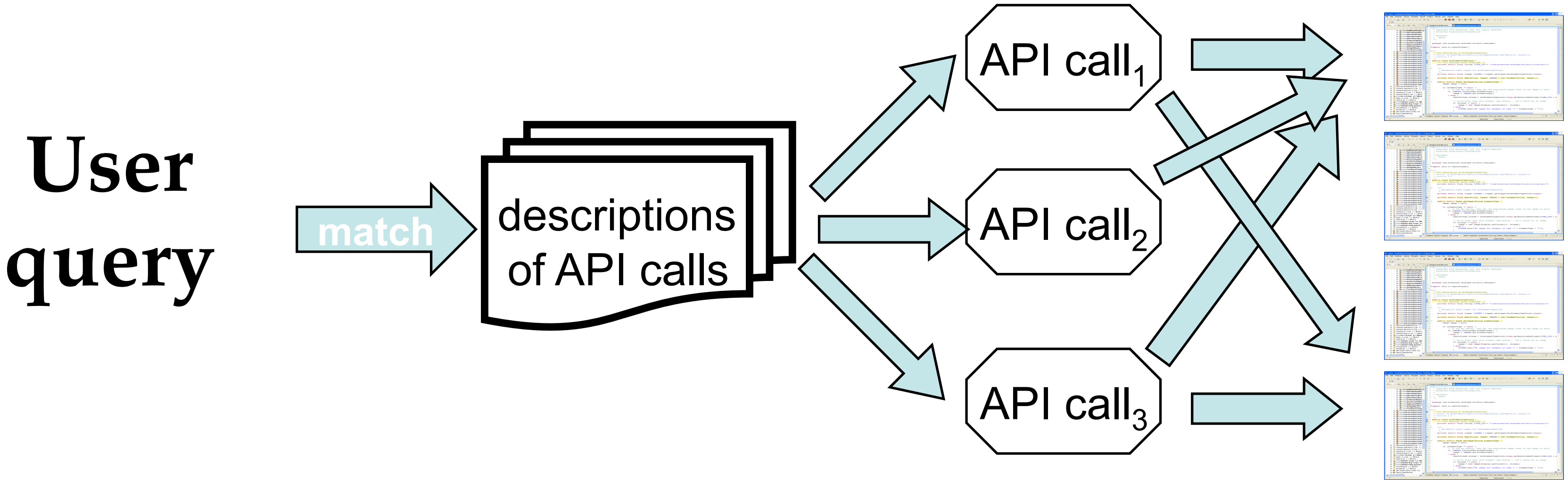

SProxy - Microsoft Visual C++ [design] - SProxy.cpp
File Edit View Project Build Debug Tools Window Help
::ProcessMessage
Clean Solution
Start Page MainFrm.cpp SProxy.cpp
(Globals)
CommandProcessor
Solution Explorer - SProxy
Ready Ln 83 Col 18 Ch 6 INS

```
            }
            break;

        case FINISHGAP:
            for( int i = 0; i < theApp.Gaps.size(); i++ )
            {
                gap = theApp.Gaps[i];
                if( gap != NULL )
                {
                    if( gap->Instances.size() > 0 )
                    {
                        pi = (gap->Instances)[0];
                        if( ap != NULL && pi->dwProcessId == ap->Num )
                        {
                            gap->m_Actuator.End();
                            LogMsg( "Terminated the GAP successfully." );
                        }
                    }
                }
            }
            break;

        case GETSTATEGAP:
            if( ap != NULL && theApp.GetState(ap->GapAlias, -1, ap->Num) )
            {
                LogMsg( "Sent the state request to the GAP successfully" );
            }
            break;

        case DOTRANSITION:
            if( ap != NULL && theApp.DoTransition(ap->GapAlias) )
            {
                LogMsg( "Started the GAP successfully" );
                Sleep( 1000 );
                if( !theApp.AckTransition(ap->DesignerAlias, ap->GapAlias) )
                {
                    LogMsg( "Failed to send the ack to the designer" );
                }
            }
            break;
```

Key observations

- While studying retrieved apps developers :
  - locate various ***API calls***
  - read information about these calls in help documents
- Help docs are supplied by the same vendors whose packages/APIs are used in software
- Programmers read and rely on these API docs
- Help docs are written and reviewed by many developers
- Help documents are usually more verbose and accurate than project descriptions

How S^3 System Works



User query

match

descriptions of API calls

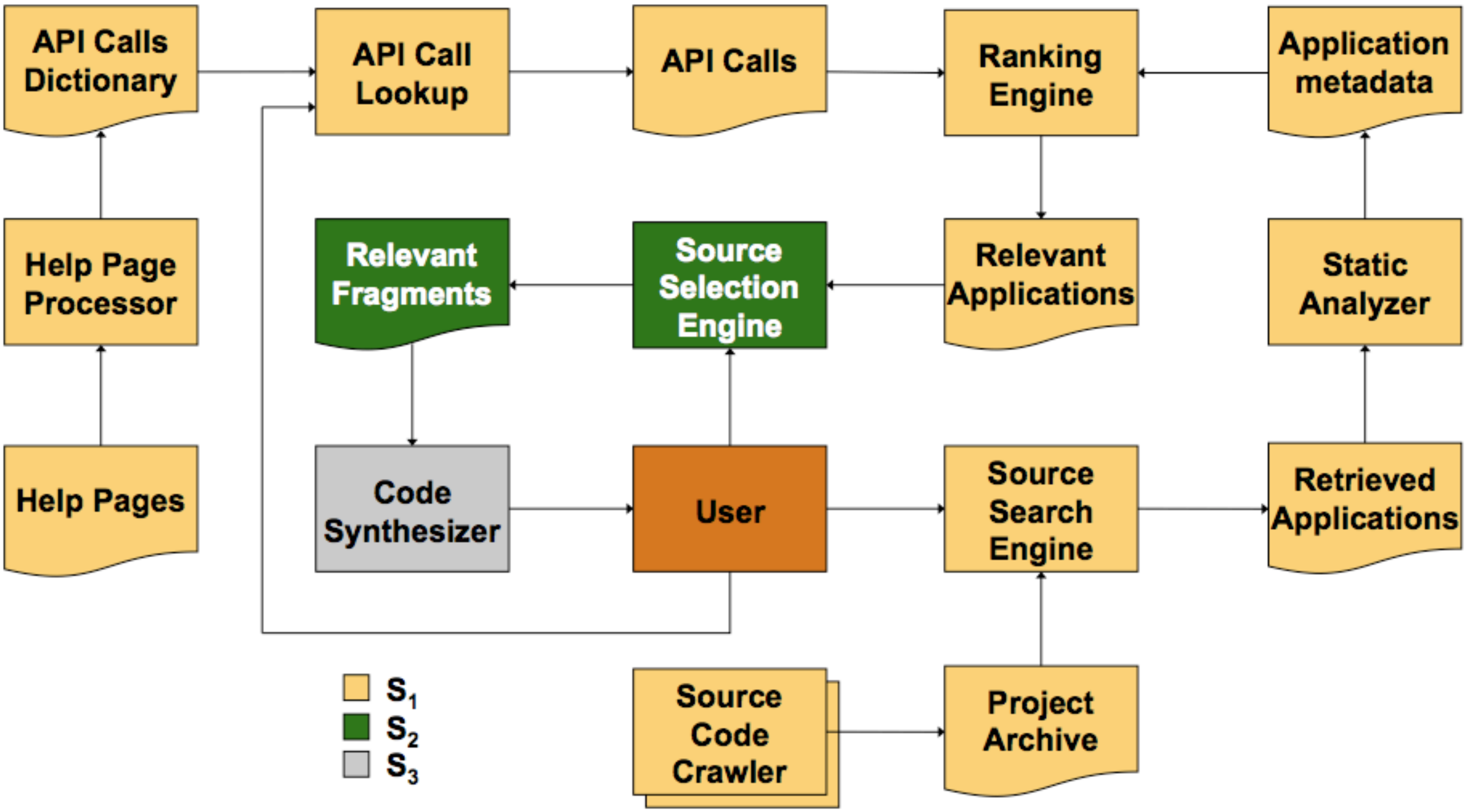
API call$_1$

API call$_2$

API call$_3$

- Automatically matching words in user queries against API help docs instead of:
  - searching in project descriptions;
  - searching in source code.
- S^3 uses help documents to produce a list of relevant API calls

S^3 Architecture



Current Status

- Restricting the scope to Java projects
- Challenges:
  - How to automatically locate and download the latest version of the software (e.g., from sourceforge)?
  - How to automatically locate the correct entry point (i.e., main) for static analysis?
  - How to reduce the time for the static analysis?
  - How and when to update API call dictionary?
  - Testing other ranking heuristics
- Evaluation is pending (some preliminary results at the poster session)

Related Work

- CodeFinder/Helgon
- ParseWeb
- CodeBroker
- Hipikat
- Automated Method Completion (AMC)
- Strathcona
- Prospector
- XSnippet
- Google code search, Krugle,…

Conclusions & Future Work

- S^3 recommends/checks relevant applications based on:
  - analysis of relevant API help documents;
  - analysis of actual API calls.
- Indexing available open-source projects and pre-computing data and control flow among API calls
- Analyzing multiple releases of the same project