Visualization of Contributions to Open-Source Projects

Andreas Schreiber
German Aerospace Center (DLR),
Institute for Software Technology
Köln, Germany
andreas.schreiber@dlr.de

ABSTRACT

We analyze visually, to what extend team members and external developers contribute to open-source projects to have a high-level impression about collaboration in that projects. For that, we record the provenance of the development process and draw the resulting property graph. Our graph drawings show, which developers are jointly changed the same files what we apply to Germany's COVID-19 exposure notification app 'Corona-Warn-App'.

CCS CONCEPTS

• **Human-centered computing** → **Graph drawings**; • **Software and its engineering** → **Open source model**; **Programming teams**.

KEYWORDS

graph visualization, software visualization, provenance, open source

ACM Reference Format:
Andreas Schreiber. 2020. Visualization of Contributions to Open-Source Projects. In *The 13th International Symposium on Visual Information Communication and Interaction (VINCI 2020), December 8–10, 2020, Eindhoven, Netherlands.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3430036.3430057

1 INTRODUCTION

A transparent and traceable team composition and development process is one of the advantages of the open-source model. Understanding the characteristics of open-source projects, where developers with different roles work together, is an important question for projects with high public interests.

The COVID-19 pandemic raises challenges for software developers to help to fight the pandemic with software systems, which must be developed under time pressure. For example, mobile apps for *contact tracing* of infected persons such as Germany's exposure notification app *Corona-Warn-App* (CWA).

To analyze, to what extend team members and external contributors contributed to CWA on GitHub, we record and store the *provenance of the software development process* [8] according to the provenance data model PROV [5] in a graph database (Section 2). We query a sub-graph for visualization with graph drawing (Section 3)—which we apply to the CWA (Section 4).

VINCI 2020, December 8–10, 2020, Eindhoven, Netherlands

ACM ISBN 978-1-4503-8750-7/20/12.
https://doi.org/10.1145/3430036.3430057

2 PROVENANCE OF SOFTWARE DEVELOPMENT PROCESSES

We extract provenance of a software development process from repositories with Git2PROV [2, 7] and store it as a property graph in the graph database NEO4J for further analysis [6].

We use the graph to answer the question: "*Which files have commits by team members as well as external contributors?*" by adding information about the contributors roles as additional edges using database queries with CYPHER; the role of each contributor, which is necessary to construct the proper CYPHER, is requested from GITHUB via their API. We make the assumption, that members of the GITHUB organization `corona-warn-app` are dedicated team members. Then we query for files, where team members and external contributor made changes at any of the files revisions and visualize the result (Figure 1; see Section 3).

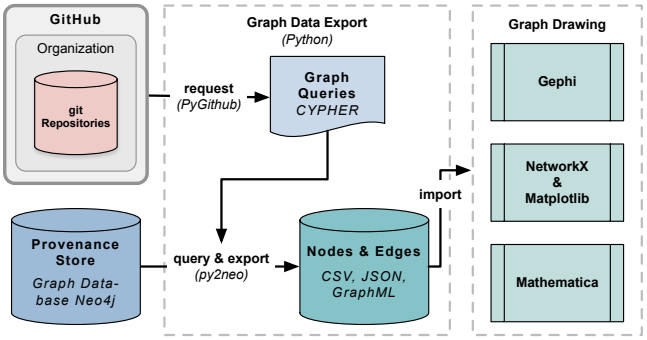

Figure 1: Querying and exporting graph data for visualization.

3 GRAPH VISUALIZATION

We visualize parts of the property graph with GEPHI [1]. During querying and exporting for visualization, we map the property graph as follows: files (i.e., PROV *entities*) and contributors (i.e., PROV *agents*) become graph nodes with two distinct colors and contributions become edges, which color depends on their property *role* (i.e., "*team member*" or "*external contributor*").

For the *coloring*, we choose distinct colors from two different qualitative color schemes generated by COLORBREWER [3]. The *size of nodes* are proportional to their degree. In our current approach, we generate two drawings for each project; one where we scale the node sizes according to the *in-degree* of file nodes and a second one where we scale according to the *out-degree* of contributors. For the *layout* we use the *ForceAtlas2* algorithm [4].