Which Smell-based Metrics Affect Spreadsheet Performance? An Analysis of Four Datasets

Alaaeddin Swidan Felienne Hermans

Delft University of Technology
Delft, Netherlands
{Alaaeddin.Swidan, F.F.J.Hermans}@tudelft.nl

Abstract

Spreadsheets are used extensively for calculations in many domains. Their easy to use and intuitive interface allow the users to build various complexities of calculations. That often lead to maintainability issues, including performance problems. Despite the number of resources containing possible spreadsheets formulas performance anti-patterns, little research has been done to validate them. In this paper, we analyze 40,122 spreadsheets from four different data sources to investigate the effect of 20 spreadsheet performance-related metrics. These metrics are chosen following a smell-driven analysis. Thereafter, our analysis constructs a linear regression model with 12 metrics that are found significant to the explanation of the spreadsheet performance. We further identify each metric contribution to the performance model. Initial Results show that the most three significant spreadsheet metrics are: the repetition of a formula over a large range (35.89%), the conditional formatting of cells (19.53%) and the calls to special Excel functions, such as lookup functions (18.84%).

1 Introduction

Spreadsheets are used commonly in practice, in various business domains [1, 2]. Despite their popularity, spreadsheets suffer from several problems. For example, spreadsheets contain a mixture of data and formula calculations, which make them hard to understand. Moreover, they have a long life span in organizations, which means they grow in size and complexity over time. Finally, spreadsheets, like software, suffer from smells that could lead to errors [3, 4, 5]. Previous research focused on improving the quality of spreadsheets: comprehension, error detection and migration among others. In this paper, we explore a new direction of the analysis of spreadsheet quality: the performance. Performance of a spreadsheet is crucial because any latency affects the sense of immediate calculation provided to the user. While there are some tools that profile spreadsheet performance [6, 7], and blog posts [8, 7] hypothesizing what might contribute to spreadsheet performance, we are not aware of any papers that attempt to measure it.

Therefore, the goal of this paper is to establish spreadsheet performance understanding using the quality metrics of the spreadsheet: its smells. To achieve this we perform a large scale analysis on 40,122 spreadsheets collected from four datasets of spreadsheets; three public datasets [9, 10, 11], and one collection of spreadsheets we crawled from the public web. With these spreadsheets, we aim to answer the following research questions:

Proceedings of the Seminar Series on Advanced Techniques and Tools for Software Evolution SATToSE 2017 (sattose.org).
07-09 June 2017, Madrid, Spain.

RQ1 What spreadsheet smell-based metrics may affect the performance?

RQ2 Which of these metrics are found to affect the performance?

RQ3 How do the significant metrics contribute to the spreadsheet performance?

In order to answer these questions, first we establish a set of metrics which describe different aspects of calculations in spreadsheets. To define these metrics, we implement *Smell-driven performance analysis (SDPA)* which is introduced by Chambers and Scaffidi [12]. In SDPA, a code smell is treated as an indicator to a performance anti-pattern. In the same time, we measure the calculation time, as the performance indicator, for the spreadsheets in our dataset. Subsequently, we build a linear regression model for the calculation time against the collected metrics. Finally, we further analyze the statistical model to assess the contribution of each metric in the performance model. The result of our analysis identifies 12 metrics which are statistically significant in describing the spreadsheet performance. Among the top three metrics contributing to the model are the repeating of formulas, the conditional formatting and the calls to special Excel functions. The main contributions of this paper are:

- A set of metrics which are related to the performance of a spreadsheet. Some of the metrics are introduced for the first time in research.
- The results of an experiment that yields a statistical model which describes the relation and contribution of these metrics to the spreadsheet performance.

2 Background and Motivation

Smell-Driven Performance Analysis
Smells in software are code segments which function correctly, but hinder the quality and maintainability of the software [13]. Several smells are related to complexities of the code design and structure, which make them suitable for detecting performance anti-patterns. Chambers and Scaffidi [12] follow this basic idea to introduce the smell-driven performance analysis (SDPA). They applied SDPA to detect performance anti-patterns in end-user applications, LabView models in particular. A similar approach in analyzing performance was followed by Wang *et al.* who detected performance anti-patterns in HPC applications [14]. In spreadsheets, there is an established research on smells and their role in hindering the quality and understanding of a spreadsheet [2, 3, 15]. We use the outcome of these research efforts, in a smell-driven performance analysis, to define a set of spreadsheet metrics that affect its performance.

Motivation
Spreadsheets are known to provide immediate results to the end-users. The smallest latency in performing the calculations hinders both the satisfaction of the end-user, and the business process behind the spreadsheet. However, little research has been done on spreadsheet performance. One research focused on the real-time generation of .NET code from a spreadsheet to speed it up [16, 17]. In their studies, a benchmarking experiment was performed which found that the generated code is faster than some other spreadsheet software, but not Excel. They did not provide explanations to the performance of Excel spreadsheets. In practice, Microsoft knowledge articles and Excel experts electronic (online) resources [18, 7] provide a mixture of information about possible causes of spreadsheet performance problems. However, these potential causes have never been evaluated in a scientific experiment. In this paper we aim to perform a scientific study that measures spreadsheet performance, and provides an understanding into which metrics contribute to the performance problem.

3 SATToSE Presentation

Under the **Work in Progress** category at SATToSE2017, we would like to present the basic concept of the paper: linking the code smells with performance. We will go through the experiment setup, the smell-based metrics extracted and the approach to extract them. We will present and discuss initial results which show a significant relation between smell-based metrics and the performance of a spreadsheet. Finally, we will describe the following action plan: To what extent does refactoring these smells help in improving the performance, in addition to improving he quality?

References

[1] Croll, G.J.: The importance and criticality of spreadsheets in the city of london. Proc. European Spreadsheet Risks Int. Grp. (EuSpRIG) 2005 82-92 ISBN:1-902724-16-X (2005)

[2] Hermans, F.: Analyzing and visualizing spreadsheets. Ph.D. thesis, Technische Universiteit Delft (2012)

[3] Hermans, F., Jansen, B., Roy, S., Aivaloglou, E., Swidan, A., Hoepelman, D.: Spreadsheets are code: An overview of software engineering approaches applied to spreadsheets. 2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER) 5 (2015)

[4] Hermans, F., Pinzger, M., van Deursen, A.: Detecting and visualizing inter-worksheet smells in spreadsheets. 2012 34th International Conference on Software Engineering (ICSE) (2012)

[5] Abreu, R., Cunha, J., Fernandes, J.P., Martins, P., Perez, A., Saraiva, J.: Smelling faults in spreadsheets. 2014 IEEE International Conference on Software Maintenance and Evolution (2014)

[6] Fastexcel v3 profiler, `http://www.decisionmodels.com/FastExcelV3Profiler.htm`

[7] Mcpherson, B.: Identifying which formulas in excel are slowing down workbook recalaculation, `http://www.nullskull.com/a/1479/identifying-which-formulas-in-excel-are-slowing-down-workbook-recalaculation.aspx`

[8] Duggirala, P.: A round-up on circular references (2010), `http://chandoo.org/wp/2010/09/16/excel-circular-references/`

[9] Fisher, MarcRothermel, G.: The euses spreadsheet corpus. Proceedings of the first workshop on End-user software engineering - WEUSE I (2005)

[10] Hermans, F., Murphy-Hill, E.: Enron's spreadsheets and related emails: A dataset and analysis. 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering (2015)

[11] Barik, T., Lubick, K., Smith, J., Slankas, J., Murphy-Hill, E.: Fuse: A reproducible, extendable, internet-scale corpus of spreadsheets. 2015 IEEE/ACM 12th Working Conference on Mining Software Repositories (2015)

[12] Chambers, ChristopherScaffidi, C.: Smell-driven performance analysis for end-user programmers. 2013 IEEE Symposium on Visual Languages and Human Centric Computing (2013)

[13] Fowler, M.: Refactoring: Improving the design of existing code. Extreme Programming and Agile Methods XP/Agile Universe 2002 pp. 256–256 (2002)

[14] Wang, C., Hirasawa, S., Takizawa, H., Kobayashi, H.: Identification and elimination of platform-specific code smells in high performance computing applications. International Journal of Networking and Computing 5(1), 180–199 (2015)

[15] Jansen, B., Hermans, F.: Code smells in spreadsheet formulas revisited on an industrial dataset. 2015 IEEE International Conference on Software Maintenance and Evolution (ICSME) (2015)

[16] Iversen, T.: Runtime code generation to speed up spreadsheet computations. Ph.D. thesis, University of Copenhagen (2006)

[17] Woodside, M., Franks, G., Petriu, D.C.: The future of software performance engineering. Future of Software Engineering (FOSE '07) (2007)

[18] Williams, C., Bokone, A., Rothschiller, C.: Excel 2010 performance: Tips for optimizing performance obstructions (2010), `https://msdn.microsoft.com/en-us/library/office/ff726673(v=office.14).aspx`