Beyond Replication: An example of the potential benefits of replicability in the Mining of Software Repositories Community

Gregorio Robles
GSyC/LibreSoft
Universidad Rey Juan Carlos
Madrid, Spain
grex@gsyc.urjc.es

Daniel M. Germán
Computer Science Dept.
University of Victoria
Victoria, Canada
dmg@uvic.ca

ABSTRACT

While in theory the mining software repositories is an area where replication is easier to perform than for other empirical software engineering fields, a review of papers presented at the Mining Software Repositories Workshop/Working Conference shows that the research studies presented do not satisfy the requirements for easy replication. In this paper, we present some possibilities that replicability may provide to this community that go beyond the verification of results presented in the original study.

Categories and Subject Descriptors

D.2.0 [**Software Engineering**]: General

Keywords

Replication, mining software repositories

1. INTRODUCTION

Replication is a fundamental task in empirical sciences and the lack of replicability is one of the main shortcomings that empirical software engineering may suffer [1]. But replication of an experiment in empirical software engineering research is a complex task to perform. As Juristo and Vegas show, it is difficult to recreate the exact conditions and context of the original experiment; furthermore, there is a high variability inherent to the fact that many experiments are performed from the observation on humans [10].

As part of the Mining Software Repositories (MSR) research community (see `msr.uwaterlooo.ca`), one of the authors of this paper thought (perhaps naïvely) that many of the problems that are a burden for replication in empirical software engineering research did not affect this area. An examination of the papers submitted to the MSR Workshop/Working Conference during the last six years told, however, a different story: very few of the studies presented at that venue are potentially replicable [8].

RESER 2010 Cape Town, South Africa

The aim of this position paper is to emphasize the importance of replication in the area of mining software repositories research. In the opinion of the authors, the contribution of replicability goes beyond the possibility of reproduction and validation of the studies, and has other benefits that the mining software repositories community should pursue.

The remainder of this paper is structured as follows: next a summary of the study on the replicability of the papers published in the MSR proceedings is offered. Then, a description of best practices that would allow replication by other research groups in the are of mining software repositories are described. Section 4 presents some ideas of why replication in the area of mining software repositories is a practice that goes beyond reproduction of previous results; these advantages are highlighted. The following section discusses how to foster replication at the MSR. Finally, conclusions are drawn.

2. REPLICABILITY OF MSR RESEARCH

Mining software repositories has become a fundamental area of research for the Software Engineering community that relies heavily on empirical studies. Software repositories contain a large amount of valuable information that includes source control systems storing all the history of the source code, defect tracking systems that host defects, enhancements and other issues, and other communication means such as mailing lists or forums.

The general process by which software repositories are mined is described in Fig. 1 [7], and includes identification of the source of data under study (usually available over the Internet), the retrieval of the data, the extraction and *cleaning* of the data (it usually comes embedded in other information such as web pages, e-mail messages,etc. and may be partial, redundant, erroneous, etc.), and its storage in a convenient format (usually a database). The final step consists of the analysis of the data, using data mining or other techniques.

Because the amount of involved data is usually very large, the whole, or at least part of the process is supported by tools or scripts developed by the researchers.

The replicability of a study in this field depends mainly on the public availability of the data of the project under study (the "raw" data) and access to the tools and scripts used by the researchers—as many of the details of the study are *embedded* into these. In addition, the preparation of the "processed dataset" involves decisions that usually are not clearly described in the paper, and those attempting to replicate the study might not be able to duplicate such pro-

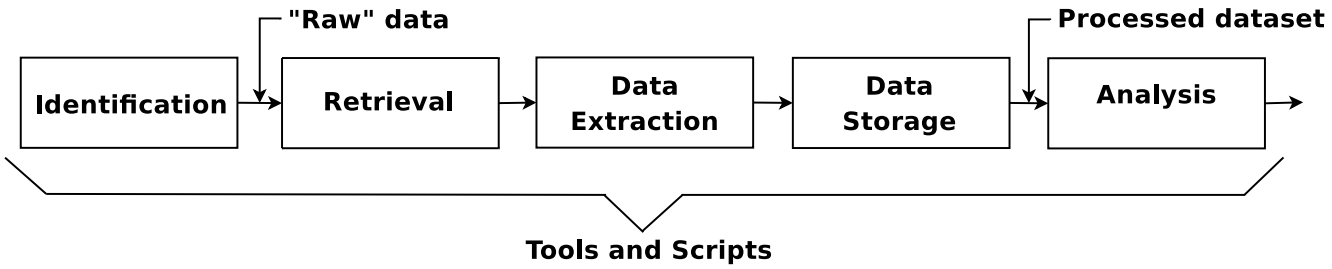

Figure 1: Typical MSR integral process: from identification of the data sources to analysis of the data.

cessed dataset. In this sense, the availability of this dataset is another condition to be met to make a study replicable.

A study of the MSR proceedings from 2004 to 2009 shows that only two out of 150 experimental papers are potentially replicable [8]. This means that most MSR papers are in general not very replication friendly. Interestingly enough, the MSR community has clearly benefited from the openness of the development process of Free/Libre/Open Source Software (FLOSS), as the majority of studies have used the "raw" data from them; and yet the same researchers have done the opposite, by being reluctant to offer a similar openness of their research process.

Other interesting results are the fact that replicability of a paper tends to decrease with the age of the paper. This is due to various reasons:

- The original sources may change, making the "raw" data difficult to obtain. This is due, for example, to technological changes in the infrastructure of FLOSS projects, such as when they change the version control system–from CVS to git—or when moving from one development forge to a new one.

- Tools and other artifacts (such as the processed dataset) are more difficult to find. For example, the websites are moved, revamped, and authors cease to maintain maintain them. The examination of the MSR papers for tools and datasets threw over a dozen "Not Found" errors when trying to access them.

3. BEST PRACTICES

One of the key findings from studying the MSR proceedings is that, in general, papers do not provide enough details that help to replicate the study they perform. Some recommended good practices are described:

"Raw" data and processed dataset

1. Even if publicly available, provide a snapshot of the "raw" data. Projects change infrastructure and the original data on which a study is based on may be not available anymore or just partially be available. This guarantees knowing the exact data used in the study.

2. Version your datasets and offer the various versions publicly. It is frequently the case that changes to the dataset are performed if used for various purposes or for different publications.

3. Indicate the time span and/or version(s) of the data under study.

4. Although there is uncertainty over the scope and applicability of copyright and other legal protections over databases, it is a good practice to specify explicitly the conditions under which the data can be used, copied and redistributed. These conditions should take into consideration the original license and restrictions of the original data.

5. Upload the dataset to a public repository, such as Science Commons[1]. This will lower the effort of maintaining all datasets used by an author.

Tools & scripts (aka "Prototypes")

1. Make your tools available to others, even if incomplete or undocumented. It is often better to have undocumented tools, than to have nothing.

2. Version your tool.

3. Indicate in the version of the tool that has been used.

4. Use a license for your software. Specify explicitly the conditions under which your software may be used, copied, modified and redistributed. A FLOSS license is recommended.

5. Use external infrastructure to support your tool. Create a project at one of the many forges (for instance, SourceForge, BerliOS, etc.) and use that infrastructure (that includes versioning, download, forums). This will reduce the maintenance effort.

4. BEYOND REPLICATION

Beyond the mere replicability of papers, there are some more profound implications from which the MSR community would benefit from.

These points embrace a *culture of sharing* that allows new researchers to join the MSR research community faster, as there would be less barriers to entry. The time required to innovate would be lower as new studies can begin on the grounds of previous data sets and tools, the amount of case studies could be significantly improved as well as the diversity of such case studies, as several points of view (even studying the same data) are possible.

Learning by doing

Replication can be a valuable method in the introduction of students to software engineering research. One of the most challenging aspects of doing research is finding the right research question to ask and developing a method to answer it.

[1]http://sciencecommons.org/

Replication obviates the need for both: the student can address the same research questions, and use the same methodology as the paper that is to be replicated.

In an on-going graduate course on Mining Software Repositories at the University of Victoria[2], students were asked to complete as a class assignment the general track of the MSR Challenge: "Discover interesting facts about the history of the FreeBSD distribution, the Ultimate Debian Database, and the GNOME desktop suite. Results should be reported as 4-page submissions, to be included in the proceedings as challenge papers."[3]. As an option, students were allowed to replicate an MSR Challenge report from previous years, either by using newer data for the same projects analyzed, or by applying the same methods on a different data set. In addition to the replication, students were asked to 1) compare their results to the ones of the paper being replicated, and 2) comment on their experience doing the replication. Students were given two weeks to complete the assignment, and allowed to work individually or in teams of two or three students. A total of 9 reports were submitted, 8 of them replications.

Only one team decided to do a replication on a different data set. Four teams chose to do replications of one of the author's paper. This was influenced because they were provided the processed data for the study, and they had direct access to the author for clarifications. Some of the challenges the students faced were those described above:

- The raw data is not enough. The data is made available by the organizers of the MSR Challenges. Unfortunately reproducing the challenge papers requires not only the data, but tools to process it, and these tools or scripts might not be available. In some instances students recreated some of the processing scripts (when they were not too complex) based on the paper description. In other cases the tools were widely available (such as the data mining framework Weka[4]).
- The data processing and analysis description is often incomplete or not clear enough. Students had to make assumptions that might affect the quality of the replication.

The major outcome of these replications was the experience gained by the students. In a nutshell, it can be summarized as follows:

- Experience using a data set. The students had to get "their hands dirty" and process and analyze data from a software project.
- Experience performing MSR research. The majority of the students had never done this type of research. In general, the reports they chose to reproduce had a simple methodology and were easy to follow.

The quality of the results varied. In few instances the replications were simple reproductions of the same methods. In other cases the students reflected on the results of the replication, in particular the differences found by using newer data than the original paper. One of the replications was accepted for publication as an MSR Challenge Report [2]. This replication demonstrated that the results of the original paper (accuracy of prediction of the lifetime of a defect) vary significantly from one project to another (and casts doubt of the generalization of the original results).

As an instructor one of the most important lessons of this experiment was that replications are a valuable research teaching tool that can rapidly introduce students to research and potentially result in publications. At the time of writing this paper, the formal student evaluations are not yet available. We expect to include them in the presentation at the time of the workshop.

High quality data sets and tool benchmarking

Although the amount of data that is available for research is very large [6] and other (public) processed data sets on software development, such as the ones offered by the MSR Challenge, FLOSSMetrics [4], FLOSSMole [3] or PROMISE [9], are being published, the number of studies concerned with the data itself is low.

Data sets must be carefully studied to recognize their limitations, shortcomings, and special circumstances. The main objective of the MSR community is the identification, retrieval and analysis of interesting datasets of software engineering data. Hence, MSR research can become a data provider for other related research communities such as software visualization and software measurement.

Another aspect from which the MSR community would benefit through replication is the possibility of benchmarking. As the mining research is data intensive, much of it is automatized. The availability of tools and data potentially makes comparison among different tools and mining processes possible.

This will allow to bring back to the research agenda some aspects that many view as *only engineering*, as innovation to achive better performance and results will be fostered. Benchmarking in the case of MSR research goes beyond Big-Oh performance issues in terms of time and memory consumption; process, heuristics and methods used for data extraction and *cleaning* could be compared as well.

5. DISCUSSION

Although, in the opinion of the authors, the benefits of replication are clear, there are some issues that require further discussion.

Fostering replication

One of the first questions that arises is to what extent should replication be considered as an obligatory characteristic of any study. This would, for instance, make many studies from industrial settings for which the availability of the data sources is generally not publicly available difficult to publish. In the opinion of the authors, not all published papers should be replicable, but a majority should be—contrary to the current state of practice, where only a few are. Scenarios where case studies combine publicly available data sources (such as the ones from FLOSS projects) with non-public industrial settings could be a plausible solution. Other cases, such as validation of findings from FLOSS projects in industrial settings are also of great value and should not be dismissed because they lack the possibility of being replicated.

A second question is related to the incentives and rewards that authors, reviewers and the community have in order to foster replication. It is beyond any doubt that offer-

[2] http://turingmachine.org/msr
[3] http://msr.uwaterloo.ca/msr2010/challenge/
[4] http://www.cs.waikato.ac.nz/ml/weka/

ing a replicable paper is a task that requires effort from the authors. The reward for this effort may be unclear, as authors lose the competitive advantage of having data and tools. Replicability could be a key characteristic for any research to be accepted, but in that case it is not clear how far reviewers should verify and evaluate replicability; a potentially acceptable level of replicability is the one described in this position paper ("raw" data, tools and processed dataset made publicly available). One of the duties of the reviewers is to verify that such data is available, and appears to be complete.

There are other rewards—beyond getting a paper accepted—that may motivate authors to make their studies more replicable. Citations are considered an indication of quality of a paper; similarly, the number of papers that replicate the study could also be taken into consideration. By making the data and tools available, other researchers might improve on the methods and techniques described in the paper, hence "building on the shoulders of others".

Supporting replication

From our study on the MSR proceedings, we have noticed that there is a need to have standardized ways to indicate where data sets, tools and other important details for replication can be found. In addition, we have found that a specific infrastructure for replication would be desirable. As researchers publish their work, even if they offer the data and tools it is based on, they usually do not spend much time maintaining it. So, in the same manner we have the IEEE Library we could use a FLOSS forge (such as SourceForge) where such tools could be stored and made available to anybody who requires them.

Finally, we believe that current practices are not mature enough to define standards, such as the creation of metadata that describes existing data.

Privacy Concerns

There are many challenges related to the ideas that we have presented in this short paper, but one of the most evident ones is the privacy and anonymity concerns. This is a complex issue that requires thoughtful consideration. Lessons might be learnt from the network measurement community who use the tactic of IP address anonymization [5].

6. CONCLUSIONS

In this position paper we have presented our view on the potential benefits that the mining software repositories community would gain by fostering the replicability of their studies. We have summarized previous research that shows that the potential replicability of this community is very low at its current state and have proposed some good practices that would enhance this situation.

In addition, we have argued that, in the case of mining software repositories research, replication has more benefits than validating previous research results. It is the opinion of the authors that embracing a *culture of sharing* would raise the quality of the research. In this sense, we have depicted some examples of this benefits, such as learning by doing, having high quality data sets and the possibility of comparing tools and mining processes.

Finally, we have presented some open topics that have to be addressed by the mining software repositories community in order to take advantage of the benefits of replication.

Acknowledgments

The work of Gregorio Robles has been funded in part by the European Commission, through projects FLOSSMetrics, FP6-IST-5-033982, QUALOSS, FP6-IST-5-033547, and Qualipso, FP6-IST-034763. The work of D. M. German has been supported by the Natural Sciences and Engineering Research Council of Canada. The authors would also thank the attendants of the Mining Software Archives Workshop organized in March 2010 by Harald C. Gall and Andreas Zeller in Monte Veritá for their insightful comments and suggestions. The four anonymous reviewers of the RESER workshop also provided us with very detailed and inspiring feedback.

7. REFERENCES

[1] V. R. Basili, F. Shull, and F. Lanubile. Building knowledge through families of experiments. *IEEE Trans. on Software Engineering*, 25(4):456–473, 1999.

[2] G. Bourgi, C. Treude, D. M. German, and M. A. Storey. A Comparative Exploration of FreeBSD Bug Lifetimes. In *International Working Conference in Mining Software Repositories–MSR Challenge Reports (MSR 2010)*, 2010. To be presented.

[3] M. Conklin, J. Howison, and K. Crowston. Collaboration using OSSmole: A repository of FLOSS data and analyses. In *Proceedings of the International Workshop on Mining Software Repositories*, pages 126–130, St. Louis, Missouri, USA, May 2005.

[4] I. Herraiz, D. Izquierdo-Cortazar, and F. Rivas-Hernández. Flossmetrics: Free/libre/open source software metrics. In *CSMR*, pages 281–284, 2009.

[5] W. John and S. Tafvelin. Analysis of internet backbone traffic and header anomalies observed. In *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 111–116, New York, NY, USA, 2007. ACM.

[6] A. Mockus. Amassing and indexing a large sample of version control systems: Towards the census of public source code history. In *MSR*, pages 11–20, 2009.

[7] G. Robles. *Empirical Software Engineering Research on Libre Software: Data Sources, Methodologies and Results*. PhD thesis, Escuela Superior de Ciencias Experimentales y Tecnología, Universidad Rey Juan Carlos, 2006.

[8] G. Robles. Replicating MSR: A study of the potential replicability of papers published in the Mining Software Repositories proceedings. In *Proceedings of the 2010 Working Conference on Mining Software Repositories*, 2010. accepted.

[9] J. Sayyad Shirabad and T. J. Menzies. The PROMISE Repository of Software Engineering Databases. School of Information Technology and Engineering, University of Ottawa, Canada, 2005.

[10] S. Vegas, N. Juristo, A. Moreno, M. Solari, and P. Letelier. Analysis of the influence of communication between researchers on experiment replication. In *ISESE '06: Proc. of the 2006 ACM/IEEE International Symposium on Empirical Software Engineering*, pages 28–37, 2006.