MSSF
Professional and Research Practice
CA 640
Jane Horgan

Critical Review of Tesseract

cand. Dipl. Inf. Tobias Müller <muellet2@>, 59212333

11th June 2013

Disclaimer: Submitted to Dublin City University, School of Computing for module CA640: Research Skills, 2009. I hereby certify that the work presented and the material contained herein is my own except where explicitly stated references to other material are made.

Abstract

This critical review of a paper, which presents Tesseract and was handed in for the ICSE 2009, focusses on strength and weaknesses of the idea behind Tesseract: Visualising and exploring freely available and loosly coupled fragments (mailing lists, bug tracker or commits) of Free Software development. Tesseract is thus a powerful data miner as well as a GUI to browse the obtained data.

This critique evaluates the usefulness of Tesseract by questioning the fundamental motivation it was built on, the data which it analyses and its general applicability.

Existing gaps in the original research are filled by conducting interviews with relevant developers as well as providing information about the internal structure of a Free Software project.

1 Introduction

As part of coursework for a Masters in Security and Forensic Computing (MSSF) students were asked to critically review a paper from the International Conference of Software Engineering 2009 (ICSE). I chose to review *Tesseract: Interactive Visual Exploration of Socio-Technical Relationships in Software Development* [Sarma et al., 2009]. I will refer to them as *the authors* and *the paper* throughout this text as the authors and the paper are the main matter in this review.

Tesseract is a program that builds and visualises a social network based on freely available data from a software project such as mailing lists, bug tracker or commits to a software repository. This network can be interactively explored with the Tesseract tool. This tool shows how communication among developers relates to changes in the actual code. The authors used a project under the GNOME umbrella named *Rhythmbox* to show their data mining and the program in operation. GNOME is a Free/Libre Software Desktop[de Icaza et al., 1998] used as default by many Linux distributions including the most popular ones, i.e. Ubuntu and Fedora[1]. To assess Tesseracts usability and usefulness, the authors interviewed people not related to Rhythmbox asking whether Tesseract was usable and provided useful information.

The paper was particularly interesting for me because the authors analysed data from the GNOME project. As I am a member of that development community[2], I wanted to see how their approach can or cannot increase the quality of the project. Another focus was to help their attempt to improve GNOME by highlighting where they may have gaps in their knowledge of its internals.

During this critique, I will show that some assumptions were made that do not hold for Free/Libre and Open Source Software (FLOSS) in general and for GNOME in particular either because the authors simply did not have the internal knowledge or did not research carefully enough. Also I will show that the used data is not necessarily meaningful and I will attempt to complement the lacking data by presenting the results of interviews I conducted with actual GNOME developers. This will show how to further improve Tesseract by identifying new usage scenarios. Lastly, this text will question the general usefulness of Tesseract for the majority of Free Software projects.

2 Presenting Tesseract

Tesseract is a program that collects, analyses, cross-links and visualises data from a software project such as mailing list archives, source code repositories or bug tracker. Loosely coupled fragments of distributed software development (i.e. bug reports, posts to mailing lists or changes to source code) can be connected and interactively explored. Files which are related semantically but not syntactically can thus be identified, because if a set of files are often changed together, it is likely that they have a nontrivial relationship although the code itself does not show this connection.

Assuming that the software itself is well designed and works fine, I am not going to criticise the implementation itself. Instead, I am going to question the usefulness of the application by critically reviewing

[1] cmp. http://www.ubuntu.com or http://www.fedoraproject.org

[2] tobiasmue@gnome.org

- what data was collected,
- how it was obtained and
- whether that data is meaningful.

Using a well-established name When first confronted with the name *Tesseract*, people may reasonably confuse it with the OCR software that exists for more than 20 years now[Wikipedia, 2009]. At the time of writing, a simple web search for "Tesseract" results in the first five hits being the OCR software and would thus have been sufficient to find out whether the name was already taken. This begs the question: did the authors deliberately choose the name despite the fact that some other project uses it, or did they not research at all whether it was already in use. Either way, this is not good practice because it creates confusion, especially since the OCR Tesseract is well known. The choice of a name does of course not influence the usefulness of the software or the idea behind it.

Unproven Motivation The authors claim that "*understanding [...] the [...] congruence between social and technical aspects of a project is vital*"[Sarma et al., 2009, p. 23] to motivate and justify their development of Tesseract. However, the referenced work that is supposed to prove their claim evaluated closed source and highly commercial products only. It is not shown that Free/Libre Open Source Software projects have the same development models or requirements as commercial closed source products. In fact, due to their commercial nature, these products tend to have different requirements than Free Software projects such as time-to-market deadlines. Thus, the work of Cataldo et al ([Cataldo and Herbsleb, 2008] or [Cataldo et al., 2006]) cannot easily be used to prove that visualising data helps a Free Software project at all. Nevertheless, experience shows that if people, who are technically skilled or leading a project, communicate more often the project itself will benefit.

Chats not taken into account In the paper, it is assumed that everything is publicly archived and available, when in fact most communication is not. The importance of communication via real time chats (esp. IRC), which is only archived to a small extent, is shown in [Shihab et al., 2009]. Other important communication methods include private email, personal meetings in offices or on conferences. As archives of these chats between developers either do not exist or are not publicly available, the data being collected cannot represent the social network to a sufficient extent [Aranda and Venolia, 2009, p. 9]. I do not claim that there is a solution to this problem and in fact I do think the authors do their best in gathering data, but the paper does not mention this problematic fact, let alone identify this as a threat to the validity of their results.

Chosen small project Knowing the GNOME project very well, it is interesting to see that the authors chose to visualise the data of the Rhythmbox module. Compared to other GNOME modules, it has a rather small code base and short commit history[3]. Unfortunately, the authors did not state why they tested Tesseract with Rhythmbox and not with a bigger mod-

[3] cmp. http://git.gnome.org/cgit/rhythmbox/log/

ule like "Evolution" or "GTK+". The latter projects actually do need help in development while Rhythmbox is in a good shape. I doubt that they made that decision randomly. They may have wanted to avoid bigger projects because it caused problems either on the data collection or the data visualisation side.

Usefulness of bug data Virtually every Free Software project that is mature and important enough has a group of people that dedicatedly deal with incoming bug reports (called *Bugsquad*) [Linstead and Baldi, 2009]. It refines the reports and requests information from the reporter until the bug report is good enough (cmp. [Bettenburg et al., 2008]) to save developers and reporters time, since the former can deal with bug reports which contain enough information while the latter get quick feedback in a language they actually understand. Most of the time, a Bugsquad member is not a member of the development team so tracking communication patterns from the bug report database[4] is not necessarily meaningful. The authors did not acknowledge the existence of non-developers to change metadata of a bug report. Tesseract may thus display developers as inactive although they are not (false-negative). However, if a developer is shown as active, the result is valid (no false-positive).

Asking non GNOME people The people who were asked about usability and usefulness testing were not associated with the GNOME project in any way. Hence, they did not have any experience or knowledge of how development is done within that special community. According to the authors [Sarma et al., 2009, p. 31], this

> was a deliberate choice as we did not want past experience or personal information about the project to influence the developer in their investigation.

To test the usability of a program it is not crucial to know how a project works, but it certainly is for assessing the usefulness of displayed data. The interviewees could not know whether the data that Tesseract showed was of any practical use for fixing bugs, advance the state of the art, or even if it is correct at all. Thus, deliberately choosing people who have not been involved in the project that corresponds to the data set to evaluate "*how real-life developers would use [...] Tesseract for their day-to-day use*" cannot yield satisfactory results. Instead, the question they can answer with this approach is how a project manager would use Tesseract. As the authors admit in their conclusion, they did not find an answer to their original question but rather that it "*benefits new developers and managers*"[Sarma et al., 2009, p. 32].

To answer their original research question, I conducted an informal interview with two GNOME developers asking what benefits they see in using software like Tesseract and how they would use it on a day-to-day basis. Since neither the Tesseract program nor the interview data were publicly available at the time of writing, I was not able to ask the GNOME developers the very same questions or to allow them to try out the Tesseract program. Instead, I presented screenshots and a high level description of Tesseract. The interviewees acknowledged the fact that it helps to find key people of a project. However, they

[4] Bugzilla is the most dominant web based bug tracking solution and is widely adopted among Free Software projects

do not appreciate that from a developers, but rather from a recruiting (head-hunter) point of view.

According to the interviewees the information Tesseract displays is not of much use for a developer, because it discourages proper testing of the changes to the source code. They prefer to interactively explore the source code by modifying it instead of "*clicking through a GUI full of bulletpoints and arrows*". Both agreed that they would be more willing to use Tesseract if it could explore the data of more than one given project at the same time (cross-project), but rather to see who the most active developer is, to replace or extend social coding websites such as ohloh.net instead of having a help for fixing a given bug:

> It could help stimulating ones ego and replace Kudos[5] to find out who best contributor is. Open Source is about pushing the ego anyway.

Since I conducted only two interviews, the results may not be considered representative.

Inaccurate Data The authors claim that 10% of their collected data could not be associated with its corresponding developer. Given that other researchers such as Bird et al in [Bird et al., 2008] managed to associate just 30% - 70% correctly while examining 5 major Free Software projects, it would be interesting to know how the authors actually managed to cross-link their data to get 90% covered. Since the authors did not outline their method or algorithm it is impossible to verify or replicate their results.

[5] a give-away on the social coding site ohloh.net that developers can send each other

Another detail that is omitted is the time and bandwith their data extraction needed and whether it is possible to update the data set incrementally. Given that Tesseract holds and visualises a large amount of data the obvious questions are how much memory is used during runtime and how well the memory consumption scales.

Generality Since Tesseract mines only publicly available data, it cannot be used in some scenarios where the project simply does not have the raw data. Many projects make use of so-called *canned-hosting* [Fogel, 2005] so that the required data cannot be accessed, i.e. because the database is hosted on an external server owned by SourceForge[6] and not by the project. As the majority of Free Software projects make use of canned-hosting, Tesseract can only be useful for a small fraction of the Free Software projects; namely those who house their own infrastructure.

Other work While the authors list different tools that almost implement the desired functionality, namely mapping social and technical artifacts, they did not mention an existing tool built from the GNOME community itself (sic!). It is not clear, why the authors wrote their own proprietary program instead of improving the existing approach named *Pulse* mainly written by Sean McCance[McCance, 2008]. I doubt that the authors were not aware of that approach whose development was even funded by Google in 2009[Google, 2009].

[6] sf.net is the biggest canned-hosting provider

3 Conclusion

Based on screenshots provided in the paper, the Tesseract software seems to be well written and the visualisation apparently works fine. However, I have shown that the idea behind it is not unique as other projects try to implement the same functionality. Although the authors did not publish their implementation, Tesseract may nevertheless be the first program that succcessfully cross-links and visualises data from a FLOSS project.

While other researches have had problems linking data from various sources with its corresponding developer, the authors presented a powerful method that is able to leave only 10% of the data unassociated. Reproducing their results was not possible as they did not present details of their method.

The usefulness of Tesseract is reduced to a great extent by two facts. Firstly, the assumptions made by the authors were shown to not necessarily hold for a FLOSS project. Secondly, since the majority of the Free Software projects use canned-hosting, the authors powerful data extraction method is not of much use to these projects.

I have identified not asking people related to the GNOME project as a threat to the validity of the papers result and filled that gap by conducting interviews with GNOME developers. That also uncovered new usage scenarios as well as ways to improve Tesseract.

References

[Aranda and Venolia, 2009] Aranda, J. and Venolia, G. (2009). The secret life of bugs: Going past the errors and omissions in software repositories. pages 298–308.

[Bettenburg et al., 2008] Bettenburg, N., Just, S., Schröter, A., Weiss, C., Premraj, R., and Zimmermann, T. (2008). What makes a good bug report? pages 308–318.

[Bird et al., 2008] Bird, C., Pattison, D., D'Souza, R., Filkov, V., and Devanbu, P. (2008). Chapels in the bazaar? latent social structure in OSS. *Submitted to the Sixteenth ACM SIGSOFT International Symposium on Foundations of Software Engineering*.

[Cataldo and Herbsleb, 2008] Cataldo, M. and Herbsleb, J. D. (2008). Communication networks in geographically distributed software development. In *Proceedings of the ACM 2008 conference on Computer supported cooperative work*, pages 579–588, San Diego, CA, USA. ACM.

[Cataldo et al., 2006] Cataldo, M., Wagstrom, P. A., Herbsleb, J. D., and Carley, K. M. (2006). Identification of coordination requirements: implications for the design of collaboration and awareness tools. In *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, pages 353–362, Banff, Alberta, Canada. ACM.

[de Icaza et al., 1998] de Icaza, M., Lee, E., Mena, F., and Tromey, T. (1998). The GNOME desktop environment. In *Proceedings of the annual conference on USENIX Annual Technical Conference*, pages 38–38, New Orleans, Louisiana. USENIX Association.

[Fogel, 2005] Fogel, K. F. (2005). *Producing Open Source Software: How to Run a Successful Free Software Project*. O'Reilly Media. http://producingoss.com/en/producingoss.pdf.

[Google, 2009] Google (2009). Integrate bugzilla into pulse for GNOME. http://socghop.appspot.com/gsoc/student_project/show/google/gsoc2009/gnome/t124022403484.

[Linstead and Baldi, 2009] Linstead, E. and Baldi, P. (2009). Mining the coherence of GNOME bug reports with statistical topic models. In *Mining Software Repositories, 2009. MSR '09. 6th IEEE International Working Conference on*, pages 99–102.

[McCance, 2008] McCance, S. (2008). Pulse. http://blogs.gnome.org/shaunm/2008/03/03/pulse/.

[Sarma et al., 2009] Sarma, A., Maccherone, L., Wagstrom, P., and Herbsleb, J. (2009). Tesseract: Interactive visual exploration of socio-technical relationships in software development. In *Proceedings of the 2009 IEEE 31st International Conference on Software Engineering*, pages 23–33. IEEE Computer Society.

[Shihab et al., 2009] Shihab, E., Jiang, Z. M., and Hassan, A. (2009). On the use of internet relay chat (IRC) meetings by developers of the GNOME GTK+ project. In *Mining Software Repositories, 2009. MSR '09. 6th IEEE International Working Conference on*, pages 107–110.

[Wikipedia, 2009] Wikipedia (2009). Tesseract (software) — wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Tesseract_(software)&oldid=330098415 [Online; accessed 8-December-2009].

License