Data Article

Software reusability dataset based on static analysis metrics and reuse rate information

Michail D. Papamichail*, Themistoklis Diamantopoulos, Andreas L. Symeonidis

Intelligent Systems and Software Engineering Laboratory, Electrical and Computer Engineering Dept., Aristotle University of Thessaloniki, Greece

ARTICLE INFO

Article history:
Received 26 September 2019
Received in revised form 8 October 2019
Accepted 14 October 2019
Available online 19 October 2019

Keywords:
Reusability assessment
Code reuse
Static analysis metrics
Developer-perceived reusability

ABSTRACT

The widely adopted component-based development paradigm considers the reuse of proper software components as a primary criterion for successful software development. As a result, various research efforts are directed towards evaluating the extent to which a software component is reusable. Prior efforts follow expert-based approaches, however the continuously increasing open-source software initiative allows the introduction of data-driven alternatives. In this context we have generated a dataset that harnesses information residing in online code hosting facilities and introduces the actual reuse rate of software components as a measure of their reusability. To do so, we have analyzed the most popular projects included in the maven registry and have computed a large number of static analysis metrics at both class and package levels using SourceMeter tool [2] that quantify six major source code properties: complexity, cohesion, coupling, inheritance, documentation and size. For these projects we additionally computed their reuse rate using our self-developed code search engine, AGORA [5]. The generated dataset contains analysis information regarding more than 24,000 classes and 2000 packages, and can, thus, be used as the information basis towards the design and development of data-driven reusability evaluation methodologies. The dataset is related to the research article

DOI of original article: https://doi.org/10.1016/j.jss.2019.110423.

* Corresponding author.
E-mail address: mpapamic@issel.ee.auth.gr (M.D. Papamichail).

https://doi.org/10.1016/j.dib.2019.104687
2352-3409/© 2019 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
entitled "Measuring the Reusability of Software Components using Static Analysis Metrics and Reuse Rate Information" [1].
© 2019 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

1. Data

The provided dataset contains three files: projects_information.csv, classes.csv, and packages.csv. The first file (projects_information.csv) contains general information (artifact url, github repository url,
| #  | Maven Artifact                                                                 | Classes | Packages |
|----|--------------------------------------------------------------------------------|---------|----------|
| 1  | http://mvnrepository.com/artifact/ch.qos.logback/logback-classic              | 145     | 30       |
| 2  | http://mvnrepository.com/artifact/com.android.support/appcompat-v7           | 299     | 13       |
| 3  | http://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-annotations | 18      | 5        |
| 4  | http://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-databind | 535     | 24       |
| 5  | http://mvnrepository.com/artifact/com.google.code.gson/gson                   | 158     | 11       |
| 6  | http://mvnrepository.com/artifact/com.google.guava/guava                      | 1701    | 24       |
| 7  | http://mvnrepository.com/artifact/com.google.gwt/gwt-user                     | 4177    | 242      |
| 8  | http://mvnrepository.com/artifact/commons-beanutils/commons-beanutils         | 127     | 9        |
| 9  | http://mvnrepository.com/artifact/commons-codex/commons-codex                  | 83      | 11       |
| 10 | http://mvnrepository.com/artifact/commons-collections/commons-collections       | 467     | 23       |
| 11 | http://mvnrepository.com/artifact/commons-logging/commons-logging              | 27      | 6        |
| 12 | http://mvnrepository.com/artifact/dom4j/dom4j                                 | 160     | 16       |
| 13 | http://mvnrepository.com/artifact/javax.mail/mail                              | 120     | 7        |
| 14 | http://mvnrepository.com/artifact/javax.servlet/servlet-api                   | 28      | 6        |
| 15 | http://mvnrepository.com/artifact/joda-time/joda-time                          | 220     | 10       |
| 16 | http://mvnrepository.com/artifact/junit/junit                                 | 272     | 35       |
| 17 | http://mvnrepository.com/artifact/joda-time/joda-time                          | 181     | 18       |
| 18 | http://mvnrepository.com/artifact/org.apache.commons/commons-lang3            | 235     | 17       |
| 19 | http://mvnrepository.com/artifact/org.apache.httpcomponents/httpclient        | 390     | 31       |
| 20 | http://mvnrepository.com/artifact/org.apache.maven/maven-core                 | 281     | 65       |
| 21 | http://mvnrepository.com/artifact/org.apache.maven/maven-plugin-api           | 16      | 9        |
| 22 | http://mvnrepository.com/artifact/org.clojure/clojure                         | 368     | 7        |
| 23 | http://mvnrepository.com/artifact/org.clojure/clojure/script                  | 0       | 1        |
| 24 | http://mvnrepository.com/artifact/org.clojure/tools.nrepl                     | 4       | 4        |
| 25 | http://mvnrepository.com/artifact/org.codehaus.jackson/jackson-core-asl       | 89      | 15       |
| 26 | http://mvnrepository.com/artifact/org.codehaus.jackson/jackson-mapper-asl     | 446     | 22       |
| 27 | http://mvnrepository.com/artifact/org.easymock/easymock                       | 88      | 5        |
| 28 | http://mvnrepository.com/artifact/org.eclipse.jetty/jetty-server             | 215     | 11       |
| 29 | http://mvnrepository.com/artifact/org.hibernate/hibernate-core                | 41      | 5        |
| 30 | http://mvnrepository.com/artifact/org.hibernate/hibernate-entitymanager       | 57      | 11       |
| 31 | http://mvnrepository.com/artifact/org.hibernate/hibernate-validator           | 3212    | 331      |
| 32 | http://mvnrepository.com/artifact/org.javassist/javassist                    | 389     | 18       |
| 33 | http://mvnrepository.com/artifact/org.junit/junit                              | 400     | 68       |
| 34 | http://mvnrepository.com/artifact/org.maven/maven-templates                   | 400     | 68       |
| 35 | http://mvnrepository.com/artifact/org.mockito/mockito                          | 52      | 23       |
| 36 | http://mvnrepository.com/artifact/org.mockito/mockito-module-junit4           | 21      | 8        |
| 37 | http://mvnrepository.com/artifact/org.projectlombok/lombok                    | 357     | 23       |
| 38 | http://mvnrepository.com/artifact/org.reflects/reflections                    | 83      | 8        |
| 39 | http://mvnrepository.com/artifact/org.scalajr/scalajr-test-interface         | 0       | 1        |
| 40 | http://mvnrepository.com/artifact/org.scalacheck/scalacheck                  | 0       | 1        |
| 41 | http://mvnrepository.com/artifact/org.scalatest/scalatest                    | 0       | 1        |
| 42 | http://mvnrepository.com/artifact/org.scalacoverage/scalac-coverage-plugin    | 0       | 1        |
| 43 | http://mvnrepository.com/artifact/org.scalacoverage/scalac-coverage-runtime   | 0       | 1        |
| 44 | http://mvnrepository.com/artifact/org.slf4j/jcl-over-slf4j                    | 8       | 6        |
| 45 | http://mvnrepository.com/artifact/org.slf4j/jul-to-slf4j                      | 1       | 4        |
| 46 | http://mvnrepository.com/artifact/org.slf4j/slf4j-api                         | 25      | 7        |
| 47 | http://mvnrepository.com/artifact/org.slf4j/slf4j-simple                     | 6       | 4        |
| 48 | http://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-test | 95   | 21       |
| 49 | http://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-web | 470  | 77       |
| 50 | http://mvnrepository.com/artifact/org.springframework/spring-aop             | 205     | 21       |
| 51 | http://mvnrepository.com/artifact/org.springframework/spring-beans           | 278     | 16       |
| 52 | http://mvnrepository.com/artifact/org.springframework/spring-context         | 558     | 69       |
| 53 | http://mvnrepository.com/artifact/org.springframework/spring-context-support  | 99      | 17       |
| 54 | http://mvnrepository.com/artifact/org.springframework/spring-core            | 463     | 34       |
| 55 | http://mvnrepository.com/artifact/org.springframework/spring-jdbc            | 208     | 20       |

(continued on next page)
source code folder, number of classes and number of packages) regarding the analyzed maven projects in order to enable on-demand dataset creation and thus facilitate analyses that target projects with certain characteristics in terms of size. Table 1 presents the information included in projects_information.csv. Files classes.csv, and packages.csv contain the values of the computed static analysis metrics regarding all classes and packages included in the aforementioned projects along with their reuse rate information. The reuse rate information is an integer value computed for each component (using the API provided by AGORA [5]) that corresponds to its number of reuse occurrences. A sample record of classes.csv file and packages.csv file is presented in Table 2. In addition, each record included in classes.csv, and packages.csv files contains the project of the respective component along with its long name and name. At this point, it is worth noting that the dashes ("-".) included in Table 2 denote that a certain attribute is not computed for the respective level of granularity.

The description of the aforementioned computed static analysis metrics along with their associated source code property and their level of computation can be found at Table 2 of [1].

### Table 2
Example Record of classes.csv and packages.csv

| Attribute   | Value (classes.csv) | Value (packages.csv) |
|-------------|---------------------|----------------------|
| Project     | ch.qos.logback/logback-classic | ch.qos.logback/logback-classic |
| LongName    | ch.qos.logback.classic.AsyncAppender | ch.qos.logback.classic.boolex boolex |
| Name        | AsyncAppender       | -                    |
| LCOM5       | 2                   | -                    |
| NL          | 1                   | -                    |
| NLE         | 1                   | -                    |
| WMC         | 5                   | -                    |
| CBO         | 2                   | -                    |
| CBOI        | 0                   | -                    |
| NII         | 0                   | -                    |
| NOI         | 4                   | -                    |
| RFC         | 8                   | -                    |
| AD          | 0.33                | 0.625                |
| CD          | 0.43                | 0.350649             |
| TCD         | 0.43                | 0.350649             |
| CLOC        | 14                  | 81                   |
| TCLOC       | 14                  | 81                   |
| DLOC        | 14                  | -                    |
| PDA         | 1                   | 5                    |
| DIT         | 0                   | -                    |
| LLOC        | 18                  | 150                  |
| LOC         | 29                  | 275                  |
| TLOC        | 29                  | 275                  |
| NG          | 0                   | 4                    |
| TLLOC       | 18                  | 150                  |
| TNA         | 1                   | 4                    |
| TNG         | 0                   | 4                    |
| TNM         | 4                   | 8                    |
| TNOS        | 7                   | 59                   |
| TNPM        | 2                   | 4                    |
| NCL         | -                   | 3                    |
| ReuseRate   | 6                   | 4                    |
2. Experimental design, materials, and methods

The dataset construction process involves the following distinct steps:

- The first step involves the selection of the projects to be analyzed. Towards this direction, and in an effort to create a dataset that can be used to evaluate the reusability of software components, we retrieved information regarding the 100 most popular projects included in the maven registry. The list of the most popular maven projects is available through the website of the maven central repository [3].

- After having selected the projects to be analyzed, the next step involves acquiring their source code. Given that maven does not directly provide the source code of the projects, we manually created the mappings between the popular projects and the associated GitHub repository. During this step and in an effort to maintain the purity of the dataset, we filtered out the projects that could not be associated with a certain repository. As a result, we resorted to analysing 65 projects.

- Using the aforementioned mappings, the next step involves analyzing the source code. Upon downloading the source code of each project from GitHub, we perform static analysis using the SourceMeter static analysis tool [2], which enables the computation of a series of metrics that quantify six source code properties: complexity, cohesion, coupling, inheritance, documentation and size. The analysis was performed in both class and package levels. The complete list of metrics along with their associated source code property and their level of computation are shown in Ref. [1].

- Finally, we use our self-developed code search engine AGORA [5], in order to calculate the reuse rate of all components (classes and packages) included in the dataset, which depends on the analysis of import statements. AGORA is a powerful source code index, which embeds syntax aware capabilities and has an open API that allows performing queries and easily integrating it with other tools. The index of AGORA is populated with approximately 3000 of the most popular Java repositories of Github, where popularity is determined by the number of stargazers. An example query that can be used in order to find all the possible declarations of a sample component is demonstrated in Figure 1 of [1].

Acknowledgments

This research has been co-financed by the European Regional Development Fund of the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH – CREATE – INNOVATE (project code: T1EDK-02347).

Conflict of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Supplementary data

Supplementary data related to this article can be found at https://doi.org/10.1016/j.dib.2019.104687.

References

[1] D. Michail, Papamichail, Themistoklis Diamantopoulos, and Andreas L. Symeonidis. Measuring the reusability of software components using static analysis metrics and reuse rate information, J. Syst. Softw. (2019), https://doi.org/10.1016/j.jss.2019.110423.
[2] SourceMeter. https://www.sourcemeter.com/, 2019. (Accessed 24 September 2019).
[3] MvnRepository. https://mvnrepository.com/popular, 2019. (Accessed 24 September 2019).
[4] GitHub. https://github.com/, 2019. (Accessed 24 September 2019).
[5] AGORA Code Search Engine. http://agora.ee.auth.gr, 2019. (Accessed 24 September 2019).