Delft University of Technology
Software Engineering Research Group
Technical Report Series

Enron's Spreadsheets and Related Emails: A Dataset and Analysis

Felienne Hermans and Emerson Murphy-Hill

Report TUD-SERG-2014-021

TUD-SERG-2014-021

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: This paper is currently under review.

Enron's Spreadsheets and Related Emails: A Dataset and Analysis

Felienne Hermans
Delft University of Technology
Mekelweg 4
2628 CD Delft, the Netherlands
f.f.j.hermans@tudelft.nl

Emerson Murphy-Hill
North Carolina State University
890 Oval Drive
Raleigh, North Carolina, USA
emerson@csc.ncsu.edu

***Abstract*—Spreadsheets are used extensively in business processes around the world and as such, a topic of research interest. Over the past few years, many spreadsheet studies have been performed on the EUSES spreadsheet corpus. While this corpus has served the spreadsheet community well, the spreadsheets it contains are mainly gathered with search engines and as such do not represent spreadsheets used in companies. This paper presents a new dataset, extracted for the Enron Email Archive, containing over 15,000 spreadsheets used within the Enron Corporation. In addition to the spreadsheets, we also present an analysis of the associated emails, where we look into spreadsheet specific email behavior.**

Our analysis shows that 1) 24% of Enron spreadsheets with at least one formula contain an Excel error, 2) there is little diversity in the functions used in spreadsheets: 76% of spreadsheets in the presented corpus only use the same 15 functions and, 3) the spreadsheets are substantially more *smelly* than the EUSES corpus, especially in terms of long calculation chains. Regarding the emails, we observe that spreadsheets 1) are a frequent topic of email conversation with 10% of emails either sending or referring spreadsheets and 2) the emails are frequently discussing errors in and updates to spreadsheets.

I. Introduction

Spreadsheets are an important type of software. Scaffidi and colleagues estimate that there are more than 55 million end user programmers in the US alone [1].

In many ways, developing a spreadsheet is similar to writing code. Both entail analyzing data, manipulating operations on that data and understanding dependencies between different parts of the program. One of the main differences is that in software engineering many methods and techniques have been constructed that support developers in managing complexity and understanding existing artifacts. Although recent efforts to transfer software engineering methods to spreadsheets have been relatively successful [2], there is still a lot to be gained, as spreadsheet errors remain common.

For example, Reinhart and Rogoff reported on their analysis of data in Excel in a working paper that concluded that countries with high economic debt-to-GDP ratios have slow economic growth [3]. Politicians have used this result to implement debt-reducing measures across Europe, in an attempt to increase economic growth [4]. However, when the original spreadsheet was shared with scientists doing a replication study, it was revealed that the original spreadsheet contained a selection error, which reversed the paper's main findings about debt and growth [5].

As a consequence of the importance of spreadsheets, significant existing research has studied spreadsheets in a variety of contexts. For example, at ICSE 2007, Abraham and Erwig evaluated GoalDebug, a spreadsheet debugger [6]; at ICSE 2012; we ourselves presented a method for detecting and visualizating spreadsheet smells [7]; and at ICSE 2014, Dou and colleagues presented AmCheck, a tool that finds and repairs ambiguous computations [8].

A significant amout of prior work on spreadsheets does formative or evaluative studies using the EUSES corpus: a set of 4,498 spreadsheets published in 2005. Abraham and Erwig use the EUSES corpus to motivate their approach [6]; We ourselves used EUSES to set thresholds for our code smells [7]; and Dou et al. use EUSES to determine how often ambiguous computation occurs [8].

One of the reasons that EUSES is so commonly used, is that it is the best there is, there is no other corpus of similar size. Some researchers have tried to get access to spreadsheets from industry, but companies are reluctant to share them. Firstly, the contents of the spreadsheets might hold confidential information, such as pricing models, which companies want to keep out of the hands of competitors or customers. Secondly, organizations are afraid detailed studies of their spreadsheets might reveal errors and make them end up in the Europeans Spreadsheet Risk Interest Group's list of Horror Stories[1].

In this paper, we introduce a new spreadsheet corpus obtained from industry. It differs from the EUSES corpus in a number of ways:

- Although EUSES is the largest spreadsheet corpus today, it is relatively small by modern software repository standards; EUSES has about 4.5 thousand spreadsheets, while Sourceforge lists 350 thousand software projects[2] and OpenHub lists about 666 thousand software projects.[3]
- Most EUSES spreadsheets were obtained through the public world-wide-web, what we might call *open source* spreadsheets; the remaining 97 spreadsheets were ob-

[1] http://www.eusprig.org/horror-stories.htm
[2] http://sourceforge.net/blog/sourceforge-myths/
[3] https://www.openhub.net/explore/projects

tained largely from textbook examples. What is missing is a substantial set of *closed source* spreadsheets, that is, spreadsheets that were not intended to be made available to the public.
- The EUSES corpus is not publicly available. To use it, "you must be a researcher in the field of software engineering, end-user programming, human-computer interaction, or usability",[4] and even then, the researcher must explicitly ask for a copy by email.

This paper presents a dataset of spreadsheets and emails about spreadsheets for researchers to explore. The dataset was extracted from the Enron Email archive [9], which is a large set of email messages which were made public during the legal investigation concerning the Enron corporation.

The contributions of this paper are as following:
- An industrial dataset of over 15,000 spreadsheets.
- An analysis of these spreadsheets, including the use of named ranges, built-in and user defined functions.
- A dataset of over 65,000 emails either having a spreadsheet as an attachment or talking about spreadsheets.
- An analysis of these emails, including an analysis of discussed errors and updates.

II. The Dataset

A. Obtaining the emails

First, we requested the most recent version of the dataset, via this website[5]. We got access to v1.3, last updated 29 July, 2013. This version contains 130 folders, one per employee, each containing one or more Personal Storage Table (.pst) files. Pst files are *"is an open proprietary file format used to store copies of messages, calendar events.[...] The open format is controlled by Microsoft who provide free specifications and free irrevocable technology licensing."*[10] The Enron dataset contains 190 of such pst files, totaling 53 Gb in size. The pst files together contains 752,605 eml files, representing an email or note.

B. Extracting the spreadsheets

A single pst file can be opened with, for example, Outlook which can then list all emails with attachments to be subsequently copies. However, performing this operation manually for 190 files is quite labor intensive. Hence, we used Systool's Outlook Attachment Extractor[6] to obtain the spreadsheets from the pst files. The Enron set contains 265,586 attachment files (32.3 Gb), of which 51,572 are Excel files. Among those files are 16,189 unique spreadsheets, based on MD5 file hashes.

C. Spreadsheet analysis

We processed all spreadsheets with the Spreadsheet Scantool, developed at Delft University of Technology. This tool runs on the previously developed Breviz core [2], made for spreadsheet visualization and smell detection. The Scantool gathers metrics on spreadsheet, worksheet and formula level about references, errors and used Excel user defined functions.

III. Basic Characteristics of the Spreadsheets

Table I shows an overview of the characteristics of the spreadsheets of Enron, compared to EUSES. Out of the 16,189 unique files, 15,770 spreadsheets could be analyzed. The remaining 419 files were password protected, corrupt, or otherwise unreadable. In total, our set of analyzable spreadsheets contains of 15,770 files. Their average file size is 113.4 Kb. Their average file size is 113.4 Kb. The biggest file is 41 Mb.

A. Worksheets

In total the 15,770 spreadsheets contain 79,983 worksheets, which is an average of 5.1 worksheets per spreadsheet. EUSES has an average of 3.7 worksheets per spreadsheet. Figure 1 shows the distribution of worksheets over the spreadsheets. The x axis lists the number of worksheets, while the y axis shows the number of spreadsheets that contain that number of worksheets. As we can see from this figure, many spreadsheets have one or three worksheets, which is due to the fact that three is the default number of worksheets Excel puts into any newly created spreadsheet. However, it is not uncommon to have more worksheets: there are 1,652 spreadsheets in the corpus with 10 or more worksheets. The spreadsheet with the most worksheets has 175 worksheets.

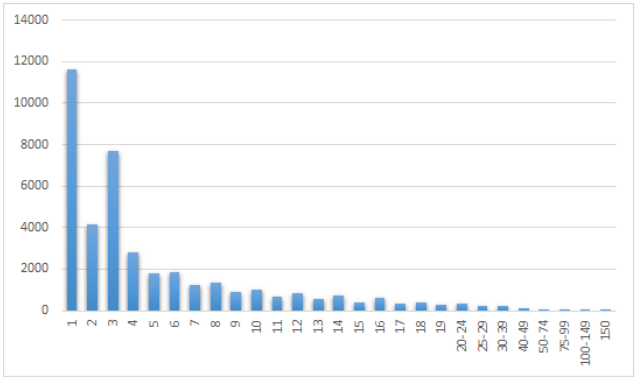

Fig. 1. Distribution of the number of worksheets over the spreadsheet files.

B. Formulas

The 15,770 spreadsheets together have 97,636,511 non-empty cells of which 20,277,835 are formulas. This is an average of 6,191 cells and 1,286 formulas per spreadsheet.

Not all spreadsheets in the Enron set contain formulas: there are 9,120 spreadsheets containing formulas, which is about half of all spreadsheets in the set. Together the spreadsheets with formulas contain 20,277,835 formulas and 913,472 unique formulas. Unique formulas are defined in terms of having an equal relative R1C1 notation[7] and intuitively means

[4] http://eusesconsortium.org/resources.php
[5] http://info.nuix.com/Enron.html
[6] http://www.systoolsgroup.com/outlook-attachment-extractor.html
[7] http://office.microsoft.com/en-us/help/about-cell-and-range-references-HP005198323.aspx

TABLE I
AN OVERVIEW THE SPREADSHEETS IN THE ENRON AND EUSES SET

	EUSES	Enron
Number of spreadsheets analyzed	4,447	15,770
Number of spreadsheets with formulas	1,961	9,120
Number of worksheets	16,853	79,983
Maximum number of worksheets	106	175
Number of non-empty cells	8,209,095	97,636,511
Average number of non-empty cells per spreadsheet	1,846	6,191
Number of formulas	730,186	20,277,835
Average of formulas per spreadsheet with formulas	372	2,223
Number of unique formulas	65,143	913,472
Number of unique formulas per spreadsheet with formulas	33	100

TABLE II
AN OVERVIEW OF EXCEL ERRORS

Error Type	Explanation
#DIV/0!	Trying to divide by 0
#N/A!	A formula or a function inside a formula cannot find the referenced data
#NAME?	Text in the formula is not recognized
#NULL!	A space was used in formulas that reference multiple ranges; a comma separates range references
#NUM!	A formula has invalid numeric data for the type of operation
#REF!	A reference is invalid
#VALUE!	The wrong type of operand or function argument is used

the user has dragged the formula down or right to repeat its calculation on following rows or columns. This means the average spreadsheet *with formulas* contains 2223 formulas of which 100 are unique. In the EUSES set, spreadsheets have an average of 4,186 non-empty cells, 372 formulas and 33 unique formulas, which is substantially lower.

C. Excel errors

It is impossible to determine what cells in the set are errors, as we do not know what was the intention of the formula, so we cannot detect semantic errors. Syntactical errors in the formulas are not possible, as Excel does not allow a user to input a syntactically incorrect formula. However, there is potential for some automated analysis however, by looking at Excel errors (#DIV/0 or #REF).

In a sense, we can compare these errors to *run-time errors* in software. The inputted formulas are syntactically correct, but faulty input (like #DIV/0 or #NUM) or missing references (#REF) result in unwanted behavior when the formula is executed. Table III list all built-in Excel errors[8].

[8]http://www.dummies.com/how-to/content/excel-formulas-and-functions-for-dummies-cheat-she.html

TABLE III
SPREADSHEETS CONTAINING EXCEL ERRORS IN THE ENRON SET

Error type	Spreadsheets	Formulas	Unique Formulas
#DIV/0!	580	76,656	4,779
#N/A	635	948,194	6,842
#NAME?	297	33,9365	29,422
#NUM!	52	4,087	178
#REF!	931	18,3014	6824
#VALUE!	423	11,1024	1751
Total	2,205	1,662,340	49,796

To get insight into the robustness of the Enron spreadsheets, we analyze the number of formulas that result in an Excel error. In total, we have found 2,205 spreadsheets that contained as least one Excel error, which amounts to 24% of all spreadsheets with formulas (14% of all spreadsheets) They together contain 1,662,340 erroneous formulas (49,796 unique ones), which is 585.5 (17.5 unique ones) on average. There were 755 files with over a hundred errors, with the maximum of errors in one files on 83,273 errors. Table III lists the number of spreadsheets, formulas and unique formulas in the Enron set that suffer from given Excel errors.

1) Errors and their dependents: For the erroneous cells, we have analyzed their dependents, meaning those cells that use the erroneous cell as in input. The more cells depend on the erroneous cell, the more impact this error will have on the spreadsheet as a whole and the more 'dangerous' we should consider the formula. Of the 49,796 unique formulas, there are 29,324 with one of more dependents. This is 59% of all unique error formulas, which indicates that spreadsheet errors can have an impact on many other cells in the spreadsheet. On average, erroneous cells have 9.6 other formulas depending on them.

D. Use of Built-In Functions

In addition to basic metrics and Excel errors, we also investigated what functions are used within the formulas. In

TABLE IV
NUMBER OF SPREADSHEETS AND PERCENTAGES OUT OF 9,012 SPREADSHEETS USING FUNCTIONS

Distinct built-in functions	# Spreadsheets	Percentage
≤1	1,822	2.20%
≤2	3,489	38.7%
≤3	**4,922**	**54.6%**
≤4	5,983	66.4%
≤5	6,850	76.0%
≤10	**8,534**	**94.7%**
≤15	8,868	98.4%
≤20	8,962	99.4%
≤25	8,996	99.8%
≤30	9,008	100.0%
≤35	9,012	100.0%

total, there are 9,012 spreadsheets containing formulas with functions (108 spreadsheets contain only "referring" formulas, like =A1 and hence do not use built-in functions). Figure 2 shows the number of used functions for all spreadsheets, where we can see that spreadsheets in the set use a remarkably low number of functions per spreadsheet. The spreadsheet with the most distinct functions, only uses 34 different ones.

Table IV shows the same data. There is is easier to observe that more than half (4,922 spreadsheets = 54.6%) of the spreadsheets with functions uses only three or fewer functions. 8,534 spreadsheets (94.7%) use fewer than 10 built-in functions. This shows that the complexity of spreadsheets does not necessarily stem from in the use of different built-in functions that Excel provides. In fact, over the entire corpus with 913,472 different formulas only 134 of are ever used, out of the 329 functions that Excel provides[9].

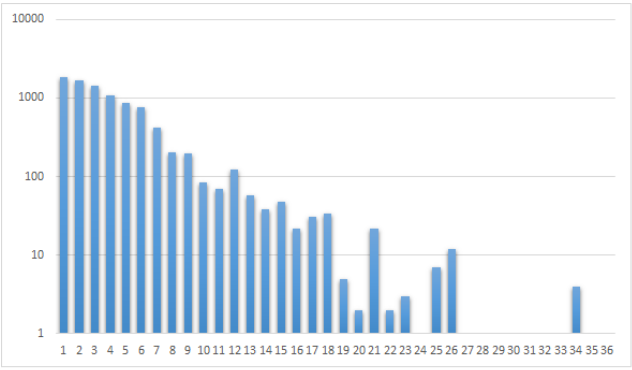

Fig. 2. Distribution of the number of distinct built-in functions over spreadsheet files on a logarithmic scale.

[9]http://office.microsoft.com/en-001/excel-help/excel-specifications-and-limits-HP005199291.aspx

TABLE V
LIST OF MOST USED FUNCTIONS, THE NUMBER OF SPREADSHEETS IN WHICH THEY OCCUR AND CORRESPONDING PERCENTAGES

Rank	Functions	# Spreadsheets	Percentage
1	SUM	6493	72.0%
2	+	5571	61.8%
3	-	4866	54.0%
4	*	3527	39.1%
5	/	3112	34.5%
6	IF	1827	20.3%
7	NOW	1501	16.7%
8	AVERAGE	879	9.8%
9	VLOOKUP	763	8.5%
10	ROUND	606	6.7%
11	TODAY	537	6.0%
12	SUBTOTAL	385	4.3%
13	MONTH	325	3.6%
14	CELL	321	3.6%
15	YEAR	287	3.2%
	Any above	8961	99.4%

There is little variety in which of the 134 functions are used too. Table V shows the 15 most used functions in the corpus, that appear in 8,916 = 99% of all spreadsheets that contain at least one function.

Even more lack of diversity is shown by the Table VI. This table shows the number of spreadsheets and the corresponding percentage that contain *only* the listed function, cumulatively. As an example: 5,733 spreadsheets (63.6%) contain only the top 9 functions: SUM, +, -, /, *, IF, NOW, AVERAGE and VLOOKUP. As can be seen from Table VI, 75,6% of all spreadsheets *only* use the top 15 functions. This means that most spreadsheets construct their formulas from a small and similar set of functions.

E. Use of User-Defined Functions

We also analyzed the use of user-defined functions, functions created with the use of Visual Basic for Applications code. These are not that common, only 47 of the spreadsheets use them. If the are used, most often (41 out of 47 files) only one is used and in 36 of the cases, the defined user-defined function is only used in one unique formula. From this we can conclude that spreasheet users at Enron mainly rely on built-in Excel functions for their calculations. Interestingly enough, there are a number of files that use a self-defined 'Concat' function that basically implements the built-in concatenate function.

F. Use of Named Ranges

Named ranges in spreadsheets have been a research topic for a while, with authors arguing that it is better not to use them [11, 12, 13] In practice it seems they are not used so frequently, only 721 of the spreadsheets use them, which

TABLE VII
NUMBER AND PERCENTAGE OF SPREADSHEETS IN THE ENRON CORPUS THAT SUFFER FROM AT LEAST ONE OF THE SPREADSHEET SMELLS FOR THREE THRESHOLDS.

Smell	Number of files > 70%	> 80%	> 90%	Percentage > 70%	> 80%	> 90%
Multiple References	3,308	2,680	1,955	36.3%	29.4%	21.5%
Multiple Operations	2,709	2,375	1,258	29.8%	26.1%	13.8%
Duplicated Formulas	441	383	352	4.8%	4.2%	3.9%
Long Calculation Chain	2,030	1404	1,000	22.3%	15.4%	11.0%
Conditional Complexity	5,05	420	379	5.5%	4.6%	4.2%
Embedded Constant	2,541	1,652	1,213	27.9%	18.1%	13.3%
Any of the above smells	4,504	3,667	2,772	49.5%	40.3%	30.5%

TABLE VI
THE MOST USED FUNCTIONS, THE NUMBER OF SPREADSHEETS IN WHICH **ONLY** THEY OCCUR AND CORRESPONDING PERCENTAGES, LISTED CUMULATIVELY

Rank	Functions	# Spreadsheets	Percentage
1	SUM	578	6.4%
2	+	1259	14.0%
3	-	2262	25.1%
4	/	2625	29.1%
5	*	3959	43.9%
6	IF	4260	47.3%
7	NOW	5322	59.1%
8	AVERAGE	5664	62.8%
9	**VLOOKUP**	**5733**	**63.6%**
10	ROUND	5990	66.5%
11	TODAY	6182	68.6%
12	SUBTOTAL	6480	71.9%
13	MONTH	6520	72.3%
14	CELL	6774	75.2%
15	YEAR	6812	75.6%

amounts to 8.0% of all spreadsheets with functions. In total, these 721 files contain 4,719 named ranges, which is an average of 6.5 ranges per spreadsheet. Most commonly (in 181 spreadsheets = 25% of range using spreadsheets) files contain only 1 named range. Over half of the spreadsheets (384 = 53%) contain fewer than 5 named ranges.

If we look into the number of cells that a named range refers to, we spot a remarkable pattern. The vast majority of them (3,160 = 67.%) consist of 1 cells only. This seems to indicate that rather than namong actual ranges in spreadsheets, Excel users have a need to define and use 'variables'.

IV. REPLICATION OF PREVIOUS SPREADSHEET WORK

In addition to some basic characteristics of the spreadsheet, we also compare how this Enron set relates to previous analyses we have done on the EUSES.

A. Smells

In previous work we have analyzed so-called *smells* in spreadsheet formulas [7, 15, 14] Other researchers have added to our catalog of smells [16]. For this repeated analysis, we have detect all smells described in [15] and also add a smell from [16].

Smell	> 70%	> 80%	> 90%
Multiple References	23.8%	18.4%	6.3%
Multiple Operations	21.6%	17.1%	6.3%
Duplicated Formulas	10.8%	7.1%	3.7%
Long Calculation Chain	9.0%	7.9%	3.3%
Conditional Complexity	4.4%	3.0%	1.1%
Any of the above smells	42.7%	31.4%	19.7%

Fig. 3. Percentage of spreadsheets in the EUSES corpus that suffer from at least one of the five spreadsheet smells in EUSES corpus, for the three thresholds ([14])

Briefly summarizing the of our previous work: we have set thresholds for the smells based on the EUSES, such that 30% of the formulas were marked smelly at the lowest level, 20% at a moderate and 10% at a high level. As you can see in Figure 3 this resulted in 42.7 % of the spreadsheets to be smelly. For a more extensive background of the spreadsheet smells and the metrics that calculate them, we refer to [14].

We have repeated this analysis for the spreadsheets of Enron, and found they are smelly to a much higher extent then the EUSES files. Figure VII shows the number of files and their percentages that have cells with metrics values above the thresholds as we set them in [15]

In these results, two things stand out. Firstly, the percentage of files with duplication is lower. Duplication means a part of a formula is repeated within another formula, for example, the formula SUM(A1:A5)+B7 has the duplication smell if there is another formula in the worksheet that contains one of the subformulas of this one, like SUM(A1:A5)+B12.

What further catches the eye is the fact that many files (22%) suffer from the 'Long Calculation Chain' smell. This means formulas depend on a long string of other formulas as input. We know from our previous work that it is exactly this that makes spreadsheets hard to comprehend: *"..it is difficult to get a global sense of the structure of the spreadsheet, which requires tracing the dependencies among the cells. Many users in our study described awkward pencil and paper procedures for tracing cell dependencies in debugging spreadsheets."*[17]

We looked into this issue a bit more and found one formula

with a calculation chain of no less then 1205 cells, with 2,401 indirect dependents. Figure 4 shows part of the spreadsheet in which this formula occurs. As you can see, the user had to trace 250 columns to find out the formula depends on a higher row too, after which these have to be inspected, and this patthern repeats several times. However, not all dependents are exactly the same, in some places of the chain different operations are used. Hence, fully understanding this formula's meaning, means tracing along all those other cells, scattered over the spreadsheet. Obviously, not all 'smelly' cells inhibited such a long calculation chain.

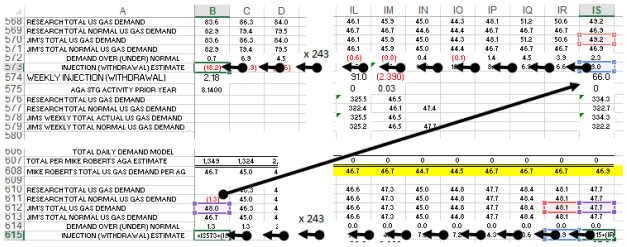

Fig. 4. A formula with a calculation chain of 1,205 cells

In total there are 41,367 *unique formulas* that have a calculation chain of 5 or longer, and are hence marked as smelly. Most of them (21,377 = 52%) have a calculation chain of 5 steps. However, long chains are not that uncommon, 9,471 formulas (23% of smelly unique formulas, 1% of all unique formulas) have a chain of over 7 steps, which is interesting, as psychology research has shown that 7 is about the number of items we can store in our short term memory. If we have to memorize more, the cognitive overhead becomes very high [18].

B. Tests

In additional previous work [19], we have examined the use of *test formulas* in spreadsheets. Test formulas are *"...of the form IF(A1<16,"OK", "margin cannot be over 16") The intention of this formula clearly differs from a calculation formula, like SUM(A1:A5). In the test formula, the user is using the conditional formula construction to express a test and some explanation on why the test could fail.* In the EUSES corpus, 8.8% of spreadsheet files contains such test formulas. In the Enron set, we have found 884 files with tests, which amounts to about 9.6% of the files. This is interesting. Even though the Enron files are vastly larger and more complex, they are not better tested than the EUSES set.

V. The Related Emails

To get insight into the context of the spreadsheets as well, we present an analysis of the email accompanying them.

A. Approach

Again, we started with v1.3, the most recent version of the dataset containing 130 folders with 190 pst files, containing 752,605 eml files from 190 different mailboxes. For the text analysis, we filtered out the 'Contacts' and 'Drafts' folders of the mailboxes. This left us with 717,102 actual email files. The emails span a period of about 15 months, from August 200 to December 2001.

To analyze the emails, we used the pattern.en Python library[10] With pattern, we filtered all emails containing the words: model, spreadsheet, spreadsheets, excel, or worksheet. We also filtered emails that contained a spreadsheet as attachment.

VI. Analysis of the Emails

A. Attachments

In the 717,102 emails, we found 44,214 emails which contained at least one .xls file as attachment, which totals to 6.2% of all emails. As the set contains emails over a 15 month time period, this means about 100 emails with spreadsheets attached were sent each day, so we can safely conclude that emailing spreadsheets was common practice at Enron.

This practice is known to result in serious problems in terms of accountability and errors, as people do not have access to the latest version of a spreadsheet, but need to be updated of changes via email. For example, in 2011, Kern Country in California misjudged their taxable property worth by $1.26 billion, because a 'wrong spreadsheet' was used[11].

B. Emails describing spreadsheets

In addition to emails containing an actual spreadsheet as attachment, there we also emails just talking about spreadsheets. In total there were 24,765 emails (3.5%) with a mention of one of the following words: model, spreadsheet, spreadsheets, excel, or worksheet, while not having a spreadsheet attached. This means out of the total set of 717,102 emails there are 68,979 either sending or mentioning a spreadsheets, which is 9.6% of emails.

In some cases, the emails describe actual instances of spreadsheets without attaching them: *"In the final phase of developing [...] I have planned a one day site visit [...] to finalize the spreadsheet for EOTT crude oil tank environmental compliance"*

"The spreadsheet which is utilized in the UK also has curves mapped to a20 2-3-year old set of Factors derived from US Nat Gas"

One could argue that such vague referral to spreadsheets is even more error prone as it is not entirely clear what spreadsheet is meant. In other cases, people requested a certain spreadsheet via email *"Can you please send us your excel spreadsheets that were used for generated the graphs??"* which indicates those spreadsheets were stored locally and not made available on, for example, a network share.

Finally, there we also cases in which not a concrete spreadsheet was discussed, but spreadsheets were taked about in a more generic context. For example, a vacancy for the

[10] http://www.clips.ua.ac.be/pages/pattern-en

[11] http://www.bakersfieldcalifornian.com/business/x986931070/County-overlooks-then-finds-taxable-property-worth-1-26-billion

position of trader was described as follows:

"ESSENTIAL REQUIREMENTS: Advanced Excel spreadsheet skills."

This is especially is interesting, as later in the email it is explained that, for this position as trader, *"Trading experience is a plus"* only.

C. Emails describing problems with spreadsheets

Furthermore, we have looked at at what other words those emails contain. We especially have an interest in emails that discuss problems with spreadsheets. Hence we also analyzed the occurrence of words related to errors within the 68,979 spreadsheet related emails. We searched the emails for the following words: error, mistake, problem, discrepancy, anomaly, anomalies, incorrect, bug, fault and failure.

These words occur quite common in the Enron set. Of all 68,979 emails that discuss spreadsheets, 4,140 contain words related to errors, which is 6.0% of all spreadsheet related emails and 0.6% of all emails. From the emails, we get the impression that problems with spreadsheets are common, it does not look like a phenomenon that needs further immediate attention. For example:

"For yet another day we seem to be having problems including all the schedules in our EPE schedule sheet."

"The minimum runtime is the one that is garnering the most attention, but there is another parameter that would appear to be modeled incorrectly."

"This was the original problem around the pipe option spreadsheets which we discovered yesterday and the reason why the numbers did not match between the old and new processes."

"The EOL deal will error out in Spreadsheet - Natural Gas, therefore you won't see it erroring out under Sitara."

There are some cases in which spreadsheets were explicitly reviewed: *"Dear Louise and Faith, I had a review of the spreadsheet and noticed an error in allocation of value for the Central Maine Power deal ."*

Sometimes, people discuss the testing of spreadsheets: *"The analytical approach is implemented in a spreadsheet and fully tested already so there will be no problems with the algorithm itself."*

D. Emails describing modification of spreadsheets

Finally, we know that different versions of spreadsheets being emailed around pose a threat to spreadsheet correctness. Hence, we also looked for emails containing the following: new version, update, change, revision, revising, revised. Out of the total of 68,979 email concerning spreadsheets, 14,084 (20.4%) contain these change related words (2.0% of all emails). A few example extracted from the emails:

"Please, find attached an excel spreadsheet that was reviewed and updated by the environmental team"

"Five of the thirteen teams which have EOTT facilities have been inspected and appropriate changes made to the database and spreadsheet."

"I have attached the most recent update of the tank spreadsheet for you to pass on to the teams"

These quotes indicate how common it is to pass emails around, and even have other send them further. This point to the lack of a version control system, and the reliance on people to email spreadsheets around.

E. Emails as documentation

While inspecting the emails related to spreadsheets, we found that the emails were often used to describe the functionality of the attached spreadsheets, a few examples:

*"The value applicable for next year will be calculated as follows: PFC ratio for next year 3D (Applicable percentage * NOx Budget for next year)/ Banked allowances 3D (10% * 100)/ 25 3D 0.4 This implies that 40% of 25 will be available at face value but the balance 60% of 25 will be available at 50% (variable in the Inputs sheet) of the face value!"*

"Doug, I have looked through the preliminary 2002 Corporate Allocations spreadsheet and summarized for European Government Affairs as follows : (All in thousands) Plan 2001 Actual YTD 2001 Plan 2002 Government Affairs Environment 70 41 86 Environmental Policy & Compliance 76 44 27 Managing Director Government Affairs 0 0 200 Total 146 85 313"

Sometimes, even documentation is provided about spreadsheets not even attached to the messages: *"This directory also hosts the traders models, position managers, one off spreadsheets and custom databases."*

VII. Discussion

In the above sections, we have described the an industrial set of spreadsheets and emails extracted from the Enron archive. Arguably, because the data set was obtained by subpoena rather than voluntarily, it is an accurate depiction of a slice of the information industry. In this section, we discuss a variety of issues that affect the applicability and suitability of the proposed approach.

A. Cleaned dataset

The Enron dataset has been cleaned before it was made available. During this cleaning process, potentially sensitive, personal information was removed from it, including credit card numbers, personal contact details, resumes, Social Security or other national identity numbers, dates of birth and information of a highly personal nature such as medical or legal matters. In total, over 10,000 emails were removed from the set for these reasons [20].

We believe however this cleaning hardly affects the analysis of emails related for spreadsheets, as they were most likely not deemed personal by the researchers cleaning the set.

B. Age of the data

The emails in the set were send between 2000 and 2001, which is over a decade ago. However, we believe many of our findings, like frequent emailing, complex formulas with little built-in functions and few named ranges, still hold for today's spreadsheets. Since spreadsheets have a long life-span [21], many of the spreadsheets in use today will stem from years ago.

C. Error analysis

When analyzing the Excel errors, one can of course wonder which of these errors are 'real errors', the type endangering the correct functionality of the spreadsheet. Some of the built-in errors could be the result of missing values. This holds for #DIV/0! and #N/A. The other errors however indicate wrong use of built-in functions, missing names or missing references and as such can not be anticipated use, hence they point at some type of vulnerability.

VIII. Related Work

While EUSES is the largest spreadsheet corpus of which we are aware, there are a few other, smaller corpora. Two prominent corpora are the Galumpke and Wall corpora, containing 82 and 150 spreadsheets respectively, both derived from classroom experiments [22]. Powell and colleagues survey other corpora used in field audits, each audit examining between 1 and 30 spreadsheets [23]. To our knowledge, none of these corpora are publicly available. These existing corpora underscore the need for larger, publicly available spreadsheet corpora.

On the other hand, in the general field of software engineering, corpora of many kinds of software abound, such as through GitHub,[12] Open HUB,[13], and SourceForge.[14] Moreover, some presently open-source projects used to be closed-source, such as Netscape Nagivator,[15] SimCity,[16] and Adobe Flex.[17] Arguably, each of these projects allows researchers an opportunity to study closed-source software.

Efforts related to our research in analyzing smells in the the spreadsheets include efforts focused on the automatic identification of code smells by means of metrics. Marinescu [24] for instance, uses metrics to identify *suspect* classes, those classes that might have design flaws. Lanza and Marinescu [25] explain this methodology in more detail. Alves *et al.* [26] focus on a strategy to obtain thresholds for metrics from a benchmark. Olbrich *et al.* furthermore investigates the changes in smells over time, and discusses their impact [27].

[12] https://github.com
[13] https://www.openhub.net/
[14] http://sourceforge.net/
[15] http://en.wikipedia.org/wiki/Netscape_Navigator
[16] http://www.simcity.com/
[17] http://www.adobe.com/products/flex.html

Furthermore, there are papers on spreadsheet metrics, which also measure properties of spreadsheets. In 2004, Bregar published a paper presenting a list of spreadsheet metrics based on software metrics [28]. He however does not provide any justification of the metrics, nor did he present an evaluation. Hodnigg and Mittermeir [29] propose several spreadsheet metrics of which some are similar to Bregar's. Their metrics are divided into three categories: general metrics, such as the number of formulas and the number of distinct formulas; formula complexity metrics, such as the number of references per formula, and the length of the longest calculation chain; and finally metrics related to user defined functions and external sources. Hole *et al.*[30] also propose an interesting approach to analyze spreadsheets in terms of basic spreadsheet metrics, such as the number of functions used, the presence of charts and the complexity of program code constructs to predict the level of the spreadsheet creator.

Other work related to ours include papers which describe desirable properties of spreadsheets. Raffensberger [31], for instance advises to merge references that occur only once. He furthermore states that unnecessary complex formulas with many operations and parenthesis should be avoided. Rajalingham *et al.* [32] also propose guidelines to improve spreadsheet quality, which they base on principles of software engineering. Secondly, there are papers that address common errors in spreadsheets, like [33, 34], together with their causes. Powell *et al.* for instance [35] names conditional formulas among the top three of commonly occurring spreadsheet error categories. Furthermore there is related work on finding anomalies on spreadsheets, for instance the work on the UCheck tool [36, 37, 38]. UCheck determines the type of cells, and locates possible anomalies based on this type system.

We ourselves have worked on spreadsheet smells in previous work [7, 15, 14]. In those papers we have explored both spreadsheet smells at the low level of formulas as in a spreadsheets structure. Those papers followed our earlier work, in which we visualized spreadsheets by means of class diagrams [40] and dataflow diagrams [21]. Recently, other work on spreadsheet smells has been published [16], that aims at smells in values, such as typographical errors and values that do not follow the normal distribution.

IX. Conclusion

The goal of this paper is to give researchers and others access to industrial spreadsheets and related emails to more deeply understand the problems and challenges around spreadsheets. To that end, this paper presents a dataset of over 15,000 spreadsheets and 65,000 emails related to those spreadsheets. We have performed a preliminary analysis of all the spreadsheets and emails ourselves and we believe that this paper is a valuable first step to obtain more insight into the actual use of spreadsheets within companies, with the following contributions:

- An industrial dataset of over 15,000 spreadsheets.
- An analysis of these spreadsheets, including the use of named ranges, built-in and user defined functions.

- A dataset of over 65,000 emails either having a spreadsheet as an attachment or talking about spreadsheets.
- An analysis of these emails, including an analysis of discussed errors and updates.

From these analyses, we conclude that:

- 24% of Enron spreadsheets with formulas contain an Excel error
- There is remarkably little diversity in the functions used in spreadsheets: we observe that there is a core set of 15 spreadsheet functions which is used in 76% of spreadsheets
- The spreadsheets in this set are substantially more *smelly* than the EUSES corpus, especially in terms of long calculation chains.
- Spreadsheet use within companies in common, with 100 spreadsheets email around *per day!*
- Spreadsheets are commonly shared via email, with about 10% of emails concerning spreadsheets, either in topic or as attachment.

X. Future work

This paper gives rise to many directions for future work. First of all, we believe this paper lays ground for more detailed analyses of the both the Enron spreadsheets and the related emails. We encourage this by sharing all data[18], enabling others to both replicate our analyses and perform new ones. A few analyses we think would be especially interesting are, for instance, sentiment analysis on the spreadsheet emails. In addition to more analyses on the Enron set, this paper shown the need for several other lines of work:

A. Version control for emails

When analyzing the emails we observed again that emailing spreadsheets is still the default way of sharing them. While the Enron set is a bit dated, our experience in industry confirms that this is still very common, despite the frequent use of SharePoint within companies that does do some basic form of version control. Hence, this paper underlines the need for more spreadsheet specific version control systems. While there is currently a UK based company[19] that tries to address this, we believe there are still many open challenges to be answered by research.

B. Mining emails for documentation

When analyzing the emails in more detail, we found that often emails contain description of spreadsheets. One interesting angle could be a plugin in Excel that connects with an email client like Outlook and fetches emails corresponding to the spreadsheet the user is currently working with to help understanding it.

Furthermore, we will take a more detailed look into the email, connecting the described errors to actual cells within the spreadsheets, in order to facilitate more studies into the root cause of industrial spreadsheet errors.

[18] www.felienne.com/enron

[19] https://spreadgit.com/

C. Support for using variables

As described above, the typical way in which named ranges are used—i.e. referring to only one cell—indicates the need for variables in spreadsheets, separated from cells. We have observed before that spreadsheet users can feel the need to observe certain values, in our of our previous studies, a subject stated: "it is easy to always have the value in sight when working on the spreadsheet, because then I can see the values I am calculating with" [7]. The results of this paper again give credibility to the hypothesis that in to a certain extent the grid structure of the spreadsheet could be too restrictive. Hence, it would be interesting to explore models in which the grid is mixed with variables or even other programming concept.

References

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

[2] F. Hermans, "Analyzing and visualizing spreadsheets," Ph.D. dissertation, Delft University of Technology, the Netherlands, 2013.

[3] C. M. Reinhart and K. S. Rogoff, "Growth in a time of debt," National Bureau of Economic Research, Tech. Rep., 2010.

[4] J. Cassidy, "The reinhart and rogoff controversy: A summing up," April 2013, the New Yorker.

[5] P. Coy, "Faq: Reinhart, rogoff, and the excel error that changed history," April 2013, businessWeek.

[6] R. Abraham and M. Erwig, "Goaldebug: A spreadsheet debugger for end users," in *Proceedings of the 29th international conference on Software Engineering*. IEEE Computer Society, 2007, pp. 251–260.

[7] F. Hermans, M. Pinzger, and A. v. Deursen, "Detecting and visualizing inter-worksheet smells in spreadsheets," in *Proceedings of the 2012 International Conference on Software Engineering*. IEEE Press, 2012, pp. 441–451.

[8] W. Dou, S.-C. Cheung, and J. Wei, "Is spreadsheet ambiguity harmful? detecting and repairing spreadsheet smells due to ambiguous computation," in *Proceedings of the 36th International Conference on Software Engineering*. ACM, 2014, pp. 848–858.

[9] B. Klimt and Y. Yang, "Introducing the enron corpus." in *CEAS*, 2004.

[10] (2014, Oct.) Wikipedia. [Online]. Available: http://en.wikipedia.org/wiki/Personal_Storage_Table

[11] R. McKeever and K. McDaid, "How do range names hinder novice spreadsheet debugging performance?" *CoRR*, vol. abs/1009.2765, 2010. [Online]. Available: http://arxiv.org/abs/1009.2765

[12] ——, "Effect of range naming conventions on reliability and development time for simple spreadsheet formulas," *CoRR*, vol. abs/1111.6872, 2011. [Online]. Available: http://arxiv.org/abs/1111.6872

[13] R. McKeever, K. McDaid, and B. Bishop, "An exploratory analysis of the impact of named ranges on the debugging performance of novice users," *CoRR*, vol. abs/0908.0935, 2009. [Online]. Available: http://arxiv.org/abs/0908.0935

[14] F. Hermans, M. Pinzger, and A. van Deursen, "Detecting and refactoring code smells in spreadsheet formulas," *Empirical Software Engineering*, pp. 1–27, 2014.

[15] ——, "Detecting code smells in spreadsheet formulas," in *Proc. of ICSM '12*, 2012, pp. 409–418.

[16] J. Cunha, J. P. Fernandes, J. Mendes, and J. S. Hugo Pacheco,

"Towards a catalog of spreadsheet smells," in *Proc. of ICCSA'12*. LNCS, 2012.

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

[18] G. A. Miller, "The magical number seven plus or minus two: some limits on our capacity for processing information." *Psychological review*, vol. 63, no. 2, pp. 81–97, March 1956.

[19] F. Hermans, "Improving spreadsheet test practices," in *Center for Advanced Studies on Collaborative Research, CASCON '12, Toronto, ON, Canada, November 18-20, 2013*, J. R. Cordy, K. Czarnecki, and S. Han, Eds. IBM / ACM, 2013, pp. 56–69. [Online]. Available: http://dl.acm.org/citation.cfm?id=2555531

[20] A. Cassidy and M. Westwood-Hill. Removing pii from the edrm enron data set: Investigating the prevalence of unsecured financial, health and personally identifiable information in corporate data. [Online]. Available: http://www.nuix.com/images/resources/case_study_nuix_edrm_enron_data_set.pdf

[21] F. Hermans, M. Pinzger, and A. van Deursen, "Supporting professional spreadsheet users by generating leveled dataflow diagrams," in *Proc. of ICSE '11*, 2011, pp. 451–460.

[22] R. R. Panko, "Two corpuses of spreadsheet errors," in *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*. IEEE, 2000, pp. 8–pp.

[23] S. G. Powell, K. R. Baker, and B. Lawson, "A critical review of the literature on spreadsheet errors," *Decision Support Systems*, vol. 46, no. 1, pp. 128–138, 2008.

[24] R. Marinescu, "Detecting design flaws via metrics in object-oriented systems," in *Proc. of TOOLS '01*. IEEE Computer Society, 2001, pp. 173–182.

[25] M. Lanza, R. Marinescu, and S. Ducasse, *Object-Oriented Metrics in Practice*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.

[26] T. L. Alves, C. Ypma, and J. Visser, "Deriving metric thresholds from benchmark data," in *Proc. of ICSM '10*. IEEE Computer Society, 2010, pp. 1–10.

[27] S. Olbrich, D. S. Cruzes, V. Basili, and N. Zazworka, "The evolution and impact of code smells: A case study of two open source systems," in *Proc. of ESEM '09*, 2009, pp. 390–400.

[28] A. Bregar, "Complexity metrics for spreadsheet models," in *Proc. of EuSpRIG '04*, 2004, p. 9.

[29] K. Hodnigg and R. Mittermeir, "Metrics-based spreadsheet visualization: Support for focused maintenance," in *Proc. of EuSpRIG '08*, 2008, p. 16.

[30] S. Hole, D. McPhee, and A. Lohfink, "Mining spreadsheet complexity data to classify end user developers," in *Proc. of ICDM '09*. CSREA Press, 2009, pp. 573–579.

[31] J. Raffensperger, "New guidelines for spreadsheets," *International Journal of Business and Economics*, vol. 2, pp. 141–154, 2009.

[32] K. Rajalingham, D. Chadwick, B. Knight, and D. Edwards, "Quality control in spreadsheets: a software engineering-based approach to spreadsheet development," in *Proc. HICSS '00*, 2000, pp. 133–143.

[33] Y. Ayalew, M. Clermont, and R. T. Mittermeir, "Detecting errors in spreadsheets," in *Proc. of EuSpRIG '00*, 2000, pp. 51–62.

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

[35] S. Powell, K. Baker, and B. Lawson, "Errors in operational spreadsheets: A review of the state of the art," in *Proc. of HICCS '09*. IEEE Computer Society, 2009, pp. 1–8.

[36] R. Abraham and M. Erwig, "Ucheck: A spreadsheet type checker for end users," *Journal of Visual Languages and Computing*, vol. 18, pp. 71–95, 2007.

[37] C. Chambers and M. Erwig, "Automatic detection of dimension errors in spreadsheets," *Journal of Visual Languages and Computing*, vol. 20, pp. 269–283, 2009.

[38] M. Erwig, "Software engineering for spreadsheets," *IEEE Software*, vol. 26, pp. 25–30, September 2009.

[39] R. Abraham and M. Erwig, "How to communicate unit error messages in spreadsheets," in *Proc of WEUSE '05*, 2005, pp. 1–5.

[40] F. Hermans, M. Pinzger, and A. van Deursen, "Automatically extracting class diagrams from spreadsheets," in *Proc. of ECOOP '10*, 2010, pp. 52–75.

TUD-SERG-2014-021
ISSN 1872-5392