

Enriching Source Code by Empirical Metadata

Karol Rástočný
Slovak University of Technology in Bratislava
Ilkovičova 2, 842 16 Bratislava, Slovakia
karol.rastocny@stuba.sk

Mária Bieliková
Slovak University of Technology in Bratislava
Ilkovičova 2, 842 16 Bratislava, Slovakia
maria.bielikova@stuba.sk

Categories and Subject Descriptors

D.2.8 [Software Engineering]: Metrics

Keywords

Information tag, empirical metadata, developer activity

1. CONTEXT

Empirical metadata that describe activities of developers with source code are often stored as logs with only basic references to source code. These approaches deals with several problems. At first, logs are attached to whole source code files and it is hard to analyse the amount of collected data. The second problem is caused by dynamism of source code. If we have only historical logs about activities of developers over source code, we could not be sure that the source code still exists or it has been at changed.

2. GOAL

If we transform logs to empirical metadata that are directly assigned to software artefacts and we can guarantee validity of these metadata, then we can assign meaning to empirical metadata, define empirical software metrics and analyse characteristics of the source code in a particular time (a project state). This idea can be used as a solution of problem of usability of empirical software metrics, with which many tools challenge [1].

3. METHOD

To address this idea we store empirical metadata in form of information tags created by a *tagger* from the stream of developers' activities. We also proposed robust location descriptor that exactly identifies position in the source code [2]. In this work we describe the approach to real-time processing developers' activities and enriching source code with empirical metadata calculated from these activities.

We collect empirical data via software tools and extensions for integrated development environments and web browsers,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ESEM '14, September 18–19, 2014, Torino, Italy.

Copyright 2014 ACM 978-1-4503-2774-9/14/09 ...\$15.00.

that collect developers' activities. Activities are processed on the server as a stream of events and calculated empirical metadata are stored to information tags repository. So we store only collected data from time window which is necessary for matching event sequence templates that identify significant activities of developers that should be assigned to source code as information tags (empirical metadata).

4. RESULTS

We realized proposed approach in the tagger based on processing linked stream data [3]. An employing linked stream data gives us possibility to utilize advantages of linked data inference with a combination of simple specification of event sequence templates in a SPARQL-like language. The tagger consists of repository of tagging rules, linked stream data generator and processor and tagging actions executor.

Tagger creates information tags based on tagging rules that have specified stream queries and tagging actions, that are stored in *repository of tagging rules*. Stream queries match sequences of events in the stream and selecting queried data. Following of a positive evaluation of queries, tagging actions uses queried data and create information tags.

5. CONCLUSIONS

We performed preliminary performance evaluations. We have registered query which selects objects that consist of five RDF triples. During the evaluation we repeatedly posted 100,000 objects to the tagger and we measured response time of the tagger. This evaluations proves that the tagger can process a stream of developers' activities in real-time.

6. ACKNOWLEDGMENTS

This work was partially supported by grants No. VG1/0675/11 and the grant No. APVV-0233-10.

7. REFERENCES

- [1] P. Johnson. You can't even ask them to push a button: Toward ubiquitous, developer-centric, empirical software engineering. In *Visions for Software Design and Productivity: Research and Applications*, p. 5, 2001.
- [2] K. Rástočný and M. Bieliková. Metadata anchoring for source code: Robust location descriptor definition, building and interpreting. In *DEXA*, volume 8056 of *LNCS*, pp. 372–379. Springer, 2013.
- [3] J. F. Sequeda and O. Corcho. Linked Stream Data: A Position Paper. In *Proc. of the 2nd Int. Workshop on Sema. Sensor Networks*, pp. 148–157. CEUR-WS, 2009.