Internet-Scale Code Search

Rosalva E. Gallardo-Valencia
University of California, Irvine
rgallard@ics.uci.edu

Susan Elliott Sim
University of California, Irvine
ses@ics.uci.edu

Abstract

Internet-Scale Code Search is the problem of finding source on the Internet. Developers are typically searching for code to reuse as-is on a project or as a reference example. This phenomenon has emerged due to the increasing availability and quality of open source and resources on the web. Solutions to this problem will involve more than the simple application of information retrieval techniques or a scaling-up of tools for code search. Instead, new, purpose-built solutions are needed that draw on results from these areas, as well as program comprehension and software reuse.

1. Introduction

Open source is the practice of distributing source code along with the executable code for a computer program. The increasing availability of high quality open source code on the Internet is changing the way software is being developed [9]. It has became commonplace to search the Internet for source code in the course of a software development project.

Developers are increasingly using an Opportunistic Software Systems Development (OSSD) approach to put together software pieces that they found. This approach is used to face the market demands of delivering software quickly and with more functionality [6]. Although, these software pieces provide functionality that programmers need to include in a system, often, they are unrelated and were not designed to work jointly.

Developers who are using these approaches search the Internet looking for open source to reuse in their projects. We will refer to this specific type of source code search as Internet-Scale Code Search. Locating the right component for as-is reuse or a reference example at the right time can have significant impact on how the project progresses.

The Internet-Scale Code Search has many similarities with other areas of research and we should build on their contributions. These areas include software reuse, code search, information retrieval, and program comprehension. We will discuss the similarities and differences with these areas.

In this paper, we argue that Internet-Scale Code Search is a new kind of problem. Not only is Internet-Scale Code Search more than the sum of the parts, different kinds of technological possibilities are available due to emerging computational practices.

2. What is Internet-Scale Code Search?

Internet-Scale Code Search is searching the Internet for source code to help solve a software development problem. Results from a web-based survey have shown that developers search for code on the Internet with the motivation of finding a piece of software to reuse or a reference example to use as a guide. The target piece of software varies on size ranging from a block (a few lines of code), a subsystem, and a system [11]. Some examples of search targets from a previous empirical study are summarized below.

Table 1. Examples of software pieces classified by motivation and target size.

	As-Is Reuse	**Reference Example**
Block	Code snippets, wrappers, parsers	To learn language syntax and idioms
Sub system	Algorithms, data structures, GUI widgets, libraries	To help in the implementation of algorithms, data structures, GUI widgets. To aid in the use of libraries
System	Stand-alone tools, ERP packages, DBMS	To get ideas about an existing similar system

In the cases where developers are searching for a component for as-is reuse, the search parameters are more tightly defined, and developers are most often looking for a piece of functionality, that is, a portion of

code that will perform a particular task. This type of search target was also evident in the studies by Chen et al. [1], Madanmohan and De' [4]. Developers preferred components that could be used as-is, with little or no modification. In fact, they avoided components that required an understanding of the inner workings.

Developers are also motivated to search for a reference example of how to use or do something. In other words, software developers are using the web as a giant desk reference manual. While this kind of knowledge reuse has been acknowledged in the software reuse literature, it has been overshadowed by as-is component reuse. Searches for reference examples are qualitatively different from those for reusable components. The underlying problem being solved is different and so too are the selection criteria.

Some tools that support Internet-Scale Code Search are available on the Web. These tools include Google Code Search[1], Koders[2], Krugle[3], and Sourcerer[4] among others. Although these tools already help developers to find open source, a better understanding of the challenges behind code search on the Internet can suggest improvements to these tools.

2.1. Motivating examples

Here, we present some motivating examples of Internet-Scale Code Searching. These are composite descriptions based on data collected in our earlier empirical study [11].

Waldo was writing a Java program to send out meeting notifications by email. His program needed to send notifications of meetings in participants' local time zones. To do this, he needed to use the Calendar classes, which have a complex interface and can be used in many different ways. Rather than reading the Javadocs, he searched the web for examples of how the classes were used. Waldo found a number of useful blog posts and tutorials that gave him the information that he was looking for.

The example above describes a developer looking for a reference example for a subsystem, i.e. the Calendar classes in Java. In such cases, general-purpose search engines, such as Google and Yahoo, do reasonably well. The code that Waldo found was surrounded by natural language explanations that matched his search keywords. It should also be noted that he was not looking for a program element or identifier.

Wenda was looking for an implementation of the Trie tree data structure in C. A Trie tree is an ordered tree data structure where the keys are strings. She started out using a general-purpose search engine, but got too many matches. She added "C" as a search term to reduce the number of matches, but this did not help at all. She tried some code-specific search engines, but 'c' appeared a lot, so she switched to filtering by programming language. Also, "trie" was a substring of retrieval, so these too were a false start. Wenda ultimately found what she needed by going to a site were developers share ideas and resources with each other, such as www.codeproject.com. Here, she found a number of annotated examples that she could reuse as-is in her project.

This second example depicts a developer looking for a subsystem-sized reusable component. She wanted source code that implemented a well-understood abstract data structure. The main goal behind these searches is that the source code is commonly available and saves time. This example illustrates ways in which both general-purpose and code-specific search engines fall short. "C" was not a good search term, because it is too short and too common. "Trie" and "tree" were not much better. Also, it was difficult to judge the suitability of the various matches returned for Wenda's project. While some of the code returned by the code search engines had good comments, they generally lacked instructions for (re)use. As well, extracting the code and incorporating into her project would have required a non-trivial amount of work. In the face of uncertainty regarding the costs and benefits of adapting unfamiliar code, the safest option is often to implement it yourself.

3. Comparison with code search

Code search typically occurs within an Integrated Development Environment while working on the source code for a single project. This activity is often done during the development and maintenance of software, and involves searching for specific program elements in a software project. Developers have mainly four motivations to search for pieces of software: defect repair, code reuse, program understanding, and impact analysis. The pieces of software they are looking for are declaration, definition, use, and all uses of functions, variables, and classes [8].

Internet-Scale Code Search involves looking in not just one project but in a great number of different open source projects. In addition, Internet-Scale Code

[1] http://www.google.com/codesearch/

[2] http://www.koders.com/

[3] http://www.krugle.com/

[4] http://sourcerer.ics.uci.edu/sourcerer/search/index.jsp

Search will also search for source code in other types of information besides to source code repositories. It also includes searching on web pages, forums, mailing lists, and other sources.

Internet-Scale Code Search is an activity that is not restricted to the maintenance of software; this type of search expands to different phases in the software development process, such as feasibility study, analysis, design, implementation, testing, and maintenance.

Internet-Scale Code Search will build on the contributions from research into code search activity and scale it where possible to the Internet. However, we believe the usefulness of searching for program elements or certain kinds of identifiers will have limited applicability. More often, developers are looking for functionality or knowledge, and not for where a method or variable is declared.

4. Comparison with information retrieval

The area of information retrieval focuses on finding material of an unstructured nature, usually text, that satisfies an information need from within large collections stored on computers [5]. Internet-Scale Code Search is different because the material that developers are searching for is source code, which is structured in nature due to the fact that it follows strict syntax rules specific to a programming language.

Information retrieval commonly allows keyword-based searches. However, developers are searching for source code in terms of features, functionality, and requirements; source code is written in a programming language, while search keywords are in natural language. The only place where natural language appears in source code is within comments. Consequently, searches for code tends to rely on the comments and the text surrounding an excerpt of source code. Blocks of code on web pages have more descriptive text around them than in a version control repository. For these reasons, general-purpose search engines work surprisingly well in code search. Still, there is room for improvement, as our second motivating example illustrates.

The effectiveness of conventional information retrieval techniques can be attributed to both human ingenuity and metadata. Developers often find creative ways to make use of the tools available. For instance, they use general-purpose search engines to find repositories or caches where they can search further. Metadata, we believe, will play a major role in the design and implementation of improved Internet-Scale Code Search engines.

5. Comparison with software reuse

Software reuse is the process of finding and using existing components or libraries in the creation of new software [3]. It is now common to create software by hacking, mashing, and gluing together existing open source code [2]. Although, software has not been built from scratch since function libraries were invented, the Internet, open source, and Opportunistic Software System Development have significantly increased the scope and scale of source code being reused.

The Internet-Scale Code Search process differs from software reuse, where the recommended process is to identify the requirements and use them to evaluate the suitability of candidate libraries. Instead, the requirements are defined iteratively based on the available functionality. This process more resembles engineering design than looking for a set of lost keys. The former process proceeds by optimizing constraints in a cost effective manner. The latter process seeks to find an object that is known to exist, is well defined, and can only be located in a limited number of places. In other words, software developers acquire an understanding of what they are looking for by searching.

Software reuse contributes knowledge about the different facets of reuse, such as substance, scope, technique, and products [7]. Internet-Scale Code Search can use this knowledge when developers are looking for open source code on the Internet opportunistically.

6. Comparison with program comprehension

Program comprehension research has focused on the cognitive theories that help us understand how programmers comprehend software in a single body of source code and on the tools to aid users in their comprehension tasks [10]. A key step in the Internet-Scale Code Search process is the evaluation of thousands of candidate matches that have been returned by a search engine in order to find the right piece of open source code to incorporate into a project. This evaluation requires developers to understand the source code, but in contrast with program comprehension, this evaluation involves discerning the characteristics of a candidate piece of code without becoming entangled in the internals.

In conventional program comprehension, developers use source code and documentation to understand the program [10]. We believe that program comprehension in Internet-Scale Code Search is very different. Developers tend not to look at the source

code when selecting a component for reuse, but rather, they rely on surface features and external information sources. Preliminary research has identified two kinds of judgments. Relevance judgments are made by software developers while identifying promising candidates among the available matches. These decisions are rapid, taking only a few seconds, and use relatively little information. Suitability judgments are made when determining whether a promising candidate is appropriate for the software project. These decisions are slower and typically involve a careful cost-benefit analysis. These judgments are made based on characteristics of the open source project, fellow users, price, terms of license, documentation, and functionality.

7. Summary

In this paper, we argued that Internet-Scale Code Search is a novel problem, in need of novel solutions. Although, it is similar to a number of existing problems, research is needed to combine and create research contributions. This emerging field is similar to software reuse, source code searching, information retrieval, and program comprehension.

Internet-Scale Code Search is similar to software reuse, because developers are often looking for code to reuse as-is on their projects. But they also look for code to use as reference examples. Source code searching and Internet-Scale Code Search have in common the fact that developers are searching for source code. But, in the former developers are typically looking for program elements within a single project. In the latter, they look in a great number of open source projects on the Internet. Like information retrieval, Internet-Scale Code Search involves searching large collections. An important difference is that developers are searching for source code and not for natural language text in unstructured documents. Some program comprehension is needed during Internet-scale code search, but not the kind normally performed during software maintenance. During Internet-Scale Code Search, developers need to evaluate thousands of candidate matches using superficial information and from different sources.

In summary, Internet-Scale Code Search is a new problem that has arisen as a result of evolving technologies and software development practices. The solution to this problem will require similar innovation and creativity to both use existing results and to create know-how.

8. References

[1] Weibing Chen, Jingyue Li, Jianqiang Ma, Reidar Conradi, Junzhong Ji, and Chunnian Liu. “An empirical study of software development with open source components in the Chinese software industry.” *Software Process: Improvement and Practic*e, 13:89–100, January 2008.

[2] Bjorn Hartmann, Scott Doorley, and Scott R. Klemmer. “Hacking, mashing, gluing: A study of opportunistic design.” *Technical Report CSTR 2006-14*, Department of Computer Science, Stanford University, September 2006.

[3] C. W. Krueger. “Software Reuse.” *ACM Computing Surveys*, 24(2):131-184, June 1992.

[4] T.R. Madanmohan and Rahul De’. “Open source reuse in commercial firms.” *IEEE Software*, 21(6): 62–69, 2004.

[5] C. Manning, P. Raghavan, and H. Schutze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[6] Cornelius Ncube, Patricia Oberndorf, Anatol W. Kark, "Opportunistic Software Systems Development: Making Systems from What's Available," *IEEE Software*, 25 (6): 38-41, Nov./Dec. 2008.

[7] Johannes Sametinger. *Software Engineering with Reusable Components*. Springer, New York, 1997.

[8] S. E. Sim, C. L. A. Clarke, and R. C. Holt. “Archetypal source code searches: A survey of software developers and maintainers.” *In Proceedings of the 6th International Workshop on Program Comprehension*, page 180, Los Alamitos, CA, 1998. IEEE Computer Society.

[9] Diomidis Spinellis and Clemens Szyperski. “Guest editors’ introduction: How is open source affecting software development?” *IEEE Software*, 21(1):28–33, 2004.

[10] Margaret-Anne D. Storey. “Theories, tools and research methods in program comprehension: past, present and future.” *Software Quality Journal*, 14(3):187–208, 2006.

[11] Medha Umarji, Susan Elliott Sim, and Cristina V. Lopes. “Archetypal internet-scale source code searching.” *In Barbara Russo*, editor, OSS, page 7, New York, NY, 2008. Springer.