Delft University of Technology
Software Engineering Research Group
Technical Report Series

Breviz: Spreadsheet Visualization and Quality Analysis

Felienne Hermans, Martin Pinzger, and Arie van Deursen

Report TUD-SERG-2011-020

TUD-SERG-2011-020

Published, produced and distributed by:

Software Engineering Research Group
Department of Software Technology
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Mekelweg 4
2628 CD Delft
The Netherlands

ISSN 1872-5392

Software Engineering Research Group Technical Reports:
http://www.se.ewi.tudelft.nl/techreports/

For more information about the Software Engineering Research Group:
http://www.se.ewi.tudelft.nl/

Note: Accepted for publication in the Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing, 2011, IEEE CS Press.

Breviz: Spreadsheet Visualization and Quality Analysis

Felienne Hermans
Delft University of Technology
Email: f.f.j.hermans@tudelft.nl

Martin Pinzger
Delft University of Technology
Email: m.pinzger@tudelft.nl

Arie van Deursen
Delft University of Technology
Email: arie.vandeursen@tudelft.nl

***Abstract*—Thanks to their flexibility and intuitive programming model, spreadsheets are widely used in industry, often for business-critical applications. Similar to software developers, professional spreadsheet users demand support for understanding spreadsheets, and analyzing their quality.**

This paper describes Breviz, a tool that visualizes spreadsheets as leveled dataflow diagrams, showing users all worksheet in a spreadsheet and the data that is flowing between them. Initially, we used these diagrams to support spreadsheet users in explaining and transferring spreadsheets to colleagues. In further studies of the tool however, we have seen that the visualization is also useful to help detect anomalies in the spreadsheet.

I. Introduction

Spreadsheets are used heavily in industry, for a large variety of tasks, ranging from inventory registration to financial administration. A study from the year 2005 shows about 23 million American workers use spreadsheets, which amounts to about 30% of the workforce [1]. In spite of their frequent use, several researchers have published on the problems surrounding spreadsheets, for instance the difficulty of working with spreadsheets[2] and their high error rates [3].

In previous research we have focused on overcoming the difficulties around spreadsheet use, by interviewing spreadsheet professionals to obtain a list of their most prevalent problems and corresponding information needs [4].

The results showed the biggest problem with spreadsheets was transferring a spreadsheet to a colleague, called a transfer scenario. In case of a transfer scenario, the majority of the professional spreadsheet users wanted to know how the the different worksheets within a spreadsheet are related. In answer to that, we have created a tool called Breviz that visualizes spreadsheets as leveled dataflow diagrams, where the highest levels shows all worksheets and their relationships. Figure 1 shows an example of a dataflow diagram generated from a university spreadsheet to calculate exam results. In this diagram each block represents a worksheet. Furthermore an arrow going from Worksheet A to Worksheet B represents there is a formula on Worksheet B that is referring to a cell on worksheet A. The thickness of the arrow indicates the number of formulas that are connecting two worksheets.

During user studies with Breviz we have found that it is very suitable to support spreadsheet users in a transfer s[illegible] our own observations showed that Breviz and the data flow visualization are also useful. useful for analyzing the quality of spreadsheets and identifying possible errors. The remainder of this paper elaborates on an small case study to support this claim.

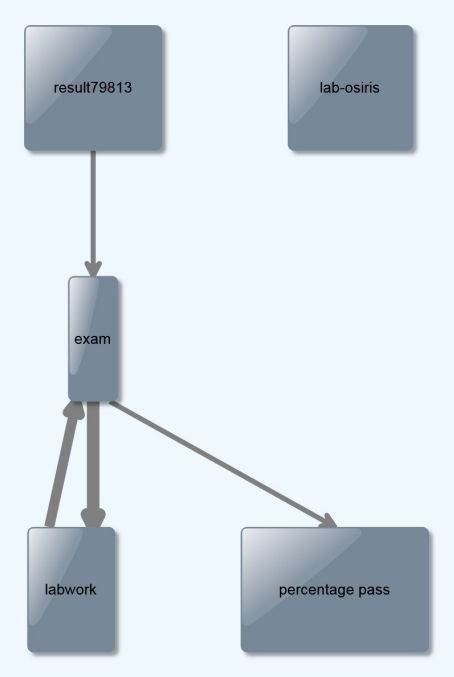

Fig. 1. A spreadsheet dataflow visualization

II. Motivating Case Study

This section describes an example of a spreadsheet, of which the visualization raises questions about the quality of its design.

The spreadsheet in question is a real life spreadsheet used to calculate the grades for a course taught at our university. Figure 1 shows the data flow diagram of the spreadsheet as extracted with the Breviz tool. The spreadsheet consists of five worksheets. From the dataflow visualization some aspects of the spreadsheet immediately catch the eye. Firstly, one of the worksheets lab-osiris is not connected to the other sheets, this sheet contains the data from Osiris, the university's grading [illegible] This information could pose a threat to the correctness of the spreadsheet, since someone working with the spreadsheet

might mistakenly think that all scores are updated when updating the information form Osiris. Besides confusing, this worksheet is also making the spreadsheet unnecessarily large. Such a lonely worksheet is somewhat comparable to the 'lazy class' code smell, a class that is not doing enough to pay for itself, and should be refactored away, since it makes the software system bigger. By the same argument we believe that 'lazy worksheets' are bad practice within spreadsheets. In the example we asked the professor about this lazy sheet, and it turned out that it was indeed no longer necessary; it was replaced in function by *result78913* and it should have been removed.

Determining the presence of lazy worksheets without Breviz would require the user to select all cells for all worksheets one by one and checking their dependents. This is time consuming and error prone.

Secondly the fat loop between *exam* and *labwork* stands out. The name *exam* could suggest that the worksheet only contains data about the exam, however apparently also information regarding the lab work. When inspecting the worksheets, and interviewing the professor who created the spreadsheet we found that *exam* is actually an aggregation of the scores for the labwork and the exam results, so the name could mislead a user of this spreadsheet.

Worksheet *labwork* in turn obtains the names of the students from the exam. Again this reeks of bad practice; getting the names from the worksheet *exam*, while outputting the results back to the worksheet with names could easily confuse users of this spreadsheet. Here again we can see a parallel to code smells in the source code. This smell corresponds to the 'inappropriate intimacy' smell which indicates that two classes are too interested in each other. We suspect a similar fact is true for two worksheets in a spreadsheet; if they are too connected, things might not have been modeled in a logical way. Furthermore this tight coupling could have an effect on maintainability; if one is going to change formulas on the one worksheet, one might need to adapt the other worksheet.

III. Locating spreadsheet smells

After we performed this small study and some additional similar studies; our confidence in this method of identifying weak points in a spreadsheet's design was confirmed. Based on our findings, we created a preliminary list of *spreadsheet smells*, and adapted Breviz in such a way that it can automatically detect the chance of a spreadsheet smell. At this point we only support the lazy worksheet and the inappropriate intimacy smell. We are curently working on the translation of other known code smells into the area of spreadsheets; and subsequently also their automatic detection. Figure 2 shows the output of Breviz with the smell detection module enabled.

IV. Conclusion and contributions

This paper describes Breviz, a tool that visualizes spreadsheets as dataflow diagrams and further explores the possibility of using this tool to identify 'smells' in spreadsheets. Preliminary studies on this new goal of Breviz, such as the one we describe in this paper have increased our confidence of Breviz' applicability for this goal. However, a more elaborate study is needed to prove this hypothesis.

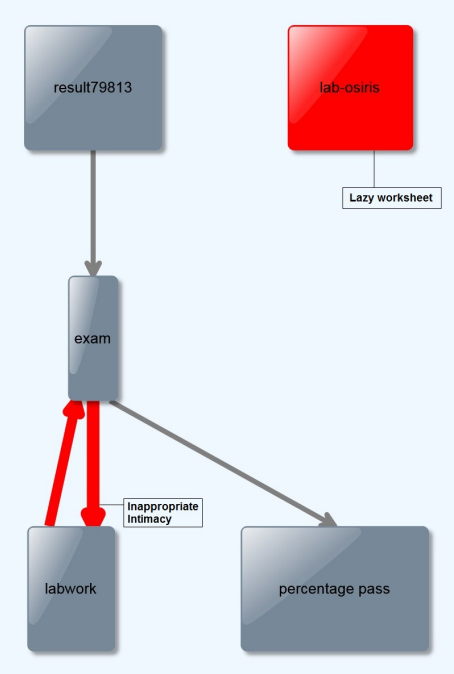

Fig. 2. A spreadsheet dataflow visualization

Since spreadsheets are commonly used in industry, Breviz could support many spreadsheet users in analyzing the structure of spreadsheets. With our new focus on analyzing the quality of spreadsheet structure, taking experience from code smells and anti-patterns, we hope to broaden the scope of finding errors in spreadsheets; since research in that area currently is mainly aimed at finding errors at the formula level.

References

[1] C. Scaffidi, M. Shaw, and B. A. Myers, "Estimating the numbers of end users and end user programmers," in *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 2005, pp. 207–214.

[2] B. Nardi and J. Miller, "The spreadsheet interface: A basis for end user programming," in *Proceeding of Tthe IFIP Conference on Human-Computer Interaction (INTERACT)*. North-Holland, 1990, pp. 977–983.

[3] R. Panko, "What we know about spreadsheet errors," *Journal of End User Computing*, vol. 10, no. 2, pp. 15–21, 1998.

[4] F. Hermans, M. Pinzger, and A. van Deursen, "Supporting professional spreadsheet users by generating leveled dataflow diagrams," in *Proceedings of the 33th International Conference on Software Engineering*, 2011, to appear.

TUD-SERG-2011-020
ISSN 1872-5392