12th EuSpRiG Annual Conference

Spreadsheet Governance - Policy and Practice

University of Greenwich, London 14-15 July 2011
European Spreadsheet Risks Interest Group

EuSpRIG 2011 Conference

Spreadsheet Governance – Policy and Practice

Conference Co-Sponsored By:

CIMCON Software
Information Systems Audit & Control Association
(Northern England Chapter)
Baker Tilly Corporate Finance LLP
UWIC
University of Greenwich
Auditnet.org
Systems Modelling
Operis Plc
KHK Analytics

Proceedings Editors:
Simon Thorne & Grenville Croll

July 2011
University of Greenwich
Proceedings of EuSpRIG 2011 Conference

"Spreadsheet Governance – Policy and Practice

First published 2011

Printed in United Kingdom by Five Star Printing Ltd, Claydon, Suffolk, UK

Held at the University of Greenwich, London.

July 14th-15th 2011

Copyright: European Spreadsheet Risks Interest Group
www.eusprig.org

ISBN : 978-0-9566256-9-4

All rights reserved. Apart from fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, no part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form by any means, electronic, mechanical, photocopied, recording or otherwise, without prior permission of the Publisher who is the Committee of the European Spreadsheet Risks Interest Group (www.eusprig.org).

Conference Organised by
EuSpRIG Committee

The use of registered names, logos, trademarks etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant laws and regulations and therefore free for general use.

No responsibility is assumed by the Publisher for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions or ideas contained in the materials herein.

Cover design by David Ward
EuSpRiG 2011 Conference

“Spreadsheet Governance – Policy and Practice”

EUSPRIG 2011 ORGANISING COMMITTEE

Pat Cleary (Chair), University of Wales Institute Cardiff, UK
Patrick O’Beirne, Systems Modelling, Co. Wexford, Ireland
David Colver, Operis PLC, London, UK
Simon Thorne, University of Wales Institute Cardiff, UK
Grenville J. Croll, KHK Analytics, Bury St Edmunds, UK

EUSPRIG 2011 PROGRAMME COMMITTEE

Simon Thorne (Chair), University of Wales Institute Cardiff, UK
Grenville J. Croll, KHK Analytics, Bury St Edmunds, UK
Jocelyn Ireson-Paine, Spreadsheet Factory
Francoise Tort, STEF ENS Cachan, Paris, France
Eric Bruillard, STEF ENS Cachan, Paris, France
David Ball (Co-opted) University of Wales Institute Cardiff, UK
David Ward, Baker Tilly, London, UK
David Colver, Operis PLC, London, UK
TABLE OF CONTENTS

Author
Nancy Wu
Prowvirtu. Inc.

Dr. Kevin McDaid, Dr Ronan MacRuairi, Mr. Neil Clynch, Mr. Kevin Logue, Mr. Cian Clancy, Mr. Shane Hayes Spreadsheet Research Group, Dundalk Institute of Technology

Tom Grossman
University of San Francisco

Nancy Coster, Linda Leon, Lawrence Kalbers, and Dolphy Abraham
Loyola Marymount University, Los Angeles, Alliance University, Bangalore

Gordon Guthrie, Stephen McCrory
hypernumbers.com

Daniel Kulesz
Institute of Software Technology, University of Stuttgart, Germany

Ruth Mckeever, Kevin McDaid
Dundalk Inst Tech, Ireland

Jácome Miguel Cunha
University of Minho.

Derek Flood, Rachel Harrison, Kevin McDaid
Oxford Brookes & Dundalk Institute of Technology

Zbigniew Przasnyski, Linda Leon, and Kala Chand Seal
Loyola Marymount University, Los Angeles

Grenville J. Croll
KHK Analytics

José Pedro Correia, Miguel A. Ferreira, Software Improvement Group

Bill Bekenn and Ray Hooper
Fairway Associates Ltd

K.S. Preeti, Vijit Singh, Sushant Bhatia, Ekanish Preet Singh, Manu Sheel Gupta
Independent developer Software for Education, Entertainment and Training Activities

Paper
Leveraging User Profile and Behavior to Design Practical Spreadsheet Controls...

Spreadsheets in Financial Departments

Towards Evaluating the Quality of a Spreadsheet: The Case of the Analytical Spreadsheet Model

Controls over Spreadsheets for Financial Reporting in Practice

Beyond The Desktop Spreadsheet

From Good Practices to Effective Policies for Preventing Errors in Spreadsheets

Effect of Range Naming Conventions on Reliability and Development Time for Simple Spreadsheet Formulas

An Empirical Study on End-users Productivity Using Model-based Spreadsheets

Spreadsheets on the Move: An Evaluation of Mobile Spreadsheet Apps

In Search of a Taxonomy for Classifying Qualitative Spreadsheet Errors

An Insight into Spreadsheet User Behaviour through an Analysis of EuSpRIG Website Statistics

Requirements for Automated Assessment of Spreadsheet Maintainability

Workbook Structure Analysis – “Coping with the Imperfect”

Spreadsheet on Cloud – Framework for Learning and Health Management System

Proceedings of the EuSpRIG 2011 Conference "Spreadsheet Governance – Policy and Practice"
ISBN : 978-0-9566256-9-4
Copyright © 2011, European Spreadsheet Risks Interest Group (www.eusprig.org)
Dear Colleagues,

You are very welcome to the Twelfth Annual Conference of the European Spreadsheet Risks Interest Group. This year’s theme is "Spreadsheet Governance – Policy and Practice". This reflects more research bridging the gulf between theory and practice, which has always been our focus.

Currently we have about 3,000 unique visitors to our website per month and an active YahooGroup.

I would like to acknowledge with gratitude our sponsors who have supported us financially and professionally. Our logo sponsors this year, as last year, are Cimcon, and I recommend you to visit their exhibition area and learn about their developing products. We depend also on the support of ISACA Northern England chapter, the University of Greenwich and the University of Wales Institute in Cardiff.

It is my pleasure to once again acknowledge the keen work of our conference and programme organiser Dr Simon Thorne from the University of Wales Institute Cardiff, and David Chadwick of our host university of Greenwich. The committee also depends upon the wise counsel and active support of Ray Butler of ISACA Northern England Chapter, David Colver of Operis PLC, Grenville Croll of KHK Analytics (our previous chair), Angela Collins of BDO (our secretary), Morten Siersted of F1F9, Françoise Tort of the École Normale Supérieure de Cachan, Jocelyn Paine of Spreadsheet Factory, David Ward of Baker Tilly, Dr Kevin McDaid of the Dundalk Institute of Technology, Bill Bekenn of Fairway Associates, Angus Dunn, consultant, and Tie Cheng. Representatives of these organisations have contributed a great amount of expertise in the organising of this conference, the publicity, the proceedings, and much more committee work in the background.

A packed programme of seventeen speakers includes both old friends, Dean Buckner of the UK FSA, Kevin McDaid’s Spreadsheet Research team from Ireland, and Tom Grossman’s team from the USA; and we especially welcome new faces from India, Italy, Portugal, and the Netherlands.

Proceedings of the EuSpRiG 2011 Conference "Spreadsheet Governance – Policy and Practice"
ISBN : 978-0-9566256-9-4
Copyright © 2011, European Spreadsheet Risks Interest Group (www.eusprig.org)
You are going to hear practical recipes, challenges to existing orthodoxy on spreadsheet practices, an exposé of the real world of spaghetti spreadsheets, and detailed technical analysis.

Thank you for your interest and participation, and we look forward to a stimulating and interactive conference!

Kind Regards

Patrick O’Beirne, EuSpRIG Chair 2011
http://www.eusprig.org
http://groups.yahoo.com/group/eusprig
Leveraging User Profile and Behavior to Design Practical Spreadsheet Controls for the Finance Function

Nancy Wu
www.backofficemechanics.com

Abstract: Recognizing that the use of spreadsheets within finance will likely not subside in the near future, this paper discusses a major barrier that is preventing more organizations from adopting enterprise spreadsheet management programs. But even without a corporate mandated effort to improve spreadsheet controls, finance functions can still take simple yet effective steps to start managing the risk of errors in key spreadsheets by strategically selecting controls that complement existing user practice.

Spreadsheet Usage Continues to Thrive in Finance.

Spreadsheets are an indispensable tool for organizations today, particularly in finance. Due to their flexibility, computing power and widespread availability, spreadsheets are often the answer for finance functions that need to supplement their financial system(s) quickly and inexpensively. It is often easier, faster and more cost effective to create an analysis in Microsoft Excel or similar applications than to reconfigure an enterprise resource planning (“ERP”) system to perform the same task. In the last ten years, the heavy reliance on spreadsheets for financial reporting, budgeting, forecasting and planning purposes are consistently observed and echoed by researchers, scholars and business practitioners alike [Panko 1998, Rittweger 2010]. And there is little sign of this trend abating in the immediate future. In 2011, the Financial Executive Research Institute and Robert Half Management Resources jointly published *Benchmarking the Finance Function* (Thompson 2010). Of the 207 public and private sector senior executives surveyed for this benchmarking study, more than 60% of finance functions:

- Still relies on Excel for budgeting and planning
- Still relies on Excel for long term planning
- Do not use an account reconciliation tool or system to reconcile accounts. As manual account reconciliations are mostly completed in Excel, the author deduces that these organizations continue to use spreadsheets to reconcile accounts.

While it might not be possible or desirable to phase out all spreadsheets, organizations can still enjoy the benefits of such a powerful and versatile application by recognizing and mitigating the risks associated with a poorly controlled spreadsheet environment. The intent behind a spreadsheet management program is to treat important spreadsheets as stand-alone applications and apply the same due diligence as other enterprise applications during development and in production. In other words, verifying that business critical spreadsheets have built-in controls to validate input data, ensure logic accuracy, and manage changes to the spreadsheet’s infrastructure.

While the Risks are Acknowledged, Many Organizations Have Not Implemented Formal Spreadsheet Management Programs.
While more and more organizations, including several published as EuSpRIG case studies\(^1\) are demonstrating that a spreadsheet management program can be implemented in an efficient and practical manner at both ends of the cost spectrum, many more organizations continue to operate in a loosely controlled spreadsheet environment. Cost of consulting and software licenses aside, their hesitancy to change this status quo may be attributed to the belief that an effort to govern something that impacts so many people on a day-to-day basis cannot help but be intrusive to end users and hence disruptive to the business. Especially among organizations that have not experienced a significant spreadsheet incident, a program that restricts how spreadsheets are created and used may appear to bear too large of an impact upon individuals in return for benefits that, while readily acknowledged (e.g., reduced risk, increased productivity), cannot be easily quantified to build a business case. In addition, spreadsheets belong to no single department, no single function. Rather, they belong to individuals and groups within all functions, all departments. And understandably, these organizations may find that the challenges that come with an initiative to control an application without a central owner are not insignificant. Other spreadsheet risk practitioners echoed similar sentiments in a prior case study: “Ultimately we formed the view that the central difficulty in controlling end user applications is not discovering them or mitigating their risks…The central difficulty lies in formulating, and getting accepted, an organizational response...All aspects of [the end user computing] project required central coordination, but there is as yet no consensus as to the appropriate location of that coordination function. Despite all the evidence of risk and inefficiency, few managers set about addressing these problems, largely, we believe, because they lack the skills and support to do so (Chambers 2008).

**Success of Informal Spreadsheet Initiatives Hinges Largely Upon User Acceptance.**

For finance and other functions in want of stronger spreadsheet governance, but find their organizations unable to launch an organization wide program, it is still possible to improve the spreadsheet environment by implementing within the function one specific component of an end-to-end spreadsheet program\(^2\).

One specific component in building an effective program is to **define the set of controls** that will govern spreadsheets. With few exceptions, these controls occur within the spreadsheet themselves. And once defined, the expectation is that they will be built into critical\(^3\) spreadsheets. There are numerous controls that can be effective at preventing and detecting errors when incorporated into spreadsheet design and usage. But upon introduction, certain controls are more likely to be assimilated into normal use (and hence become sustainable) while other controls meet more user resistance. Not surprisingly, spreadsheet controls that fall into the former category are often those that complement existing user practice. In the absence of a corporate edict in the form of a formal spreadsheet program that mandates compliance, minimizing intrusion to the stakeholders through careful spreadsheet control design is critical to gaining user acceptance and making the controls “stick.”

**The Finance and Accounting Function’s Profile as it Relates to Spreadsheets**
It is the author’s experience that many finance and accounting functions do not consistently utilize professional developers to design spreadsheets that process critical financial data. Instead, the task falls upon casual spreadsheet designers – individuals with varying degrees of Excel expertise who lack formal instruction in programming and systems design. Very often they are also the users or reviewers of the spreadsheets who must create spreadsheets to support their primary job functions, such as reconcile balance sheet accounts or review the budget to actual variance at month-end. Some of these individuals have undergone employer sponsored Excel training. While others do not have access to formal training programs through work, the internet provides free content that is easily searchable for anyone who is interested in learning Excel. But often, both formal corporate programs and self-study content (e.g., the internet or “how to” books) primarily emphasize features and functionalities in Excel and do not focus on spreadsheet design from a risk and internal controls perspective.

In finance, there also tends to be a preponderance of review and due diligence activities that are done “offline” on paper. Once completed, spreadsheets and supporting documents are printed and submitted for review. The reviewer relies on the printed versions of the spreadsheet to look for mistakes and provide feedback. Because of these user characteristics, the most implementable controls will be those that:

1. Require no advanced Excel technical knowledge – While some casual spreadsheet designers have a strong interest in building Excel expertise, most spreadsheet users and creators just want to “know enough to get by” in order to focus on their core competencies within finance. Introducing controls that require more advanced technical skill does not fit the profile of the typical user and will have a higher likelihood of creating resistance.

2. Enhances review activities on paper – Controls that are deliberately designed to complement and improve review activities that are performed on paper can be placed into use immediately without disrupting existing processes. Consequently, controls that require the individual to change habitual practices will require more education and persuasion.

Three Examples of Suitable Spreadsheet Controls

Based on the profile illustrated above, three examples of powerful spreadsheet controls that fit the above criteria are described below. Note that this should not be viewed as an effort to extract the “best” spreadsheet controls in any and all circumstances, but rather it is a demonstration of how understanding the behaviour of a targeted audience can shape the spreadsheet management approach so that internal control recommendations are more practical and relatable to the individuals who must work with them.

*Data Validity Checks for Key Inputs and Calculations (a.k.a. Check Cells)*

The objective of employing data validity checks, or check cells, is to validate accuracy or reasonableness of a body of data. Depending on the level of precision demanded by the user, check cells which come from a separate source are inputted into designated cells and compared to the data from the main spreadsheet body. How comparable the values are to each other (and in some cases, whether the values match exactly) allows the user...
to evaluate whether the main spreadsheet data produced the expected and correct values, or if an error may have occurred. Examples of check cells include hash totals that should agree with number of input cells imported into the file or a value from an independent source. Additional examples for financial models are described in Self-Checks in Spreadsheets: a Survey of Current Practice by David Colver of Operis Group. Designing validity checks to “gut check” portions of the spreadsheet is not only helpful to the user when updating the spreadsheet, but they are especially good for the reviewer. Seeing the results of various sets of cross-checks from secondary sources improve the quality and efficiency of the review process since independently derived conclusions can help the reviewer frame his or her evaluation of the spreadsheet data.

Clear Data Placement and Labels

The objective of promoting clear data placement and labels is to treat spreadsheets more as applications (i.e., formal) and less as digital scratch pads (i.e., informal) simply through better data organization. Examples of good organization include grouping all input cells to one area and calculation cells in a separate area. Another example includes visually distinguishing input cells, logic cells and output cells for the reader. Examples of good labeling include clearly demarcating column or row content and unit of measure (currency v. volume, thousands v. millions, etc.). Also, bodies of data that are outdated or no longer relevant should be removed from the spreadsheet. A well-organized and clearly labeled spreadsheet is not only less prone to misunderstanding and errors and more efficient to review on paper, but the file also becomes more transferable as new users can learn the contents of the spreadsheet more quickly than with data that is jumbled and vaguely identified.

Display of Constants

First mentioned in the New Guidelines for Spreadsheets [Raffensperger, 2000], the objective of displaying constants is to clearly call out the assumptions that feed into the spreadsheet’s calculations. Spreadsheets that display constants have key manual inputs such as growth projection rates, interest or discount rate and other assumptions in identified in dedicated and distinct cells. Subsequent calculations made based on these constants then reference the cells using formulas. The opposite of this practice is to place numeric constants within the formula cells. The constants become part of the formula string and in effect become “buried.” Displaying constants in dedicated and distinct cells increases user efficiency and accuracy as changes in constants need only to be updated once to affect multiple calculations. Reviewers also benefits as they can easily see the assumptions used right on the spreadsheet instead of being hidden within formulas.

A Case Study

The three controls illustrated above are not only effective but are also simple and straightforward to implement. So it is not unusual to spot them in certain critical spreadsheets on occasion. But even among finance functions that have already taken informal steps, whether knowingly or otherwise, to build in these controls, tremendous improvement opportunities still exist through more consistent and thoughtful implementation. To illustrate, recently the author was invited by the finance functions of two publicly traded U.S. companies to review a selection of spreadsheets that were
considered critical to finance and accounting operations and to provide recommendations on improving the spreadsheet environment. Both organizations were concerned about the health of their business critical finance spreadsheets. Although neither organization had experienced a financial statement error caused by faulty spreadsheets, each group wanted to better understand the control mechanisms within the spreadsheets that would detect any material errors and prevent such an incident from occurring.

While the scope of the review encompassed a broader examination of these spreadsheets’ design, high level findings as well as the review criteria pertaining to the three particular controls are summarized below.

Table 1: Results for Publicly Traded Global Financial Services Organization (2010)

<table>
<thead>
<tr>
<th>Nature of Spreadsheet</th>
<th>File Creation Date</th>
<th>Total No. of Occupied Cells</th>
<th>No. of Calculation Cells</th>
<th>Data Validity Checks for Input and Calculation</th>
<th>Clear Data Placement and Labels</th>
<th>Display of Constants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cash Flow (1 of 2)</td>
<td>2008</td>
<td>4,146</td>
<td>1,427</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Cash Flow (2 of 2)</td>
<td>2003</td>
<td>2,146</td>
<td>1,637</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Consolidated Financial Statements</td>
<td>2008</td>
<td>21,684</td>
<td>41</td>
<td>Yes*</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Equity Rollforward</td>
<td>2005</td>
<td>1,924</td>
<td>1,334</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Other Financials Document #1</td>
<td>2004</td>
<td>1,436</td>
<td>1,085</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Trial Balance</td>
<td>2005</td>
<td>301,787</td>
<td>284,374</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Other Financials Document #2</td>
<td>2007</td>
<td>307</td>
<td>38</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Financial Statement for 10Q Filing (Select Portion)</td>
<td>2005</td>
<td>1,329</td>
<td>283</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

*Calculation check cells were deemed unnecessary in this case.

Table 2: Results for Publicly Traded Global Manufacturing Organization (2011)

<table>
<thead>
<tr>
<th>Nature of Spreadsheet</th>
<th>File Creation Date</th>
<th>Total No. of Occupied Cells</th>
<th>No. of Calculation Cells</th>
<th>Data Validity Checks for Input and Calculation</th>
<th>Clear Data Placement and Labels</th>
<th>Display of Constants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Shares Outstanding Calculation</td>
<td>2000</td>
<td>56,500</td>
<td>38,500</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Lease Amortization Calculation</td>
<td>2006</td>
<td>420,834</td>
<td>122,040</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Inventory reserve calculation (1 of 4)</td>
<td>1997</td>
<td>2,523</td>
<td>883</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Inventory reserve calculation (2 of 4)</td>
<td>1999</td>
<td>2,178</td>
<td>1,590</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Inventory reserve calculation (3 of 4)</td>
<td>1999</td>
<td>106</td>
<td>65</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Inventory reserve calculation (4 of 4)</td>
<td>1999</td>
<td>1179</td>
<td>259</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
Table 3: Review Criteria

<table>
<thead>
<tr>
<th>Condition</th>
<th>Criteria for “YES”</th>
<th>Criteria for “NO”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Validity Checks for Input and Calculation</td>
<td>Check cells are placed within the spreadsheet to verify the accuracy of inputs and the reasonableness of calculated outputs. In spreadsheets with both inputs and calculations, both types of check cells must be present.</td>
<td>Check cells do not exist or are inadequate. Examples of inadequate check cells include: - Circular referencing - Broken link / Error value - Check cells built in for inputs but not for calculated outputs.</td>
</tr>
<tr>
<td>Clear Data Placement and Labels</td>
<td>Spreadsheet houses only data required to execute the task. Cell values are clearly labelled and presented logically. Calculation cells are not mixed in with input cells unless they are visually distinct. No hidden data.</td>
<td>Disorganized representation of information. Data is hidden in sheets, columns and/or rows. Existence of irrelevant or out-dated bodies of data.</td>
</tr>
<tr>
<td>Display of Constants</td>
<td>Hard coded values are displayed independently and labelled in the spreadsheet. Formulas reference the cell location.</td>
<td>Hard coded values are embedded in formula construction without clear labelling. For example: “=780000*.35” where “780000” is the operating income and “*.35” is the corporate tax rate.</td>
</tr>
</tbody>
</table>

What the Results Mean

Table 4: Rate of Compliance with Review Criteria (Derived from Test Results in Tables 1 and 2)

<table>
<thead>
<tr>
<th>Company</th>
<th>Data Validity Checks for Input and Calculation</th>
<th>Clear Data Placement and Labels</th>
<th>Display of Constants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Publicly Traded Global Financial Services Organization</td>
<td>25%</td>
<td>25%</td>
<td>75%</td>
</tr>
<tr>
<td>Publicly Traded Global Manufacturing Organization</td>
<td>33%</td>
<td>33%</td>
<td>33%</td>
</tr>
</tbody>
</table>

At the time of fieldwork, neither organization had adopted an end-user computing policy. Yet the results illustrate that each of the three controls was implemented successfully in at least two spreadsheets in both organizations, thus further validating their suitability and ease of implementation for the finance function. The results also show that implementation is inconsistent. In total, only two spreadsheets (13% of total) contained all three controls, and five spreadsheets (36% of total) had none of the three built into the file. So while both groups were already designing these three simple yet powerful controls into key spreadsheets, there are still many opportunities to incorporate them into all critical spreadsheets.

Conclusion

Spreadsheet usage continues to thrive in finance with little signs of slowing. In a recent benchmarking study, more than 60% of finance executives surveyed still rely primarily on Excel for budgeting, long term planning and account reconciliations [Thompson 2010]. An effective spreadsheet management program can lead to more reliability in the financial reporting process, less time dedicated to manually intensive mechanical review as well as more capacity available for other value-added activities in finance. But the perceived disruption to the organization, particularly to spreadsheet users and reviewers,
and the decentralization of spreadsheet ownership may be preventing more organizations from taking action at an enterprise level. Absent a formal effort to manage important spreadsheets, finance functions in want of stronger spreadsheet governance should not be discouraged to act. Because by understanding the function’s user profile and tailoring spreadsheet controls to complement user practice, it is possible for finance to successfully implement simple yet effective spreadsheet controls within its own sphere of influence, such as Data Validity Checks, Clear Data Placement & Labels and Display of Constants. While these measures are straightforward and undemanding in nature, the author’s experience shows that they are far from being fully exploited by organizations today.

Footnotes


3. A spreadsheet’s criticality is determined by how important the spreadsheet’s purpose and output is to the organization. Within finance, the most critical spreadsheets are typically those that can materially affect the financial statements.

4. Concluded based on the author’s professional experience. As further evidence, the author inserted “Excel Training” in Google’s search bar and reviewed the training content offered by the top three search results and the top sponsored result as listed in the companies’ websites: CTS Training, Free Training Tutorial Dot Com, Ozgrid and Microsoft Office – Excel 2007 Training Courses. All were primarily focused on functionality.
References


Spreadsheets in Financial Departments:
An Automated Analysis of 65,000 Spreadsheets using the Luminous Technology

Dr. Kevin McDaid, Dr Ronan MacRuairi, Mr. Neil Clynch, Mr. Kevin Logue, Mr. Cian Clancy, Mr. Shane Hayes
Spreadsheet Research Group
Dundalk Institute of Technology, Dundalk, Co. Louth, ROI,
kevin.mcdaid@dkit.ie

ABSTRACT
Spreadsheet technology is a cornerstone of IT systems in most organisations. It is often the glue that binds more structured transaction-based systems together. Financial operations are a case in point where spreadsheets fill the gaps left by dedicated accounting systems, particularly covering reporting and business process operations.

However, little is understood as to the nature of spreadsheet usage in organisations and the contents and structure of these spreadsheets as they relate to key business functions with few, if any, comprehensive analyses of spreadsheet repositories in real organisations.

As such this paper represents an important attempt at profiling real and substantial spreadsheet repositories. Using the Luminous technology an analysis of 65,000 spreadsheets for the financial departments of both a government and a private commercial organisation was conducted. This provides an important insight into the nature and structure of these spreadsheets, the links between them, the existence and nature of macros and the level of repetitive processes performed through the spreadsheets.

Furthermore it highlights the organisational dependence on spreadsheets and the range and number of spreadsheets dealt with by individuals on a daily basis. In so doing, this paper prompts important questions that can frame future research in the domain.

1. INTRODUCTION

Despite the low level of research in the domain, spreadsheet technology is as fundamental to the smooth running of every financial department as the dedicated financial accounting system that may be in place.

However, with its poor level of control and visibility, spreadsheet usage often slips beneath the radar of management. As a result, an organisation is rarely aware of its dependence on spreadsheets, and of the nature of these spreadsheets and their relevance to key business processes. Furthermore they are rarely aware of the usage of spreadsheets by individual employees and whether this usage is appropriate to the level and experience of the employee.

Dundalk Institute of Technology, along with CUSPA Technologies, has developed a suite of technologies, termed Luminous, which includes a tool for rapidly scanning and analysing large repositories of spreadsheets to discover an organisation’s dependence on spreadsheet usage for key business processes. The tool has been successfully deployed at a number of organisations including
one at which in excess of a quarter of a million spreadsheets were analysed containing 4 billion cells of information.

This paper outlines the results of the tool’s application to the spreadsheet repositories of the financial departments of two firms, one a government sector company of approximately 400 employees, and the second a private commercial company with annual turnover in excess of 300 million euro. In total more than 65,000 spreadsheets were analysed. For the purposes of confidentiality, the names of companies, individuals, and spreadsheet files have been altered.

The paper profiles the spreadsheets in terms of size, cell content and use of functions and other features. It summarises the use of macros and measures the extent of external links. Crucially it examines the use of spreadsheets across business processes over time. Finally, it highlights numerous poor practices involving the linking of spreadsheets.

To the best of our knowledge this paper represents one of the first attempts to comprehensively profile real spreadsheet usage through direct analysis of complete real and current repositories. As such it represents important work in the domain of spreadsheet usage.

The remainder of this paper is structured as follows: Section 2 provides the background for this analysis. Section 3 presents the technology used for collating and analysing data. Section 4 presents results of the respective repository scans. Section 5 discusses the work and draws conclusions, and Section 6 concludes this paper.

2. BACKGROUND

This paper analyses the centrally stored spreadsheet repositories associated with the activities of the financial departments of two large firms.

The first firm, operating in the government sector, has in excess of 400 employees with approximately 10 people employed in financial management and administrative roles. For this paper we name this company Company A. The second firm operates in the Wholesale Food sector employing 1700 people with an annual turnover of approximately 350 million euro. The financial function is larger, with up to 20 people. For this paper we name this company Company B.

As financial departments, the activities of the individuals of the firms will be similar, dealing with core financial and accounting systems. The main difference is that Company B would be involved in significant sales activity, whereas Company A would not. As with most firms spreadsheets are used in many instances act as the glue between the formal systems that allows their smooth, or not so smooth, integration, and in other instances as the tool of choice for reporting. Finally, spreadsheets are also used to automate self contained business processes, separate from the more structured and controlled systems.

The analysis is not conducted on all servers and PC’s for the firms, instead it is pointed only at the key shared repositories. These represent the primary storage areas for the organisations. For Company A, the repository is located on a server running Novell Netware 6.5 with machines accessing this repository typically running Windows XP, and using Microsoft Office 2003. For Company B, the repository is located on a Windows 2003 Server. Machines accessing this repository are running either a version of Windows 2000 or Windows XP and using Microsoft Office 2003 or Office 2007.

The Luminous application found 12,378 spreadsheet files for Company A and 57,446 files for Company B.
Related Research

We understand that a number of vendors of Enterprise Spreadsheet Management software, including CIMCON at Conference of European Spreadsheet Research Interest Group 2010, have presented summary information of server based repositories. However, we are not aware of any other formally published research that profiles entire spreadsheet repositories, particularly with regard to the business function of the organisation.

That said, numerous studies have examined real spreadsheets. In particular Powell, Baker and Lawson have examined for correctness a very small sample of spreadsheets from 5 different firms [1]. The spreadsheets were selected by the firm and this cannot be considered as a random sample from the organisations. Other authors have also examined real spreadsheets for the purposes of determining reliability. This work is summarised by Panko in [2] with a number of repositories of spreadsheet errors presented by the same author in [3]. The most complete survey of spreadsheet usage based on survey methods is given in [4], also by Powell, Baker and Lawson.

The EUSES consortium has gathered a repository of approximately 4000 spreadsheets for research purposes. They have used this set in a number of publications including [5]. The question arises as to how well this group reflects use of spreadsheets in industry. We investigate this repository and compare it with the spreadsheets we have analysed.


3. TECHNOLOGY

The Luminous tool is a standalone application that can analyse large repositories of Microsoft Excel spreadsheets in a highly efficient manner. Importantly, it does not require an installation of Microsoft Office on the host machine to run and can be pointed at a single or a number of mapped server and/or local locations. Its efficiency has been demonstrated by the successful application to other repositories of in excess of quarter of a million spreadsheets covering in excess of 100Gb of spreadsheet information.

Luminous analyses spreadsheets on a file by file basis. It does not process files of type older than Microsoft Excel 95, and files that are password protected can only be analysed if passwords are provided. For the analyses of Company A and Company B passwords were not provided and so password protected files were not analysed.

Luminous provides a wealth of information on each spreadsheet including file level information such as size, user who last saved the spreadsheet, date file created, date last modified and location.

In addition, based on an individual analysis of each cell and object in each spreadsheet the application provides details of cell content, use of inbuilt Excel function usage, existence of objects such as pivot tables etc, extent and nature of links between and across spreadsheet files. Furthermore, Luminous provides an add-on to the tool allows the analysis of macros contained in spreadsheets.

4. RESULTS

The analyses of the repositories are summarised in this section. Given the extent of information gathered through the application and the size of the repositories, this paper represents an initial summary rather than a complete analysis of the data. Following further analysis we intend to publish
more detailed results. Data for Company A is first presented followed by data for Company B. There is some analysis included for Company B that is not, due to pressure of space, included for Company A.

4.1 Company A

File Size

*Luminous* detected 12,378 workbooks, successfully processing 12,161 (98%), amounting to 11Gb of data. Of the remaining 217 workbooks, 157 are pre-1995 Excel files not supported by the scan technology; 12 are password protected; and the remaining 48 were not processed due to being either corrupt or of an older format. Table 1 below shows a distribution of file sizes. Note that the largest file was 280Mb, and that 13% of files are greater than 1Mb in size.

Table 1: Distribution of File Size

<table>
<thead>
<tr>
<th>File Size</th>
<th>Number of Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;=10Kb</td>
<td>75</td>
</tr>
<tr>
<td>10Kb - 50Kb</td>
<td>5,122</td>
</tr>
<tr>
<td>50Kb - 100Kb</td>
<td>1,956</td>
</tr>
<tr>
<td>100Kb - 500Kb</td>
<td>2,839</td>
</tr>
<tr>
<td>500Kb - 1 Mb</td>
<td>557</td>
</tr>
<tr>
<td>1Mb - 10 Mb</td>
<td>1,520</td>
</tr>
<tr>
<td>10Mb - 50Mb</td>
<td>62</td>
</tr>
<tr>
<td>50Mb - 100Mb</td>
<td>10</td>
</tr>
<tr>
<td>100Mb - 150Mb</td>
<td>2</td>
</tr>
<tr>
<td>150Mb - 200Mb</td>
<td>17</td>
</tr>
<tr>
<td>&gt;200Mb</td>
<td>1</td>
</tr>
</tbody>
</table>

| Percentage | 1% | 42% | 16% | 23% | 5% | 12% | 1% | 0% | 0% | 0% |

Structure of Spreadsheets

The analysis of individual worksheets indicates that on average a file contains approximately 160 used columns and 1,500 used rows spread over 7 sheets involving approximately 10,000 cells on average. However the median values are lower with 3 sheets, 185 rows and 23 columns illustrating the skewed nature of the distribution for this data.

The breakdown by file of the number of rows, columns and sheets across the repository is presented in Table 2 below. It shows the number of spreadsheets which contain more than the stated number of rows, columns or sheets. Interestingly it shows that 18% spreadsheets contain more than 10 worksheets. In this context we will later examine how workbooks with a large number of sheets are used to manage repeated business process activities over time.

Table 2: Distribution of Percentage of Rows, Columns and Sheets for Company A

<table>
<thead>
<tr>
<th>Criteria</th>
<th>&lt;=10</th>
<th>&gt;10</th>
<th>&gt;50</th>
<th>&gt;100</th>
<th>&gt;250</th>
<th>&gt;500</th>
<th>&gt;750</th>
<th>&gt;1,000</th>
<th>&gt;10,000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Worksheets</td>
<td>82%</td>
<td>18%</td>
<td>1%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Columns</td>
<td>21%</td>
<td>79%</td>
<td>32%</td>
<td>20%</td>
<td>9%</td>
<td>4%</td>
<td>1%</td>
<td>1%</td>
<td>0%</td>
</tr>
<tr>
<td>Rows</td>
<td>2%</td>
<td>98%</td>
<td>79%</td>
<td>64%</td>
<td>43%</td>
<td>32%</td>
<td>24%</td>
<td>20%</td>
<td>3%</td>
</tr>
</tbody>
</table>

It is interesting to examine the content of the workbooks in that we found that of the 119 million cells containing information 43% contained text data, 25% contained numeric data and 32% contained formulas.
Use of Inbuilt functions

Inbuilt functions in Microsoft Excel are a powerful feature of the spreadsheet application. Our analysis showed that there was no use of the following functions: INDIRECT, TRANSPOSE, INDEX, MATCH, ISERROR, MMULT and AVERAGEIF; and that the IRR function was used in just one cell in the entire repository. Table 3 presents information on the number of cells that reference each of the used functions.

Table 3: Occurrence of Functions for Company A

<table>
<thead>
<tr>
<th>Criteria</th>
<th>OFFSET</th>
<th>VLOOKUP</th>
<th>HLOOKUP</th>
<th>NPV</th>
<th>SUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percentage of files with function</td>
<td>0.01%</td>
<td>11.41%</td>
<td>1.23%</td>
<td>0.02%</td>
<td>65.90%</td>
</tr>
<tr>
<td>Max number of functions in individual file</td>
<td>35</td>
<td>229,638</td>
<td>18,584</td>
<td>2</td>
<td>29,317</td>
</tr>
<tr>
<td>Average number per file when present</td>
<td>35</td>
<td>14,698</td>
<td>9,617</td>
<td>2</td>
<td>723</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Criteria</th>
<th>SUMIF</th>
<th>AVERAGE</th>
<th>COUNT</th>
<th>COUNTIF</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percentage of files with function</td>
<td>0.26%</td>
<td>0.26%</td>
<td>1.17%</td>
<td>0.09%</td>
<td>14.41%</td>
</tr>
<tr>
<td>Max number of functions in individual file</td>
<td>763</td>
<td>42</td>
<td>18</td>
<td>414</td>
<td>114,841</td>
</tr>
<tr>
<td>Average number per file when present</td>
<td>474</td>
<td>12</td>
<td>1</td>
<td>264</td>
<td>6,430</td>
</tr>
</tbody>
</table>

The data shows that, for example, 0.09% of the spreadsheets use the COUNTIF function with an average of 264 cells in these workbooks using the function. One file included 414 cells that call the function.

The most popular functions are SUM, IF and VLOOKUP. Surprisingly, while VLOOKUP is called in 11% of all files compared with 66% for the SUM function, the average number of times the VLOOKUP function is used in files which contain the function is 14,698 compared with 723 for the SUM function.

Importantly, on examination of some of the files with a large number of VLOOKUP calls, we found that these workbooks often consisted of large sets of disparate data extracted from applications such as accounting systems and involving the matching of these data sets to standard lists. The question must arise as to the efficiency of these operations and the potential for investment in automated solutions, either embedded in Excel or not, to reduce the time and to increase the reliability of these tasks.

The application detected that 2.7% of files contained pivot tables, with an average of 2 present per file for those files with pivot tables, and that 1% of files contain active auto filters with an average of 4 in use. The application also detected 570 workbooks (4%) that contain a macro of some sort. More detailed macro analysis is presented later for Company B.

External Links

Spreadsheets can be linked together through formulas that refer to other spreadsheets. In this way chains of spreadsheets can share information to perform complex business processes. For Company A we found that 26.0% (3,162) of all spreadsheets contained at least one external link. For those with external links we found that each spreadsheet contains on average 4,270 cells that refer to other spreadsheets. More interestingly, although the average number of external workbooks referred to is 2, there are 85 workbooks that refer to more than 10 external workbooks. Additionally, 17 spreadsheets contained in excess of 100,000 cells that reference other external workbooks.
As an indication of the internal structure we found that 25.8% of spreadsheets contain worksheets that link to other worksheets in the workbook with an average of 1,266 cells that link to other worksheets in the same workbook.

4.2 Company B

*Luminous* detected 57,446 workbooks and successfully processed 53,645 workbooks (93%), amounting to 40Gb of data. Of the remaining 3,801 workbooks 536 are pre-1995 Excel files with 3,263 files password protected, and the remaining 2 were not processed due to being corrupt.

**Filesize**

Table 4 shows the distribution of file size for Company B. The largest file is 105Mb, and there are 19% of files greater than 1Mb in size, higher than the 14% quoted for Company A. For both companies the percentage of files below 50kb is just above 40%. Note that the median filesize was 85kb.

<table>
<thead>
<tr>
<th>File Size</th>
<th>&lt;10Kb</th>
<th>10Kb - 50Kb</th>
<th>50Kb - 100Kb</th>
<th>100Kb - 500Kb</th>
<th>500Kb - 1Mb</th>
<th>1Mb - 10Mb</th>
<th>10Mb - 50Mb</th>
<th>50Mb - 100Mb</th>
<th>100Mb - 150Mb</th>
<th>150Mb - 200Mb</th>
<th>&gt;200Mb</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Files</td>
<td>26</td>
<td>21,549</td>
<td>6,921</td>
<td>11,899</td>
<td>2,931</td>
<td>9,755</td>
<td>550</td>
<td>13</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Percentage</td>
<td>0.1%</td>
<td>40.2%</td>
<td>12.9%</td>
<td>22.2%</td>
<td>5.5%</td>
<td>18.2%</td>
<td>1.03%</td>
<td>0.02%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
</tr>
</tbody>
</table>

**Structure of Spreadsheets**

As with Company A the average number of sheets in a workbook is approximately 7. For Company B, the average number of rows used was higher, at 2,300 compared to 1,500 (for Company A); however, Company B had a lower number of columns used – 90 compared to 150. The distribution of values for Company B is shown in Table 5 below.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>&lt;=10</th>
<th>&gt;10</th>
<th>&gt;50</th>
<th>&gt;100</th>
<th>&gt;250</th>
<th>&gt;500</th>
<th>&gt;750</th>
<th>&gt;1000</th>
<th>&gt;10000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Worksheets</td>
<td>87.3%</td>
<td>12.7%</td>
<td>1.7%</td>
<td>0.8%</td>
<td>0.2%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>Columns</td>
<td>24%</td>
<td>76%</td>
<td>31%</td>
<td>19%</td>
<td>9%</td>
<td>3%</td>
<td>2%</td>
<td>1%</td>
<td>0%</td>
</tr>
<tr>
<td>Rows</td>
<td>2%</td>
<td>98%</td>
<td>77%</td>
<td>63%</td>
<td>45%</td>
<td>33%</td>
<td>26%</td>
<td>22%</td>
<td>4%</td>
</tr>
</tbody>
</table>

Thirteen per cent of spreadsheets contained in excess of 10 worksheets, less than the 18% for Company A. On the other hand there were 82(0.2%) workbooks each containing more than 250 worksheets. Company A contained no such workbooks.

The average number of used cells was higher, at 17,000 approximately with 40% of cells containing text, 37% of cells containing numeric data and 23% containing formulas based on an analysis of almost 900 million cells of information.
Use of inbuilt functions

For Company B there was no use of the following functions: TRANSPOSE, MATCH, MMULT, AVERAGEIF, NPV and IRR.

As with Company A, the most used functions were SUM, VLOOKUP and IF with 71% of files containing at least one SUM function, 14% containing at least one IF function and 6% containing at least one VLOOKUP. Again, the use of VLOOKUP was extensive for the files in which it was used. The use of the IF function is also very high with almost 25,000 cells using it for files where it is present. As with Company A, the use of other functions in Company B was minimal, apart from the ISERROR function which occurs in almost 3% of files.

Finally, the use of pivot tables was higher at 4% with the use of auto filters significantly higher at nearly 5% of workbooks. More detailed results for all functions and pivot tables and autofilters is shown in Table 6.

Table 6: Number of Occurrences of Functions for Company B

<table>
<thead>
<tr>
<th>Criteria</th>
<th>OFFSET</th>
<th>VLOOKUP</th>
<th>HLOOKUP</th>
<th>NPV</th>
<th>SUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percentage of files with function</td>
<td>0.86%</td>
<td>6.45%</td>
<td>0.48%</td>
<td>0.00%</td>
<td>70.50%</td>
</tr>
<tr>
<td>Max number of functions in individual file</td>
<td>9,958</td>
<td>284,896</td>
<td>42,744</td>
<td>0</td>
<td>69,349</td>
</tr>
<tr>
<td>Average number per file when present</td>
<td>150</td>
<td>4,862</td>
<td>825</td>
<td></td>
<td>456</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Criteria</th>
<th>SUMIF</th>
<th>AVERAGE</th>
<th>COUNT</th>
<th>COUNTIF</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percentage of files with function</td>
<td>2.62%</td>
<td>1.62%</td>
<td>1.70%</td>
<td>0.64%</td>
<td>14.38%</td>
</tr>
<tr>
<td>Max number of functions in individual file</td>
<td>8,076</td>
<td>2,977</td>
<td>30,702</td>
<td>3,648</td>
<td>845,928</td>
</tr>
<tr>
<td>Average number per file when present</td>
<td>882</td>
<td>215</td>
<td>43</td>
<td>211</td>
<td>24,842</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Criteria</th>
<th>INDIRECT</th>
<th>INDEX</th>
<th>ISERROR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percentage of files with function</td>
<td>0.02%</td>
<td>0.14%</td>
<td>2.81%</td>
</tr>
<tr>
<td>Max number of functions in individual file</td>
<td>12,584</td>
<td>148</td>
<td>59,845</td>
</tr>
<tr>
<td>Average number per file when present</td>
<td>1,572</td>
<td>138</td>
<td>2,898</td>
</tr>
</tbody>
</table>

External Links

For Company B, the percentage of files with links to external workbooks was lower, but still significant. 17% of spreadsheets access data from a separate workbook, with each of these workbooks accessing three other workbooks on average. In terms of internal structure, we found that for both Company A and Company B, 26% of spreadsheets contain worksheets that link to other worksheets in the workbook. However, Company B has a higher average of 1,630 cells that link to other worksheets in the same workbook.

The question of the extent and integrity of external links is an important one, and Luminous technology helps to answer this via production of network maps to explain the links between files. We show below one real example (names of files have been minimally altered) where the file Summary Update.xls accesses data from 7 other files. However, there are issues with two of those files. In the first case Copy of Repair and Rental.xls does not exist in the location being called and, in the second case, Food Rec.xls exists but the specific sheet being called does not. We have labelled this as a ghost sheet in the diagram.
We have found that this is not an isolated case and that many of the external links for spreadsheets are effectively broken. Furthermore, there are many cases of files on the server linking to files on local machines. While it would be natural for spreadsheets on local machines to draw data from a central spreadsheet located on the server, it is certainly important to monitor and ensure the integrity of links going in the opposite direction.

In addition we found that many links exist to files that reside in temporary storage spaces on the local machines, many of which indicate links to spreadsheets that were attached to emails and opened from the associated temporary storage locations. This practice indicates a clear lack of awareness of file management.

We acknowledge that many of the links may be legacy ones that may well not have a material impact on the correctness of the spreadsheet for its current use, but the research indicates that this is an area that merits further attention. To that end we intend to conduct more detailed analysis in this area for future publication.

![Link Map with Ghost File and Ghost Sheet](image)

Figure 1: Link Map with Ghost File and Ghost Sheet

There is also a clear issue with the complexity of the link maps that can arise. We present in Figure 2 below a second real example. In this example, *Day Bank Report.xls* has 9 immediate source files, and is a precedent for over 250 other workbooks! One of these dependents, *Daily FX.xls*, has 45 dependents, one of which, *Treasury Dec.xls*, has an invalid data source.

While spreadsheets can be linked to create, what could be termed spreadsheet information systems, the management of such a network will be very problematic, particularly if more than one user is responsible for the different files.
Macro analysis

We include a brief analysis of the macros in the workbooks for Company B. Of the 53,645 workbooks 1,750 workbooks (3%) contain a macro with 1,734 analysed in detail. The average number of lines of code per macro was 56, with 134 containing over 250 lines and 84 workbooks containing over 500 lines of macro code.

Spreadsheets by Date and User

It is interesting to examine the creation and saving of spreadsheets by time and user. For company B the date of last modification for the files starts at 1997 through to March 2011. We show in Figure 3 below a plot of the number of files by date created and by date last modified for months since January 2010. The graph shows that between 600 and 1000 files are created and last modified each month.
Concentrating on the files that have been modified since January 2010 we plot the number last saved by each user, showing the number for each user. The plot ignores those files automatically generated through export from accounting and other systems. In total there are 66 different users in the data set, with only 43 users with more than 10 spreadsheets. Interestingly, the 8 heaviest spreadsheet users are each the last saved user for more than 400 spreadsheets over the 15 month period.
Spreadsheets can be used for single once off tasks. They can also be used for business activities that repeat every day, month, week or quarter or even year. On closer examination we found that the use of spreadsheets for repeated activities can be achieved through the creation of a new workbook for each period or through the creation of a new sheet for each period.

Through examination the folder structure and file names, which were very detailed and well structured, we established that the level of use of spreadsheets for repeated activities was highly significant particularly for monthly, weekly and even daily accounting tasks. This provides a key insight into the issue of spreadsheet proliferation.

We also found, through examination of sheet names, that many activities are repeated through creation of an additional sheet for each period. We found one example of a workbook with 356 sheets, each representing a week from late 2003 through to 2011.

5. DISCUSSION & FINDINGS

This work is very much investigative in nature. At the outset we did not embark on the work with any prior hypotheses as to extent and nature of the repositories we would be analysing. The work seeks to provide an overview, using empirical methods, of the spreadsheets. The work does not explore the usage and development processes for the spreadsheets. While there were discussions with the Chief Financial Officer and with the Head of Finance of Company B, there were no detailed exchanges on the content and function of the spreadsheets within business processes. However, we presented these results to the senior staff, and they were very much in agreement with our findings, and further work is ongoing with Company B to seek to improve the organisation’s use of spreadsheets within their processes.

There is much further analysis of the data that is possible, particularly with regard to the extent of links that are obsolete or are linked to temporary internet files. It is also planned that we will examine the information to establish the any relationships that may exist between key factors. In addition, there
are many ways in which the tool might be improved. One interesting avenue would involve the analysis of cell formulas to separate cells that perform computations from ones that merely relay information contained in other cells possible originating in different worksheets or workbooks. Another improvement would be to measure the number of unique or root formulas that includes each of the functions as well as the number of times these unique formulas are repeated.

A summary of our key findings and the lessons for the research and practitioner community follows. In some cases these findings are unsurprising, but in others they may provide the stimulus for important areas of future research. In considering these findings, we acknowledge the threats to the external validity of these results based on the fact that the results are for just two departments in a single country in two entirely different economic sectors. We would argue that the users of the spreadsheets are very much non-experts, with a very small number of more expert users. However, no formal profiling was conducted.

We also acknowledge that the analysis presented here is very much a high level view. However, and most importantly, we believe that this work represents a natural starting point for the analysis of spreadsheet use in organisations and that the work is important given the lack, to our knowledge, of similar detailed work.

Our key findings are:

- In these organisations the use of excel functions is largely restricted to SUM, IF and VLOOKUP. However the use of VLOOKUP is, to us, surprisingly large with many spreadsheets including functionality, implemented through VLOOKUP, to match very large lists often extracted automatically from accounting systems. Discussions with managers indicated that the development of formulas with VLOOKUP is not due to all users, but rather to a smaller number of more advanced users. However, these spreadsheets are often shared, and given the complexity and issues with VLOOKUP, this may represent a significant risk.

- The prevalence of the IF function is also significance. This is not surprising but to us the extent of its use, on average nearly 25,000 times for Company B, is.

- Although the users are not advanced, there are significant links between workbooks, and the integrity of these links is a clear issue, given that many are calling local files and many other calling files in temporary folders. When presented with link diagrams, managers expressed concern at some of the links which were to older versions of files of ones from previous years and seemed to be unaware of the impact of renaming of sheets and workbooks. The extent of the chains of spreadsheets that result surprised us and, given that many users are unaware of managing links, there are significant issues with the integrity of the links. We acknowledge that many of these links may be legacy ones without any impact on the materiality of the spreadsheet.

- The organisations use spreadsheets extensively for repeating business activities such as weekly, monthly and even, in the case of Company B, daily accounts. This is often performed through copying of the workbook or worksheet for the preceding period and the replacement with up to date values. There are clearly efficiency issues with these activities, and the question for the organisation must be whether a structured automated solution might be better. Alternatively, there may well be an argument to the development of templates or solutions developed through VBA technology.

- Finally, while the majority of spreadsheets are relatively small in size, the organisations develop a high number of large and possibly unmanageable spreadsheets, often arising from the repeated business activities as highlighted in the previous bullet point.
In our view a number of specific lessons can be learned from this research:

1. Researchers and practitioners need to address the extensive use of VLOOKUP in spreadsheets to both establish its reliability and ensure its efficient and effective usage. The work raises the need for technologies that serve to combine, integrate and consolidate large datasets in spreadsheets. In that regard the PowerPivot functionality in Excel 2010 may reduce somewhat the dependence on the VLOOKUP function.

2. There is a need for better knowledge, visibility and management of links between spreadsheet workbooks. There may well be commercial technologies that help in this regard but the issue in the first instance is one of awareness and understanding.

3. There are efficiency issues with the repetition of tasks in spreadsheets, and organisations seem to be unaware of the potential of alternative solutions or use of advanced features of spreadsheets to improve matters.

4. Discussions with managers on the use of VLOOKUP and IF functions highlighted the disparity in knowledge among users of spreadsheets across the organisations. They also highlighted the sharing of workbooks containing elements that not all users were entirely familiar with. This prompts the need for focussed training to counter this disparity and for structuring and protection of workbooks to reduce the associated risks.

More generally, this research prompts the question as to whether the focus of spreadsheet research fits the profile of spreadsheet development in industry as highlighted here. This research shows that many of the spreadsheets are large with limited use of complex functions. There are real efficiency issues around the development and use of this type of spreadsheet. To date, possibly for practical reasons, most of the controlled experiments conducted by researchers such as Panko[2] and Bishop and McDaid[8] deal with smaller spreadsheets.

To guide future research it is important to have access to real and entire spreadsheet repositories. The only publicly available repository is the EUSES one which represents a disparate collection with a large number produced by students for spreadsheet assignments. There is a question as to how representative they are of spreadsheet use in general. We applied the Luminous technology to this set of spreadsheets with interesting results. While we do not have space to detail all the results we found that in general, the EUSES spreadsheets are significantly smaller than the organisations’ ones analysed with on average 31 columns and 296 rows. There was a much lower use of VLOOKUP, SUM and IF with less than 0.2% of files containing VLOOKUP, and, given the disparate sources, there was a very low level of external linking.

This certainly highlights the need for commercial practitioners and organisations to make available spreadsheets for research purposes.

6. CONCLUSION

The use of spreadsheets in real firms is certainly well understood by many expert practitioners who make a living in this domain. However, to the best of our knowledge this work represents the first detailed analysis of complete spreadsheet repositories for two finance departments.
Using the highly efficient Luminous technology we analysed in excess of 65,000 spreadsheets. The analysis was very much empirical in nature concentrating on the structure, functional usage, linking, macro content as well as user and temporal file information.

The analysis found that while many of the spreadsheets were large, they contained a low level of use of in built functions outside of the SUM function. However, there was evidence of relatively high usage of the VLOOKUP and IF functions, extensive linking of spreadsheets and the widespread repetitive use of spreadsheets for periodic business activities. The paper highlights the disparity in spreadsheet knowledge within firms and the lack of knowledge of advanced features. It also highlights potential deficiencies with publicly available spreadsheet repositories and calls for practitioners and organisations to make real spreadsheets available for research purposes.

ACKNOWLEDGEMENT

This work was supported by Enterprise Ireland through Project CFTD/2009/0216.

REFERENCES

Spreadsheets on the Move:
An Evaluation of Mobile Spreadsheets

Derek Flood\textsuperscript{1}, Rachel Harrison\textsuperscript{1}, Kevin McDaid\textsuperscript{2}
\textsuperscript{1}Oxford Brookes University,
\{derek.flood, rachel.harrison\}@brookes.ac.uk
\textsuperscript{2}Dundalk Institute of Technology,
kevin.mcdaid@dkit.ie

ABSTRACT

The power of mobile devices has increased dramatically in the last few years. These devices are becoming more sophisticated allowing users to accomplish a wide variety of tasks while on the move. The increasingly mobile nature of business has meant that more users will need access to spreadsheets while away from their desktop and laptop computers. Existing mobile applications suffer from a number of usability issues that make using spreadsheets in this way more difficult. This work represents the first evaluation of mobile spreadsheet applications. Through a pilot survey the needs and experiences of experienced spreadsheet users was examined. The range of spreadsheet apps available for the iOS platform was also evaluated in light of these users' needs.

1 INTRODUCTION

Advances in technology have enabled mobile devices to allow users to accomplish a wide variety of tasks while away from traditional computing equipment (e.g. desktop or laptop computers). Traditional desktop applications are also now being ported to mobile devices, allowing users to do their computing while on the move.

One such application is the spreadsheet. With modern smart phones such as the iPhone from Apple, and the Curve from BlackBerry, users can now access their spreadsheets from anywhere. However, the small nature of these devices, required for portability has introduced a number of limitations that have caused severe usability problems. Section 2 outlines some of these issues and outlines how these issues impact upon the usability of mobile spreadsheet applications.

This paper is the first work to examine the needs of experienced spreadsheet users for accessing spreadsheets while away from traditional computing devices. By better understanding the needs of these users, more effective mobile spreadsheet applications can be developed that not only meet the needs of these users but also provide a more pleasing user experience.

A pilot survey has been conducted with experienced spreadsheet users. This survey was designed to assess not just the needs of these users but also their previous experience with mobile spreadsheets. In light of the needs of these users, existing spreadsheet applications available on the Apple iOS platform were evaluated. The research methodology employed during this study can be found in Section 3.

The results of this study have shown that there is a need for access to spreadsheets on mobile devices. Existing applications however, suffer from a number of issues that make them difficult to use and therefore provide a poor overall user experience. The full results of this study can be found in Section 4, while Section 5 outlines the lessons that have been learned during the study.

Section 6 outlines some threats to the validity of this work. The study presented here was conducted with a small number of participants which will limit the generalisability of these results. In addition to this only experienced spreadsheet users were targeted for the study. In the future a more extensive study will be conducted with all types of spreadsheet users.
In addition to this further studies will be conducted examining the usability of mobile spreadsheet applications. This study will allow for a deeper understanding of the usability issues associated with these applications. This study is outlined in Section 7 while Section 8 concludes this paper.

2 RELATED WORK

The rapid progression of technology has led to an increase in the number of mobile applications available. Although these applications offer a number of advantages in terms of portability and convenience they do so at the cost of usability. Zhang and Adipat (2005) have highlighted a number of issues that affect the usability of mobile applications:

- **Mobile Context**: When considering mobile applications the user is not tied to a single location. This will also include interaction with nearby people, objects and environmental elements which may distract a user’s attention.

- **Connectivity**: With mobile devices connectivity is often slow and unreliable and therefore will impact the performance of mobile applications which utilise these features.

- **Small Screen Size & Different Display Resolution**: In order to provide portability mobile devices contain very limited screen size meaning that the amount of information that can be displayed is drastically reduced.

- **Limited Processing Capability and Power**: In order to provide portability, mobile devices often contain less processing capability and power. This has the effect of limiting the types of applications that are suitable for mobile devices.

- **Data Entry Methods**: The input methods available for mobile devices are difficult and require a certain level of proficiency. This problem increases the likelihood of erroneous input and decreases the rate of data entry.

The above limitations of mobile devices further aggravate existing usability issues in the spreadsheet application. The limited screen size on mobile devices requires the user to perform considerably more navigation when looking at large spreadsheets. This may cause users to find it difficult to conceptualise the overall spreadsheet and to see how the section on-screen fits with this overall picture.

Flood et al. (2008), have identified navigation as an issue that affects the performance of people debugging spreadsheets through voice recognition technology. By addressing this issue it was found that the performance of users debugging spreadsheets could be increased. It was also found that participants audited more cells with the improved navigation system, which is an important aspect of the debugging process.

Mobile devices generally do not contain a traditional keyboard as the size required would be too large to enable portability. Some devices incorporate a physical keyboard which utilises small keys while other devices use touch screen technology to present a keyboard to the user on screen. These keyboards also require the physical keys to be smaller than traditional keyboards to fit all keys on screen. The iOS platform addresses this issue by providing users with three separate keyboards; one containing letters, one containing numbers and some special characters and a third containing additional special characters.

Chen et al. (2010) conducted an evaluation of users entering text on a small size QWERTY keyboard. This evaluation required 15 participants to enter a passage of text using the small sized keyboard. On average participants used 540 keystrokes to enter the passage of text. The most prevalent type of error made by these participants during the task was a key ambiguity error, which occurred when a user entered a character other than the target character. It was found that on average, participants made
about 9 key errors on the first typing task. It is also worth noting that all participants made at least one error of this type during the study.

Errors of this type, when made on a spreadsheet, may result in a misspelled word or in an incorrect reference in a cell formula, which could alter the bottom line value of a spreadsheet substantially. It has been shown repeatedly that even on desktop computers errors like this persist. Two independent studies (Panko 1998; Powell, Baker et al. 2009) have found that over 85% of the evaluated spreadsheets contained errors.

The limited processing power of portable devices has meant that existing spreadsheet applications may not function correctly when run on these devices. In an attempt to address this issue a number of developers have created spreadsheet apps which scale down the level of functionality to enable users to view and use spreadsheets in a mobile context. Most of these applications however, are limited in terms of functions available and spreadsheet size.

3 RESEARCH METHODOLOGY

During February 2011 a pilot survey was conducted among experienced spreadsheet users to investigate the need for mobile spreadsheet applications and to identify the issues that exist with mobile spreadsheet applications. Participants, recruited through the European Spreadsheet Risk Interest groups’ mailing list, were asked to complete a short survey featuring 24 questions on their experience of using mobile spreadsheet apps and their need for such applications.

In addition to the survey, an evaluation of existing spreadsheet apps for the iOS platform was also conducted. This study examined the usability of existing spreadsheet applications as well examining the range of features they contain. The analysis of the suitability of these applications for the user’s needs is presented here. A full report can be found in (Flood, Harrison et al. 2011).

3.1 Survey on mobile spreadsheets

The aim of this survey was to examine the extent to which mobile spreadsheet applications are required and used. The usability issues associated with existing spreadsheet applications are also investigated. To meet these aims three primary research questions were established:

- **RQ1:** To what extent is access to spreadsheets required while away from traditional computing devices?
- **RQ2:** To what extent is access to spreadsheets used while away from traditional computing devices?
- **RQ3:** What issues affect the usability of mobile spreadsheet applications?

To meet the first research question, a number of aspects were considered. In addition to asking participants if they required access to spreadsheets while away from a traditional computing devices, the survey examined the purposes for which participants required access to the spreadsheet. A deeper understanding of these purposes would allow future mobile spreadsheet applications to optimize the interface for the most common purpose, therefore improving the overall usability of the application.

A number of usability issues can be traced back to the extensive range of features found in modern applications. By identifying the most common features future spreadsheet applications could be designed so as to prioritise access to these features and make them easier to access. Although this will make other features difficult to access, it should produce a more usable application overall.

The biggest advantage of mobile applications is that they can be used anywhere, in any context. In many cases the context will impact upon the users’ level of attention and it is therefore important to consider the contexts in which these applications are needed. Previous research (Schildbach and...
Rukzio 2010) has demonstrated that by altering the target size of a button depending on the context of the user, the performance of the user can be increased.

While RQ1 examines the needs of experienced spreadsheet users, RQ2 examines how users have actually used mobile spreadsheet apps in the past. RQ 3 was designed to examine specific attributes of the usability of current mobile spreadsheet applications. Using participants’ prior experience with mobile spreadsheet applications, the survey examined the level of satisfaction participants have had with existing mobile spreadsheet applications. Participants were also asked about what aspects of the mobile spreadsheet application they enjoyed. Including this information gave the participants an opportunity to highlight some of the positive attributes of these applications that could be utilised in developing future applications.

3.2 Spreadsheet App Evaluation

To better understand existing mobile spreadsheet applications, a systematic evaluation of these apps was conducted. This evaluation focused on apps available for the iOS platform available on mobile devices by Apple. It was decided to focus on this platform initially as Apple is one of the leading providers of mobile devices. It is planned to extend this evaluation to other platforms such as the Blackberry OS and Google Android. The results of this initial evaluation are summarised here in the context of participants’ needs of a mobile spreadsheet application.

The evaluation protocol used during the evaluation is summarised below.

1. **Identify all potentially relevant applications.** There are a number of ways to conduct a search for appropriate applications, including a standard web search, and current software distribution methods make this increasingly easy. Most of the major mobile phone platforms now have an associated online application store. As this work is focused on spreadsheet apps for the iOS operating system, a search of the Apple App store was conducted. The search string “Spreadsheet” was used during this search.

2. **Remove light or old versions of each application.** Many software developers release trial versions of their systems, which are often free. Some of these versions include only a subset of the functionality offered by the full application whilst others allow full access to the application but for a limited time period. These types of applications should be removed if the full version of the app is also included within the search results.

3. **Identify the primary operating functions and exclude all applications that do not offer the required functionality.** The primary operating functions include frequently used functions and also occasionally used functions that are essential for the correct operation of the system in a desired context. For example, the initial system setup might include language and currency settings that would depend upon the country of use. The primary functionality of interest is to allow a user to perform spreadsheet tasks on a mobile device.

4. **Identify all secondary functionality within the remaining apps.** In addition to the primary operating functions, mobile apps will offer users a range of secondary functionalities which can enhance the application. A thorough knowledge of these functions will enable the application developers to see what functionality is available and may present opportunities for additional functionality to be included in future applications.

5. **Install each of the remaining applications, and test each of the tasks using Keystroke level modelling.** Keystroke Level Modelling (KLM) is a well established technique (Card, Thomas et al. 1983) for estimating the time taken to complete certain tasks. This will provide a quantitative measure of the difference between the efficiency of applications. KLM was used to measure the average number of interactions required to enter each of a set of 5 single digit numbers as well as the number of interactions to enter a subsequent formula to total these five numbers.
4 RESULTS

4.1 Survey Participants

In total fourteen participants took part in the online survey. The participants were recruited through the European Spreadsheet Risk interest groups’ mailing list which is comprised of approximately 750 email addresses, featuring members from both academia and industry, who share in interest in the use of spreadsheets and the risks associated therein.

All participants use spreadsheets on a daily basis. When asked which of the following options best describes their level of expertise with spreadsheets; Novice, Intermediate or Expert. 71% of participants rated themselves as experts while 21% rated themselves as Intermediate. The remaining 7% rated themselves as novice spreadsheet users.

To better understand the nature of the participants’ use of spreadsheet participants were asked “In the last week, approximately how many times did you use a spreadsheet”. The responses to this question varied widely, from 6 to over 100 times. As the amount of time spent using a spreadsheet can vary per usage, participants were also asked to estimate how long they spend on average per use of a spreadsheet. Participants were asked to select one of the following options or to enter their own value: Less than 30 minutes, 30 – 60 minutes, 1 – 2 hours, 2 – 3 hours, 3 – 4 hours, over 4 hours. It can be seen from Figure 1 that most participants spend between 30 – 60 minutes per spreadsheet use.

![Duration of use](image)

**Figure 1: Average Duration of spreadsheet use**

4.2 Survey Results

The need for mobile spreadsheets

*RQ1: To what extent is access to spreadsheets required while away from traditional computing devices?*

Approximately 79% of participants said that they required access to a spreadsheet while away from a desktop or laptop computer. These results indicate that there is a strong need for mobile spreadsheet applications.

Purpose of spreadsheet app usage

It is important to consider the reasons why mobile spreadsheets are to be used. To identify this, participants were asked “for what purpose did you need it [access to a mobile spreadsheet while away from traditional computing devices]”. Participants were asked to select all that applied from the following options: *To view spreadsheet data, To change a spreadsheet and To create a new spreadsheet.*
Table 1 shows the percentage of participants who required a mobile spreadsheet application for each purpose. It can be seen that the majority of participants required spreadsheets for viewing and/or changing an existing spreadsheet. Only 21.4% of respondents said that they needed to create a spreadsheet on a mobile device.

<table>
<thead>
<tr>
<th>Purpose</th>
<th>% of respondents</th>
</tr>
</thead>
<tbody>
<tr>
<td>To view spreadsheet data</td>
<td>64.3</td>
</tr>
<tr>
<td>To change a spreadsheet</td>
<td>64.3</td>
</tr>
<tr>
<td>To create a new spreadsheet</td>
<td>21.4</td>
</tr>
</tbody>
</table>

Table 1: Percentage of participants by spreadsheet purpose

A critical component of spreadsheet applications is the ability to transfer data to and from the device easily. Existing spreadsheet apps for the iOS platform allow users to transfer data in a number of ways. The easiest way for users to transfer spreadsheets to the mobile device is through email. When participants receive an email including a spreadsheet, they can open the spreadsheet directly from within the email application. In addition to this some spreadsheet apps allow users to email spreadsheets directly from within the app.

Other methods of data transfer require users to be connected to the device containing the spreadsheet. The most common means is through Wi-Fi, where users can access the mobile device through a web based interface which allows users to transfer files between devices. This approach requires users to type in a specific IP address into the browser after ensuring both devices are connected to the same wireless network.

One final approach offered by a small number of apps (25%) is the use of online storage, such as Google spreadsheets. These applications allow users to log into their Google account and download their spreadsheets to their mobile device. This approach supposes that users have such accounts and store their spreadsheets in these locations. One of the applications evaluated does not allow users to transfer data to and from the device and is solely designed for use on the device.

A number of applications provide access to on-line spreadsheet solutions such as Google spreadsheets. As the access to the internet is not always available on mobile devices these applications were not considered. All of the applications evaluated allow users to operate the spreadsheet on the device alone once it has been downloaded.

Context of spreadsheet usage

The biggest advantage of mobile devices is the range of contexts in which they can be used. Whether out walking or travelling to work, users can access a wide variety of services that allow them to accomplish a range of tasks. In order to determine in what context mobile spreadsheets would be used participants were asked “In what context did you need access to the spreadsheet where no desktop or laptop computer was available”. Participants provided a wide variety of contexts in which they needed access to spreadsheets where traditional computers would be impractical. Some of the contexts cited are presented below:

“Discussing changes in assumptions”
“Could not get to laptop - but had iphone with me.”
“Inbound email onto a mobile device”
“Daily commute”
“To demonstrate data to clients”

The broad range of contexts outlined above is from a small population sample. It is believed that a more extensive survey will produce an even broader range of contexts in which spreadsheets are
required. The context in which the application is being used is an important factor as this will have an impact upon the appropriate design of the system.

**Required features on a mobile spreadsheet app**

As mobile devices are limited in terms of processing power, a full spreadsheet application would be slow and resource intensive. Most mobile spreadsheet applications therefore only supply a subset of the available spreadsheet functionality. In order to determine which functionality is important to them, participants were asked which features they required.

<table>
<thead>
<tr>
<th>Feature</th>
<th>% of participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formula</td>
<td>64.3</td>
</tr>
<tr>
<td>Data Features</td>
<td>57.1</td>
</tr>
<tr>
<td>Graphs/Charts</td>
<td>28.6</td>
</tr>
</tbody>
</table>

**Table 2: Features Required**

Table 2 shows the percentage of participants who needed each feature. It can be seen the most needed feature is formula. Functions can be used to simplify the creation of such formulae. A broad range of functions are available within the mobile spreadsheet apps evaluated. The number of functions varies considerably by app, with some offering a single function, while others offer up to 146 functions. The range of financial functions available is quite limited, with most applications only offering less than 5 functions. MarnerCalc is an exception to this offering users access to 18 financial functions.

Only 25% of the apps evaluated allow users to use graphs and charts. Other features such as the sort function are more common among the apps, appearing in 50% of those evaluated. These results would indicate that existing applications are focused on real users’ needs. It should be noted however that none of the apps evaluated featured the ability to filter data.

In addition to the features included in Table 2, some participants also stated that they required Visual Basic for Applications (VBA) macros, which are often used to enhance the functionality of spreadsheets. During this evaluation of mobile spreadsheet apps however, it was found that none of the applications offered this functionality.

**Frequency of mobile spreadsheet apps**

<table>
<thead>
<tr>
<th>Frequency</th>
<th>% of participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daily</td>
<td>0.0%</td>
</tr>
<tr>
<td>Weekly</td>
<td>21.4%</td>
</tr>
<tr>
<td>Monthly</td>
<td>14.3%</td>
</tr>
<tr>
<td>Less than once a month</td>
<td>50.0%</td>
</tr>
<tr>
<td>Never</td>
<td>14.3%</td>
</tr>
</tbody>
</table>

**Table 3: Frequency by which access to mobile spreadsheets are needed**

Table 3 shows the frequency with which participants have needed access to spreadsheets while away from a desktop or laptop computer. It can be seen that most participants needed access to the spreadsheet less than once a month. 21.43% of the participants did say that they needed this access on a weekly basis. These findings would indicate that memorability is an important aspect of the usability of mobile spreadsheet apps, as most participants use them on an infrequent basis.

**The use of mobile spreadsheets**

*RQ2: To what extent is access to spreadsheets used while away from traditional computing devices?*
In addition to examining the participants’ need to access spreadsheets while away from desktop or laptop computers, the survey also examined participants’ previous experience using spreadsheets on a mobile device. Of the 14 participants included within the study only 42.9% said they have previously accessed spreadsheets on a mobile device. The following results are based on only these participants.

**Purpose of spreadsheet app usage**

<table>
<thead>
<tr>
<th>Purpose</th>
<th>% of participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>To view spreadsheet data</td>
<td>83.3%</td>
</tr>
<tr>
<td>To change a spreadsheet</td>
<td>50.0%</td>
</tr>
<tr>
<td>To create a new spreadsheet</td>
<td>0.0%</td>
</tr>
</tbody>
</table>

Table 4: Purpose of mobile spreadsheet usage

Table 4 shows the percentage of participants who have used a mobile device for each purpose. These results indicate that the most common use of spreadsheets on a mobile device is for viewing existing spreadsheet data. Only half of the participants have used spreadsheets for changing a spreadsheet. None of the participants have created a spreadsheet on a mobile device. These results highlight the need for mobile spreadsheet apps to allow users to transfer and access existing spreadsheets easily.

**Context of spreadsheet usage**

As was seen when examining the need of mobile spreadsheet applications, participants have used mobile spreadsheet applications in a broad range of contexts for a wide variety of tasks. One participant remarked that they have used a mobile device to change assumptions to perform sensitivity analysis while another participant wanted to read a spreadsheet that had been received in an email. Participants have also used mobile spreadsheet applications on their daily commute, a task that may be impractical with conventional laptop computers, due to their cumbersome nature in certain environments (public transport for example).

**Used features on a mobile spreadsheet app**

<table>
<thead>
<tr>
<th>Feature</th>
<th>% of participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formula</td>
<td>66.7%</td>
</tr>
<tr>
<td>Data Features</td>
<td>50.0%</td>
</tr>
<tr>
<td>Graphs / Charts</td>
<td>16.7%</td>
</tr>
</tbody>
</table>

Table 5: Features Used

The features used by participants on a mobile spreadsheet app are presented in Table 5. It can be seen that the most common feature used by the participants is formulae, a central component of a spreadsheet. Data features such as sort and filter options were used by half of the participants. Only 16% of participants used graphs or charts on a mobile device. These results would indicate that more complex features such as graphs and charts are less important to participants than more basic functionality such as formula. The use of data features is typically associated with only certain spreadsheets, i.e. those which contain tables of data. The relatively high frequency of their use would indicate that participants are using this type of spreadsheet frequently on mobile devices.

**Frequency of use**

When asked how often they use spreadsheets on a mobile device, 33% of participants who have used spreadsheets on a mobile device said that they use them on a weekly basis. The majority of these participants however replied that they would only need spreadsheets on a mobile device less than once a month. This infrequency of usage would indicate that the use of a mobile spreadsheet app should be memorable. There are a number of ways in which this could be achieved, for example by replicating existing interface elements from more traditional spreadsheet applications.
The usability of mobile spreadsheets

RQ3: What issues affect the usability of mobile spreadsheet applications?

A number of definitions of usability exist. The ISO define usability as “Extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use” (ISO/IEC 1998). However, Nielsen (Nielsen 1993) also identified a number of attributes that this definition does not include but are important in terms of usability; Learnability, Memorability and Errors.

The authors have identified seven attributes of usability of mobile applications; Effectiveness, Efficiency, Satisfaction, Learnability, Memorability, Simplicity and Cognitive load. Due to the nature of the research method it was not possible to evaluate participants’ experience of all of these attributes. However, the following attributes of usability were examined; Effectiveness, Efficiency, Satisfaction, and Simplicity.

Effectiveness

To evaluate effectiveness participants were asked if they could accomplish their desired task using the mobile spreadsheet application. Participants were asked to select one of three options to answer this question; Yes, No or Partially.

<table>
<thead>
<tr>
<th>Could complete</th>
<th>% of participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>33.33%</td>
</tr>
<tr>
<td>No</td>
<td>16.67%</td>
</tr>
<tr>
<td>Partially</td>
<td>50.00%</td>
</tr>
</tbody>
</table>

Table 7: Effectiveness of mobile spreadsheet apps

Table 7 shows that only 33% of the participants could accomplish their desired task using a mobile spreadsheet application. In addition to this a further 50% could partially complete their desired task. Only 16% of the participants were unable to complete their task. These results indicate that the standard of mobile spreadsheet applications is low for the tasks that participants wish to accomplish. If mobile spreadsheet apps are to be successful their effectiveness will need to be increased dramatically.

Efficiency

The efficiency of participants was not evaluated through the online survey. Participants did describe using spreadsheets on a mobile device as being tedious; indicating the efficiency of these applications is quite poor. During the evaluation of existing spreadsheet apps, the efficiency of these applications was measured using a KLM analysis (Card, Thomas et al. 1983) of creating a simple spreadsheet.

This evaluation found that the efficiency of these applications varied by as much as 100%. To input 5 single digit numbers and then total these numbers in the evaluated apps took between 19 and 36 keystrokes depending on the application. To enter a single digit on some applications took up to 4 keystrokes. The reason for this is that these applications require users to select a cell, open the keyboard, select the numerical keypad and then enter the number. When the application then moves to
the next cell the keyboard is automatically closed forcing users to reopen the keyboard when on this cell.

It was found that some of the spreadsheet apps evaluated allowed users to use shortcuts to enter the sum formula. When selected from a shortcut menu item, the system will automatically infer that the user wants to sum all the numbers within the selected column and will enter the formula to do so. Other applications however require the user to manually type in the formula, which can be tedious and time consuming on the touch screen.

During the evaluation it was found that one spreadsheet app had reversed the usual presentation of numbers and letters to denote the rows and columns. While most traditional spreadsheet applications number the rows and identify columns through the use of letters, this app used letters to denote the rows and numbered the columns. By altering the users’ preconceptions, a new potential of error is introduced especially when users are switching between conventional spreadsheet applications and mobile spreadsheet apps.

**Satisfaction**

The satisfaction of users was measured on a five point Likert scale, where 1 indicated very satisfied and 5 indicated very unsatisfied. Of the six participants who had used mobile spreadsheet apps, the average rating was 3.33, indicating that participants were slightly dissatisfied with the existing applications. There are a number of reasons why users are dissatisfied with these applications.

When asked if they experienced any errors while using the mobile spreadsheet app, 50% of the participants said that they did experience errors. The remaining participants answered no to this question. However a subsequent question asked participants to list all of the problems that they experienced. 83% of the participants listed at least one problem for this.

The most common problem quoted by the participants related to the size of the device. Most participants had problems viewing large spreadsheets on the small screen. One participant also found it difficult to progress with their task. Another participant described spreadsheets on a mobile device as being *Useful but tedious.*

Despite the issues outlined above, some participants had some positive comments to make about the use of spreadsheets on a mobile device. The most common positive comment related to the ability to view the spreadsheet on the mobile device. The high frequency of this type of comment would indicate that participants had low expectations of spreadsheets on these devices and were pleased that they at least existed and therefore are willing to put up with a lot of errors and limitations without it having a large negative impact on the satisfaction they had with the device.

**Simplicity**

Simplicity refers to how well a user can complete their task without errors. When asked if they made any errors while using a mobile spreadsheet app, 50% of participants confirmed that they did. In addition to this participants were also asked if the device was sufficient for their needs. Half of the participants said that the device was sufficient while half disagreed saying it was insufficient. The main reason participants found it insufficient was that the size of the screen was too small. One participant also remarked that there was no access to VBA and therefore the device was insufficient to meet their particular needs.

**5 SUMMARY**

It has been found that there is a strong need for mobile spreadsheet apps. Approximately 79% of the participants surveyed said that they have needed access to a spreadsheet while away from a traditional computing device. The rapid increase in the power of mobile devices and the relatively low development costs associated with mobile applications has enabled the spreadsheet application to be ported to mobile devices.
The limitations imposed by the miniature nature of these devices has meant that mobile spreadsheet applications suffer from a number of usability issues which limit the usefulness of these applications and make them tedious for users to use. The biggest issue identified by participants has been the very small screen size of these devices, meaning that users can only view a small portion of a spreadsheet at a time. The severity of this issue could be reduced through the use of a mini map, which shows a scaled down version of the spreadsheet with the current area being displayed highlighted in the bottom right hand corner of the screen. Similar approaches have been used for looking at images, web pages and other large documents (Burigat, Chittaro et al. 2008).

It was also found that despite the many limitations of mobile spreadsheet applications, users were only slightly dissatisfied with the existing mobile spreadsheet applications. This result shows that the convenience of accessing the spreadsheet on a mobile device out-weighs the negative usability aspects experienced.

Despite the limitations outlined above, spreadsheets can be used on mobile devices. It was found that the most efficient app was Spreadsheet by AppAuthors [http://www.appauthors.com]. This application allowed the test spreadsheet to be created with the fewest interactions, and offers users the ability to sort data and to use freeze panes to keep headings displayed when looking at data. This app however does not allow users to view or create charts.

6 THREATS TO VALIDITY

The results presented above are based on a small number of participants which limits the generalisability of these results. It is hoped to conduct a more extensive survey in the future, sampling not just experienced spreadsheet users but also novice spreadsheet users. This broader sample will enable a more general picture of the need for mobile spreadsheet apps to be attained.

The pilot survey outlined above contained a limited subset of the existing spreadsheet features. A more detailed set of features should be used to determine which features are required by users and which features would not be beneficial if included within a mobile spreadsheet app. Similarly an investigation into the functions most needed by participants would enable mobile apps to better facilitate users’ needs.

The subjective nature of surveys will limit the accuracy of the reported frequency of errors made by participants on mobile devices. A more accurate measure would be to monitor the participants while using a mobile spreadsheet app and record the errors that are made by participants. This will allow for a more detailed understanding of the errors that are made while using mobile spreadsheet apps.

The evaluation of existing mobile spreadsheet apps is limited to those available on the iOS platform. Other mobile platforms such as the RIM featured on Blackberry devices, Android from Google and Windows Mobile may also feature additional spreadsheet applications which would need to be evaluated to provide a complete picture of mobile spreadsheet applications. The devices on which these platforms operate will also vary, providing different modes of input; such as traditional keyboard or touch screen or a combination of both.

7 FUTURE WORK

The pilot survey presented above has shown that there is a need for mobile spreadsheet apps. It is intended to run a more extensive survey including not just experienced spreadsheet users but also novice users. This more extensive survey will allow for a more general understanding of the need for mobile spreadsheet apps.

It is also planned to extend the current survey to examine a more extensive range of spreadsheet features and functions and to determine which of these are most important for mobile spreadsheet users. With this understanding it would be possible to design a mobile spreadsheet app that is
optimised for the real needs of users and therefore puts people at the centre of mobile application
development.

To better understand the issues that are experienced by mobile spreadsheet users, it is intended to run
a controlled experiment which asks participants to use a mobile spreadsheet app to retrieve
information from an existing spreadsheet and to create a new spreadsheet on the device. By recording
the users’ actions while they are performing these tasks it will be possible to see what procedures
users employ while interacting with the device and to see what errors they make during these common
spreadsheet tasks.

Using this information a deeper insight into user behaviour can be gained and used to make
recommendations as to how mobile spreadsheet apps should be designed to produce a more usable
mobile spreadsheet app.

8 CONCLUSIONS

This research has shown that there is a need for mobile spreadsheet apps. A large proportion of
participants surveyed, 78%, have said that they have needed access to spreadsheets while away from
traditional computing devices. The primary needs of these users are to view or change existing
spreadsheets with only 21% of participants saying they would need to create a spreadsheet on a
mobile device.

Along with examining the need for mobile spreadsheet applications, this work also examined the
extent to which existing applications are used. It was found that approximately half of the participants
that required access to spreadsheets while away from traditional computing devices have used mobile
spreadsheet applications. Of these participants only 33% could accomplish their desired task, with a
further 50% stating that they could partially complete their task. To better understand why this was
the case, the usability of these applications was examined.

It was found that existing apps suffer from a number of issues, most predominantly due to the small
screen size of most mobile devices. Participants’ most common complaint of these devices was that
the screen size was too small. One participant also remarked that a lack of a global view was a
problem. Despite these issues, it was found that participants were not dissatisfied with mobile
applications. The convenience afforded by the ability to view spreadsheets on a mobile device
outweighs the usability issues associated with mobile spreadsheet applications.

9 REFERENCES

zoomable user interfaces with overviews. Proceedings of the 10th international conference on Human computer
interaction with mobile devices and services. Amsterdam, The Netherlands, ACM.
Associates.
Guidance on usability.
Computing (JOEUC) 21(3).
Proceedings of the 12th international conference on Human computer interaction with mobile devices and services.
Lisbon, Portugal, ACM.
Towards Evaluating the Quality of a Spreadsheet: The Case of the Analytical Spreadsheet Model

Thomas A. Grossman
Vijay Mehrotra
Johncharles Sander

tagrossman@usfca.edu, vmehrotra@usfca.edu, jgsander@dons.usfca.edu

University of San Francisco, School of Management
2130 Fulton Street, San Francisco, CA  94117-1045, USA

ABSTRACT
We consider the challenge of creating guidelines to evaluate the quality of a spreadsheet model. We suggest four principles. First, state the domain—the spreadsheets to which the guidelines apply. Second, distinguish between the process by which a spreadsheet is constructed from the resulting spreadsheet artifact. Third, guidelines should be written in terms of the artifact, independent of the process. Fourth, the meaning of “quality” must be defined. We illustrate these principles with an example. We define the domain of “analytical spreadsheet models”, which are used in business, finance, engineering, and science. We propose for discussion a framework and terminology for evaluating the quality of analytical spreadsheet models. This framework categorizes and generalizes the findings of previous work on the more narrow domain of financial spreadsheet models. We suggest that the ultimate goal is a set of guidelines for an evaluator, and a checklist for a developer.

Keywords: Spreadsheet engineering, spreadsheet design, spreadsheet analysis, spreadsheet analytics
1. INTRODUCTION

This paper seeks to advance our understanding of a very important question: If I open a spreadsheet I have not seen before, how can I determine whether it is a “good spreadsheet”? From the literature and conversations with practitioners, there is no clear agreement regarding this question.

Discussions at EuSpRIG conferences over the years have approached this question in many ways, sometimes at a high level and sometimes at a very detailed level, but never at a level where clear and coherent agreement could be achieved. It is evident that the apparently simply question of the quality of a spreadsheet is, in fact, very complex. Indeed, the only agreed-upon answer to “is this a good spreadsheet?” would seem be the two words “It Depends”!

The purpose of this paper is to change the conversation regarding “what makes a good spreadsheet”. We think that spreadsheet experts agree more than they disagree. In this paper, we make an attempt to identify the areas of agreement and construct a framework that, we hope, can encompass existing spreadsheet development methodologies and recommendations. Our ultimate goal is to unify the many perspectives, approaches, assumptions, and special cases that exist in the literature and in the hard-won habits of practitioners, to provide a springboard for detailed methodologies for specific types of spreadsheets.

1.1. Overview

We start by articulating a set of principles that we believe are essential for evaluating the quality of a spreadsheet. We then apply the principles to a specific situation, the domain of analytical spreadsheet models. In the conclusions we discuss our hopes for a dialog around the ideas in this paper, and where the work might go next.

In section 2 we propose four principles related to evaluating spreadsheet quality:

1. Any guidelines require a clear definition of where they apply; that is, one must define the “domain” of spreadsheets to which the guidelines apply.
2. The meaning of “quality” must be defined.
3. It is essential to distinguish between the spreadsheet “artifact” and the “process” by which that artifact was created.
4. Spreadsheet quality should be evaluated in terms of the spreadsheet artifact, not the process.

We believe that these principles are essential for any guidelines for evaluating spreadsheet quality.

In sections 3 – 12 we apply the principles for a specific class of spreadsheet. In section 3 we define the domain of “analytical spreadsheet model” and explain the quality criteria that we use. In section 4 we discuss the wisdom of practice that informs the following sections. In section 5 we provide an overview of an initial framework for evaluating the quality of an analytical spreadsheet model. In sections 6 – 12 we discuss elements of the framework.

In the final section, we discuss where we might go next. In particular, it is our hope to develop useful, detailed guidelines for analytical spreadsheet models that are broadly
consistent with the existing guidelines for narrow sub-domains such as large-scale financial planning models.

1.2. Contribution
This paper makes the following contributions.

1. Proposes four principles for evaluating the quality of a spreadsheet.
2. Defines a new domain of analytical spreadsheet models
3. Provides a rigorous framework and terminology for discussing the design quality of an analytical spreadsheet model.
4. Shows how the existing more narrow sub-domain of large-scale financial planning models fits into this domain.
5. Generalizes the existing literature and practice wisdom for large-scale financial planning models to the broader domain of analytical spreadsheet models
6. Hypothesizes on the nature of future contributions, specifically guidelines for evaluating the quality of analytical spreadsheet models, and a checklist for the developers of such models.

This paper does not propose a detailed evaluation methodology, as it would be premature to do so. We must first get the framework right. This paper does not provide an example of evaluating the quality of a specific spreadsheet because that would be premature.

2. PRINCIPLES FOR EVALUATING SPREADSHEET QUALITY

We propose four principles for any evaluation of spreadsheet quality.

2.1. A Clear Domain Definition is Required
The world of spreadsheets is vast and diverse (for example, see Jelen 2005, Grossman, Mehrotra and Özlık 2007, and Powell, Baker and Lawson 2009), ranging from business models to quilting patterns to decorative art. It is difficult to imagine a set of universal guidelines that apply to all spreadsheets. The specificity and usefulness of guidelines increases as the scope of spreadsheets under consideration decreases. Thus, any recommendations will be useful only for a limited set (or “domain”) of spreadsheets.

Therefore, it is essential that a clear domain definition be provided for any spreadsheet guidelines. One of the characteristics of the human experience is a tendency for the word “spreadsheet” to be used to mean “the types of spreadsheets that I and my friends tend to build”. This is perfectly natural, but for the field of spreadsheet engineering to advance it is necessary that guidelines and methodologies carefully define the domain to which they apply. However, the existing literature is characterized by a lack of clean domain definitions and this confounds our ability to understand where methodologies apply.

2.2. Define What is Meant by “Quality”
It is well known in the discipline of operations management that the term “quality” has many different meanings. For example, “quality” can refer to conformance to specification, compliance with tolerances, fitness for use, a count of defects, the extensiveness of features, cost, exclusivity of access, or other meanings. Thus, any activity related to quality must carefully define what aspect of quality is of interest.
This definition of quality will depend upon the purpose of the spreadsheet and the goals of the evaluator. For example, a quilting pattern spreadsheet might be evaluated on how easily it can be observed while sewing, whereas a financial model might be evaluated on accuracy and how easily it can be modified. The relevant quality dimensions will depend upon the purpose of the spreadsheet and the goals of the evaluator. It is essential that these quality dimensions not be assumed obvious or otherwise left unstated. Any evaluation of spreadsheet quality be preceded by an explicit definition of what dimensions of quality are to be evaluated.

2.3. Distinguish between the Artifact and the Process
The most powerful spreadsheet development methodologies are those promulgated for large-scale financial planning models (FAST 2010, Swan 2008, and SSRB 2005, compared in Grossman and Özlük 2010). These very detailed methodologies provide process guidelines for the development of spreadsheet models. The power of these methodologies arises in part because the spreadsheet itself is intimately connected with the way that it was constructed; the layout is intended to support construction, and the construction is intended to build the layout. We hypothesize that a rigorous, detailed process is only possible for a rigorous, standardized design.

However, this tight coupling between process and the artifact that results from the process makes it difficult to compare methodologies. The quality of a spreadsheet model – the artifact – is viewed through the lens of the design process. The effectiveness of the design process is viewed through the lens of its ability to rapidly and robustly create the standardized design. In order to achieve any level of generality in guidelines, it is essential that we separate our thinking about the process of creating a spreadsheet and our study of the spreadsheet artifact. We must distinguish between the “what” and the “how”.

2.4. Evaluate Quality in Terms of the Spreadsheet Artifact
Any useful discussion of spreadsheet quality must be flexible enough to allow for different designs. This means that we must break away from the constraints of process. Spreadsheet quality needs to be defined as an attribute of the spreadsheet, without any knowledge of how the spreadsheet was constructed.

For devotees of a deep spreadsheet development methodology, this can be a challenge, because the quality of a spreadsheet model – the artifact – is naturally considered in the context of the construction methodology. This serves to confuse the evaluation of spreadsheet quality, because quality is in part defined as adherence to an a priori standardized design. We need to define quality independent of the constraints of any particular design.

3. THE CASE OF THE ANALYTICAL SPREADSHEET MODEL

We illustrate the above principles for a particular domain (principle 1) and for a particular set of quality criteria (principle 4). We limit our examination to the spreadsheet artifact, and do not include any process guidelines (principles 2 and 3).

3.1. Domain Definition: The Analytical Spreadsheet Model
We start by defining the domain. We then present a framework that we intend to be helpful for evaluating the quality of any analytical spreadsheet model, regardless of how it was constructed.
The domain is based on our experience teaching in American business schools where we see spreadsheet models grounded in the discipline of management science, as well as models in finance, marketing, supply chain, and other business disciplines. These models, to us, have much in common and we find that our advice to students is similar across a variety of models. Therefore we define the domain as follows.

We define an analytical spreadsheet model as (1) a spreadsheet computer program that (2) implements a mathematical model (3) for purposes of analysis that (4) serves as an organizational asset that is (5) employed in a larger business context. An analytical spreadsheet model has the following properties:

1. As a spreadsheet computer program, the analytical spreadsheet model is programmed in a spreadsheet language such as Excel.
3. The analytical spreadsheet model exists to enable analysis upon it by changing inputs and observing outputs. [Grossman 2008, Powell and Baker 2010, Spreadsheet Analytics 2010].
4. As an organizational asset, the analytical spreadsheet model is intended for use not only by the author but also by the author’s immediate colleagues and/or successors; usage could be intensive use for a short period, or routine use over time.
5. The analysis is performed in the context of a larger business context that has important organizational goals, hence the outputs of the spreadsheet are designed to meet the needs of people besides the author.

We note that an analytical spreadsheet model is different than a “data-driven” spreadsheet model that starts with a large set of numeric values, and then seeks to extract insight from them. A spreadsheet that relies heavily on business intelligence tools such as sort, filter, and pivot table will not normally be considered an analytical spreadsheet model.

An analytical spreadsheet model can be constructed purposefully, or it can arise without intention as a “legacy spreadsheet application” (Grossman, Mehrotra and Özlük 2007). The level of spreadsheet engineering investment in an analytical spreadsheet model is more than that required for an informal “quick and dirty model” or “personal productivity tool”, but less than that required for a formal “spreadsheet application” [Grossman 2007] that is to be deployed to multiple less-sophisticated users.

Note that analytical spreadsheet models are primarily about the business logic embedded in the model. Models that are primarily data-driven, with a large dataset and relatively few cell formulas do not fall in this domain.

Analytical spreadsheet models are used in business, including supply chain management, market research, worker scheduling, finance, as well as in engineering and science. Analytical spreadsheet models are taught extensively in business school courses on management science and quantitative analysis. Examples of analytical spreadsheet models across many areas of business can be found in Powell and Baker 2010, Winston & Albright 2008, Ragsdale 2010, and Interfaces 2008, and a rich variety of finance examples can be found in Benninga 2008. The spreadsheets described in Read and Batson 1999 and Tennent and Friend 2001, and the large-scale financial planning models of FAST 2010, Swan 2008, and SSRB 2005 are analytical spreadsheet models. Most or all of the spreadsheets discussed in Croll 2005 are analytical spreadsheet models. Grossman,
Mehrotra and Özlük 2007 describe analytical spreadsheet models across diverse industries. We indicate the domain of analytical spreadsheet models in Figure 1.

![Figure 1: Venn Diagram for Analytical Spreadsheet Models](image)

We emphasize that this paper considers the spreadsheet as an artifact; we consider the quality of the spreadsheet model in isolation from the manner in which it was created. Thus, we do not address the process by which the spreadsheet is constructed. For example, we do not discuss recommendations for how to go about creating cell formulas, nor how to test or code-inspect a spreadsheet to detect any errors. We note that our guidelines are consonant with the process recommendations of FAST 2010, Swan 2008, and SSRB 2005.

3.2. Criteria for Quality

We are interested in the properties of a “high-quality” analytical spreadsheet model. What are the “right” quality dimensions for an analytical spreadsheet model? This question will undoubtedly spark debate. In the interests of initiating a constructive discussion, we propose that the following quality dimensions. A high-quality analytical spreadsheet model must be:

1. Suitable for efficient analysis
2. Readable (can be understood by a non-author; see McConnell p. 842)
3. Transferable (to another analyst or the author’s successor)
4. Accurate (compute the mathematical model without error)
5. Reusable (can easily be employed again with different data)
6. Modifiable (can be extended or adapted to new circumstances)

These six quality dimensions come directly from the definition of the analytical spreadsheet model. They are, in our view, necessary (we repeat the definition of an analytical spreadsheet model) for a spreadsheet computer program that implements a mathematical model for purposes of analysis that serves as an organizational asset that is employed in a larger business context.

We believe that obtaining agreement on the dimensions of quality will aid in the creation of guidelines and recommendations. In the framework of guidelines that appears in later
sections, each element is devised to contribute to one or more of these quality dimensions.

4. LITERATURE AND THE WISDOM OF PRACTICE

This paper started when we sought to articulate for the benefit of our business students what makes a high-quality analytical spreadsheet. There is no empirical research on this topic. Therefore, as in many practical disciplines, these guidelines are based in the scholarly tradition of “the wisdom of practice” (see Weimer 2006). This paper articulates the wisdom of practice from the literature and our experience, along with our tacit knowledge accumulated from years of experience in industry and academia engaging in business consulting, modeling, software development, and spreadsheet engineering.

We integrate the published results from the spreadsheet engineering community (especially the European Spreadsheet Risks Interest Group), and draw upon the principles of the mainstream software engineering literature. It is our intention that the framework presented here is consistent with the recommendations for business models in Read and Batson 1999 and financial models in Tennent and Friend 2001, and the detailed guidelines for large-scale financial planning models of FAST 2010, Swan 2008, and SSRB 2005, summarized in Grossman 2010.

5. FRAMEWORK FOR EVALUATING THE QUALITY OF AN ANALYTICAL SPREADSHEET MODEL

Our ultimate goal is a set of practical, useful guidelines for evaluating the quality of an analytical spreadsheet model. These guidelines should be specific enough to be useful, but broad enough to encompass the detailed recommendations of the existing methodologies. We believe that any set of useful guidelines for evaluating the quality of an analytical Spreadsheet model should address the following elements:

- Modular design
- Structured design
- Design for input-output usage
- Disciplined information flow
- Distinct inputs module(s) with certain properties
- Distinct model computations module(s) with certain properties
- Distinct reports module(s) with certain properties

We provide a sketch of each of these elements in the following sections.

6. MODULAR DESIGN

A module is a set of similar things. A module can contain subsidiary modules (sub-modules). A module can be a few cells; a portion of a worksheet; an entire worksheet; a set of worksheets; or an entire workbook. A top-level module is a module that is not a sub-module. A bottom-level module is a module that contains no sub-modules.
Each module should have a clear purpose. A module with distinct activities contains sub-modules, each with a clear purpose.

The need for input-output usage discussed above requires that there be three distinct top-level modules: Inputs, Model Computations, and Reports. The Inputs module receives all model inputs in raw form, and adapts them as necessary to the needs of the Model Computations module. The Model Computations module performs the computations necessary to implement a mathematical model that is known to the author. The Reports module presents model results in a form that is convenient to the consumer.

7. STRUCTURED DESIGN

Structured design is the purposeful arrangement of the analytical spreadsheet model into a set of connected building blocks called “modules”. Structured design is a standard technique from traditional software engineering used to manage complexity. Structured design is essential for readability, accuracy, reusability, and modifiability.

In addition, structured design for an analytical spreadsheet model supports input-output usage; is modular; has disciplined information flow; and comprises an Inputs module, a Model Computations module, and a Reports module.

8. DESIGN FOR INPUT-OUTPUT USAGE

For efficient analysis using an analytical spreadsheet model, the analyst must be able to focus on the values for model inputs and resulting model outputs, without being distracted by “internal” model calculations. (I.e., the analyst interacts with the analytical spreadsheet model the same way he interacts with traditional shrink-wrap software, a decision support system, or web analytics software.) The spreadsheet must be designed so that:

- analysis does not involve observation of the model logic.
- inputs are grouped together separate from calculations.
- outputs are grouped together separate from calculations.

Design for input-output usage supports suitability for analysis, and transferability.

Note that design for input-output usage implies that a “calculator” design, where inputs are entered then used for calculation, and more inputs are entered then used for calculation, is unsuitable for efficient analysis. Such a spreadsheet should be converted to an input-output design prior to service as an analytical spreadsheet model.

9. DISCIPLINED INFORMATION FLOW

Information flows through the modules via a direct, non-circuitous path. At the top level, information flows from the Inputs module to the Model Computations module to the Reports module.

Disciplined information flow is summarized in the structure chart [Structure Chart 2011] of Figure 2.
Disciplined information flow insures suitability for input-output usage, and supports modifiability as any future changes to the model logic in the Model Computations module are isolated from the Inputs and Reports.

10. INPUTS MODULE

The Inputs module manages all the inputs necessary for model computations. The design and construction of the Inputs module is a spreadsheet engineering exercise that requires little domain expertise, although some domain expertise can be required for quality assurance.

The Inputs module receives all model inputs in raw form, and adapts them as necessary to the needs of the Model Computations module. The design of the Inputs module can have sub-modules.

We would suggest that it is useful to distinguish among sub-modules. Based on experience with a wide variety of models, we suggest that it is valuable to provide as necessary distinct sub-modules for different types of inputs including Source Data (external inputs that are outside the organization’s control, presented in their original form); Assumptions (inputs that are judgmental, tendentious, or frequently change); Decision Variables (business choices under the organization’s control); and Input Pre-Processing (changes to the other inputs to make them more useful). A spreadsheet model may include other modules as appropriate.

11. MODEL COMPUTATIONS MODULE

The Model Computations module performs the computations necessary to implement a mathematical model that is known to the spreadsheet developer. The logic of the model resides in the Model Computations module. The Model Computations module is the place where the domain expertise of the developer is expressed as a spreadsheet computer program. Tennent and Friend 2001 and Swan 2008 refer to the Model Computations module as the “workings”.

The Model Computations module takes information from the Inputs module, performs computations, and ultimately generates the information required by the Reports module.

What might be flexible, robust guidelines for the Model Computations module? It should allow for one or more worksheets; this could be generalized as sub-modules for different blocks of computations, which would allow multiple worksheets or only a single worksheet. There should undoubtedly be a guideline related to the arrangement of inputs to make them most suitable for computation, or perhaps bringing (“echoing” or “linking”) distant cells to a place close to the computation, to generalize the concept of “ingredients” from FAST 2010.

12. REPORTS MODULE
The Reports module contains all the outputs from the model. The Reports module and sub-modules should be designed and formatted to meet the needs, preferences, and habits of the consumer. Its design may require little or no domain expertise, although domain expertise can be required when devising new, non-standard reports.

The Reports module should do no or only trivial computations, and contain primarily links to cells in the Inputs and Computations modules.

13. CONCLUDING COMMENTS

Our intention is to advance the way we think about the quality of spreadsheets to recognize that diversity in spreadsheets requires multiple, focused evaluation schemes; to make more general certain existing, narrow spreadsheet engineering recommendations, and ultimately provide useful guidelines for evaluating quality. Our intent is to provide the basis for constructive discussion that will lead, in time, to a framework and set of actionable guidelines for analytic spreadsheet models, and also provide an approach for devising guidelines for other domains of spreadsheet models.

We believe that when evaluating spreadsheet quality it is essential to observe four principles. 1) Define the domain. 2) Define what is meant by “quality” prior to evaluating it. 3) Distinguish between the spreadsheet-as-artifact and the process of constructing the spreadsheet. 4) Evaluate spreadsheet quality based only on the spreadsheet itself. We wonder whether there might be other such principles.

We defined a domain of “analytical spreadsheet models”. We indicated the quality dimensions that seem to be important. We ask whether these are in some sense the “right” quality dimensions for this class of models. We then provided a framework for evaluating the quality of an analytical spreadsheet model. This framework obviously requires more detail, but before we add we must first ask whether the elements of the framework are correct.

The next steps are to finalize the framework, add detail, and consider how each element can be meaningfully evaluated or measured. In addition, further work will be required to consider what spreadsheet engineering properties should be in place to ensure reusability, for example cell protection and other coding details which might or might not be considered appropriate, depending on circumstances that need to be circumscribed.

It is our hope that the newly-defined domain of analytical spreadsheet models and the framework serve as a “broad church” that encompasses best practices from the literature. Where it is perceived not to do so will represent powerful opportunities to advance our knowledge.

We emphasize that we consider in this paper only the evaluation of the artifact—the quality of an analytical spreadsheet model. Further research is needed to provide general guidelines on other aspects of the spreadsheet engineering of analytical spreadsheet models, including construction approaches and the mechanics of entering cell formulas and other information into the spreadsheet, and design aspects such as style, visual formats, descriptors, labels, units of measure, titles, authorship information, cell formulas, choice of functions, source code integrity, and visibility.
REFERENCES


Controls over Spreadsheets for Financial Reporting in Practice

Nancy Coster*, Linda Leon*, Lawrence Kalbers*, and Dolphy Abraham**
*Loyola Marymount University, Los Angeles, California (USA)
**Alliance University, Bangalore, India
Nancy.Coster@lmu.edu

ABSTRACT

Past studies show that only a small percent of organizations implement and enforce formal rules or informal guidelines for the designing, testing, documenting, using, modifying, sharing and archiving of spreadsheet models. Due to lack of such policies, there has been little research on how companies can effectively govern spreadsheets throughout their life cycle. This paper describes a survey involving 38 participants from the United States, representing companies that were working on compliance with the Sarbanes-Oxley Act of 2002 (SOX) as it relates to spreadsheets for financial reporting. The findings of this survey describe specific controls organizations have implemented to manage spreadsheets for financial reporting throughout the spreadsheet’s lifecycle. Our findings indicate that there are problems in all stages of a spreadsheet’s life cycle and suggest several important areas for future research.

1 INTRODUCTION

It is broadly accepted that errors are prevalent in spreadsheets [Panko, 2006]. Spreadsheet risk can be defined as the likelihood of adverse operational or financial consequences resulting from use of a spreadsheet. To date, most spreadsheet research has focused primarily on understanding and mitigating spreadsheet risks associated with quantitative errors that occur during system development, the first part of the system’s life cycle. This line of research investigates and often suggests implementation of more formal software engineering techniques during the creation of a spreadsheet [Leon, Abraham & Kalbers, 2010; Grossman & Ozluk, 2010; Panko, 2006]. While there are definite risks of developing an erroneous spreadsheet, there are additional and just as serious risks associated with the maintenance and operational use of the spreadsheet throughout the later parts of its life cycle.

We surveyed 38 companies working on compliance with the Sarbanes-Oxley Act of 2002 (SOX) as it relates to spreadsheets associated with financial reporting. We collected information about the controls and processes they have implemented in their organization as well as the difficulties/challenges they have encountered. This paper presents the findings of this survey. While these findings describe spreadsheets associated with financial reporting, the identification of effective controls and processes are applicable to other key spreadsheets in an organization and therefore should be considered in developing best practices for IT governance. Similarly we expect all organizations will encounter the same difficulties that these
organizations are facing, but many non-regulated companies will avoid addressing risk until there is an easy and effective way of dealing with the problem.

This paper is outlined as follows: the next section provides an overview of spreadsheet controls and the accountability SOX introduces that motivates organizations to implement controls and processes for spreadsheet development and use. The survey methodology is then discussed and the results of the survey are described. Finally, conclusions and suggestions for practice and future research are made as we identify the areas where companies are struggling to effectively control spreadsheets.

2 OVERVIEW OF SPREADSHEET CONTROLS AND SOX ACCOUNTABILITY

2.1 Background

Panko and Halverson [1996] outlined a taxonomy of spreadsheet research issues as a three dimensional cube, where one important dimension was life cycle stage. As a first step in the outlined research, they created a separate taxonomy of development and testing error types [Panko and Halverson, 2001]. Panko and Aurigemma [2010] revised this taxonomy but noted that the lifecycle dimension was not addressed in either version: their studies did not examine a spreadsheet’s ongoing use after development. In the taxonomy proposed by Rajalingham, Chadwick & Knight [2000], errors that end-users can make, such as data entry errors or interpretation errors, as well as the user’s intention to create fraud, were considered. This taxonomy represented a first attempt to define and classify spreadsheet risks during operational use. Since then several governance issues have been identified which contribute to the risk of a spreadsheet after its development, such as maintenance, documentation, version control, privacy issues and separation of duties.

Basic spreadsheet programs lack the embedded logic and data controls necessary to prevent errors and misuse during operational use, so organizations need to apply manual or automated control processes to help mitigate spreadsheet risks by ensuring that appropriate tools are used to minimize, detect, and resolve errors throughout the entire life cycle. In general, end-users are resistive to attempts to control and restrict the development, sharing and use of self-generated models. The challenge is to identify effective controls that can help an organization improve the integrity of its spreadsheets without the controls being prohibitively time-consuming or expensive to implement and without interfering with the benefits of the spreadsheet medium.

Surveys show that most organizations have no formal policies to ensure the integrity of its operational spreadsheets [Panko, 1998; Caulkins, Morrison & Wiedemann, 2007; Lawson et al., 2009]. Companies reported that while informal guidelines were common, formal guidelines existed in only about half of the organizations. Neither the formal rules nor the informal guidelines were usually implemented and enforced throughout the development, testing, auditing, and modification stages of the spreadsheet life cycle, despite all of the literature on the prevalence of spreadsheet errors in organizations. One area where the corporate culture has changed is in financial reporting [Rittweger & Langan, 2010]. Sarbanes-Oxley regulations (SOX) hold publicly traded companies accountable for implementing and evaluating their spreadsheet controls for financial reporting. The Public Company Accounting Oversight Board’s (PCAOB) Auditing Standard 5 identifies the need for a combination of preventive and detective controls to prevent and detect errors or fraud in financial reporting [PCAOB, 2007]. In 2004, several surveys reported that 80-95% of U.S. firms use spreadsheets for financial reporting [Panko, 2006]. Thus, SOX forces many publicly traded companies to view end-user developed spreadsheet models that impact financial reporting similar to formal information systems used for financial reporting.
2.2 Background of SOX and Controls

In the U.S., as a result of various financial frauds and scandals over the past two decades, the Sarbanes-Oxley Act of 2002 (SOX) [U.S. Congress, 2002] initiated new policies, procedures, and disclosures in financial reporting for publicly held companies. As a result, when external audit firms identify material weaknesses in a company’s financial reporting process a description of the weakness or deficiency is documented in the company’s annual 10-K report. Audit Analytics is a public company intelligence service that provides detailed research on over 20,000 public companies. Based on the companies included in their database, there were 113 10-Ks that recorded material weaknesses as the result of inadequate spreadsheet controls for 77 different companies between 2004 and the first half of 2008 [Leon, Abraham & Kalbers, 2010]. For example, in 2006, Design Within Reach Inc. was identified as having the following material weakness: “Specifically, controls were not designed and in place throughout the year to ensure that access was restricted to appropriate personnel and that unauthorized modification of the data or formulas within spreadsheets was prevented” [Design Within Reach Inc. 10-K, 2006].

The external audit firms have provided documented guidance that no one in the organization is assuming accountability for spreadsheet risk management and control deficiencies [Protiviti Inc., 2008]. Ultimately though, senior management is the party that will be held accountable for the identified deficiencies. Therefore, senior executives should communicate an end-user computing policy to define the spreadsheet risk management requirements expected from the organization [PricewaterhouseCoopers, 2004]. This policy must define effective processes and enact appropriate monitoring to ensure compliance with these processes. From this policy, an operating model defining accountability, roles and responsibilities, processes, controls, and control standards can be created [O’Beirne, 2005]. Finally, the company should document the usage of the controls and processes outlined in the operating model.

It is advisable for companies to adopt a framework as a foundation for developing policies and procedures for spreadsheet controls. Many companies and auditors have adopted Control Objectives for Information and Related Technology (CobiT) [IT Governance Institute, 2007] to address IT compliance for SOX [Blum, 2005]. Other useful guidance for the development and assessment of spreadsheets also exists. The Institute of Internal Auditors recently issued a practice guide for user-developed applications (UDAs), which includes guidelines for controlling and auditing UDAs using a risk-based assessment of financial, operational, and compliance materiality [Institute of Internal Auditors, 2010]. PricewaterhouseCoopers [2004] proposes that organizations use a high-level five step process to manage spreadsheet risk:

1. Create an inventory of spreadsheets that are in the scope of SOX regulations
2. Perform a risk assessment of financial misstatement (materiality and likelihood) by evaluating the use and complexity of the spreadsheet
3. Determine the necessary level of controls for “key” spreadsheets
4. Evaluate existing controls for each spreadsheet
5. Develop action plans for remediating control deficiencies

General types of controls that can be considered include change controls, version controls, access controls, input controls, security and integrity of data, documentation, development lifecycle, back-ups, archiving, logic inspection, segregation of duties, and overall analytics [PricewaterhouseCoopers, 2004]. The accountability that SOX imposes makes it critical for companies to consider how these different types of controls should be implemented in their operations, which includes defining who should be responsible for their implementation and for monitoring their effectiveness. It is critical that an organization clearly define the roles and responsibilities of different organizational stakeholders, which includes developers, business users, business owners (who are defined as the people responsible for having the spreadsheet developed), IT and IS security officers, independent review groups, the accounting
department, and internal auditors. It is often the case however that one person performs several roles, such as the business owner, developer, user and reviewer.

3 SURVEY METHOD AND SAMPLE

We conducted an online survey of 38 U.S. publicly traded companies to study how organizations define the roles and responsibilities of different stakeholders for various types of controls. In particular, we investigated what these companies were doing to comply with SOX, what roles various stakeholders played, which stakeholders were responsible for ensuring processes were implemented and which processes organizations found most challenging to control. The survey included items related to the seven-stage model of spreadsheets: designing, testing, documenting, using, modifying, sharing, and archiving [Lawson et al., 2009]. The survey questions were designed to elicit responses about material and/or critical spreadsheet applications used in the financial reporting process, where “material” and/or “critical” spreadsheet applications were defined as being significant to the financial statements and/or footnotes, and probably identified as “in scope” for purposes of SOX compliance. A longer and a shorter version of the survey were developed. This was done to increase participation. The short survey provided an option to continue and complete the long survey.

We had several objectives in selecting our sample from the population of U.S. public companies. First, we wanted a sample of public companies that varied in size and industry. This was intended to provide a better opportunity to generalize to all public companies. Second, we desired a person from each participating company to complete the survey who had the necessary knowledge of SOX compliance and spreadsheet controls in their organization. This was intended to increase the reliability of the responses.

Several approaches were taken to elicit responses to meet our objectives. First, participants were sought by posting a link to the electronic survey on local LinkedIn groups associated with the Information Systems Audit and Control Association and the Institute of Internal Auditors. Second, a list of the 200 largest public companies in Southern California and a random sample from the S&P 1500 were called in an attempt to identify the contact name of a qualified individual in the organization. After obtaining a specific name and email address, an email was sent to each contact who had expressed a willingness to participate. Last, some personal contacts of the authors were emailed. The emails opened with the sentence, “You have been identified as the person who is most knowledgeable about how spreadsheets associated with financial reporting for SOX compliance are being managed in your company” to emphasize the need for an appropriate person to complete the survey. The email also briefly described the study and provided a link to the online surveys. All participants were given a choice between the shorter and longer versions of the anonymous survey.

There were a total of 38 respondents to our online surveys—26 responded to the longer survey and 12 to the shorter version. The total sample population cannot be estimated due to the various approaches taken, therefore no response rate can be calculated. Table 1 presents the 15 different industries represented in the sample. Manufacturing and entertainment had the largest number of respondents, representing 23.7% and 15.8% of the sample, respectively. The size of the companies responding, measured in assets, ranged from less than $24 million to over $100 billion, with about 74% of companies with assets greater than $1 billion (see Table 2). Of the 25 respondents that provided their job title, the majority (16) were associated with internal audit, followed by SOX or General Compliance positions (5). In addition, 15 respondents were also the person responsible for SOX Compliance and/or spreadsheet controls in the organization. These results indicate that respondents were at an appropriate level of the organization and knowledgeable about controls over spreadsheets for purposes of this study.
4 FINDINGS

In this section we report the results of the survey. There were 26 respondents that answered the longer survey version, and 12 that answered the shorter version. Therefore, there was a maximum of 26 or 38 responses for items from the longer survey and the total responses, respectively. In order to increase participation, participants were allowed to skip questions that they did not feel comfortable answering. Thus the responses for some items are lower than the overall number of participants for the survey. Results are presented in percentages and/or numbers. Care is taken to show the total responses in each case.

Table 1 Industries Represented in the Sample

<table>
<thead>
<tr>
<th>Industry</th>
<th>Response Percent</th>
<th>Response Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aerospace &amp; Defense</td>
<td>5.3%</td>
<td>2</td>
</tr>
<tr>
<td>Construction</td>
<td>2.6%</td>
<td>1</td>
</tr>
<tr>
<td>Engineering &amp; Related Services</td>
<td>2.6%</td>
<td>1</td>
</tr>
<tr>
<td>Entertainment</td>
<td>15.8%</td>
<td>6</td>
</tr>
<tr>
<td>Financial Services</td>
<td>7.9%</td>
<td>3</td>
</tr>
<tr>
<td>Health Care</td>
<td>5.3%</td>
<td>2</td>
</tr>
<tr>
<td>Insurance</td>
<td>2.6%</td>
<td>1</td>
</tr>
<tr>
<td>Manufacturing</td>
<td>23.7%</td>
<td>9</td>
</tr>
<tr>
<td>Real Estate</td>
<td>2.6%</td>
<td>1</td>
</tr>
<tr>
<td>Restaurants</td>
<td>5.3%</td>
<td>2</td>
</tr>
<tr>
<td>Retail Services</td>
<td>5.3%</td>
<td>2</td>
</tr>
<tr>
<td>Software Development</td>
<td>7.9%</td>
<td>3</td>
</tr>
<tr>
<td>Telecommunications</td>
<td>2.6%</td>
<td>1</td>
</tr>
<tr>
<td>Travel/Leisure</td>
<td>2.6%</td>
<td>1</td>
</tr>
<tr>
<td>Other</td>
<td>7.9%</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 2 Asset Size of Companies in the Sample

<table>
<thead>
<tr>
<th>Asset Size</th>
<th>Response Percent</th>
<th>Response Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0 - $24MM</td>
<td>2.6%</td>
<td>1</td>
</tr>
<tr>
<td>$25MM - $99MM</td>
<td>5.3%</td>
<td>2</td>
</tr>
<tr>
<td>$100MM - $999MM</td>
<td>18.4%</td>
<td>7</td>
</tr>
<tr>
<td>$1 billion - $99 billion</td>
<td>65.8%</td>
<td>25</td>
</tr>
<tr>
<td>&gt; $100 billion</td>
<td>7.9%</td>
<td>3</td>
</tr>
</tbody>
</table>

4.1 Use of Spreadsheets after SOX

SOX emphasizes internal controls and the documentation of those internal controls. 22 out of 24 survey respondents (92%) report that spreadsheets used in the financial reporting process are of the same level or higher level of importance in the post SOX era. The number of spreadsheets that respondents reported using has not decreased as expected. 18 out of 23 respondents (78%) reported using the same number or more spreadsheets in the financial reporting process since the implementation of SOX. One might expect that after SOX, companies would either increase the controls over spreadsheets used in the financial
reporting process or decrease the number of spreadsheets used in this area. However, 21 out of 24 respondents (88%) indicated they did not have a computing policy specific to spreadsheets, demonstrating lack of a comprehensive plan to mitigate spreadsheet risk. Our survey results further indicate that the operating controls over spreadsheets used in financial reporting are also still lacking in many areas including change management, version management, access control, and the development process.

4.2 Self-Identified Areas of Controls Difficult to Implement

We asked the 38 respondents to identify the top three processes where implementing appropriate controls for critical spreadsheets used in financial reporting are most difficult. As shown in Figure 1, respondents were most concerned with change management, version management and access control. They were most confident with the backup process. Respondents proceeded to answer survey questions that supported their initial levels of concern or confidence in these areas as detailed in the following paragraphs. The detailed items about each area provided more insight into the perceived strengths and weaknesses of the controls. In some areas, detailed responses confirmed the strongest areas of concern. In other areas, however, later survey items suggested weaknesses in areas that were not identified by many respondents as difficult areas.

![Figure 1 # of Times Process Identified as a Top 3 Area of Difficulty for Implementing Controls](image)

**Change Management**

Change management was the number one self-identified area of concern in implementing controls for critical spreadsheets in financial reporting as reported by 23% of the 38 respondents. Given this, it is not surprising that over one half of the 31 respondents to the questions shown in Table 3 reported no documented procedures (including use of IT and built-in spreadsheet controls) to address issues related to change management. If a company did have a documented procedure in place, the accounting department was the person or department largely responsible for ensuring the policy was followed. Two respondents listed their spreadsheet review group as the responsible department. 87% of respondents (27/31) reported they did not have a policy to describe the job title or skill requirement of the person responsible for changing the spreadsheet.
Table 3 Modifications to Spreadsheets

<table>
<thead>
<tr>
<th>Is there a documented procedure (including use of IT and built-in spreadsheet controls) to:</th>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. Prevent unwanted changes to a spreadsheet?</td>
<td>39%</td>
<td>61%</td>
</tr>
<tr>
<td>b. Request changes to a spreadsheet?</td>
<td>29%</td>
<td>71%</td>
</tr>
<tr>
<td>c. Test accuracy of changes made to a spreadsheet?</td>
<td>39%</td>
<td>61%</td>
</tr>
<tr>
<td>d. Track changes made to a spreadsheet?</td>
<td>26%</td>
<td>74%</td>
</tr>
</tbody>
</table>

**Version Management**

The second area respondents identified as a top concern when implementing controls over spreadsheets for financial reporting was version management. 61% of respondents (19/31) stated they did not have a documented procedure to limit access to the most recent version of the spreadsheet.

**Access Control**

Access control closely followed version management as the third area where respondents reported it was difficult to implement spreadsheet controls in financial reporting. Their answers to questions in both the development and usage phases supported this concern. 84% of respondents (26/31) stated that there was no policy to describe the job title or skill requirement of the person responsible for using a newly developed spreadsheet. Similarly when asked if they had a documented procedure to set up appropriate access levels for different identified users of the spreadsheet 71% (23/32) answered no.

**Input Control**

As shown in Figure 1, input control was an area selected by some respondents as one of their top three control concerns. However, our results indicate 61% of respondents (19/31) actually have documented procedures to validate that data inputs and outputs are complete and accurate for manual and systematic downloads. Those respondents with documented procedures in place would likely not list input controls as a top concern. For the 61% of respondents that have procedures in place, more than half of them listed the accounting department as the department or person responsible for ensuring the procedure is followed.

**Back Up Process**

While companies surveyed identified areas where it was difficult to implement controls in financial reporting, by not selecting an area we interpret this to mean they feel more confident with spreadsheet controls in that area. The backup process was the least selected area, thereby indicating the greatest level of confidence which is supported by answers to additional survey questions. 87% of respondents (27/31) reported backing up critical spreadsheets on a regular basis. 77% (23/30) reported limiting access to archived files. The majority of respondents reported the IT department was responsible for ensuring the backup procedures are followed.

**Review Process**
The review process was self-reported as a concern when implementing controls for spreadsheets used in financial reporting, but it was not in the top three concerns. Respondents’ answers to subsequent survey questions supported their placement of this process as a mid-level concern. 68% of respondents (21/31) reporting no policy in place to describe the job title or skill requirements of the person responsible for reviewing the spreadsheet after new spreadsheets are developed. Over one half of the respondents (18/31) indicated the process used to review a spreadsheet included an auditing review checklist or protocol that describes the types of tests/reviews to be done. 35% (11/31) stated their review process included a procedure that generates an audit trail. For those respondents who have these review processes in place, the accounting department and internal auditor were largely identified as the department or person responsible for such review.

Further adding to the concern within the review process is the level of the reviewers’ domain experience. 30% of respondents (7/23) reported a minimal level or no domain experience. The reviewers were stronger in their spreadsheet experience with 61% (14/23) reporting advanced to expert levels. If respondents utilize the review phase as a compensating control, hoping to catch possible errors made in the development stage before they would impact the financial reporting process, it is important for reviewers to have a higher level of domain knowledge than currently reported.

Development Process

The development process and testing process are two areas respondents did not identify as top areas of concern, selected by even less respondents than the review process. However, based on responses to later survey questions these are areas more lacking in controls than the review process. 87% of respondents (27/31) indicated there was no formal development procedure that should be followed when new spreadsheets for financial reporting are developed. 94% (29/31) stated there was no policy that described the company’s styles, design and documentation standards. Furthermore, 87% of respondents (27/31) said they did not have a policy to describe the job title or skill requirements of the person responsible for developing the spreadsheet. For the 13% of respondents who did report having a policy to describe the job title or skill requirement, all reported the accounting department as the department or person responsible for ensuring the procedure is followed. The absence of formal development procedures is an issue, and this coupled with the developers’ lack of accounting knowledge when creating spreadsheets critical for the financial reporting process has the potential to lead to serious errors in output. Our results indicated the spreadsheet developer’s domain experience was often weak. 35% of respondents (8/23) reported minimal or no domain experience while 30% (7/23) reported only a moderate level.

4.3 Current and Future Controls

It is clear from the responses documented in section 4.2 that control weaknesses exist in some areas for a number of companies. We provided respondents in the longer version of the survey a list of internal controls and asked them to identify the effective controls or tools currently implemented within their company. We also asked them to identify those controls or tools they planned to implement in the future. The results are shown in Table 4.

For current effective internal controls and tools, 7 of the 15 listed were implemented in over half of the responding companies. 79% stated they have files secured in drives and server folders with limited access. This indicates respondents have taken measures to limit access to spreadsheets, but as discussed above they struggle with access controls in the development and usage phases. In addition to limiting access to the drives and folders, they also keep these areas well organized. 67% of respondents reported effective controls in logically structured directories and folders for business units, cycles, and type of
spreadsheets. Consistent with our findings, 63% of respondents report having a formal review process and 58% also report having input controls to ensure data integrity and a password to update the spreadsheet. However, independent review groups, required Excel Track Changes, external tools, developer training and a spreadsheet computing policy stating design standards are some of the controls and tools used in 25% or fewer of the responding companies.

For those respondents who stated future plans, the number one future control respondents plan to implement is Excel Track Changes followed by formal review processes. However, the implementation percentage for any future control was less than 28% for each control. There does not appear to be a popular control or solution that companies are eager to implement. It is particularly troublesome that both a spreadsheet computing policy for stating design standards and mandated training for developers are low in current and planned implementation.
### Table 4 Internal Controls Organizations Considered for Implementation

<table>
<thead>
<tr>
<th>Internal Controls or Tools</th>
<th>Percent of Companies that Currently Implement Tool</th>
<th>Percent of Companies that Plan to Implement Tool in Future</th>
</tr>
</thead>
<tbody>
<tr>
<td>Files secured in drives &amp; server folders with limited access</td>
<td>76.9%</td>
<td>11.5%</td>
</tr>
<tr>
<td>Logically structured directories/folders for business units, cycles, and type of spreadsheets</td>
<td>65.4%</td>
<td>11.5%</td>
</tr>
<tr>
<td>Formal review process</td>
<td>57.7%</td>
<td>23.1%</td>
</tr>
<tr>
<td>Input controls that ensure data integrity</td>
<td>57.7%</td>
<td>15.4%</td>
</tr>
<tr>
<td>Password required to update spreadsheet</td>
<td>57.7%</td>
<td>15.4%</td>
</tr>
<tr>
<td>Cell protection (required)</td>
<td>50.0%</td>
<td>15.4%</td>
</tr>
<tr>
<td>More than one person responsible for data and maintenance</td>
<td>46.2%</td>
<td>3.8%</td>
</tr>
<tr>
<td>Independent review groups</td>
<td>23.1%</td>
<td>19.2%</td>
</tr>
<tr>
<td>Excel Track Changes (required)</td>
<td>19.2%</td>
<td>26.9%</td>
</tr>
<tr>
<td>Spreadsheet computing policy stating design standards</td>
<td>11.5%</td>
<td>15.4%</td>
</tr>
<tr>
<td>Mandated training for developers</td>
<td>7.7%</td>
<td>3.8%</td>
</tr>
<tr>
<td>Third party auditing software</td>
<td>7.7%</td>
<td>15.4%</td>
</tr>
<tr>
<td>Spreadsheet data consolidated into databases managed by IT</td>
<td>7.7%</td>
<td>11.5%</td>
</tr>
<tr>
<td>Third party tools for access, version, change, and archive support</td>
<td>3.8%</td>
<td>11.5%</td>
</tr>
<tr>
<td>Spreadsheet converted into server-based application</td>
<td>0.0%</td>
<td>11.5%</td>
</tr>
<tr>
<td>No Stated Plans</td>
<td>-</td>
<td>26.9%</td>
</tr>
</tbody>
</table>

#### 4.4 Spreadsheet Outcomes

The emphasis on internal controls for the purposes of SOX is to increase the probability of financial statements that are materially correct. Critical weaknesses in controls over spreadsheets used for the purpose of preparing financial reports have the potential to lead to public disclosure of weaknesses in controls by external auditors. Perhaps even worse, errors in spreadsheets used in financial reporting may cause material errors in the financial statements. We asked two questions related to these possibilities. First, 46% (11/24 respondents) indicated that "internal reviews uncovered lapses or non-compliance with established protocols for spreadsheet controls." Second, 21% (5/24 respondents) answered yes to the question, "have internal reviews documented financial statement errors related to spreadsheet errors?" As mentioned earlier, compensating controls or final reviews by qualified individuals may reduce the ultimate risk of material errors in financial statements. However, the responses to these two items support the other findings that stronger internal controls are needed in many corporations.

#### 5 CONCLUSIONS AND SUGGESTIONS FOR PRACTICE AND FUTURE RESEARCH

The Sarbanes-Oxley Act of 2002 has provided significant motivation for public companies to develop and tighten controls over spreadsheets used for financial reporting. There are potential serious negative consequences of poorly designed spreadsheets, including public disclosure of significant weaknesses in controls and materially misstated financial statements. Our findings demonstrate that companies continue to use spreadsheets for financial reporting. However, even with such a strong incentive for companies to have strong controls, many weaknesses in controls exist. Formal policies and procedures are still lacking in most companies for most of the stages of spreadsheets. More than half, and often most, of the companies report no policy in place to describe the required qualifications for individuals who develop, modify, review, or use spreadsheets. The results show that while individuals developing and reviewing spreadsheets have a reasonably high level of spreadsheet experience, their domain knowledge tends to be much lower. In the case of financial reporting, which can involve knowledge of complex accounting rules, this is of great concern. Though compensating controls may be in place at the final review stage.
before information goes into financial reports, stronger controls at earlier stages of the process would reduce the risk of non-compliance and errors.

Our findings indicate that practitioners can improve controls in several areas. More formal policies and procedures that set requirements for processes and expertise for domain knowledge and spreadsheet expertise are needed, particularly in the development, review, and use stages. We note again that the weaknesses found in this study are for controls in an area that is highly regulated and visible. We would further suggest that practitioners consider and apply similar analyses to operational spreadsheets, where errors may lead to poor business decisions.

Our findings suggest several important areas for future research. Though more research is now being done beyond the development stage, our results indicate that there are problems in all stages of a spreadsheet’s life cycle. Attempts made by some organizations to control certain processes do not appear to be sufficient. Future surveys need to query organizations for more technical details about the various controls being implemented. Finally, further research comparing the impact of domain knowledge and spreadsheet expertise is also needed. The findings from this line of research will help organizations plan and implement policies that impact training for spreadsheet developers, design review, version control and auditing of spreadsheet models in all application areas within their organizations.
6 REFERENCES


Institute of Internal Auditors (2010), Global Technology Audit Guide (GTAG®) 14, Auditing User-developed Applications.


ABSTRACT

Hypernumbers is a new commercial web-based spreadsheet. It addresses several risk factors in deploying spreadsheets.

Traditional risk management of spreadsheets has focused on run-time risk - incorrect formulae, accidental overwriting of formula, unapproved adaption of the spreadsheets structure by an unapproved person and so on. The sources and lifecycles of spreadsheet errors are by now quite well known.

Current risk management has three major mechanisms to address risk. Firstly, to instrument spreadsheets using their scripting capabilities to impose certain behaviours on them. Secondly, to minimize risk by using operating procedures and standards which are shown to be less error-prone that ad-hoc spreadsheet development methods. These are training intensive and hard to police. Thirdly, to inspect and audit spreadsheets in operation.

Hypernumbers is attempting to address spreadsheet risk in two radically new ways. The core approach is not to mitigate risk but to engineer out risky activities.

The curtilage of the spreadsheet has been extended to include much of the operational environment in which conventional spreadsheets are used. This process absorbs and subsumes much of the ‘best practice’ and end-user disciplines that are conventionally used to address risk.

Traditional spreadsheets have no clear barrier between data and programme instructions – in the Turing sense – or between input data and output information. Hypernumbers enables those barriers to be simply and reliably re-imposed, thus draining off a whole category of run-time errors. This separation allows spreadsheet usage to be split into two distinct phases – development, where audit and testing can reduce errors, and deployment, where what would be risky practices in other spreadsheet paradigms are simply engineered out.

Using these techniques, ‘applications’ consisting of 10,000+ spreadsheets used by tens of users can be, and have been, safely built.

1 A BRIEF HISTORY OF SPREADSHEETS

Spreadsheets arose in the swansong days of the mainframe and really exploded only with the rise of personal computer [ROIZEN, 2010]. The run-time environment was un-networked personal computers in a single user mode.

The early spreadsheet war was decisively won by Microsoft Excel, and the second phase began – a battle between proprietary software and the open source spreadsheets. This
was, from a technical perspective, a re-run of the previous war, as all the open source contenders (including what became Open Office) had their roots in the un-networked, single user world.

The third war is in progress: a battle between web-enabled desktop spreadsheets and their earth-bound counterparts – the collaboration wars. Collaboration being, in the context of online spreadsheets, narrowly defined as two or more people editing the same spreadsheet sequentially.

This common architectural history explains the commonality of solutions to spreadsheet risk across all the contenders.

The conventional spreadsheet is a networked object – shared inputs, shared outputs – which is distributed via a variety of ad-hoc mechanisms because it is based on non-networked technologies.

2 HYPERNUMBERS – AN OVERVIEW

Hypernumbers resembles a spreadsheet in the way that a stick insect resembles a stick – it is a purposeful, but misleading resemblance. A better example would be the telephone dialer on a smart phone. A smart phone is a computer pretending to be a mobile phone, and a mobile phone is a two-way radio pretending to be a telephone. However, across the technological changes of the last century there has been a consistent and long-lived user mental-model – the dialer, the core user interface into the world of telephony.

The intention of Hypernumbers has been to enable the end-user to do new things, whilst preserving and building on the familiar mental model of a spreadsheet.

Hypernumbers aims to address the common problems arising from the ad-hoc distribution of spreadsheets (file systems, e-mail, etc, etc) by building on a networked base that eliminates them as a class of error.

3 THE PROBLEMS

3.1 Overview

This section will enumerate a set of common spreadsheet problems to provide a sound analytical basis for the discussion of risk management approach developed so far in Hypernumbers.

3.2 Co-ordination

Spreadsheets are typically used for co-ordination of activities, often by aggregation. This is usually achieved by using shared folders, collaborative technologies like Google Docs or Sharepoint, or sometimes by e-mail.

The negotiation of locking of spreadsheets for update is usually manual and tiresome.
3.3 Proliferation

Proliferation takes two characteristic forms.

Individual spreadsheet files are duplicated, ‘the one on the work server’, ‘the one I emailed home to work on at the weekend’, ‘the one I sent you for your input’.

Individual departments or work units use spreadsheets to build ad-hoc operational systems. These departmental spreadsheets ecosystems overlap in scope and become impossible to manage.

3.4 Permission Management

The conventional desktop spreadsheet has very primitive permission management – it tends to be all or nothing – you can edit the whole spreadsheet or not see it. There is a password protection layer built into modern spreadsheets but it is ad-hoc and not based on role-based access. In addition, if you can have physical possession of a spreadsheet, but not the password, you can eventually force access to it. It is not true permission management.

3.5 Version Control

The problems of version control are close cousins of the problems of proliferation. Publication and management of templates is difficult to do properly. Individual spreadsheets have no sense of their own history (am I version 1.0 or version 1.1?). Versioning has to be imposed by external systems, or operational discipline, and both are error prone.

3.6 Verification

Spreadsheets contain an admixture of input data, programme instructions (formulae) and output data. This makes it hard for verification to be sustained. The person who develops a spreadsheet may be appropriate trained, the template so produced can go through an audit process, but ultimately the end-user can then trample on that good work. The lack of access control and audit means that anytime a spreadsheet has ‘been passed around’ its correctness is automatically suspect. This makes verification somewhat like painting the Forth Railway Bridge, the end is only the beginning again.

3.7 Audit And Data Management

Audit is hard to do with traditional spreadsheets. As users are not identified intrinsically in the interaction with the spreadsheet; it is well-nigh impossible to build a picture of what this person did to that spreadsheet let alone what this person did to those spreadsheets.

Because the spreadsheet file is an uncontrolled object it is hard to ensure that it is backed up, reliably deleted and other conventional data management operations.
4 ADDRESSING THE PROBLEMS

4.1 Introduction

This paper contains just a short overview of the features of the Hypernumbers platform that pertain to addressing common problems in spreadsheet deployment. A fuller account can be found elsewhere [McCRORY, GUTHRIE, 2011].

4.2 Business Architecture (And Technical Architecture)

Overview

The basic framework adopted to address these problems was a complete platform business architecture based on a new technical architecture.

Web Centralisation

The core technical architecture chosen was a web-based platform with role-based access.

Spreadsheet proliferation is eliminated by making the spreadsheets accessible from anywhere. This aspect is not substantially different from the Google Docs or Sharepoint approach to centralisation. Central role-based access, however, makes it possible to build a set of sheets where Alice can edit her sales data, Bob can edit his marketing data, and Charlie can edit his fulfilment data, the whole aggregating seamless, and in an error free manner, to a report for the CEO.

Hierarchical Structure

Traditional spreadsheets, both desktop and web, are organised as a set of tabs (siblings). By contrast it was decided that Hypernumbers sheets should be organised in a hierarchical space that resembles a file system. A particular spreadsheet might have the URL:

http://example.com/accounts/2011/invoices/inv00000001/

This can be thought of as the equivalent of a spreadsheet called inv0000001.xls in directory accounts -> 2011 -> invoices

The advantage of this organisational structure is that it enables a whole class of special queries. It makes it possible to write functions that correspond to add up all the unpaid invoices or show me a list of all invoices that are more than 30 days overdue.

A traditional spreadsheet can be considered in terms of x-y coordinates, 5 columns along 6 rows down. These novel queries, z-queries, operate in ‘the third dimension’, the z-dimension – they can be thought of as a go through this pile of spreadsheets on my desk and add the totals up.

An Operational Platform

To address many of the basic management and housekeeping risks it was decided to deliver Hypernumbers as a PaaS – Platform As A Service. The platform was defined to include certain intrinsic capabilities (daily backup, clustered high-availability service, central log-on and authentication, automatic logging) that would drain some of the swamps of operational risk associated with large spreadsheet bases.
Problems Addressed

The business and technical architecture of Hypernumbers was intended to addresses the following problems in whole or in part:

- proliferation
- audit and data management

4.3 Views And Permissions

Overview

There are three core roles in spreadsheet use:

- the maker – the person who designs the business logic and structure and creates the formulae
- the data inputer – the person (or system) who provides or inputs data to drive the business logic
- the output consumer – the person (or system) who receives and acts on the outputs of the spreadsheet

A decision was taken to bake these three roles into the structure of Hypernumbers. These three roles thus have a dedicated view of the data, respectively:

- the spreadsheet view
- the wikipage view
- the webpage view

In addition there are a couple of minor views:

- the table view
- the log view

Spreadsheet View

The maker can use a spreadsheet view of a particular page – it is familiar to all spreadsheet users with the appropriate cell and selection behaviours, keyboard short cuts, wizards and so on and for forth.

Webpage View

The output consumer can use a webpage view of a particular page – nothing can be changed, no spreadsheet operations can be performed. As this was regarded as a fundamental activity, it was decided to make creating the webpage views as easy as possible – the process was finally reduced to a single click.

Wikipage View

Data imputers can be provided with a wiki page. The maker can mark cells on a page as wiki page inputs (either simple text boxes or drop down menus). When the page is rendered as a wikipage these cells (and these cells only) can be edited by the end user. A decision was taken to make it easy to turn spreadsheets into web forms with a single click.
To keep the roles separate, the wikipage view also entirely prevents the user from making any structural changes (row and column insertions or deletions).

Tables

A common use case of spreadsheets is as primitive tabular data – each row is considered ‘a thing’. The table view is simple row editor designed to expedite this usage.

Because this is a pervasive use case, a simple forms construct was developed which makes their creation easy.

Logs

It was decided that forensic activities should be automatic and not optional or bolted on.

As a result, by default all user actions are logged – in two different respects. There is a log view which provides real time access to the historical values of particular cells on particular pages which makes it possible to reconstruct the history of a particular value.

There is another narrative log which documents the interactions of a particular user across multiple pages. This is not yet integrated into the body of the system, but provides detailed forensic audit information.

Permissions

From a technical perspective permissions could be granular down to particular actions on particular cells. However usability experiments found that this was impractical. Simple spreadsheet pages would expose upward of 15,000 securable API points in the viewable portion of the page.

As a result of these tests a decision was taken that permissions should be applied at a view level. If a person can see a particular view of a particular page then they can perform all the actions that that view implies.

Problems Addressed

The intention was that the views and permissions functionality of Hypernumbers should address the following problems in whole or in part:

- co-ordination
- permission management
- audit

Views perform the role of separating out the various classes of user. Permissions ensure that the various classes are locked down. This combination is intended to address directly the bulk of common spreadsheet risks.

4.4 Functions

Overview

At the core of the spreadsheet model is the common programming precept of putting =function(arg1, arg2,...) in a cell.
Hypernumbers have taken as a design principle Blaise Pascal’s famous dictum [PASCAL, 1656-1657]: *je n'ai fait celle-ci plus longue parce que je n'ai pas eu le loisir de la faire plus courte*.

Hypernumbers endeavoured to implement this precept in all respects. Considerable effort has gone into ensuring that all the various types of user needs, from web controls to database queries, html menus to debugging, all can be fulfilled by invoking functional expressions in cells.

**Layout**

Layout items for webpages and wikipages are implemented in functions. For example:

`=html.box.4x8("Title", "This is the body text", "Footer")`

This creates a styled box 4 standard cells wide and 8 standard cells high with a title, a body and a footer. The set of layout functions will be expanded to include standard web idioms like tabbed boxes, accordion boxes, slideshows and the like.

**Navigation**

Navigation on web pages is provided by menus. These are created by functions with the following syntax:

`=html.menu(ref to sub-menu1, ref to sub-menu2, ....)`

Other common web navigational items like crumb trails are provided by other functions. Future work will include making menu functions that read the underlying page and permissions structure of the Hypernumbers website to create dynamic menus.

**Business Logic**

At the core of the business logic is a large set of 100+ Excel-compatible functions. The current set of implemented functions is based on the ODF Small Group [OASIS OPEN DOCUMENTS, 2011], with some additional functions from the Medium and Large Group.

These have the same parameters (and throw the same errors) as their Excel and Open Office equivalents.

**Forms And Transactions**

Various elements for traditional web forms can be specified by functions in cells, for instance, radio boxes, text input boxes, dropdown lists and so on. These can be grouped into transactions by parameters. This approach enables data input to be managed, aggregated and validated without tedious audit. Because the functions are themselves programmable the transactions exposed to the end user are themselves programmable, providing a flexible, yet secure mechanism for handling data input.

---

1 I have made this letter longer than usual, because I lack the time to make it short
Database

Database-like functionality is provided by so-called z-functions. Some of these take the form of additional parameter types for conventional functions. Consider the function

\[=\text{sum}(\ldots, \ldots)\]

It behaves as you would expect:

\[=\text{sum}(1, 2, \text{a3}, \text{!page!a4})\]

The reference to other sheets can also be by a standard ‘web-style’ addressing scheme where the Lotus-1-2-3 bangs are replaced by forward slashes:

\[=\text{sum}(1, 2, \text{a3}, /\text{page}/\text{a4})\]

The page reference approach is totally generalised to the arbitrarily deep hierarchy of hypernumbers using a conventional web syntax with .. being the parent page and . being the current page.

In addition to all this (mostly) familiar syntax, sum can also take z-references:

\[=\text{sum}(1, 2, \text{a3}, /\text{some}/\text{page}/[\text{a1 > 44}]/\text{b7})\]

The expression /some/page/[a1 > 44]/b7 is a z-expression. Imagine there are 4 pages on a Hypernumbers site:

/some/page/bleh/
/some/page/blah/
/some/thing/blurg/
/another/page/

The z-expression would be evaluated segment by segment against these pages to see if they match:

- the first three match the segment /some/
- the first two then go on to match the second segment /page/
- at this point the strange segment /[a1 > 44]/ is applied. The brackets [...] simply indicate that it is a z-segment. The expression a1 > 44 is evaluated on each page. If it returns true that page matches, if it returns false it doesn’t. The values of the cell b7 on every matching page is then included in the sum

The expressions in z-queries are simply valid excel-compatible spreadsheet expressions that evaluate to either true or false. They can use the full panoply of functions (with some minor exceptions). So for instance expressions like [if(or(a1 > 0, b7 = “failed”), true, false)] are valid z-segments.

These database functions (when combined with structural functions and templates) enable complex applications to be built.

Integration

Because web spreadsheets should be first-class web citizens there are functions like:

\[=\text{facebook.comments}(\text{facebook_app_id})\]

This puts integrated facebook comments on webpages or wikipages. Other integration functions for twitter, and such-like web properties are available.
Structural Functions

There are a number of structural functions. A typical one would be:
=create.button(buttonTitle, params)

This function takes a specification of a webpage to create. It typically builds it from a template, under a structural name and then forwards the user to that page.

An example of a specification is:
=create.button("Prepare New Day", "/some/page/[blank, date, yyyy]/[blank, date, mm]/[day_sheet, date, dddd]/")

This would create the following pages:
/some/page/2011/ from a blank template
/some/page/2011/apr/ from a blank template
/some/page/2011/apr/21/ from the day_sheet template

The person who clicked the button would be redirected to the page:
/some/page/2011/apr/21/

More sophisticated parameters enable the manipulation of views and groups at run time, as well as the creation of more complex sets of subpages.

Problems Addressed

These functions play their part in making it possible to address the following problems:
• co-ordination
• proliferation

None of them address these problems directly, but between them they make it possible to build sophisticated web-based applications that are usable by multiple people simultaneously.

However the primary purpose of the functional set is not to directly solve problems, but to make it as easy as possible for existing spreadsheet users to extend their repertoire of actions by presenting what are logically new activities in an already understood and mastered form.

4.5 Templates And Structural Change

Overview

In order to address issues pertaining to checking and verification of spreadsheets a decision was taken to make templates the basic units of development. They work the same way as templates in desktop spreadsheets – except their invocation is trivially programmable.

Template Creation

Templates are created by simply saving spreadsheet pages as templates. As well as saving the data, formulae and formatting the template includes special attributes (whether a cell is a wiki-editable cell for instance) as well as permissions and views. A page can be
created as a wiki page for the sales department. All pages created from that template will then by default be wiki pages for the sales department.

**Structural Functions**

There is a need to enable repeatable abstractions in the creation of spreadsheets [PEYTON-JONES, BLACKWELL, BURNETT, 2003]. Templates can be used to fulfil part of this need if they can be invoked in a programmatic manner.

To this end, structural functions can be used to create buttons on webpages or wikipages which then instantiate new spreadsheets from templates at named places in the hierarchy. z-queries can then be used to aggregate the information captured in those templates, or to populate navigational items like html menus.

**The Development Process**

This use of templates and structural functions was designed to enable a novel development process to be implemented. An application can be built in spreadsheets. As part of the development process it can be audited and tested extensively and subject to quality measures.

By contrast it can be deployed as an application. In normal daily use the end-users only see webpages or wikipages – they can neither see, nor change, the underlying ‘programme’ aspects. They can enter data into the application and see the appropriate outputs. They can create new instances of wikipages and webpages by pressing structural buttons. The results of their interaction with those new pages can be automatically aggregated and summarise up the hierarchy.

The problem of data integrity at run time then becomes one of validating data entry. Was the right data entered? Was all the data entered? By building validation into the templates and using ‘double entry’ techniques, the data validation step becomes much more manageable.

**Problems Addressed**

The rational was that templates and structural change should help address the following problems:

- permission management
- verification
- version control

This clear separation of the different roles of different staff members with respect to the spreadsheets designs out several major classes of common spreadsheet errors.

**5 SUMMARY**

The intention was that Hypernumbers should be a powerful new way for creating secure, error free and manageable applications with spreadsheet skills.

After a prolonged development process Hypernumbers is now being deployed in production for the first commercial clients. Details are unfortunately commercially confidential, but there are very promising results with large scale applications consisting
of 10,000+ spreadsheet pages for use by tens of different users in different roles have been successfully created.

6 REFERENCES

Raymond R Panko (2008) *What We Know About Spreadsheet Errors*  

Heidi Roizen (2010) former CEO of T/Maker an early spreadsheet company for the Mac, personal communication

http://documentation.hypernumbers.com/contents/welcome/other-formats.html

Blaise Pascal (1656-1657), *Lettres Provinciales* no. 16.


From Good Practices to Effective Policies for Preventing Errors in Spreadsheets

Daniel Kulesz
Institute of Software Technology, University of Stuttgart, Germany
daniel.kulesz@informatik.uni-stuttgart.de

ABSTRACT

Thanks to the enormous flexibility they provide, spreadsheets are considered a priceless blessing by many end-users. Many spreadsheets, however, contain errors which can lead to severe consequences in some cases. To manage these risks, quality managers in companies are often asked to develop appropriate policies for preventing spreadsheet errors. Good policies should specify rules which are based on „known-good“ practices. While there are many proposals for such practices in literature written by practitioners and researchers, they are often not consistent with each other. Therefore no general agreement has been reached yet and no science-based „golden rules“ have been published.

This paper proposes an expert-based, retrospective approach to the identification of good practices for spreadsheets. It is based on an evaluation loop that cross-validates the findings of human domain experts against rules implemented in a semi-automated spreadsheet workbench, taking into account the context in which the spreadsheets are used.

1 INTRODUCTION

Since their invention in the late 1970s, spreadsheet software packages have been valued as a priceless blessing by millions of end-users. Thanks to the huge flexibility they provide, end-users can shape their own computing solutions even for complex computing tasks. This does not only come in handy for individuals and small businesses who do not have the budget for ready-made solutions. Spreadsheets are used on a large scale in enterprises as well – especially in situations where the IT department fails to provide acceptable solutions on time or budget.

There is a steadily growing body of evidence which suggests that many of the spreadsheets in use today suffer from a high percentage of errors [Panko, 2008] [Powell et al., 2007]. Several horror stories from the past like [Godfrey, 1995] indicate that errors in spreadsheets can lead to wrong results which, in turn, can result in costly wrong decisions. Although many executives and senior managers rate the impact of spreadsheet errors as less critical [Caulkins et al., 2008], several governments regard them as a serious threat and have already reacted by issuing laws like Sarbanes Oaxley Act 404, Basel II or Solvency II which try to confine the uncontrolled use of spreadsheets.

Apart from the fact that laws can only address the spreadsheet quality problem in limited scope, there is no generally accepted recommendation among researchers either. After a
critical review of the literature on spreadsheet errors, Powell, Baker and Lawson [Powell et al., 2008] draw several alarming conclusions about the current „state of the art“:

- There is no generally accepted way of counting and classifying errors in spreadsheets. Although a few taxonomies exist (i.e. [Panko and Halverson, 1996] [Rajalingham et al., 2000]), all of them are not clearly defined, untested and their applicability is limited to a few, implicitly defined contexts. (Note: Some of this criticism was meanwhile addressed in [Panko and Aurigemma, 2010])

- The real quantitative (economic) impact resulting from spreadsheet errors is largely unknown.

- Existing studies about the frequency of spreadsheet errors are not comparable due to a lack of standardization.

- There is very little insight about where spreadsheet errors originate from and how to prevent them. Existing studies have been suggestive and based mostly on observations in laboratory experiments which cannot be applied to operational spreadsheets in the field.

- Approaches which claim to be effective in the detection of errors in spreadsheets are not described with enough details. Furthermore, it is mostly unknown how they compare in terms of effectiveness and efficiency.

Despite these significant uncertainties in spreadsheet error research, the number of commercial offerings for addressing spreadsheet quality problems is increasing steadily. The solutions offered range from error-detection tools, end-user training, consultancy for company policies, refactoring or reengineering of existing spreadsheets, to full migration projects towards „professional“ software. There is also a large amount of literature available where experienced practitioners proclaim „good“ and „poor“ practices regarding development, exploitation and management of spreadsheets. Some prominent examples include [Read and Batson, 1999], [Raffensberger, 2008], [O’Beirne, 2005], [Bovey et. al., 2009] or [Powell and Baker, 2010]. Particularly remarkable about these recommendations is that their authors have different opinions about some of the discussed practices.

This vast gap between practice and science leads to endless debates about basic principles for the proper handling of spreadsheets on various abstraction levels. For instance, on the process level there is a basic discussion about the right mixture between issuing regulatory steps and preserving the end-users’ freedom. On the implementation level, basic questions such as whether to break down a spreadsheet into separate areas, to use named ranges or to put multiple values in a single cell tend to split parties into separate camps.

At first glance, it seems obvious that the spreadsheet community must proceed towards an agreement on Best Practices for spreadsheets. But it is not clear whether this demand can be satisfied at all: Colver [Colver, 2004] and Grossmann [Grossman, 2002] argue that spreadsheet practices are always context-dependent, while Dunn [Dunn, 2010] claims that there must exist a large body of universal practices. At least, there seems to be a general agreement that certain practices which can be recommended in certain contexts do exist.
The current situation is not satisfactory, especially for quality managers who are expected to issue policies for the use of spreadsheets in their companies. They have a sound claim for science-based practices which reduce both the frequency and the impact of errors in their spreadsheets. We need to find a way to measure which practices lead to which kind of effects on spreadsheet quality when applied in which contexts. After identifying such practices, we should be able to gain more insight into the question whether good practices exist which can be applied independently of the spreadsheet's context.

This paper proposes an expert-based, retrospective approach to the identification of good practices for spreadsheets. Since the approach relies on a tool which is still under development, we cannot provide any concrete results yet. Nevertheless, we would like to present some details about our ongoing work and compare its conceptual direction with other approaches.

2 APPROACH

Our approach is based on a retrospective analysis of typical, existing spreadsheets – not on a greenfield strategy. There are numerous promising „Best Practice“ approaches to building better spreadsheets from scratch by using fundamentally different development techniques like [Dunn, 2009], [Paine, 2001] or [Grossman and Özlük, 2010] or switching to alternative technologies like [Miller, 2010]. We have chosen not to follow this path for several reasons:

- Time is a critical factor. There are incredible amounts of poor spreadsheets in use today. Of course, rewriting them from scratch might pay off somewhere in the future. But you won't get it done in the near future. We need a cheap yet effective short-term solution for the immediate problems.

- Convincing users is hard because users are conservative. Even if you succeed in developing a better spreadsheet approach, you still have to convince the end-users of its usefulness. This is rather unlikely to happen: Several conceptually promising approaches to re-defining the nature of a spreadsheet like Lotus Improv or its successor Quantrix Modeler never managed to reach and convince such masses as VisiCalc's logical successors Lotus 1-2-3 and Microsoft Excel did.

- Experience is an asset, and learning new paradigms is hard. Even if you manage to convince the end-users of a new paradigm's benefits, they will still have to learn to cope with it in practice. Their experience with existing tools and techniques could become worthless if the new paradigm is too different from what they know and how they use to work.

Therefore, instead of searching for new practices to (re)create better spreadsheets, we are trying to find out which of the already known practices are favourable. Because many of these practices have penetrated some of today's spreadsheets, we are attempting to investigate these spreadsheets, identify the practices used there and review their effectiveness based on the quality of the spreadsheets.

Figure 1 illustrates our approach. Briefly, it works as follows: First, we specify a policy
consisting of a set of rules based on suggested practices for spreadsheets from literature. Then we ask a human domain expert to review a given spreadsheet and to provide us with a subjective, general impression about it. In case the domain expert finds any errors, we ask him to report them, too. In parallel, we feed the spreadsheet into our workbench. The workbench checks the spreadsheet for any violations of the previously provided policy and summarizes them in a report. The report includes detailed information about which rules were violated. We repeat the process with other spreadsheets from the same domain, maintaining the same policy.

Finally, we compare the domain expert’s impressions with the rule violations reported by the workbench. This allows us to identify rules which correlate with the ratings of the expert. A “perfect” practice for the used context would be represented by a rule which does not report violations for spreadsheets the expert rated as “good” but does report violations for spreadsheets the expert rated as “poor”, without producing false positives or false negatives.

![Diagram](image)

*Figure 1: Our concept for evaluating spreadsheet practices*

### 2.1 Applicability

It is the nature of our retrospective analysis that it is based on the *current state* of the
inspected spreadsheet. The approach takes into account the fact that in many cases the steps suggested to be followed by spreadsheet practices leave visible traces in the final outcome. This manifestation makes it possible to verify whether the practices have been followed later on. For instance, it is possible to inspect all formulae in each row and column of a spreadsheet to check whether the practice “Use one formula per row or column” [Read and Batson, 1999] has been followed. In almost the same manner we could check dependencies between cells by analyzing their formulae to see whether the practice “Refer to the left and above” [Read and Batson, 1999] was obeyed.

Our approach is not limited to a specific taxonomy for the classification of spreadsheet errors or quality problems. We leave this choice completely up to the human domain expert who is needed for the manual inspection step in most cases. Theoretically, it would be possible to skip this step provided we had a comprehensive error log for a „known bad“ spreadsheet beforehand. Unfortunately, such error logs are very unlikely to be found in spreadsheet samples from the field.

It also does not matter what sort of spreadsheet is to be analyzed and how complex it is – as long as the domain expert is able to understand it, it's fine. But we demand that the domain expert must not be the author of the inspected spreadsheet as spreadsheet users are often over-confident about their own work [Brown and Gould, 1987] [Panko, 2003]. It is obvious that different experts might have different perceptions of „good“ and „poor“ spreadsheets. Our underlying assumption is that if many experts rate a specific spreadsheet as „poor“, there must be something wrong with it.

2.2 Limitations

Currently, our approach does not investigate prior states of the inspected spreadsheet. Thus, we cannot derive any conclusions about the process it has undergone. This includes the spreadsheet's initial creation as well as any manipulations performed on it until it reached the current state. As a consequence, the approach does not allow the evaluation of process-oriented development practices like „Write your application in the earliest version of Excel that you expect it to run in“ [Bovey et. al., 2009] or „Create and run test cases covering all logic paths“ [O'Beirne, 2005]. Without analyzing prior versions of spreadsheets we also cannot distinguish between „bad“ spreadsheets and „good spreadsheet that go bad during usage“ [Baxter, 2010].

Another limitation is that our approach relies on a tool for the automated checking of spreadsheets against previously specified rules. Therefore, we can only evaluate spreadsheet practices that can be checked by computer programs.

3 SPREADSHEET WORKBENCH

The development of our spreadsheet workbench aims at offering a multi-purpose „Swiss-Army-Knife“ tool for the quality assurance of spreadsheets. Unlike other tools which are mostly implemented as commercial add-ins for existing spreadsheet programs, our workbench
is a stand-alone, cross-platform tool written in Java and targeted to a broad audience:

- **End-users** can find quality problems in their spreadsheets before sharing their spreadsheets with peers or basing decisions on them.

- **Quality Managers** can define organizational policies for spreadsheets based on concrete practices accepted by their organization.

- **Spreadsheet auditors** can check whether both external and the internal end-users' spreadsheets comply with the organizational policies.

- **Researchers** can use the workbench as a tool to verify or falsify hypotheses about effectiveness and efficiency of spreadsheet practices.

Providing a detailed description of all the workbench's features would exceed the scope of this paper. Instead, we want to provide an example which illustrates how researchers following a retrospective analytical approach (like, but not limited to, an approach as described above) could use the workbench.

### 3.1 Creating a new policy

Policies are represented in the workbench by so-called „scenarios“ while spreadsheet practices are implicitly defined by so-called „rule checkers“. For instance, there could be a rule checker which checks for the presence of hardcoded constants in formulae.

A scenario defines a set of concrete rule checkers to be used. The rule checkers themselves also have customizable parameters, so a scenario represents a catalogue of individually configured rule checkers Our current UI prototype for this customization is shown in Figure 2.

In the initial version of the spreadsheet workbench, end-users are not able to specify new rule checkers themselves. But unlike tools such as Spreadsheet Professional [Spreadsheet Innovations, 2011] or XL Analyst [Codematic, 2011] the rules are not hardcoded. Instead, the workbench provides a Java API which allows professional programmers to implement their own rule checkers as plug-ins. For later versions of the workbench it is planned to replace this mechanism by a more end-user friendly facility.
We assume that a professional programmer has already implemented four rule checkers, which check for the following criteria:

- Are there hardcoded constants which are used in multiple formulae?
- Are there formulae in cells which do not have cell protection enabled?
- Are there formulae which refer to the right or below?
- Are there cells which consist only of one or more blanks (spaces)?

The first step for creating a new scenario is to select the desired rule checkers. To simplify matters we assume that we want to use the first two of the above rules for a scenario named „quarterly financial reports“. Each rule checker comes with a default configuration for its customizable parameters. For most rule checkers, their parameters allow to define thresholds and exceptions.

For instance, the „constants in formulae“ checker could have the option to ignore certain constants. This could come in handy if the spreadsheets in a scenario deal with percentages and thus often contain the constant „1“. Some practitioners argue that it is easier to read and verify formulae which contain values and not cell references in formulae, if the affected value

![Figure 2: UI Prototype of the spreadsheet workbench's rule checker customization dialog](image-url)
is short, simple and not expected to change (i.e. because it is a natural constant).

3.2 Running the analysis

After finishing the customization of the scenario, the workbench can be instructed to „run“ the scenario on a set of given input spreadsheets. Internally, the workbench makes use of an abstract spreadsheet model which allows it to bind to various spreadsheet formats and locations using external APIs. For the initial version, it is planned to support Microsoft Excel workbooks, OpenOffice.org Calc documents and online spreadsheets from GoogleDocs.

3.3 Reviewing the results

Once the workbench is done with the analysis of the input spreadsheets, it summarizes its findings in a report. Figure 3 illustrates our current prototype for this „report view“. The findings presented in the report can be filtered and grouped by various criteria. Currently this includes the cells which are affected, the rule checker which reported the findings and the affected spreadsheet.

The workbench also provides a detailed finding view. Using the API offered by the workbench, rule checkers can provide a textual description of the finding, a general explanation why this could lead to problems and a suggestion how the problem could be remedied by using alternative constructs. The workbench's API also allows each rule checker to render own visualizations for its findings.

3.4 Refining the scenario

After reviewing the workbench's report, researchers can further refine a scenario by adding or removing rule checkers, tweaking their configurations and re-running the analysis. After several iterations it should be possible to end up with a scenario which is mostly consistent with the issues which the domain expert identified in the manual review. In case this scenario even complies with ratings by other experts from the same domain, there is a chance that we have identified a promising candidate for a domain-specific policy. Shall the scenario even lead to findings which are consistent with the manual review of experts from other domains, then the practices represented by the scenario's rules are probably good candidates for „universally“ good spreadsheet practices.
CONCLUSION AND FUTURE WORK

We have proposed an approach towards the identification of good practices for spreadsheets. It is based on an evaluation loop that cross-validates the findings of human domain experts against rules implemented in a semi-automated spreadsheet workbench. Due to its retrospective nature, the approach cannot analyze process-oriented practices, though. For some process-oriented practices we could partly overcome this limitation by extending the analysis on prior versions of spreadsheets as well.

The workbench's development is still underway, so we cannot provide any evidence that the approach will work as expected. But given the case it will, there is a chance for getting closer to the goal of offering a science-based catalogue of accepted good practices for spreadsheets. This would be a vast improvement over the current situation of having to choose from conflicting practitioner recommendations.
5 ACKNOWLEDGEMENTS

The author would like to thank Michael Starzmann who contributed greatly to the development of the spreadsheet workbench described in this paper. The author is also grateful to Jochen Ludewig and Kornelia Kuhle for their constructive criticism of earlier versions of this paper.

6 REFERENCES


Raffensberger, J. (2008): "The Art of the Spreadsheet", http://john.raffensperger.org/ArtOfTheSpreadsheet, 2:00pm 05/25/11


Effect of Range Naming Conventions on Reliability and Development Time for Simple Spreadsheet Formulas

Ruth McKeever, Kevin McDaid
Dundalk Institute of Technology,
Dundalk, Ireland.
catherineruth.mckeever@dkit.ie, kevin.mcdaid@dkit.ie

ABSTRACT

Practitioners often argue that range names make spreadsheets easier to understand and use, akin to the role of good variable names in traditional programming languages, yet there is no supporting scientific evidence. The authors previously published experiments that disproved this theory in relation to debugging, and now turn their focus to development. This paper presents the results of two iterations of a new experiment, which measure the effect of range names on the correctness of, and the time it takes to develop, simple summation formulas. Our findings, supported by statistically significant results, show that formulas developed by non-experts using range names are more likely to contain errors and take longer to develop.

Taking these findings with the findings from previous experiments, we conclude that range names do not improve the quality of spreadsheets developed by novice and intermediate users.

This paper is important in that it finds that the choice of naming convention can have a significant impact on novice and intermediate users’ performance in formula development, with less structured naming conventions resulting in poorer performance by users.

1 INTRODUCTION

With the current estimate of Excel users worldwide estimated at 400m the importance of spreadsheets cannot be overstated. In 2007, Microsoft sold 71 million licenses for Microsoft Office [Microsoft, 2007]; in 2008 they sold 120 million [Microsoft, 2008]. An investigation into the importance and use of spreadsheets in the City of London [Croll, 2005] is summed by the following quote: “Excel is utterly pervasive. Nothing large (good or bad) happens without it passing at some time through Excel”. Spreadsheet development is increasingly recognized by authors as programming. For example, Burnett et al [2004] declare that “spreadsheet languages are the most widely used end-user programming languages to date—in fact, they may be the most widely used of all programming languages”. Panko and Halverson [1996] state that “quite simply, spreadsheeting is quite a bit like programming”.

The power and flexibility of spreadsheets make them an indispensable tool in modern business, at the cost of reliability. The ease with which any user can become a developer leads to widespread uncontrolled use, as they are not subject to rigorous testing procedures favoured by professional programmers. Consequently, spreadsheet development is known to be highly unreliable. Spreadsheets that are unregulated and untested are used for critical decision-making, leading to media reports of accidental and
intentional errors leading to fiscal and reputational losses. A number of examples can be found on the EuSpRIG website [EuSpRIG, 2010]. Reliability is arguably the most important aspect of spreadsheet quality. The reliability of a spreadsheet is measured by the accuracy of the information produced, and is damaged by the number and magnitude of errors that commonly occur. Range names are frequently recommended by academics and practitioners to improve the reliability of spreadsheets, yet no empirical evidence has been cited to support these recommendations.

The purpose of this study is to evaluate empirically the effect of different range naming conventions on spreadsheet reliability, and development time, by conducting two experiments. Both experiments have the same design, however the second improves on certain issues that arose during the first. This work represents the first scientific attempt to establish the benefits or otherwise of range names in the development of a spreadsheet. It is important in the context of the extensive use of spreadsheets and the recommendations of practitioners. Furthermore, when combined with earlier work, the paper presents important conclusions on the use of range names by non-expert users. The findings are clear; contrary to published opinion we find no evidence to endorse the use of range names in the development of reliable spreadsheets by novice and intermediate users. In particular, we find that poorly structured names have a particularly harmful effect on formula quality. While this may seem obvious to a practitioner, this paper presents empirical evidence to support this finding.

Section 1 of this paper begins by summarizing the important literature in this area, including experiments previously published by the authors that investigated the impact of range names on debugging. Section 3 details the methodology behind the experiment design, and in Section 4 the authors present in detail the results of both experiments. Section 5 comprises a discussion of the results, and Section 6 contains the conclusion, including proposals for future work.

2 LITERATURE REVIEW

It is widely acknowledged that spreadsheet use is ubiquitous in industry, however quality is often overlooked. Researchers frequently suggest that software programming practices should be applied to spreadsheets, in an attempt to advance quality. For example, “if our goal is to teach spreadsheeters how to develop spreadsheets professionally, we may be able to draw on what we already know from program development” [Panko and Halverson, 1996]. One such suggestion is that range names could be used in place of cell references, transferring the advantages of good variable names to the spreadsheet domain.

Naming of Cell Ranges

Range names are a powerful feature of Excel. In a spreadsheet, a range is defined as a cell or group of cells. A range name is a name given to a range, which can then be used throughout the workbook in place of the cell reference. It is often suggested that they can make a spreadsheet easier to understand and to develop. There is no existing research to back up this suggestion, and the aim of this project is to carry out the research to test this theory.

The advantages of naming conventions in programming are the subject of many studies. Keller [1990] established that people have less difficulty reading programs that follow a defined naming scheme, although they cannot explain why they find them easier to read. The legacy of using one-letter variables in mathematics is thought to be one cause of poor naming [Rowe, 1985]. An interesting theory is spelling mistakes: “if people make
spelling mistakes for words whose correct spelling they have seen countless times, it is certain that developers will make mistakes, based on the same reasons, when typing a character sequence they believe to be the spelling of an identifier” Jones [2008].

Range names are recommended by a number of authors, in published conference papers and journals. For example, Bewig [2003] states: “although it takes a little more work initially to create names, it should be clear they make formulas easier to write and to read. This is especially true in large spreadsheets where you may have scores of references.” The same author, in a subsequent paper [Bewig, 2005], advocates the proper construction of range names for eliminating the problem of referring to the wrong cell while constructing formulas, and states “well chosen names are the first and best form of documentation”. Grossman et al. [2009] illustrate how range names can be used to replace complex nested-if formulas with the lookup technique. As nested-if formulas are considered risky, the lookup technique uses names to make the logic required simpler, more visible and therefore less risky.

Range names are also frequently discussed by practitioners, in newsletters, websites and blogs. In CompAct, a newsletter for the Society of Actuaries, Campbell [2009] makes the extreme statement, that in order to improve adaptability “any cells being used in formulas should be referred to as named ranges.” An example of expert practitioners who have found names to be particularly useful is OPERIS. A spreadsheet consultancy firm based in London, OPERIS have made range names central to their spreadsheet development methodology. Support is provided by OAK, the OPERIS Analysis Kit [OPERIS, 2009] which has many features for handling names.

Not all practitioners are in favour of names. Panko and Ordway [2005] warn that range names “should be considered potentially dangerous until research on using range names is done.” Blood [2006] criticizes names for making formulas needlessly long and difficult to audit, stating that names are unnecessary if spreadsheet models are well designed.

**Range names and debugging**

Previously the authors carried out two experiments [McKeever and McDaid, 2010, McKeever et al., 2009] that investigated the impact of range names on the debugging performance of novice users. These were both adapted from a design first used in a study by Howe and Simkin [2006], and later used by Bishop and McDaid [2007]. Participants were given a spreadsheet seeded with errors, and were asked to correct any mistakes they could find, directly in the spreadsheet. They were not told how many errors were in the spreadsheet, or what types of errors were included. Their cell click times were recorded by T-CAT, a “time-stamped cell activity tracking tool” [Bishop and McDaid, 2008].

The results of both these experiments reveal that range names are a hindrance to novices either debugging or developing a spreadsheet. There was a reduction in overall error finding performance in both debugging trials, indicating that the problem is not just in how names are used in a formula, but with their overall inclusion in the spreadsheet. The failure to improve reliability or speed in the development task indicates that it is dangerous to recommend them to novice or intermediate users.

**Summary**

Extensive research has been carried out on spreadsheet errors, and concluded that spreadsheets have poor levels of reliability. Range names are frequently mentioned in the literature as a potential risk mitigation measure. Most of the experts, with a few notable
exceptions, recommend their use, however there is no research at present that examines in a structured and methodical manner the impact of names on spreadsheet quality.

The authors previously showed that range names do not improve the debugging performance of novice spreadsheet users. Debugging, however, is only one aspect of spreadsheet programming, arguably less important than development. As formulas have the greatest potential for material error in a spreadsheet, we designed an experiment that directly compared the reliability of formulas developed with range names to formulas developed with cell references.

The experiment described in this paper is part of a wider research project that investigates the impact of range names on different aspects of spreadsheet quality. The authors began by focusing on spreadsheet reliability, first debugging and now formula development. We hope to continue this research and evaluate the effect of range names on all aspects of spreadsheet development and use. Future research plans are described in more detail in Section 6.1.

3 METHODOLOGY

The aim of this experiment is to evaluate the reliability and completion time of simple formulas, and how this was influenced by using different types of naming structure. We began by investigating the null hypothesis, and from this developed a number of specific research questions, set out below.

3.1 Hypothesis and Research Questions

The following hypothesis was chosen as the basis for this study:

Range names, regardless of their structure, have no impact on the reliability and completion time of simple spreadsheet formulas, developed by novice and intermediate users.

From this hypothesis we derived the following research questions:

RQ1: Do users make more mistakes using range names or cell references, when asked to develop a simple spreadsheet formula?

RQ2: Does the time it takes users to develop a simple spreadsheet formula differ for formulas using range names than for formulas using cell references?

Range names, as with programming variables, can be chosen according to various conventions. The work will examine each of the research questions above for each of the following six range naming structures:

a) Where no two names begin with the same word.
b) Where several different names begin with the same word, but end in a different word.
c) Where several different names begin and end in the same words, with a change in the number in the middle of the name.
d) Where names begin with the same word with a change in the trailing number.
e) Where several different names begin and end with the same word, with a change in the word in the middle.
f) Where names do not follow any naming convention, and are inconsistent.
The measurement for RQ1 is the number of correct formulas developed. The measurement for RQ2 is the time it takes to develop each formula.

3.2 Task Design

It was decided that the most appropriate way to address these questions would be to isolate a basic formula task, using names that followed the naming conventions specified above. Simple addition through use of the additive operator was chosen, as it is one of the most basic, well-known actions in Excel. This does not require any knowledge of Excel functions, and no users implemented the formula using the inbuilt SUM function. The spreadsheet was designed so that the participants would not require domain knowledge in order to understand the spreadsheet and the tasks.

One spreadsheet was used in this experiment, containing six worksheets. Each worksheet contained two identical tasks, based on different lists of data. For one list the participants were required to use range names for the task, for the other, cell references. Each sheet addresses one of the six naming structures, in the same order as they are listed above. All these structures were based on a combination of row and column headings. This resulted in a total of 12 tasks, six with range names, and six with cell references for each participant. Each task requires the user to select seven cells or range names.

Sheet 1: Very simple names, that follow a strict naming convention. No two names begin with the same word. Examples include ArnottsSales, ClearysSales, TopshopSales.

Sheet 2: Two names can begin with the same word. Examples include TopshopGP, TopshopNP. After the first implementation of this experiment, it was decided that GP and NP were too similar, as there was only a difference of one letter and the last letter of each name was the same. The words used were changed to Gross and Net, for example, TopshopGross and TopshopNet.

Sheet 3: Names follow a slightly more complex naming convention, whereby two names can both begin and end in the same word, but contain a different number in the middle. Examples include HMV2008Profits, HMV2009Profits.

Sheet 4: Names begin with the same word, but end in a different number. Examples include PrimarkTax2006, PrimarkTax2007.

Sheet 5: Names begin and end in the same words, but have a different word in the middle. Examples include GAPSecWages, GAPSupWages. After the first implementation of this experiment, it was decided that Sup and Sec were also too similar, hence these words were changed to Assist and Clerk.

Sheet 6: Names that do not follow any naming convention. Examples include CostsQuicksilverFixed, QuicksilverVariableCosts.

It was decided that sheet level names were most suitable, as the user would only be able to see the names that were developed for the particular task on which they are working. This was to avoid confusion, as there were 264 names in the spreadsheet. The aim was not to confuse the users, but to examine the naming conventions in isolation.

As described earlier, the study was structured on the basis of a within subject design where each participant implemented a simple formula using both cell reference and range names for each of the six range name types identified in the research questions. The order
in which the users carried out the tasks was varied by dividing the subjects randomly into
two groups i.e. Group A began with range names, then used cell references; Group B
began with cell references then moved on to range names. This was done to factor out
any learning effect, i.e. that the users might be better prepared to carry out the second task
on each sheet, as they had learned what to do from the first task.

3.3 **Operational Context**

This experiment was carried out in a computing laboratory in Dundalk Institute of
Technology. Each participant used the PCs installed in this room, running Windows 7
and Microsoft Excel 2007. The participants were randomly assigned to groups A and B,
and the researcher explained how to do the experiment. They were then observed whilst
carrying out the tasks. As in previous experiments, the T-CAT macro ran in the
background for all the participants, and recorded the time it took to complete each task.

The participants in the first implementation of this experiment, Group 1, were 15
postgraduate students from the Higher Diploma in Computing class in Dundalk Institute
of Technology, Ireland. Most of them had returned to education after a period in the
workplace. The participants in the second implementation, Group 2, were 17 second-year
Software Development students, also from Dundalk Institute of Technology. Based on a
background survey the participants in both groups were judged to be a mix of novice and
intermediate users.

3.4 **Example Tasks**

Shown in Table 1 are two examples of tasks that were given to the participants in Group
A. These tasks were included on Sheet 5. The participants in Group B were given the
same two tasks, except in Task 3 they were asked to use cell references, and in Task 4,
range names. Examples of incorrect formulas developed as answers to these tasks are
displayed in the following section.

<table>
<thead>
<tr>
<th>Task 3: With a formula that uses range names, in cell C28 calculate the total of the following store profits:</th>
<th>Task 4: With a formula that uses cell references, in cell C57 calculate the total of the following store profits:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Net profit for Powerhouse</td>
<td>Net profit for Dillons</td>
</tr>
<tr>
<td>Gross profit for Pumpkin Patch</td>
<td>Net profit for Dixons</td>
</tr>
<tr>
<td>Gross profit for Austin Reed</td>
<td>Gross profit for Dillons</td>
</tr>
<tr>
<td>Net profit for Zara</td>
<td>Net profit for Liberty</td>
</tr>
<tr>
<td>Net profit for Quicksilver</td>
<td>Gross profit for The Paper Mill</td>
</tr>
<tr>
<td>Gross profit for Quicksilver</td>
<td>Gross profit for Boots</td>
</tr>
<tr>
<td>Net profit for Topshop</td>
<td>Net profit for Etam</td>
</tr>
</tbody>
</table>

Table 1 - Example Tasks

4 **RESULTS AND DISCUSSION**

This section first examines the amount of errors made in each task, looking at whether
more incorrect formulas were developed using range names or cell references. Next, the
different type of errors that occurred are examined, with a focus on whether certain errors
were more likely where names were used. Finally there is an analysis of the time it took
the subjects to develop the formulas for each task.
Errors made per task

Table 2 shows the number of subjects who created incorrect formulas in each of the sheets on the spreadsheet, according to whether the task included range names or cell references.

<table>
<thead>
<tr>
<th>Errors</th>
<th>Group 1 (15 subjects)</th>
<th>Group 2 (17 subjects)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sheet</td>
<td>Named Ranges</td>
<td>Cell References</td>
</tr>
<tr>
<td>Sheet 1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Sheet 2</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>Sheet 3</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Sheet 4</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Sheet 5</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Sheet 6</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td><strong>12</strong></td>
<td><strong>4</strong></td>
</tr>
</tbody>
</table>

Table 2 - Errors per Task

The results indicate that participants were less effective at developing formulas using range names than using cell references. For the first group statistical analysis, using McNemar’s test for paired proportions indicates that there is significant evidence that in the case of Sheet 2 and Sheet 3 users were more effective at developing simple formulas using cell references as opposed to range names. The statistical tests for Group 1 support the following statement, referring to RQ1:

Novice and intermediate users make fewer mistakes when developing formulae using cell references than using range names where:

a) Several different names begin with the same word, but end in a different word.

b) Several different names begin and end in the same words, with a change in the number in the middle of the name.

For Group 2, there is statistically significant evidence that in the case of Sheet 5 users were more effective at using cell references to develop simple formulas. The statistical tests for Group 2 support the following statement, referring to RQ1:

Novice and intermediate users make fewer mistakes when developing formulae using cell references than using range names where several different names begin and end with the same word, with a change in the word in the middle.

The fact that these findings were inconsistent between the two experiments indicates that the changes made to the experiment design had an impact on the users’ performance. In Sheet 5 the words ‘Sup’ (supervisor) and ‘Sec’ (secretary) were changed to ‘Assist’ (assistant) and Clerk, as ‘Sup’ and ‘Sec’ were deemed too similar to represent a change in the middle word of the range name. It appears that the similarities in these words also served to confuse the subjects when using cell references, and the change to more obviously different words highlighted the effect of the naming convention, as opposed to
the effect of the actual word. In Sheet 2 the naming convention was changed for the second experiment to replace the terms ‘NP’ (net profit) and ‘GP’ (gross profit) with ‘Net’ and ‘Gross’ respectively, as ‘NP’ and ‘GP’ were not deemed sufficiently different to represent a change in the final word in the naming structure. This may explain why the subjects in Group 2 made fewer errors than the subjects in Group 1, however the column headings did not change, as they did in Sheet 5, hence we cannot offer an explanation for the increase in cell reference based errors. Likewise, no changes were made to Sheet 3.

For two naming conventions no mistakes were made using range names, while one mistake was made using cell references for each of Groups 1 and 2, although this is not statistically significant. From Group 1, for two naming conventions there were statistically significant results to conclude that they negatively affect the reliability of the formula; from Group 2, there was statistically significant results to conclude that one naming convention negatively affects the reliability of the formula. These conflicting results suggest that naming conventions have an impact on the effectiveness of range names. Taking the results of the six naming conventions combined there is evidence to suggest that range names should not be used in formulas.

**Error Types**

While the results detailed above refer to the number of incorrect formulas developed by the subjects, it is worth looking at the actual formulas to see what type of errors were made. In several cases the offending formula contained more than one error.

It was expected that the participants would make selection and omission errors under both conditions. Selection errors occur when the wrong cell or name is used in a formula; omission errors occur when a reference is left out. Looking at both Groups 1 and 2 together, of the 22 incorrect formulas developed using range names 10 contained selection errors and 14 contained omission errors. Of the 9 incorrect formulas developed using cell references, 2 contained selection errors, and 7 contained omission errors.

An unexpected finding was that some participants, while working with range names, made another type of error: in three cases a subject added an extra name to the formula. During Task 3 a subject from Group 1 added the name ZaraNP twice, and during Task 6 a subject from Group 1 included Costcutter2008Profits as well as Costcutter2009Profits. During Task 9, a subject from Group 2 included OasisClerkWages as well as OasisAssistWages. This type of error did not occur during the cell reference tasks.

Below are two examples of formulas that contain omission errors developed as answers by participants from Group 1 to the two tasks detailed in the previous section:

One participant, from Group 1, gave the following erroneous answer to Task 3:

\[=\text{PowerhouseNP}+\text{PumpkinPatchGP}+\text{AustinReedGP}+\text{ZaraNP}+\text{ZaraNP}+\text{QuicksilverNP}+\text{TopshopNP}\]

The correct answer would also include “QuicksilverGP”, and omit one instance of “ZaraNP”.

Another participant, from Group 2, gave the following erroneous answer to Task 4:

\[=\text{SUM(B35,B36,B41,C48,C51,B38)}\]

The correct answer should include cell C35, the value for Dillons Gross profit.
Time Results

Table 3 shows the average time in minutes it took the participants to complete the tasks on each sheet, according to whether the task included range names or cell references. It took Group 1 on average 0.7 minutes longer, and Group 2 0.77 minutes longer, to complete the tasks when range names were used. These times include the assimilation time for both beginning the experiment, and for each sheet. It can be presumed that the subjects would take less time to complete the second task on each sheet, as they are already familiar with the sheet since doing the first task. Dividing the participants randomly into groups A and B so that some would complete the name tasks first and others would complete the cell reference tasks first eliminated this bias.

<table>
<thead>
<tr>
<th>Times</th>
<th>Group 1</th>
<th></th>
<th>Group 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sheet</td>
<td>Named Ranges</td>
<td>Cell References</td>
<td>Named Ranges</td>
</tr>
<tr>
<td>Sheet 1</td>
<td>1.98</td>
<td>1.33</td>
<td>1.74</td>
</tr>
<tr>
<td>Sheet 2</td>
<td>1.95</td>
<td>1.30</td>
<td>2.43</td>
</tr>
<tr>
<td>Sheet 3</td>
<td>1.45</td>
<td>1.10</td>
<td>1.65</td>
</tr>
<tr>
<td>Sheet 4</td>
<td>1.35</td>
<td>1.04</td>
<td>1.57</td>
</tr>
<tr>
<td>Sheet 5</td>
<td>1.31</td>
<td>1.19</td>
<td>1.38</td>
</tr>
<tr>
<td>Sheet 6</td>
<td>3.19</td>
<td>1.04</td>
<td>4.00</td>
</tr>
<tr>
<td>Total</td>
<td>1.87</td>
<td>1.17</td>
<td>2.13</td>
</tr>
</tbody>
</table>

Table 3 - Average Time (Minutes)

Statistical analysis of the significance of the difference in the times it took to perform each task was performed using a paired T-test. For Group 1 these indicate that the time it took users to complete the tasks associated with sheets 2, 3, 4 and 6 for range names as opposed to cell references was significantly higher at a 5% level of significance. The statistical tests for Group 1 support the following statement, referring to RQ2:

Novice and intermediate users take longer to develop formulae using range names than using cell references where:

a) Several different names begin with the same word, but end in a different word.

b) Several different names begin and end in the same words, with a change in the number in the middle of the name.

c) Names begin with the same word with a change in the trailing number.

d) Names do not follow any naming convention, and are inconsistent.

For Group 2, the statistical tests indicate that the time it took users to complete the tasks associated with sheets 2, 4, 5 and 6 for range names as opposed to cell references was significantly higher at a 5% level of significance. The statistical tests for Group 2 support the following statement, referring to RQ2:

Novice and intermediate users take longer to develop formulae using range names than using cell references where:
a) Several different names begin with the same word, but end in a different word.
b) Names begin with the same word with a change in the trailing number.
c) Several different names begin and end with the same word, with a change in the word in the middle.
d) Names do not follow any naming convention, and are inconsistent.

The clear consistency in this result is that for the most structured naming convention there is no significant result to differentiate between development time for range names and cell references. For the least structured names, there are significant results from both experiments that find it takes more time to develop a formula using range names. For the various degrees of structure in-between, the findings are less conclusive, yet indicate that it is faster to develop using cell references.

For each naming convention, the average time to complete the task was higher where the participant used range names. This was statistically significant for four naming structures from each experiment. The number was particularly high for the last structure – where no naming convention was followed. This is not surprising, as the participants did not know what letter each name began with, and hence did not know where to look for the name in an alphabetically ordered list, or what letter to begin typing. This highlights the importance of following some kind of convention, if range names are chosen as a suitable practice. Further investigation should be carried out on the role of alphabetisation and the ease with which subjects can select a name from a list.

One possible reason for the increase in development time using range names is that it would naturally take longer to type a sequence of characters than to click on a cell. All participants were shown how to chose a name by clicking the *Use in Formula* button to display a list of all the names available in the worksheet. We argue that the time it takes to do this is comparable to the time it takes to click a cell. The list of names is sorted alphabetically, making it easier for the user to find the name they are looking for, unlike the list of data displayed on the worksheet, which is not sorted. During the experiment all subjects were observed following this method for choosing a name, instead of typing the name.

5 THREATS TO VALIDITY

In this section the authors first discuss the threats to construct validity, followed by internal and external validity.

**Construct Validity**

The authors contend that the hypothesis can be evaluated sufficiently by measuring the number of correct formulas, and the time it takes to develop each formula. One potential issue is that the time measurement is calculated by recording the time spent between the user entering the cell and leaving it after completing the formula. This includes the time it takes the user to read the task and comprehend the model.

Another issue regarding construct validity is the decision to measure performance on a summation task only. A wider selection of tasks would increase construct validity, but reduce internal validity when drawing comparisons between results on such a small sample size.

**Internal Validity**
In the first implementation of this experiment the words chosen for the range names on two sheets were identified as being too similar. While this may be a common occurrence in real world spreadsheets, it affects the internal validity of this study as any error made could be attributed to the similarities rather than the naming convention. Different words were chosen for the second implementation.

One issue with this experiment was that when the user clicked on a cell that was named while writing a formula, the name of the cell appeared in the formula rather than the cell reference. Unfortunately this is not something that can be caught by the T-CAT macro, as cell selections cannot be recorded while a formula is being edited. The participants were given clear instructions, and were closely monitored to ensure that they followed the instructions exactly. For Group 2, the cells that were named were also locked, so that the participants could not click on the cell to choose the reference. One participant tried to type the cell reference manually into the formula, but was immediately asked to use the range names instead. A better solution to this issue must be developed before this experiment could be rolled out to a larger group, as it is unnatural that a user would not be able to click on a cell in a working environment.

External Validity

Students were used in these experiments. Although this approach is controversial, we argue that the results can be generalized to the majority of spreadsheet users, but not experts who regularly use range names as part of their development methodology. Studies have shown that students have similar abilities to professionals, for example Galletta et al. [1993] found that spreadsheet experts did not outperform novices in finding spreadsheet formula errors. Considerable research has recognized that spreadsheets are rarely developed by professional programmers. For example, Purser and Chadwick [2006] found that 85% of survey participants developed the spreadsheets that they use. This is not to say that the developers are not professionals, but that their expertise lies within their domain rather than in programming.

The subjects who took part in these trials were taught by the researchers how to use range names. This was necessary for the purposes of the trial, but does not reflect how novice and intermediate practitioners would learn about naming in the real world. As mentioned in Section 5, they were taught how to use the Use in Formula feature to choose a name. This may not reflect how names are used in practice.

6 CONCLUSION

Described in this paper is a new experiment design, that looks at the reliability of formulas developed using structured and unstructured range names, and the time to develop such formulas. The results of two iterations of this experiment are detailed, and the findings discussed. The authors have three main conclusions.

First, there is no evidence to support the theory that range names lead to more reliable formulas. The subjects in the first group developed twelve incorrect formulas using range names, and four incorrect formulas using cell references. The subjects in the second group developed ten incorrect formulas using range names, and six incorrect formulas with cell references. With range name use there was an increase in the number of omission errors, and subjects made some addition errors, which did not occur with cell references. This leads us to conclude that range names have a negative impact on the reliability of cell formulas developed by novice and intermediate users.
Second, we found evidence that formulas developed using range names took longer to develop than formulas using cell references. For the first group, the average time it took to develop each formula was longer for all the range name tasks than the corresponding cell reference tasks. This was statistically significant for four out of six cases. For the second group, the average time it took to develop five out of six range name formulas was longer than the corresponding cell reference formula; the average times were identical for one formula. Again, this is statistically significant in four out of six cases. This leads us to conclude that range names make formulas longer to develop.

Third, we found evidence that the degree to which the range name is structured has an impact on both the reliability of, and the time it takes to develop, the formula. The most structured naming convention did not cause a significant increase in errors, or in time to develop. The formulas developed using names that followed no naming convention took significantly longer to develop, and averaged more errors (although the error rate was not statistically significant). Although the authors do not endorse range name use on the basis of the previous conclusions, if a practitioner decides to use range names in a spreadsheet we recommend they follow a strict naming convention.

The naming conventions that were found to have a significant negative impact on reliability also took significantly longer to use in a formula. This is a clear indictment of their suitability as naming conventions. Importantly, the increase in selection errors illustrates that range names do not help the user to avoid referring to the wrong cell, as is often claimed. The increase in the time it takes to develop a formula dispels the theory that range names make formulas easier to create.

These findings are consistent with the findings of previous experiments, and indicate that range names do not make formulas easier for novice and intermediate users to either develop or debug. The resulting spreadsheets are less reliable than those developed using cell references. This study is part of on-going research into the role of range names in the entire spreadsheet development process. The following section details how the authors plan to proceed with this research.

6.1 Future work

Firstly, it is crucial to the validity of this research that the scope of the project be increased to focus on professionals. We also plan to examine how range names are used in practice, and why some practitioners choose to use them. The authors have not yet looked at whether range names reduce development time when the range is located on a different sheet. If this was found to be the case then this time saving might be worthwhile to the developer. If range names do not improve the reliability of spreadsheets, or make them easier to debug, they could be used to improve other aspects of spreadsheet quality, such as understandability.

An alternative route of enquiry would be to investigate why range names so clearly confuse users. After the initial debugging experiments two factors were briefly examined as possible causes. High cognitive load was investigated, as participants must remember what each name refers to when analysing a formula, whereas with a cell reference they can see exactly where is being referred to. Overconfidence was also suggested as a contributing factor. Panko [2003] suggests that overconfidence has an impact on spreadsheet error rates, emphasizing the self reinforcing nature of risky behaviour in relation to spreadsheets: “developers who do not do comprehensive error checking are rewarded both by finishing faster and by avoiding onerous testing work”.

Proceedings of EuSpRIG 2011 Conference
Copyright © 2011 EuSpRIG (www.eusprig.org) and the Author(s)
REFERENCES


OPERIS 2009. Operis Analysis Kit. 4.0 ed.


An Empirical Study on End-users Productivity Using Model-based Spreadsheets

Laura Beckwith
HCIResearcher, Denmark
beckwith@hciResearcher.com

Jácome Cunha
Universidade do Minho, Portugal
jacome@di.uminho.pt

João Paulo Fernandes
Universidade do Minho & Universidade do Porto, Portugal
jpaulo@{di.uminho.pt,fe.up.pt}

João Saraiva
Universidade do Minho, Portugal
jas@di.uminho.pt

ABSTRACT

Spreadsheets are widely used, and studies have shown that most end-user spreadsheets contain non-trivial errors. To improve end-users productivity, recent research proposes the use of a model-driven engineering approach to spreadsheets.

In this paper we conduct the first systematic empirical study to assess the effectiveness and efficiency of this approach. A set of spreadsheet end users worked with two different model-based spreadsheets, and we present and analyze here the results achieved.

1 INTRODUCTION

Spreadsheets can be viewed as programming environments for non-professional programmers, the so-called “end users” [Nardi, 1993]. An end user is a teacher, an engineer, a secretary, an accountant, in fact almost anyone except a trained programmer. These people use computers to get their job done; often they are not interested in programming per se. End-user programmers vastly outnumber professional ones creating every year hundreds of millions of spreadsheets [Scaffidi et al., 2005]. As numerous studies have shown, this high rate of production is accompanied by an alarming high rate of errors, with some reporting that up to 90% of real-world spreadsheets contain errors [Panko, 2000, Rajalingham et al., 2001, Powell and Baker, 2003].

In order to overcome these limitations of spreadsheets, a considerable amount of research has been recently done by the human computer interaction community [Abraham and Erwig, 2006, Cunha et al., 2010, 2009b,a, Engels and Erwig, 2005, Erwig et al., 2005]. One of the promising solutions advocates the use of a Model-Driven Engineering (MDE) approach to spreadsheets. In such an approach, a business model of the spreadsheet data is defined, and then end users are guided to introduce data that conforms to the defined model [Cunha et al., 2009b]. Indeed, several models to represent the business logic of the spreadsheet have been proposed, namely, templates [Abraham and Erwig, 2006, Erwig et al., 2005], ClassSheets [Cunha et al., 2010, Engels and Erwig, 2005], relational models [Cunha et al., 2009a]. Several techniques to infer such models from a (legacy) spreadsheet data have also been studied [Abraham and Erwig, 2006, Cunha et al., 2010].

Although all these works claim that a MDE approach improves end-users productivity, there is no detailed evaluation that supports this idea besides our first attempt in [Cunha et al., 2011]. In this paper, we present a complete empirical study that we have conducted with the aim of analyzing the influence of using models in end-users spreadsheet productivity. In this study we consider two different model-based spreadsheets, as proposed in [Cunha et al., 2009a,b]. We assess end-users productivity in introducing, updating and querying data in those two model-based spreadsheets and in a traditional one. As the models we consider represent database-like spreadsheets only it should be clear that we are not analyzing all possible (types of) spreadsheets. Nevertheless, even considering spreadsheets strongly related to databases, the domain of our tests is clearly the spreadsheet environment.

In this paper we wish to answer the following research questions:

**RQ1** Do end users introduce fewer errors when they use one of the model-based spreadsheet versus the original unmodified spreadsheet?

**RQ2** Are end users more efficient using the model-based ones?

**RQ3** Do particular models lead to fewer errors in particular tasks?

The study we conducted to answer these questions is necessary and useful, since it is based on a sound experimental setting and thus allow us to draw sound conclusions for further studies on how to improve spreadsheet end users productivity.

This paper is structured as follows: in Section 2 we present the model-based spreadsheets we considered in our study. In Section 3 we describe the design of our study. We present and analyze in detail the results of our study in Section 4. Several threats to validity are discussed in Section 5 and in Section 6 we draw our conclusions.

## 2 MODEL-BASED SPREADSHEETS

There have been proposed two different techniques to tackle the problem of preventing errors in spreadsheets [Cunha et al., 2009b,a]. In order to introduce these works we will rely on the spreadsheet shown in Figure 1. This spreadsheet represents a movie renting system registering movies, renters and rents. Labels in the spreadsheet should be self-explanative.

### 2.1 The Refactored Spreadsheet Model

The spreadsheet shown in Figure 1 defines a valid model to represent the information of the renting system. However, it contains redundant information. For example, the information about the client **Paul** appears four times in the spreadsheet! This kind of redundancy makes the maintenance of the spreadsheet complex and error-prone, specially for end users. A mistake is easily made, for example, by mistyping a name and thus corrupting the data.
The same information can be stored without redundancy. In fact, in the database community, techniques for database normalization are commonly used to minimize duplication of information and improve data integrity. Database normalization is based on the detection and exploitation of functional dependencies inherent in the data [Codd, 1970]. We have adapted these techniques to work with spreadsheets: from the spreadsheet data we infer a set of normalized FDs, and from them, we compute a relational model [Cunha et al., 2009a]. A spreadsheet respecting such model is shown in Fig. 2.

The obtained modularity solves two well-known problems in databases, namely update and deletion anomalies [Codd, 1970]. The former problem occurs when we change information in one tuple but leave the same information unchanged in the others. In our example, this may happen if the user changes the rent per day of movie number mv23 from 0.5 to 0.6. In the modular spreadsheet that value occurs only once in the movie table and so that problem will never occur. The latter problem happens when we delete some tuple and lose other information as a side effect. For example, if the user deletes row 3 in the original spreadsheet all the information about movie mv1 is eliminated. Since we have a deep knowledge about the relations and relationships in the data, we can generate spreadsheets that respect them. For example, in the renter table, the generated spreadsheet will not allow the user to introduce two renters with the same number (renterNr). If that error occurs the spreadsheet system should warn the user as shown in Figure 3. Obviously, it is not possible to perform this validation in the original spreadsheet. The refactored spreadsheet not only improves modularity and detects the introduction of incorrect data, it also eliminates redundancy: the redundancy present in the original spreadsheet has been eliminated. As expected, the information about renters (and movies) occurs only once. These features should help end users to commit less errors. In the sequel, this model will be referred as refactored.

2.2 The Visual Spreadsheet Model

In [Cunha et al., 2009b], the authors proposed a technique to enhance a spreadsheet system with mechanisms to guide end users to introduce correct data. Using the relational database schema induced by the data we construct a spreadsheet environment that respects that schema. For example, for the movie spreadsheet, the system does not allow end users to introduce two different movies with the same number (movieID). Instead, it offers to the user a list of possible movies, such that he can choose the value to fill in the cell. This new spreadsheet, that we show in Figure 4, also includes advanced features which provide information to the end user about correct data that can be introduced.
We consider 3 types of advanced features. First, we consider bidirectional auto-completion of column values: based on the relational schema, we know that some columns (consequents) depend on another column (antecedents). Both antecedent and consequent columns have combo boxes that allow users to choose values instead of writing them. Using this knowledge we created a mechanism that automatically fills in consequent columns when antecedent columns are filled in. When values are written in consequent columns, the values in the antecedent columns are filtered, showing only the ones that can imply the chosen consequents. Using the bidirectional auto-completion feature the spreadsheet system guarantees that the user does not introduce data that violates the inferred model. The second feature is non-editable columns: this feature prevents the user from editing consequent columns since this could break the relationship with the antecedents. Note that, such columns are automatically filled in by selecting the corresponding antecedent. Finally, we consider safe deletion of rows: the user receives a warning when deleting of a row provokes the deletion of data not represented elsewhere.

Like in modern programming environments, the refactored spreadsheet system also offers the possibility of using traditional editing, i.e. the introduction of data by editing each of the columns. When using traditional editing the end user is able to introduce data that violates the model inferred from the previous spreadsheet data. The spreadsheet environment includes a mechanism to re-calculate the relational database model after traditional editing. This new relational model is used to guide the end user in future editing. From now on, this model of spreadsheet will be referred as the visual model.

3 STUDY DESIGN

As suggested in [Perry et al., 2000] we organized the study as follows:

1. Formulating hypothesis to test: we spent a considerable amount of time organizing our ideas and finally formulating the hypothesis presented in this work: model-based spreadsheets can help end users committing less errors editing and querying spreadsheets.

2. Observing a situation: once we got enough and appropriate qualified participants we ran the study itself. During the study, we screen casted the participants’ computers and afterwards we collected the spreadsheets they worked on.

3. Abstracting observations into data: we computed a series of statistics, that we present in detail in Section 4, over the spreadsheets participants developed during the study: we graded their performance and measured the time they took to perform the proposed tasks. All the data we used is available at the SSaPP project web page http://ssaapp.di.uminho.pt. The tasks and spreadsheets participants received are also available.

4. Analyzing the data: the enormous collection of data that we gathered was later systematically analyzed. This analysis is also presented in this paper, in Section 4.
5. Drawing conclusions with respect to the tested hypothesis: based on the results we obtained, we finally draw some conclusions. We were also able to suggest some future research paths based on our work, which are presented in the Section 6.

Our study aimed to answer if participants were able to perform their tasks with more accuracy and/or faster given the experimental environments. We used a within-subjects design, where each participant received 3 spreadsheets, one for each problem (DISHES, PROPERTIES, PROJECTS). Each of the 3 spreadsheets was randomly distributed under one of the 3 models (original, visual, refactored). Participants were asked to do various tasks in each spreadsheet: data entry, editing, and calculations. They were encouraged to work as quickly, but were not given time limits.

3.1 Methodology

Participants started the study by filling out a background questionnaire so we could collect their area of study and previous experience with spreadsheets, other programming languages and English comfort (Portuguese is their mother language). An introduction to the study was given orally in English, this was explicitly not a tutorial for the different environments because the goal was to see if even without any introduction to the various models the participants would still be able to understand and complete the tasks. The participants were asked to work as quickly and accurately as possible. Since the order of the spreadsheets was randomized, they were told that the other sitting around them might appear to be moving faster, but that some tasks were shorter than others. After 2 hours participants were stopped if they were not already finished. Following the tasks they had a post session questionnaire which contained questions assessing their understanding of the different models, (3 questions for refactored and 4 for visual).

Correct answers could only be given by participants having understood the running models. Grading the questionnaires was done as follows: a correct answer receives total points; an incorrect answer receives 0 points and an answer that is not incorrect nor (totally) correct receives half of the points. We recorded the users screens using screen capture technology. At the end of the study the users completed spreadsheets were saved and graded for later analysis.

3.2 Participants

Recruitment was conducted through a general email message to the university, asking for students with spreadsheet experience and comfort with English. Of the hundreds that responded (here was a compensation involved), participants were selected based on spreadsheet experience, comfort with English, and majors outside of computer science and engineering. In total, 38 participants finished the study with data we were able to use (25 females, 11 males, and 2 who did not answer about their gender). Two participants did not try to solve one of the proposed tasks; for these participants, we included in the study only the tasks they undertook. A few participants’ machines crashed and therefore they were eliminated from the study. The majority of participants were between 20-29 years of age, with the remaining under 20. All were students at the university. About 2/3 were working on their Baccalaureate degree, the remaining on their Masters. None were studying computer science or engineering and the most represented majors were medicine, economics, nursing and biology. A variety that is good for representing the end-user population of spreadsheets.
3.3 Tasks

The task lists were designed to include tasks that are known to be problematic in spreadsheets, which involve data insertion, edition and the use of formulas. The tasks were 1) add new information to the spreadsheet, 2) edit existing data in the spreadsheets and 3) do some calculations using the data in the spreadsheets.

Some of the tasks asked users to add many new rows of data, with the aim of a repetitive task being common in real-world situations. As we were designing the tasks, we imagined a type of data entry office scenario, where an office worker might receive on paper data which was initially filled out on a paper form and needed to be entered into a spreadsheet. This first task of data entry, in theory, should be fastest (and done with fewest entry errors) in the refactored spreadsheet. The second task, of making changes to existing data in a spreadsheet should also be easier within a refactored spreadsheet, since the change only needs to be made in one location, and therefore there would be less chance of forgetting to change it. The final task was to do some calculations using the data in the spreadsheet, such as averages, etc. This task was added because of the frequency of problems with formulas. One of the spreadsheets used in the study, PROJECTS, stores information about a house renting system (adapted from [Connolly and Begg, 2001]). This spreadsheet has information about renters, houses and their owners as well as the dates and prices of the rents.

A second spreadsheet, DISHES, contains information about sells of detergents to dish washers. Information about the detergents, prices and the stores where they are sold is present on this spreadsheets (adapted from [Powell and Baker, 2003]).

The last spreadsheet, PROJECTS, stores information about projects, like the manager and delivery date, employees instruments used (adapted from [Alhajj, 2003]).

In the task list for DISHES, 67% (39 out of 58 cells needed to be changed) of the tasks consist of inserting new data, 21% (12/58) are editing tasks and 12% (7/58) involve calculations over the data in the spreadsheet. In the task list for PROJECTS, 80% (221/277) of the tasks are for inserting new data, 7% (20/277) for edition and 13% (36/227) for calculations. Finally, for PROJECTS, inserting data tasks are 56% (64/115) of the total, whereas data editing and calculation tasks are 19% (22/115) and 25% (29/115) of the total, respectively. Grading the participants’ performance was done as follows. For tasks involving adding new data to the spreadsheet or performing calculations over spreadsheet data, whenever a participant executes a task as we asked him/her to, he/she is awarded 100% of the total score for that task; on the contrary, if the participant does not at all try to solve a particular task, he/she gets no credit for that. An intermediate situation occurs when participants try to solve a task, but fail to successfully conclude it in its entirety. In this case, the participant is awarded 50% of the score for that task. For tasks involving editing data, a value in the interval $0\% - 100\%$ is awarded according to the participants’ success rate in such tasks. Table 1 shows the number of participants that worked on each spreadsheet and each model. Note that the distribution of models and spreadsheets by the participants is homogeneous.
<table>
<thead>
<tr>
<th></th>
<th>original</th>
<th>refactored</th>
<th>visual</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISHES</td>
<td>12</td>
<td>13</td>
<td>12</td>
<td>37</td>
</tr>
<tr>
<td>PROJECTS</td>
<td>11</td>
<td>13</td>
<td>13</td>
<td>37</td>
</tr>
<tr>
<td>PROPERTIES</td>
<td>14</td>
<td>11</td>
<td>13</td>
<td>38</td>
</tr>
<tr>
<td>Total</td>
<td>37</td>
<td>37</td>
<td>38</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Participants per spreadsheet/model.

4 ANALYZING END-USER PERFORMANCE

We divide the presentation of our empiric results under two main axes: effectiveness and efficiency. In studying effectiveness we want to compare the three running models for the percentage of correct tasks that participants produced in each one. In studying efficiency we wish to compare the time that participants took to execute their assigned tasks in each of the different models. We start by effectiveness.

4.1 Effectiveness

Each participant was handed 3 different lists of tasks (insert, edit and query data) to perform on 3 different spreadsheets (DISHES, PROJECTS, PROJECTS). Each spreadsheet, for the same participant, was constructed under a different model (original, refactored, visual).

For each spreadsheet, and for each model, we started by analyzing the average of the scores obtained by participants. We shown in Figure 5 the results of such analysis.

<table>
<thead>
<tr>
<th></th>
<th>original</th>
<th>refactored</th>
<th>visual</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISHES</td>
<td>86%</td>
<td>76%</td>
<td>78%</td>
</tr>
<tr>
<td>PROJECTS</td>
<td>73%</td>
<td>68%</td>
<td>78%</td>
</tr>
<tr>
<td>PROPERTIES</td>
<td>75%</td>
<td>64%</td>
<td>62%</td>
</tr>
</tbody>
</table>

Figure 5: Global effectiveness results.

We notice that no spreadsheet model is the best for all spreadsheets in terms of effectiveness. Indeed, we may even notice that spreadsheets in the traditional style, the original model, turned out to be the best for both the DISHES and PROJECTS spreadsheets. The visual model suited the best for the PROJECTS spreadsheet.

In the same line of reasoning, there is no worst model: refactored achieved the worst results for DISHES and PROJECTS; visual got the lowest average scores for PROJECTS.

Nevertheless, these results seem to indicate that the models that we have developed are not effective in reducing the number of errors in spreadsheets, since one of them is always the model getting the lowest scores. This first intuition, however, deserves further development. For once, on the theoretical side, one may argue that original is, without a doubt, the model that end users are accustomed to. Recall that in the study, we opted to leave out participants with computer science backgrounds, who could be more sensible to the more complex models refactored and visual, preferring to investigate such
models on traditional users of spreadsheets. On the other hand, we remark that these more complex models were not introduced; a part of our study was also to learn whether or not they could live on their own.

Our next step was to investigate whether the (apparent) poor results obtained by complex models are due to their own nature or if they result from participants not having understood them. So, we studied participations that did not achieve at least 50%, which are distributed by the spreadsheet models as follows: original, 0%, refactored, 25% and visual, 21%. While in original no participation was graded under 50%, this was not the case for refactored and visual, which may have degraded their overall average results. For these participations, we analyzed the questionnaire that participants were asked to fill in after the session. The average classifications for the post session questionnaires, for participations that were graded under 50% is 24% for refactored and 31% for visual. These results show that participants obtaining poor gradings on their effectiveness, also got poor gradings for their answers to the questions assessing how they understood the models they had worked with. In fact, such participants were not able to answer correctly to (at least) two thirds of the questions raised in the post session questionnaire. From such results we can read that 1/4 of participants was not able to understand the more complex models, which might have caused a degradation of the global effectiveness results for these models. This also suggests that if these models are to be used within an organization, it is necessary to take some time to introduce them to end users in order to achieve maximum effectiveness. Nevertheless, even without this introduction, the results show that the models are competitive in terms of effectiveness: at most they are 13% worst than the original model, and for one of the spreadsheets, the visual model even got the best global effectiveness.

4.1.1 Effectiveness by Task Type

Next, we wanted to realize how effective models are to perform each of the different types of tasks that we have proposed to participants: data insertion, edition and statistics.

i) Data insertion: The results presented in Figure 6 show, for each model, how effective participants were in adding new information to the spreadsheets they received.

<table>
<thead>
<tr>
<th></th>
<th>original</th>
<th>refactored</th>
<th>visual</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISHES</td>
<td>91%</td>
<td>90%</td>
<td>81%</td>
</tr>
<tr>
<td>PROJECTS</td>
<td>76%</td>
<td>60%</td>
<td>75%</td>
</tr>
<tr>
<td>PROPERTIES</td>
<td>86%</td>
<td>67%</td>
<td>68%</td>
</tr>
</tbody>
</table>

![Figure 6: Effectiveness results for data insertion.](image)

The original model revealed to be the most effective, for all three spreadsheets, being closely followed by refactored and visual for DISHES, and by visual for PROJECTS. The refactored model, for PROJECTS, and the models refactored and visual, for PROJECTS, proved not to be competitive for data insertion, in the context of the study.
Again, we believe that this in part due to these models not having been introduced previously to the study: the insertion of new data is the task that is most likely to benefit from totally understanding of the running model, and also the one that can be otherwise most affected. This is confirmed by the effectiveness results observed for other task types, that we present next.

**ii) Data edition:** Now, we analyze the effectiveness of the models for editing spreadsheet data. The results presented in Figure 7 show that once a spreadsheet is populated, we can effectively use the models to edit its data.

<table>
<thead>
<tr>
<th></th>
<th>original</th>
<th>refactored</th>
<th>visual</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISHES</td>
<td>91%</td>
<td>82%</td>
<td>82%</td>
</tr>
<tr>
<td>PROJECTS</td>
<td>54%</td>
<td>62%</td>
<td>50%</td>
</tr>
<tr>
<td>PROPERTIES</td>
<td>65%</td>
<td>98%</td>
<td>48%</td>
</tr>
</tbody>
</table>

Figure 7: Effectiveness results for data edition.

This is the case of refactored for PROJECTS and specially for PROJECTS. original is the most effective in data editing for DISHES. visual is comparable to refactored for DISHES, but for all other spreadsheets, it always achieves the lowest scores among the three models.

**iii) Statistics:** Finally, we have measured the effectiveness of the models for performing calculations over spreadsheet data, obtaining the results shown in Figure 8.

<table>
<thead>
<tr>
<th></th>
<th>original</th>
<th>refactored</th>
<th>visual</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISHES</td>
<td>52%</td>
<td>37%</td>
<td>57%</td>
</tr>
<tr>
<td>PROJECTS</td>
<td>19%</td>
<td>76%</td>
<td>13%</td>
</tr>
<tr>
<td>PROPERTIES</td>
<td>44%</td>
<td>57%</td>
<td>51%</td>
</tr>
</tbody>
</table>

Figure 8: Effectiveness results for statistical calculations.

We can see that visual obtained the best results for DISHES, and that refactored obtained the best results for both spreadsheets PROJECTS and PROJECTS. We can also see that all models obtained the worst results for exactly one spreadsheet.

Results from i), ii) and iii) confirm that the models are competitive. On the other hand, these results allow us to draw some new conclusions: if the models are going to be used within an organization, it may not always be necessary to introduce them prior to their use. Indeed, if an organization mostly edits spreadsheet data or computes new values
from such data, and does not insert new data, then the models, and specially refactored, may deliver good results even without being explained. These results also show that it is inserting data that models need to be better understood by end users in order to increase effectiveness.

4.2 Efficiency

In this section, we analyze the efficiency results obtained in our study by the models that we have been considering in this paper.

We started by measuring, for each participant, and for each spreadsheet, the time elapsed from the moment participants started reading the list of tasks to undertake until the moment they completed the tasks proposed for that particular spreadsheet and moved on to a different spreadsheet or concluded the study. We are able of calculating these times by looking at the individual screen activity that was recorded during the study, for each participant: the participant stopping interacting with the computer signals the end of his/her work on a spreadsheet. The measured period therefore includes the time that participants took trying to understand the models they received each spreadsheet in.

Figure 9 presents the average of the overall times, for each spreadsheet and for each model. We can see that refactored and visual are competitive in terms of efficiency: participants performed fastest for DISHES in visual, and fastest, by a marginal factor, for the PROPERTIES in refactored. original got the best efficiency measurements for PROJECTS, also by a marginal factor. Again, note that no introduction to these models preceded the study.

<table>
<thead>
<tr>
<th></th>
<th>original</th>
<th>refactored</th>
<th>visual</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISHES</td>
<td>35’</td>
<td>32’</td>
<td>28’</td>
</tr>
<tr>
<td>PROJECTS</td>
<td>39’</td>
<td>40’</td>
<td>41’</td>
</tr>
<tr>
<td>PROPERTIES</td>
<td>37’</td>
<td>36’</td>
<td>40’</td>
</tr>
</tbody>
</table>

Therefore, it is reasonable to assume that the results in Figure 9 include some time overhead. In an attempt to measure this overhead, which is a consequence of participants having to analyze a new model, we extracted some information out of the participants’ screen activity. Indeed, we measured the time elapsed from the moment participants started reading, for each spreadsheet, the list of tasks to perform, until the moment they actually began editing the spreadsheet. We assume that this period corresponds exactly to the overhead of understanding each model (obviously increased by the time spent reading the list of tasks, which we are not able of isolating further, but that should be constant for any spreadsheet model, since the task list does not change with the model). These results are presented in Table 2.
We notice that there is a constant average overhead of 2 minutes for almost all models and spreadsheets, with the most significant exceptions occurring for refactored, for both DISHES and PROJECTS. In these cases, we can clearly notice an important time gap, which provides some evidence that refactored is most likely the hardest model to understand. This also comes in line with previous indications that the merits of models can be maximized if we explain them to end users. For the particular case of efficiency, this means that the results shown in Figure 9 could be further improved for the more complex models, and particularly for refactored.

### 5 Threats to Validity

As suggested by Perry et al. [Perry et al., 2000], we discuss three types of influences that might limit the validity of our study.

**Construct Validity:** Do the variables and hypotheses of our study accurately model the research questions?

i) Measuring the time overhead: when studying efficiency, we measured the overhead of understanding each model as the period of time that participants stopped interacting with a spreadsheet and started editing the next one. In this period, it might have been the case that participants, instead of being focused on understanding the new model, took the time to do something else, like resting. This could affect our conclusions in terms of efficiency. However, during the study, participants where supervised by two authors, who observed that this was not the case. Even if we were not able to spot a small number of such occurrences, the differences in the results should be minimal and so they should not affect our conclusions.

ii) Original model: In our study, we have used three spreadsheets that we have assumed to be in the original model. What we are saying is that these three spreadsheets are representative of the spreadsheets normally defined by end users. Although this set of spreadsheets may be too large to be represented by (any) three spreadsheets, we have taken DISHES, PROJECTS and PROJECTS directly, or with small changes, from other works on general purpose spreadsheets [Alhajj, 2003, Connolly and Begg, 2001, Powell and Baker, 2003].

**Internal Validity:** Can changes in the dependent variables be safely attributed to changes in the independent variable?

i) Accuracy of the analysis: Some of the inferences we make in this paper deserve further analysis. To some extent, we assume that our models could achieve better results if a tutorial has been given to the participants. In fact, we have no proof of this, but the evidences from the study seem to strongly indicate this fact. A new study is required to prove this, though.

<table>
<thead>
<tr>
<th></th>
<th>original</th>
<th>refactored</th>
<th>visual</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISHES</td>
<td>35'</td>
<td>32'</td>
<td>28'</td>
</tr>
<tr>
<td>PROJECTS</td>
<td>39'</td>
<td>40'</td>
<td>41'</td>
</tr>
<tr>
<td>PROPERTIES</td>
<td>37'</td>
<td>36'</td>
<td>40'</td>
</tr>
</tbody>
</table>

Table 2: Average overhead results.

We notice that there is a constant average overhead of 2 minutes for almost all models and spreadsheets, with the most significant exceptions occurring for refactored, for both DISHES and PROJECTS. In these cases, we can clearly notice an important time gap, which provides some evidence that refactored is most likely the hardest model to understand. This also comes in line with previous indications that the merits of models can be maximized if we explain them to end users. For the particular case of efficiency, this means that the results shown in Figure 9 could be further improved for the more complex models, and particularly for refactored.

**Construct Validity:** Do the variables and hypotheses of our study accurately model the research questions?

i) Measuring the time overhead: when studying efficiency, we measured the overhead of understanding each model as the period of time that participants stopped interacting with a spreadsheet and started editing the next one. In this period, it might have been the case that participants, instead of being focused on understanding the new model, took the time to do something else, like resting. This could affect our conclusions in terms of efficiency. However, during the study, participants where supervised by two authors, who observed that this was not the case. Even if we were not able to spot a small number of such occurrences, the differences in the results should be minimal and so they should not affect our conclusions.

ii) Original model: In our study, we have used three spreadsheets that we have assumed to be in the original model. What we are saying is that these three spreadsheets are representative of the spreadsheets normally defined by end users. Although this set of spreadsheets may be too large to be represented by (any) three spreadsheets, we have taken DISHES, PROJECTS and PROJECTS directly, or with small changes, from other works on general purpose spreadsheets [Alhajj, 2003, Connolly and Begg, 2001, Powell and Baker, 2003].

**Internal Validity:** Can changes in the dependent variables be safely attributed to changes in the independent variable?

i) Accuracy of the analysis: Some of the inferences we make in this paper deserve further analysis. To some extent, we assume that our models could achieve better results if a tutorial has been given to the participants. In fact, we have no proof of this, but the evidences from the study seem to strongly indicate this fact. A new study is required to prove this, though.
ii) Accuracy of measurements: Each task proposed to participants was individually graded. For most of the cases, this was done automatically using OpenOffice scripts. These scripts and their results were tested and checked. The cases for which an automatic grading was not possible were carefully graded by hand. All grades were validated by two authors and were randomly re-checked. Since we have more than 1400 grades, it is virtually impossible to guarantee full accuracy. This could affect the results observed for dependent variables (efficiency and effectiveness) without really the independent variables (the models considered) having changed. Nevertheless, if imprecisions exist in the grades, they should be equally distributed by the 3 models and thus they should not affect the overall results.

The measurement of times that lead to the results presented earlier was achieved by visualizing the screen casts made during the study for. Being a manual and repetitive task, it is subject to imprecisions. Also, not being able to visualize the actual participants’ behavior now may lead to imprecise measurements. We are confident that, even if there are imprecisions, such imprecisions should be distributed evenly by all measurements and thus do not influence the efficiency results or the conclusions that we draw based on them.

External Validity: Can the study results be generalized to settings outside the study?

i) Generalization: In this study we used three different spreadsheets from different domains. We believe that the results can be generalized to other spreadsheets, although probably not to all. The models we developed are not restricted to any particular spreadsheet, and thus, the results should be the same if the study was run with a different set of spreadsheets.

ii) Industrial usage: Participants were asked to simulate industrial activity: they received some data on paper that they had to register in a spreadsheet. Although we have tried to create a realistic environment for the study, it is likely that people would respond differently in an industrial context. Also, participants were University students whose technical abilities and experience surely differ from other spreadsheet users. Nevertheless, we believe that this affects no spreadsheet/model in particular. Possible impacts would affect all spreadsheets/models in the same way and thus the overall results apply. We believe that if the study was conducted on an industrial environment, the conclusions should be similar.

6 CONCLUSIONS

In this paper, we have presented the results of an empirical study that we conducted in order to assess the practical interest of models for spreadsheets.

According to [Perry et al., 2000], three topics deserve further analysis. The first is accuracy of interpretation: this study was prepared carefully and a significantly large number of end users participated in it. Our goal here was to guarantee that the results are not unknowingly influenced. For this, it also contributes the fact that we make all the elements of this study available, both in this paper and online. The second topic is relevance: MDE is one of the most significant research areas in software engineering. We adapted some techniques from this field to spreadsheets and showed that they can bring benefit not only for professional users but also for end users. The last topic is impact: our first results show that MDE can bring benefits for spreadsheet end users. This is a
promising research direction, that we believe can be further explored, particularly in contexts similar to the one of this paper. From the preparation of the study, from running it and from its results, we can summarize our main contributions as follows: we have shown that MDE techniques can be adapted for end-users software; moreover, we provided empirical evidence that models can bring benefits to spreadsheet end users; finally, we have proposed a methodology that can be reused in studies similar to the one we have conducted.

Finally, we seek to answer the research questions that we presented in the introduction of this paper, which correspond exactly to the questions our study was designed to answer.

**RQ1:** Our observations indicate that there is potential for improving end-user effectiveness using model-based spreadsheets. Even if this is not always the case, our results also indicate that deeper insight on the spreadsheet models is required to maximize effectiveness. Indeed, we believe that the effectiveness results for refactored and visual could have been significantly better if these models had been preliminary presented to the participants of our study.

**RQ2:** We observed that, frequently, the more elaborate spreadsheet models allowed users to perform faster. Nevertheless, we were not fully able of isolating the time that participants took trying to understand the models they were given. So, we believe that the observed efficiency results could also be better for refactored and visual if they had been introduced.

**RQ3:** Although this was not observed for inserting tasks, the fact is that, for editing and querying data the models did help end users. Furthermore, the results seem to indicate that the inserting data task is the one that benefits the most from better understanding the models.

With this study we have shown that there is potential in MDE techniques for helping spreadsheet end users. The study of these techniques for professional users of spreadsheets seems a promising research topic. Moreover, the use of MDE techniques in other non-professional softwares should also be investigated.

**REFERENCES**


Spreadsheets on the Move: An Evaluation of Mobile Spreadsheets

Derek Flood\(^1\), Rachel Harrison\(^1\), Kevin McDaid\(^2\)

\(^1\)Oxford Brookes University, [derek.flood, rachel.harrison]@brookes.ac.uk
\(^2\)Dundalk Institute of Technology, kevin.mcdaid@dkit.ie

ABSTRACT

The power of mobile devices has increased dramatically in the last few years. These devices are becoming more sophisticated allowing users to accomplish a wide variety of tasks while on the move. The increasingly mobile nature of business has meant that more users will need access to spreadsheets while away from their desktop and laptop computers. Existing mobile applications suffer from a number of usability issues that make using spreadsheets in this way more difficult. This work represents the first evaluation of mobile spreadsheet applications. Through a pilot survey the needs and experiences of experienced spreadsheet users was examined. The range of spreadsheet apps available for the iOS platform was also evaluated in light of these users’ needs.

1 INTRODUCTION

Advances in technology have enabled mobile devices to allow users to accomplish a wide variety of tasks while away from traditional computing equipment (e.g. desktop or laptop computers). Traditional desktop applications are also now being ported to mobile devices, allowing users to do their computing while on the move.

One such application is the spreadsheet. With modern smart phones such as the iPhone from Apple, and the Curve from BlackBerry, users can now access their spreadsheets from anywhere. However, the small nature of these devices, required for portability has introduced a number of limitations that have caused severe usability problems. Section 2 outlines some of these issues and outlines how these issues impact upon the usability of mobile spreadsheet applications.

This paper is the first work to examine the needs of experienced spreadsheet users for accessing spreadsheets while away from traditional computing devices. By better understanding the needs of these users, more effective mobile spreadsheet applications can be developed that not only meet the needs of these users but also provide a more pleasing user experience.

A pilot survey has been conducted with experienced spreadsheet users. This survey was designed to assess not just the needs of these users but also their previous experience with mobile spreadsheets. In light of the needs of these users, existing spreadsheet applications available on the Apple iOS platform were evaluated. The research methodology employed during this study can be found in Section 3.

The results of this study have shown that there is a need for access to spreadsheets on mobile devices. Existing applications however, suffer from a number of issues that make them difficult to use and therefore provide a poor overall user experience. The full results of this study can be found in Section 4, while Section 5 outlines the lessons that have been learned during the study.

Section 6 outlines some threats to the validity of this work. The study presented here was conducted with a small number of participants which will limit the generalisability of these results. In addition to this only experienced spreadsheet users were targeted for the study. In the future a more extensive study will be conducted with all types of spreadsheet users.
In addition to this further studies will be conducted examining the usability of mobile spreadsheet applications. This study will allow for a deeper understanding of the usability issues associated with these applications. This study is outlined in Section 7 while Section 8 concludes this paper.

2 RELATED WORK

The rapid progression of technology has led to an increase in the number of mobile applications available. Although these applications offer a number of advantages in terms of portability and convenience they do so at the cost of usability. Zhang and Adipat (2005) have highlighted a number of issues that affect the usability of mobile applications:

- **Mobile Context**: When considering mobile applications the user is not tied to a single location. This will also include interaction with nearby people, objects and environmental elements which may distract a user’s attention.

- **Connectivity**: With mobile devices connectivity is often slow and unreliable and therefore will impact the performance of mobile applications which utilise these features.

- **Small Screen Size & Different Display Resolution**: In order to provide portability mobile devices contain very limited screen size meaning that the amount of information that can be displayed is drastically reduced.

- **Limited Processing Capability and Power**: In order to provide portability, mobile devices often contain less processing capability and power. This has the effect of limiting the types of applications that are suitable for mobile devices.

- **Data Entry Methods**: The input methods available for mobile devices are difficult and require a certain level of proficiency. This problem increases the likelihood of erroneous input and decreases the rate of data entry.

The above limitations of mobile devices further aggravate existing usability issues in the spreadsheet application. The limited screen size on mobile devices requires the user to perform considerably more navigation when looking at large spreadsheets. This may cause users to find it difficult to conceptualise the overall spreadsheet and to see how the section on-screen fits with this overall picture.

Flood et al. (2008), have identified navigation as an issue that affects the performance of people debugging spreadsheets through voice recognition technology. By addressing this issue it was found that the performance of users debugging spreadsheets could be increased. It was also found that participants audited more cells with the improved navigation system, which is an important aspect of the debugging process.

Mobile devices generally do not contain a traditional keyboard as the size required would be too large to enable portability. Some devices incorporate a physical keyboard which utilises small keys while other devices use touch screen technology to present a keyboard to the user on screen. These keyboards also require the physical keys to be smaller than traditional keyboards to fit all keys on screen. The iOS platform addresses this issue by providing users with three separate keyboards; one containing letters, one containing numbers and some special characters and a third containing additional special characters.

Chen et al. (2010) conducted an evaluation of users entering text on a small size QWERTY keyboard. This evaluation required 15 participants to enter a passage of text using the small sized keyboard. On average participants used 540 keystrokes to enter the passage of text. The most prevalent type of error made by these participants during the task was a key ambiguity error, which occurred when a user entered a character other than the target character. It was found that on average, participants made
about 9 key errors on the first typing task. It is also worth noting that all participants made at least one error of this type during the study.

Errors of this type, when made on a spreadsheet, may result in a misspelled word or in an incorrect reference in a cell formula, which could alter the bottom line value of a spreadsheet substantially. It has been shown repeatedly that even on desktop computers errors like this persist. Two independent studies (Panko 1998; Powell, Baker et al. 2009) have found that over 85% of the evaluated spreadsheets contained errors.

The limited processing power of portable devices has meant that existing spreadsheet applications may not function correctly when run on these devices. In an attempt to address this issue a number of developers have created spreadsheet apps which scale down the level of functionality to enable users to view and use spreadsheets in a mobile context. Most of these applications however, are limited in terms of functions available and spreadsheet size.

3 RESEARCH METHODOLOGY

During February 2011 a pilot survey was conducted among experienced spreadsheet users to investigate the need for mobile spreadsheet applications and to identify the issues that exist with mobile spreadsheet applications. Participants, recruited through the European Spreadsheet Risk Interest groups’ mailing list, were asked to complete a short survey featuring 24 questions on their experience of using mobile spreadsheet apps and their need for such applications.

In addition to the survey, an evaluation of existing spreadsheet apps for the iOS platform was also conducted. This study examined the usability of existing spreadsheet applications as well examining the range of features they contain. The analysis of the suitability of these applications for the user’s needs is presented here. A full report can be found in (Flood, Harrison et al. 2011).

3.1 Survey on mobile spreadsheets

The aim of this survey was to examine the extent to which mobile spreadsheet applications are required and used. The usability issues associated with existing spreadsheet applications are also investigated. To meet these aims three primary research questions were established:

- **RQ1**: To what extent is access to spreadsheets required while away from traditional computing devices?

- **RQ2**: To what extent is access to spreadsheets used while away from traditional computing devices?

- **RQ3**: What issues affect the usability of mobile spreadsheet applications?

To meet the first research question, a number of aspects were considered. In addition to asking participants if they required access to spreadsheets while away from a traditional computing devices, the survey examined the purposes for which participants required access to the spreadsheet. A deeper understanding of these purposes would allow future mobile spreadsheet applications to optimize the interface for the most common purpose, therefore improving the overall usability of the application.

A number of usability issues can be traced back to the extensive range of features found in modern applications. By identifying the most common features future spreadsheet applications could be designed so as to prioritise access to these features and make them easier to access. Although this will make other features difficult to access, it should produce a more usable application overall.

The biggest advantage of mobile applications is that they can be used anywhere, in any context. In many cases the context will impact upon the users’ level of attention and it is therefore important to consider the contexts in which these applications are needed. Previous research (Schildbach and
Rukzio 2010) has demonstrated that by altering the target size of a button depending on the context of the user, the performance of the user can be increased.

While RQ1 examines the needs of experienced spreadsheet users, RQ2 examines how users have actually used mobile spreadsheet apps in the past. RQ 3 was designed to examine specific attributes of the usability of current mobile spreadsheet applications. Using participants’ prior experience with mobile spreadsheet applications, the survey examined the level of satisfaction participants have had with existing mobile spreadsheet applications. Participants were also asked about what aspects of the mobile spreadsheet application they enjoyed. Including this information gave the participants an opportunity to highlight some of the positive attributes of these applications that could be utilised in developing future applications.

3.2 Spreadsheet App Evaluation

To better understand existing mobile spreadsheet applications, a systematic evaluation of these apps was conducted. This evaluation focused on apps available for the iOS platform available on mobile devices by Apple. It was decided to focus on this platform initially as Apple is one of the leading providers of mobile devices. It is planned to extend this evaluation to other platforms such as the Blackberry OS and Google Android. The results of this initial evaluation are summarised here in the context of participants’ needs of a mobile spreadsheet application.

The evaluation protocol used during the evaluation is summarised below.

1. **Identify all potentially relevant applications.** There are a number of ways to conduct a search for appropriate applications, including a standard web search, and current software distribution methods make this increasingly easy. Most of the major mobile phone platforms now have an associated online application store. As this work is focused on spreadsheet apps for the iOS operating system, a search of the Apple App store was conducted. The search string “Spreadsheet” was used during this search.

2. **Remove light or old versions of each application.** Many software developers release trial versions of their systems, which are often free. Some of these versions include only a subset of the functionality offered by the full application whilst others allow full access to the application but for a limited time period. These types of applications should be removed if the full version of the app is also included within the search results.

3. **Identify the primary operating functions and exclude all applications that do not offer the required functionality.** The primary operating functions include frequently used functions and also occasionally used functions that are essential for the correct operation of the system in a desired context. For example, the initial system setup might include language and currency settings that would depend upon the country of use. The primary functionality of interest is to allow a user to perform spreadsheet tasks on a mobile device.

4. **Identify all secondary functionality within the remaining apps.** In addition to the primary operating functions, mobile apps will offer users a range of secondary functionalities which can enhance the application. A thorough knowledge of these functions will enable the application developers to see what functionality is available and may present opportunities for additional functionality to be included in future applications.

5. **Install each of the remaining applications, and test each of the tasks using Keystroke level modelling.** Keystroke Level Modelling (KLM) is a well established technique (Card, Thomas et al. 1983) for estimating the time taken to complete certain tasks. This will provide a quantitative measure of the difference between the efficiency of applications. KLM was used to measure the average number of interactions required to enter each of a set of 5 single digit numbers as well as the number of interactions to enter a subsequent formula to total these five numbers.
4 RESULTS

4.1 Survey Participants

In total fourteen participants took part in the online survey. The participants were recruited through the European Spreadsheet Risk interest groups’ mailing list which is comprised of approximately 750 email addresses, featuring members from both academia and industry, who share in interest in the use of spreadsheets and the risks associated therein.

All participants use spreadsheets on a daily basis. When asked which of the following options best describes their level of expertise with spreadsheets; Novice, Intermediate or Expert. 71% of participants rated themselves as experts while 21% rated themselves as Intermediate. The remaining 7% rated themselves as novice spreadsheet users.

To better understand the nature of the participants’ use of spreadsheet participants were asked “In the last week, approximately how many times did you use a spreadsheet”. The responses to this question varied widely, from 6 to over 100 times. As the amount of time spent using a spreadsheet can vary per usage, participants were also asked to estimate how long they spend on average per use of a spreadsheet. Participants were asked to select one of the following options or to enter their own value: Less than 30 minutes, 30 – 60 minutes, 1 – 2 hours, 2 – 3 hours, 3 – 4 hours, over 4 hours. It can be seen from Figure 1 that most participants spend between 30 – 60 minutes per spreadsheet use.

![Duration of use](image)

**Figure 1: Average Duration of spreadsheet use**

4.2 Survey Results

The need for mobile spreadsheets

*RQ1: To what extent is access to spreadsheets required while away from traditional computing devices?*

Approximately 79% of participants said that they required access to a spreadsheet while away from a desktop or laptop computer. These results indicate that there is a strong need for mobile spreadsheet applications.

Purpose of spreadsheet app usage

It is important to consider the reasons why mobile spreadsheets are to be used. To identify this, participants were asked “for what purpose did you need it [access to a mobile spreadsheet while away from traditional computing devices]”. Participants were asked to select all that applied from the following options: To view spreadsheet data, To change a spreadsheet and To create a new spreadsheet.
Table 1 shows the percentage of participants who required a mobile spreadsheet application for each purpose. It can be seen that the majority of participants required spreadsheets for viewing and/or changing an existing spreadsheet. Only 21.4% of respondents said that they needed to create a spreadsheet on a mobile device.

<table>
<thead>
<tr>
<th>Purpose</th>
<th>% of respondents</th>
</tr>
</thead>
<tbody>
<tr>
<td>To view spreadsheet data</td>
<td>64.3</td>
</tr>
<tr>
<td>To change a spreadsheet</td>
<td>64.3</td>
</tr>
<tr>
<td>To create a new spreadsheet</td>
<td>21.4</td>
</tr>
</tbody>
</table>

Table 1: Percentage of participants by spreadsheet purpose

A critical component of spreadsheet applications is the ability to transfer data to and from the device easily. Existing spreadsheet apps for the iOS platform allow users to transfer data in a number of ways. The easiest way for users to transfer spreadsheets to the mobile device is through email. When participants receive an email including a spreadsheet, they can open the spreadsheet directly from within the email application. In addition to this some spreadsheet apps allow users to email spreadsheets directly from within the app.

Other methods of data transfer require users to be connected to the device containing the spreadsheet. The most common means is through Wi-Fi, where users can access the mobile device through a web based interface which allows users to transfer files between devices. This approach requires users to type in a specific IP address into the browser after ensuring both devices are connected to the same wireless network.

One final approach offered by a small number of apps (25%) is the use of online storage, such as Google spreadsheets. These applications allow users to log into their Google account and download their spreadsheets to their mobile device. This approach supposes that users have such accounts and store their spreadsheets in these locations. One of the applications evaluated does not allow users to transfer data to and from the device and is solely designed for use on the device.

A number of applications provide access to on-line spreadsheet solutions such as Google spreadsheets. As the access to the internet is not always available on mobile devices these applications were not considered. All of the applications evaluated allow users to operate the spreadsheet on the device alone once it has been downloaded.

Context of spreadsheet usage

The biggest advantage of mobile devices is the range of contexts in which they can be used. Whether out walking or travelling to work, users can access a wide variety of services that allow them to accomplish a range of tasks. In order to determine in what context mobile spreadsheets would be used participants were asked “In what context did you need access to the spreadsheet where no desktop or laptop computer was available”. Participants provided a wide variety of contexts in which they needed access to spreadsheets where traditional computers would be impractical. Some of the contexts cited are presented below:

“Discussing changes in assumptions”

“Could not get to laptop - but had iphone with me.”

“Inbound email onto a mobile device”

“Daily commute”

“To demonstrate data to clients”

The broad range of contexts outlined above is from a small population sample. It is believed that a more extensive survey will produce an even broader range of contexts in which spreadsheets are
required. The context in which the application is being used is an important factor as this will have an impact upon the appropriate design of the system.

**Required features on a mobile spreadsheet app**

As mobile devices are limited in terms of processing power, a full spreadsheet application would be slow and resource intensive. Most mobile spreadsheet applications therefore only supply a subset of the available spreadsheet functionality. In order to determine which functionality is important to them, participants were asked which features they required.

<table>
<thead>
<tr>
<th>Feature</th>
<th>% of participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formula</td>
<td>64.3</td>
</tr>
<tr>
<td>Data Features</td>
<td>57.1</td>
</tr>
<tr>
<td>Graphs/Charts</td>
<td>28.6</td>
</tr>
</tbody>
</table>

**Table 2: Features Required**

Table 2 shows the percentage of participants who needed each feature. It can be seen the most needed feature is formula. Functions can be used to simplify the creation of such formulae. A broad range of functions are available within the mobile spreadsheet apps evaluated. The number of functions varies considerably by app, with some offering a single function, while others offer up to 146 functions. The range of financial functions available is quite limited, with most applications only offering less than 5 functions. MarnerCalc is an exception to this offering users access to 18 financial functions.

Only 25% of the apps evaluated allow users to use graphs and charts. Other features such as the sort function are more common among the apps, appearing in 50% of those evaluated. These results would indicate that existing applications are focused on real users’ needs. It should be noted however that none of the apps evaluated featured the ability to filter data.

In addition to the features included in Table 2, some participants also stated that they required Visual Basic for Applications (VBA) macros, which are often used to enhance the functionality of spreadsheets. During this evaluation of mobile spreadsheet apps however, it was found that none of the applications offered this functionality.

**Frequency of mobile spreadsheet apps**

<table>
<thead>
<tr>
<th>Frequency</th>
<th>% of participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daily</td>
<td>0.0%</td>
</tr>
<tr>
<td>Weekly</td>
<td>21.4%</td>
</tr>
<tr>
<td>Monthly</td>
<td>14.3%</td>
</tr>
<tr>
<td>Less than once a month</td>
<td>50.0%</td>
</tr>
<tr>
<td>Never</td>
<td>14.3%</td>
</tr>
</tbody>
</table>

**Table 3: Frequency by which access to mobile spreadsheets are needed**

Table 3 shows the frequency with which participants have needed access to spreadsheets while away from a desktop or laptop computer. It can be seen that most participants needed access to the spreadsheet less than once a month. 21.43% of the participants did say that they needed this access on a weekly basis. These findings would indicate that memorability is an important aspect of the usability of mobile spreadsheet apps, as most participants use them on an infrequent basis.

**The use of mobile spreadsheets**

*RQ2: To what extent is access to spreadsheets used while away from traditional computing devices?*
In addition to examining the participants’ need to access spreadsheets while away from desktop or laptop computers, the survey also examined participants’ previous experience using spreadsheets on a mobile device. Of the 14 participants included within the study only 42.9% said they have previously accessed spreadsheets on a mobile device. The following results are based on only these participants.

**Purpose of spreadsheet app usage**

<table>
<thead>
<tr>
<th>Purpose</th>
<th>% of participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>To view spreadsheet data</td>
<td>83.3%</td>
</tr>
<tr>
<td>To change a spreadsheet</td>
<td>50.0%</td>
</tr>
<tr>
<td>To create a new spreadsheet</td>
<td>0.0%</td>
</tr>
</tbody>
</table>

**Table 4: Purpose of mobile spreadsheet usage**

Table 4 shows the percentage of participants who have used a mobile device for each purpose. These results indicate that the most common use of spreadsheets on a mobile device is for viewing existing spreadsheet data. Only half of the participants have used spreadsheets for changing a spreadsheet. None of the participants have created a spreadsheet on a mobile device. These results highlight the need for mobile spreadsheet apps to allow users to transfer and access existing spreadsheets easily.

**Context of spreadsheet usage**

As was seen when examining the need of mobile spreadsheet applications, participants have used mobile spreadsheet applications in a broad range of contexts for a wide variety of tasks. One participant remarked that they have used a mobile device to change assumptions to perform sensitivity analysis while another participant wanted to read a spreadsheet that had been received in an email. Participants have also used mobile spreadsheet applications on their daily commute, a task that may be impractical with conventional laptop computers, due to their cumbersome nature in certain environments (public transport for example).

**Used features on a mobile spreadsheet app**

<table>
<thead>
<tr>
<th>Feature</th>
<th>% of participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formula</td>
<td>66.7%</td>
</tr>
<tr>
<td>Data Features</td>
<td>50.0%</td>
</tr>
<tr>
<td>Graphs / Charts</td>
<td>16.7%</td>
</tr>
</tbody>
</table>

**Table 5: Features Used**

The features used by participants on a mobile spreadsheet app are presented in Table 5. It can be seen that the most common feature used by the participants is formulae, a central component of a spreadsheet. Data features such as sort and filter options were used by half of the participants. Only 16% of participants used graphs or charts on a mobile device. These results would indicate that more complex features such as graphs and charts are less important to participants than more basic functionality such as formula. The use of data features is typically associated with only certain spreadsheets, i.e. those which contain tables of data. The relatively high frequency of their use would indicate that participants are using this type of spreadsheet frequently on mobile devices.

**Frequency of use**

When asked how often they use spreadsheets on a mobile device, 33% of participants who have used spreadsheets on a mobile device said that they use them on a weekly basis. The majority of these participants however replied that they would only need spreadsheets on a mobile device less than once a month. This infrequency of usage would indicate that the use of a mobile spreadsheet app should be memorable. There are a number of ways in which this could be achieved, for example by replicating existing interface elements from more traditional spreadsheet applications.
The usability of mobile spreadsheets

**RQ3: What issues affect the usability of mobile spreadsheet applications?**

A number of definitions of usability exist. The ISO define usability as “*Extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use*” (ISO/IEC 1998). However, Nielsen (Nielsen 1993) also identified a number of attributes that this definition does not include but are important in terms of usability; Learnability, Memorability and Errors.

The authors have identified seven attributes of usability of mobile applications; Effectiveness, Efficiency, Satisfaction, Learnability, Memorability, Simplicity and Cognitive load. Due to the nature of the research method it was not possible to evaluate participants’ experience of all of these attributes. However, the following attributes of usability were examined; Effectiveness, Efficiency, Satisfaction, and Simplicity.

**Effectiveness**

To evaluate *effectiveness* participants were asked if they could accomplish their desired task using the mobile spreadsheet application. Participants were asked to select one of three options to answer this question; *Yes*, *No* or *Partially*.

<table>
<thead>
<tr>
<th>Could complete</th>
<th>% of participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>33.33%</td>
</tr>
<tr>
<td>No</td>
<td>16.67%</td>
</tr>
<tr>
<td>Partially</td>
<td>50.00%</td>
</tr>
</tbody>
</table>

Table 7: Effectiveness of mobile spreadsheet apps

Table 7 shows that only 33% of the participants could accomplish their desired task using a mobile spreadsheet application. In addition to this a further 50% could partially complete their desired task. Only 16% of the participants were unable to complete their task. These results indicate that the standard of mobile spreadsheet applications is low for the tasks that participants wish to accomplish. If mobile spreadsheet apps are to be successful their effectiveness will need to be increased dramatically.

**Efficiency**

The efficiency of participants was not evaluated through the online survey. Participants did describe using spreadsheets on a mobile device as being tedious; indicating the efficiency of these applications is quite poor. During the evaluation of existing spreadsheet apps, the efficiency of these applications was measured using a KLM analysis (Card, Thomas et al. 1983) of creating a simple spreadsheet.

This evaluation found that the efficiency of these applications varied by as much as 100%. To input 5 single digit numbers and then total these numbers in the evaluated apps took between 19 and 36 keystrokes depending on the application. To enter a single digit on some applications took up to 4 keystrokes. The reason for this is that these applications require users to select a cell, open the keyboard, select the numerical keypad and then enter the number. When the application then moves to
the next cell the keyboard is automatically closed forcing users to reopen the keyboard when on this

cell. 

It was found that some of the spreadsheet apps evaluated allowed users to use shortcuts to enter the

sum formula. When selected from a shortcut menu item, the system will automatically infer that the

user wants to sum all the numbers within the selected column and will enter the formula to do so.

Other applications however require the user to manually type in the formula, which can be tedious and
time consuming on the touch screen.

During the evaluation it was found that one spreadsheet app had reversed the usual presentation of

numbers and letters to denote the rows and columns. While most traditional spreadsheet applications

number the rows and identify columns through the use of letters, this app used letters to denote the

rows and numbered the columns. By altering the users’ preconceptions, a new potential of error is

introduced especially when users are switching between conventional spreadsheet applications and

mobile spreadsheet apps.

Satisfaction

The satisfaction of users was measured on a five point Likert scale, where 1 indicated very satisfied

and 5 indicated very unsatisfied. Of the six participants who had used mobile spreadsheet apps, the

average rating was 3.33, indicating that participants were slightly dissatisfied with the existing

applications. There are a number of reasons why users are dissatisfied with these applications.

When asked if they experienced any errors while using the mobile spreadsheet app, 50% of the

participants said that they did experience errors. The remaining participants answered no to this

question. However a subsequent question asked participants to list all of the problems that they

experienced. 83% of the participants listed at least one problem for this.

The most common problem quoted by the participants related to the size of the device. Most

participants had problems viewing large spreadsheets on the small screen. One participant also found

it difficult to progress with their task. Another participant described spreadsheets on a mobile device

as being Useful but tedious.

Despite the issues outlined above, some participants had some positive comments to make about the

use of spreadsheets on a mobile device. The most common positive comment related to the ability to

view the spreadsheet on the mobile device. The high frequency of this type of comment would

indicate that participants had low expectations of spreadsheets on these devices and were pleased that

they at least existed and therefore are willing to put up with a lot of errors and limitations without it

having a large negative impact on the satisfaction they had with the device.

Simplicity

Simplicity refers to how well a user can complete their task without errors. When asked if they made

any errors while using a mobile spreadsheet app, 50% of participants confirmed that they did. In

addition to this participants were also asked if the device was sufficient for their needs. Half of the

participants said that the device was sufficient while half disagreed saying it was insufficient. The

main reason participants found it insufficient was that the size of the screen was too small. One

participant also remarked that there was no access to VBA and therefore the device was insufficient to

meet their particular needs.

5 SUMMARY

It has been found that there is a strong need for mobile spreadsheet apps. Approximately 79% of the

participants surveyed said that they have needed access to a spreadsheet while away from a traditional

computing device. The rapid increase in the power of mobile devices and the relatively low

development costs associated with mobile applications has enabled the spreadsheet application to be

ported to mobile devices.
The limitations imposed by the miniature nature of these devices has meant that mobile spreadsheet applications suffer from a number of usability issues which limit the usefulness of these applications and make them tedious for users to use. The biggest issue identified by participants has been the very small screen size of these devices, meaning that users can only view a small portion of a spreadsheet at a time. The severity of this issue could be reduced through the use of a mini map, which shows a scaled down version of the spreadsheet with the current area being displayed highlighted in the bottom right hand corner of the screen. Similar approaches have been used for looking at images, web pages and other large documents (Burigat, Chittaro et al. 2008).

It was also found that despite the many limitations of mobile spreadsheet applications, users were only slightly dissatisfied with the existing mobile spreadsheet applications. This result shows that the convenience of accessing the spreadsheet on a mobile device out-weighs the negative usability aspects experienced.

Despite the limitations outlined above, spreadsheets can be used on mobile devices. It was found that the most efficient app was Spreadsheet by AppAuthors [http://www.appauthors.com]. This application allowed the test spreadsheet to be created with the fewest interactions, and offers users the ability to sort data and to use freeze panes to keep headings displayed when looking at data. This app however does not allow users to view or create charts.

6 THREATS TO VALIDITY

The results presented above are based on a small number of participants which limits the generalisability of these results. It is hoped to conduct a more extensive survey in the future, sampling not just experienced spreadsheet users but also novice spreadsheet users. This broader sample will enable a more general picture of the need for mobile spreadsheet apps to be attained.

The pilot survey outlined above contained a limited subset of the existing spreadsheet features. A more detailed set of features should be used to determine which features are required by users and which features would not be beneficial if included within a mobile spreadsheet app. Similarly an investigation into the functions most needed by participants would enable mobile apps to better facilitate users’ needs.

The subjective nature of surveys will limit the accuracy of the reported frequency of errors made by participants on mobile devices. A more accurate measure would be to monitor the participants while using a mobile spreadsheet app and record the errors that are made by participants. This will allow for a more detailed understanding of the errors that are made while using mobile spreadsheet apps.

The evaluation of existing mobile spreadsheet apps is limited to those available on the iOS platform. Other mobile platforms such as the RIM featured on Blackberry devices, Android from Google and Windows Mobile may also feature additional spreadsheet applications which would need to be evaluated to provide a complete picture of mobile spreadsheet applications. The devices on which these platforms operate will also vary, providing different modes of input; such as traditional keyboard or touch screen or a combination of both.

7 FUTURE WORK

The pilot survey presented above has shown that there is a need for mobile spreadsheet apps. It is intended to run a more extensive survey including not just experienced spreadsheet users but also novice users. This more extensive survey will allow for a more general understanding of the need for mobile spreadsheet apps.

It is also planned to extend the current survey to examine a more extensive range of spreadsheet features and functions and to determine which of these are most important for mobile spreadsheet users. With this understanding it would be possible to design a mobile spreadsheet app that is
optimised for the real needs of users and therefore puts people at the centre of mobile application development.

To better understand the issues that are experienced by mobile spreadsheet users, it is intended to run a controlled experiment which asks participants to use a mobile spreadsheet app to retrieve information from an existing spreadsheet and to create a new spreadsheet on the device. By recording the users’ actions while they are performing these tasks it will be possible to see what procedures users employ while interacting with the device and to see what errors they make during these common spreadsheet tasks.

Using this information a deeper insight into user behaviour can be gained and used to make recommendations as to how mobile spreadsheet apps should be designed to produce a more usable mobile spreadsheet app.

8 CONCLUSIONS

This research has shown that there is a need for mobile spreadsheet apps. A large proportion of participants surveyed, 78%, have said that they have needed access to spreadsheets while away from traditional computing devices. The primary needs of these users are to view or change existing spreadsheets with only 21% of participants saying they would need to create a spreadsheet on a mobile device.

Along with examining the need for mobile spreadsheet applications, this work also examined the extent to which existing applications are used. It was found that approximately half of the participants that required access to spreadsheets while away from traditional computing devices have used mobile spreadsheet applications. Of these participants only 33% could accomplish their desired task, with a further 50% stating that they could partially complete their task. To better understand why this was the case, the usability of these applications was examined.

It was found that existing apps suffer from a number of issues, most predominantly due to the small screen size of most mobile devices. Participants’ most common complaint of these devices was that the screen size was too small. One participant also remarked that a lack of a global view was a problem. Despite these issues, it was found that participants were not dissatisfied with mobile applications. The convenience afforded by the ability to view spreadsheets on a mobile device outweighs the usability issues associated with mobile spreadsheet applications.

9 REFERENCES


In Search of a Taxonomy for Classifying Qualitative Spreadsheet Errors

Zbigniew Przasnyski, Linda Leon, and Kala Chand Seal
Loyola Marymount University, Los Angeles, California (USA)
zprzasny@lmu.edu

ABSTRACT
Most organizations use large and complex spreadsheets that are embedded in their mission-critical processes and are used for decision-making purposes. Identification of the various types of errors that can be present in these spreadsheets is, therefore, an important control that organizations can use to govern their spreadsheets. In this paper, we propose a taxonomy for categorizing qualitative errors in spreadsheet models that offers a framework for evaluating the readiness of a spreadsheet model before it is released for use by others in the organization. The classification was developed based on types of qualitative errors identified in the literature and errors committed by end-users in developing a spreadsheet model for Panko’s (1996) “Wall problem.” Closer inspection of the errors reveals four logical groupings of the errors creating four categories of qualitative errors. The usability and limitations of the proposed taxonomy and areas for future extension are discussed.

1. INTRODUCTION
The prevalence of errors in organizational spreadsheets that can lead to disastrous consequences is well documented, yet few organizations have formal policies for quality control of spreadsheets deployed for end-users (Caulkins et al., 2007; Panko and Ordway, 2005; http://www.eusprig.org/stories.htm). The literature on spreadsheet errors classifies errors into two general categories: quantitative and qualitative (e.g., Beaman et al., 2005; Panko & Aurigemma, 2010; Powell et al., 2008; Teo & Tan, 1999). Quantitative errors are identified as immediate incorrect numerical values or logic in the spreadsheet, while qualitative errors are associated with spreadsheet design flaws that increase the likelihood of an eventual quantitative error occurring during operational use of the spreadsheet. To fully assess the quality of a spreadsheet model and certify it for operational use, it is therefore necessary to identify the presence of both quantitative and qualitative errors in the model. In this paper we propose a taxonomy for classifying qualitative errors. The scarcity of formal taxonomies for classifying qualitative errors in the literature is the motivation for this paper. The paper first provides some background on the various types of spreadsheet errors and existing taxonomies followed by a detailed description of our own proposed taxonomy, and a discussion about its usability and the conclusion.

2. SPREADSHEET ERROR TAXONOMIES AND QUALITATIVE ERRORS
The taxonomy proposed by Panko and Halverson (1996) was the first significant attempt to create a classification of errors in spreadsheets. Since then a number of researchers have contributed to the development of spreadsheet error taxonomies, but most work has focused on quantitative errors (e.g., Panko & Sprague, 1998; Powell et al., 2008; Rajalingham et al., 2000). Very little work has been done so far on developing taxonomies to classify qualitative errors even though qualitative errors can be just as damaging to an organization’s productivity as quantitative errors (Panko and Aurigemma,
Panko and Halverson’s original quantitative error taxonomy was subsequently revised (Panko and Aurigemma, 2010; Panko and Halverson, 2001) to include the effect of the spreadsheet life cycle on errors, thereby distinguishing development and design errors from errors generated by users during the operational use phase of the spreadsheet.

Figure 1: Panko and Aurigemma (2010) spreadsheet error classification revisited

Panko and Aurigemma’s classification is extended in the boxed area of Figure 1 to indicate that qualitative errors can be generated both during the design and development as well as operational use stages of the life cycle, just like quantitative errors. In this paper, we are interested in developing a taxonomy for qualitative errors created during the spreadsheet design and development phases so that they can be detected during the testing phase before the spreadsheet is released for operational use – indicated by the broken line in Figure 1. Qualitative errors, such as entering an inappropriate input value or overwriting a formula with a number (labeled as execution errors in Figure 1), may be generated by end-users during operational use of the spreadsheet, but we argue that avoidance of such errors is dependent upon the use of appropriate spreadsheet controls and user training. Some examples of controls that can be considered to reduce user-generated qualitative errors are cell protection, data validation, and conditional formatting. (Bewig, 2005; Martin, 2005; O’Beirne, 2005).

Design related qualitative errors are difficult to classify because the exact classification of the error is often based on its context, which is dependent upon the view of the reviewers. This same difficulty has been observed by researchers attempting to outline “best practices” for spreadsheet development (e.g. Conway & Ragsdale, 1997; Raffensberger, 2003). Examples of design related qualitative errors considered in past research include semantic errors, jamming or “hard-coding” errors, duplication errors, and poor design layouts that make the model hard to read (Purser & Chadwick, 2006; Raffensberger; 2003; Rajalingham et al., 2000; Teo & Tan, 1999). Clearly, identifying and avoiding these types of errors improves the quality of the spreadsheet. However, this list is far from complete and there is a need for systematic research to identify and classify these and other types of qualitative errors that can occur in spreadsheets. Panko and Aurigemma (2010) note: “It is now time to shift our focus into qualitative errors, which may be far more common than quantitative errors, and identifying the large number of different types of errors that are possible in different life cycle stages and by people with different roles to play” (p.244).

3. PROPOSED QUALITATIVE ERROR TAXONOMY

We describe the development of a qualitative error taxonomy that should help an organization detect if a spreadsheet contains certain types of qualitative errors, which potentially can be damaging if the spreadsheet is allowed to go into operational use. The purpose of the taxonomy is to help managers
decide whether a numerically accurate spreadsheet (i.e., containing no quantitative errors) is robust enough to be in operational use. This would help tremendously in spreadsheet governance as auditors and testers of spreadsheets will have a mechanism for judging the spreadsheet quality for usability.

We took a two-pronged approach in developing the taxonomy. We first consolidated and organized the qualitative errors found in the literature using the umbrella taxonomy for qualitative errors outlined in Rajalingham et al., (2000). Second, we analyzed the qualitative errors found in a lab experiment that consisted of 104 spreadsheet models created for the “Wall Problem” described in Panko (1996). We found that the Rajalingham et al., (2000) qualitative error taxonomy was not sufficiently detailed in its practical use as a rubric for detecting and counting the various types of qualitative errors that were present in these completed spreadsheets. We therefore expanded the taxonomy by introducing specific elements that could measure the various types of qualitative errors at a more atomic level and would provide a set of rubric items for a formal grading process. Figure 2 expands the box in Figure 1 to provide a detailed picture of how our taxonomy fits in the overall classification scheme of qualitative spreadsheet errors.

**Figure 2: Proposed qualitative error taxonomy**

In the proposed taxonomy, as long as one instance of a particular type of qualitative error was present in the spreadsheet we recorded it as 1. Initially we tried to count every instance of a qualitative error but found the process to be inefficient and inconsistent across reviewers. Since qualitative errors are latent errors (Reason, 1990) that often lead to serious errors later, we argue that it is important to ensure that the presence of each type of error is flagged and recorded so that the spreadsheet model can be revisited before it is released for operational use. Depending on the domain, the mere existence of a certain type of error (e.g., hard-coding of the interest rate in a financial spreadsheet that updates based on a rolling interest rate) should trigger a complete reexamination of the spreadsheet for detection and correction of all instances of that error. The resulting list of the qualitative errors from our lab experiment and their subsequent logical groupings is shown in column 2 and column 1 of Table 1 respectively.
Table 1. A taxonomy for defining qualitative errors

<table>
<thead>
<tr>
<th>Category</th>
<th>Type of Error</th>
<th>Record the error as 1 (&quot;exists&quot;) if:</th>
<th>Explanatory Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input Data Structure</strong></td>
<td>Hard-coding/jamming values into formulas</td>
<td>At least one value is hard-coded into a formula somewhere in the model</td>
<td>1-1</td>
</tr>
<tr>
<td></td>
<td>Duplication of Input Values</td>
<td>• There is more than one place to enter the same value in the model</td>
<td>1-2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• An intermediate calculated value must be entered directly as an input in another part of the model</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Input Cells Not Clearly Identified</td>
<td>One or more input cells anywhere in the spreadsheet are not clearly identified</td>
<td>1-3</td>
</tr>
<tr>
<td><strong>Semantics</strong></td>
<td>Missing Cell Documentation</td>
<td>At least one cell where a number is displayed is not labeled</td>
<td>2-1</td>
</tr>
<tr>
<td></td>
<td>Incorrect Cell Documentation</td>
<td>At least one cell has a label that is clearly incorrect (e.g., refers to total profit when it is total cost; or states wrong units)</td>
<td>2-1</td>
</tr>
<tr>
<td></td>
<td>Ambiguous Cell Documentation</td>
<td>At least one cell that has not been identified as clearly incorrect has an ambiguous label associated with it (e.g., no units are specified, general labels such as “costs” are used)</td>
<td>2-1</td>
</tr>
<tr>
<td></td>
<td>Poor Layout for Readability</td>
<td>The spreadsheet is generally difficult to read/follow for any of the following reasons:</td>
<td>2-2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Does not read left to right and top to bottom</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Related formulas are not in physical proximity of each other</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Excessive blank spaces are used to create long arcs of precedence</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The logic of the business problem is difficult to follow</td>
<td></td>
</tr>
<tr>
<td><strong>Extendibility</strong></td>
<td>Poor Layout for Model Extension</td>
<td>• Inserting a row or column will hurt the readability of the model or</td>
<td>3-1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Extending the model will require multiple insertions and manipulation of the model</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Poor Layout for Copy/Paste</td>
<td>Layout of the logic of the model does not allow for easy copying to a new column/row</td>
<td>3-2</td>
</tr>
<tr>
<td></td>
<td>Poor Absolute/Relative Cell References for Copy/Paste</td>
<td>At least one cell included in a layout that allows for the cell logic to be easily copied to another row or column contains a formula with incorrect use or lack of use of the $ notation</td>
<td>3-3</td>
</tr>
<tr>
<td><strong>Formula Integrity</strong></td>
<td>Spurious Formulas</td>
<td>A confusing or spurious entry in a formula exists</td>
<td>4-1</td>
</tr>
<tr>
<td></td>
<td>Lack of Explicit Formulas</td>
<td>An inputted number (jammed or entered in an input cell) is the result of a calculation not carried out in the spreadsheet (even if some of the numbers used to calculate the input value were generated in the spreadsheet)</td>
<td>4-2</td>
</tr>
</tbody>
</table>
Category 1: Input Data Structure

The Input Data Structure error category focuses on inappropriate structures used for incorporating input data values into the model’s formulas and logic. This category includes the two qualitative errors most commonly referred to in existing spreadsheet literature, the hard-coding or “jamming” of a value in a formula and the duplication of an input value. In both cases, the assumed data value may initially be correctly entered thus creating no quantitative error. However, as a user interacts with the model and performs what-if analyses, the user may fail to correctly update all of the places where the input data value assumptions exist, and thereby generate an error in the bottom-line result.

**Example 1-1: Hard-coding/Jamming Values into Formulas**

Assume that three numbers representing the height, width and length of a wall are jammed into the volume formula in cell C3 of Table 2. Here the hard-coding hides the dimension inputs so that the user does not clearly see what assumptions were made for the current bottom-line output. The user may forget that the wall is assumed to be 6 ft. tall, 2 ft. wide and 20 ft. long since these inputs are not clearly documented. Furthermore, when the user wants to re-dimension the wall, the user must correctly identify which number, (6, 2 or 20), in cell C3 represents the dimension to be modified. There is a distinct risk that the wrong number will be modified.

### Table 2: Example of Hard-Coding

<table>
<thead>
<tr>
<th>Values</th>
<th>Formulas</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>2</td>
<td>Type of Wall</td>
</tr>
<tr>
<td>3</td>
<td>Lava Rock</td>
</tr>
</tbody>
</table>

Absolute constants, such as pi or the number 1 when it represents 100%, can be hard-coded without generating these risks. These are not jamming errors as these numbers cannot be changed in subsequent use and the functional purpose of the number in the formula is clear to the user.

**Example 1-2: Duplication of Input Values**

Assume that for the “Wall problem” the company requires a 30% profit margin on all of its bids. The required 30% profit margin has been jammed into both profit margin formulas in cell C14 and D14 of Table 3. This qualitative error is more serious than the simple hard-coding errors described in Example 1-1 as the user must change the profit margin in two places in order to generate the correct bottom-line result. There is a risk that the user will remember to change the profit margin correctly in only one of the two locations, forgetting about the other formula.

### Table 3: Example of Duplication

<table>
<thead>
<tr>
<th>Values</th>
<th>Formulas</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>11</td>
<td>Type of Wall</td>
</tr>
<tr>
<td>12</td>
<td>Lava</td>
</tr>
<tr>
<td>13</td>
<td>Total Costs</td>
</tr>
<tr>
<td>14</td>
<td>Required Profit Margin</td>
</tr>
<tr>
<td>15</td>
<td>Bid</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>Type of Wall</td>
</tr>
<tr>
<td>12</td>
<td>Lava</td>
</tr>
<tr>
<td>13</td>
<td>Total Costs</td>
</tr>
<tr>
<td>14</td>
<td>Required Profit Margin</td>
</tr>
<tr>
<td>15</td>
<td>Bid =SUM(C13:C14) =SUM(D13:D14)</td>
</tr>
</tbody>
</table>
error. However, duplication errors may or may not occur simultaneously with instances of hard-coding. If, say, cells C5 and D5 in the Table 3 example had been set up as data input cells for the profit margin rate and correctly referenced in C14 and D14, then there is no hard-coding error, but the need to change the profit margin in two places (assuming that they are never to be different for each wall) creates a duplication error.

Example 1-3: Input Cells Not Clearly Identified

Input assumptions may be labeled clearly with appropriate cell documentation but still be difficult for users to identify if the model layout does not properly emphasize the location of the input cells. There are several ways to clearly identify input cells for the user. The traditional modular computer programming approach, which is commonly referenced in spreadsheet literature, specifies standard formats with a separate and clearly labeled data input module. Use of a designated color may alternatively be used to identify input cells that are not separated in a distinct module.

It should be noted that this error classification should not include hard-coding or duplication errors. This error type presupposes appropriate input data structures (e.g., no jamming) but identifies layout and formatting design flaws where the input cells may not be easily recognizable or understood by the user.

Category 2: Semantics

The Semantic error category describes errors that create a distortion of or an ambiguity in the meaning of an input or output of the model. As a result, the user can make an inappropriate choice of an input assumption value and/or a misinterpretation of an output result. Common sources of semantic errors are poor documentation of assumptions and outputs, including cell formatting and labeling, as well as the readability of the layout design.

Example 2-1: Cell Documentation Problems

Cells which are not labeled create confusion. Some of these cells may be a key calculation in the model’s logic while others may be dangling cells that serve no purpose. In both cases, unlabeled cells add visual and logical clutter and confuse the user.

Cells which are incorrectly labeled or formatted often result in wrong inputs and/or decisions being made. For example, in Table 4 due to the missing documentation in row 13 and an incorrect label in row 15, a user might not realize that the required profit margin is already included in row 15 and might add a second mark-up to row 15’s results to calculate the “final bids”.

Table 4: Examples of Missing and Incorrect Cell Documentation

<table>
<thead>
<tr>
<th></th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td></td>
<td><strong>Type of Wall</strong></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td><strong>Lava</strong></td>
<td><strong>Brick</strong></td>
</tr>
<tr>
<td>13</td>
<td>$1,296</td>
<td>$1,056</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Margin</td>
<td>$388.80</td>
<td>$316.80</td>
</tr>
<tr>
<td>15</td>
<td>Total Cost</td>
<td>$1,684.80</td>
<td>$1,372.80</td>
</tr>
</tbody>
</table>

Misunderstandings arising from incorrect cell documentation can occur for inputs as well as outputs. For example, if the input cell containing the fringe benefit percentage was incorrectly formatted to display as a $ value, the user may convert the percentage input value into a dollar equivalent estimate, which would be an inappropriate unit for the model’s logic.

Cells with ambiguous documentation can be just as misleading as missing and incorrect cell documentation errors. For example, in Table 5, it is not obvious that $20.00 in cell C2 represents the wage rate per hour or if the fringe benefit in cell C3 is a percentage of the wage rate as opposed to a dollar pay rate. In both cases, the user will have to make implicit assumptions about the context of the input when a value is entered for the cell, thus creating the possibility of error.
Similarly, confusion regarding outputs can occur when the labeling or formatting is not as detailed or numerically as precise as it should be. Incorrect formatting can display a rounded value of a fraction (e.g., 0.80 displayed as 1) and thus create confusion or lead to acceptance of an incorrect result. Similarly, when a cell is labeled Costs, it may be interpreted as Total Costs, Material Costs or Labor Costs. Ambiguous cell documentation is one of the most frequent qualitative error and yet one of the hardest errors to detect. The reviewers in our study had more difficulties in consistently identifying ambiguous cell documentation errors than any other type of qualitative error. To maximize the likelihood of detecting these errors, testing needs be done by an independent party that is not familiar with the model who can be more objective about the confusion created by the ambiguity.

**Example 2-2: Poor Layout for Readability**

Conway & Ragsdale (1997) emphasize the importance of readability in spreadsheet design and present general guidelines to organize a spreadsheet for readability. They stipulate that the spreadsheet should be laid out so that the user is reading left to right and top to bottom, related formulas should be in physical proximity of each other, use of blank space should be carefully used to distinguish blocks but not create unnecessarily long arcs of precedence, and the model should be kept on one screen or worksheet if possible. Within these general guidelines however, there are still different ways that a developer may choose to organize and communicate the business logic of the problem being modeled (e.g. Raffensberger, 2003).

Some organizations have created standards that govern layouts and data input cell formats to promote consistent development practices that will make spreadsheets easier to review, test and use. If such standards exist within an organization then the evaluation for Examples 1-3 and 2-2 should be modified with respect to how well the model follows the company’s specified guidelines. If standardized layout development practices do not exist in the organization, the general impression of the model’s readability can be evaluated. In our study, existence of poor layout for readability error was recorded if the reviewer could not easily follow the business logic of the problem.

**Category 3: Extendibility**

The Extendibility error category focuses on qualitative errors that place the model’s integrity at risk when the model is to be used as a template by others and may need to be extended to make it operational for its new application. A model designed for readability by the user in the current business application may not be designed for generalization to other similar applications or for extensions of the existing problem’s dimensions, such as additional time periods or product lines. There are certain layout attributes that increase the usability of a model when basic dimension and line item modifications may be required.

**Example 3-1: Poor Layout for Model Extension**

Models must be laid out such that when a row or column is inserted to extend a particular section of the model, the insertion does not hurt the readability of the other sections. In addition, a layout should require as few insertions of rows/columns, copy commands and formula modifications as possible into the existing model sections for extending the model.

Table 6 presents a model that is considered very readable but whose readability would be compromised should the user want to include a third type of material in the bidding process. It is easy to copy the calculation logic in cells D20:E29 to a new section for the third material bid in cells G20:H29. The cost per cubic foot in row 21 however is linked back to cell B5 in the input section. To
maintain the integrity and readability of the model, a row would need to be inserted after row 5 for the new material information. With the other input assumption sections laid out horizontally, the blank row will extend (with input shading for some cells) into these sections and may be confusing for future users. In addition, the user would have to know to link the material cost in the new cell H21 to the new input cell B6. Thus the model has a poor layout for extension because the user would need at least three steps to modify the model to include a third building material and the resulting model would be less readable after the modifications.

Table 6: Example of Poor Layout for Model Extension

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>D</th>
<th>E</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Assumptions</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Lava Rock</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>$3.00</td>
<td></td>
<td>Length</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Brick</td>
<td></td>
<td>Height</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Bid Margin</td>
<td></td>
<td>Width</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>30%</td>
<td></td>
<td>Total Cubic Volume</td>
<td>240</td>
<td>Hourly Wage</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>input field</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Calculations</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Lava Rock</td>
<td></td>
<td>Brick Rock</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>$3.00</td>
<td></td>
<td>$2.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>240</td>
<td></td>
<td>240</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>$720.00</td>
<td></td>
<td>$480.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>$576.00</td>
<td></td>
<td>$576.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>$1,296.00</td>
<td></td>
<td>$1,066.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>30%</td>
<td></td>
<td>Bid Margin</td>
<td>30%</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>$1,684.80</td>
<td></td>
<td>Grand Total Bid</td>
<td>$1,372.80</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example 3-2: Poor Layout for Copy/Paste

Some models will be laid out so that rows/columns can be easily inserted without harming the readability of the model, but the layout of the logic in the model does not allow for easy copying to the new row/column. For example, in Table 7 the labels between inputs and calculations are switched. Columns D and E, that contain the cost inputs, are labeled as Lava and Brick wall types respectively, but columns F and G are labeled as Brick and Lava respectively, thereby not allowing to copy the relative formulas in column F to column G.

Table 7: First Example of Poor Layout for Copy/Paste

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Volume</td>
<td>Lava</td>
<td>Brick</td>
<td>Brick Wall</td>
<td>Lava Wall</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>$3.00</td>
<td>$2.00</td>
<td>$480.00</td>
<td>$720.00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Similarly, the labels for the cost inputs in Table 8 are placed in two consecutive rows (cells B11 and B12), but the headings for calculations are switched to columns (B15 and C15). This change in layout makes it impossible to copy the calculation logic from one cell to the other, as the relative reference relationship is lost. If another wall made of a third material is to be inserted in this spreadsheet, the user would have to insert a new row after row 12 and then manually program the logic into cell D15 as the formula in C15 cannot be extended through simple copying. This last step is considered a risky modification as it requires the user to identify correct references for the new formulas. Here, even though the spreadsheet was initially correct, a qualitative error exists in the form of the lack of extendibility through easy copy/paste and thus increases the chance of a quantitative error in subsequent use and modification of the spreadsheet.

Table 8: Second Example of Poor Layout for Copy/Paste

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>Volume (in cu.ft.)</td>
<td>240</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Lava cost/cubic ft</td>
<td>$3.00</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Brick cost/cubic ft</td>
<td>$2.00</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Lava Wall</td>
<td>Brick Wall</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Total Material Cost</td>
<td>$720.00</td>
<td>$480.00</td>
</tr>
</tbody>
</table>

Note that in evaluating the model layouts in Tables 7 and 8 for Example 3-2 (suitability for copy/paste), we did not evaluate whether the associated formulas had been correctly coded with the necessary absolute and relative references. The outcome in this example is based solely on how the various cells were positioned relative to each other.

Example 3-3: Poor Absolute/Relative Cell References for Copy/Paste

Even when the layout is designed well so that formulas can be copied and maintain appropriate relationships, the formulas may not be coded with the correct use of absolute and relative references for copying. For example the logic in cells D22:E29 in Table 6 can only be copied if references to B7, E8 and H14 are absolute and all other relationships are relative.

We only record this referencing error for formulas that could be copied given the appropriate layout of the model by assessing whether the correct use of absolute and relative references had been programmed to allow the formula to be successfully copied to other cells in the model. Based on this definition, no errors associated with Example 3-3 exist for the illustration in Table 8 as the model was not laid out to allow copying of the formulas. Determination of an error was based on the proper use of absolute or relative reference syntax in the formulas for various cells, specifically the incorrect use or absence of the $ symbol. These errors are not based on the accuracy of the existing cell formulas but rather are assessed under the assumption that a user will do a copy/paste in the future. The error occurs when the formulas in future pasted cells lose the integrity of the correct logic due to reference problems.

Category 4: Formula Integrity

The Formula Integrity error category considers how robust cell calculations are given the structure of the formula that was used to generate the cell entry. Assuming that the initial model was reasonably tested and no quantitative errors were identified, the logic and formulas associated with the cell should not compute a wrong result. However, some formula structures are vulnerable to misinterpretation and therefore are more prone to producing an eventual miscalculation once the model is put into operational use.

Example 4-1: Spurious Formulas

Formulas that use a function inappropriately create confusion about the correctness and intent of the formula. For example, if the formula \( = \text{sum}(C13+C16) \) is entered in a cell, then the use of the \textit{sum} function may lead to unexpected results due to incorrect application of the function or misunderstanding of its purpose.
function is spurious and creates unnecessary confusion. While the intent of the formula is to add the entries in cells C13 and C16, a user may inadvertently read that formula incorrectly as \textit{sum(C13:C16)}. This user would then believe that entries in cells C14 and C15 are being included in the total. A user who does notice that the parameter is indeed C13+C16 may wonder if it was supposed to have been C13:C16 and may be tempted to alter the formula to reflect a more common syntax. This will be especially confusing if additional line items are added between the current row 13 and row 16.

\textit{Example 4-2: Lack of Explicit Formulas}

Models can be created where a value that has been calculated externally by the user is entered into an input cell, without the calculation or logic displayed in the spreadsheet itself. For example, if a formula did not exist for the wall volume in cell C3 of Table 2, the user externally calculated the volume of the wall and then input the resulting calculation in cell C3 of the model. This lack of explicit formula weakens the integrity of the model’s results as there is no process to ensure that the user knows how to correctly calculate the volume of a wall and therefore may enter an erroneously calculated input into the model. This situation is even more serious if an externally calculated input is based on intermediate calculations generated in the model.

4. DISCUSSION

In developing the taxonomy presented in this paper, we referred to the guidelines suggested in Powell et al., (2008). They suggest that a well-developed classification should have three characteristics: (1) it should specify the purpose for which it was created and the context in which it is meant to be used, (2) each category should be clearly defined and examples provided, and (3) the classification should be tested in the relevant context and evidence provided that different people classify errors consistently. Our main context for investigating design qualitative errors was usability of the model by end users throughout the spreadsheet life cycle as described in the discussion of Figure 2. The taxonomy’s ultimate purpose is to allow researchers and organizations to focus on identifying controls that can be used to mitigate the risks associated with each type of qualitative error associated with the design and development phase of spreadsheet models. We have also provided a clear definition for each type of error that can be detected during the testing phase of a spreadsheet model and provided examples of each type. The consistency of the taxonomy is not yet tested but an inter-rater reliability study is in progress.

Developing this taxonomy also has highlighted the issue of different approaches for counting the various qualitative errors. Unlike the more mature literature on quantitative errors where benchmarks for error counting are well established, the existing literature on qualitative errors does not yet have a consensus. For reasons of efficiency an organization may wish to adopt our methodology of simply flagging the existence of a type of error. It is also possible to count the number of cells that contain a particular type of error and calculate the cell error rate (CER) for each error type. In practice, an organization would need to define or adopt an error counting methodology that is appropriate for their purposes in evaluating the spreadsheet. However, for direct and meaningful comparisons outside the organization and for the research community at large, standardization of qualitative error counting is essential. We hope that our proposed taxonomy will be a starting point for researchers to develop and explicitly state a standardized qualitative error counting and classification methodology, so that future results can be meaningfully compared and measures can be recommended for reducing spreadsheet qualitative errors.

There are several limitations with the proposed taxonomy at present. The main limitation is that it is currently restricted to qualitative errors encountered in problems of a similar nature to the Wall problem. Possible other types of qualitative errors that result from use of features such as linking formulas between files, VBA, manual procedures for executing macros or copy/pasting data into the model, and use of array formulas and range names was not evaluated because these features are typically associated with modeling more complex problems. A classification of errors associated with use of these features could not be obtained from our experiment due to the simple nature of the “Wall” problem. The taxonomy should therefore be expanded and tested on problems that require larger models with more advanced features often embedded in an organization’s large and mission-
critical spreadsheets. Future research then would consider how the existing categories and subcategories in the proposed taxonomy could be extended and better defined so that more of a consensus emerges on all the possible types of qualitative errors.

5. CONCLUSION

Qualitative errors in spreadsheets are as serious as quantitative errors and their study is equally important. However, to date, there has been little discussion in the literature regarding qualitative errors. We have made an attempt to develop a qualitative error taxonomy based on the previous work in the literature and errors obtained from an experiment with the “Wall problem.” We envision this research as a starting point for focusing on the definitions and counting schemes for the study of qualitative errors that can be used by the research community to move towards a broader, and generally accepted taxonomy for qualitative errors in a spreadsheet.

REFERENCES


An Insight into Spreadsheet User Behaviour through an Analysis of EuSpRIG Website Statistics

Grenville J. Croll
EuSpRIG – European Spreadsheet Risks Interest Group
grenvillecroll@gmail.com

ABSTRACT

The European Spreadsheet Risks Interest Group (EuSpRIG) has maintained a website almost since its inception in 2000. We present here longitudinal and cross-sectional statistics from the website log in order to shed some light upon end-user activity in the EuSpRIG domain.

1 INTRODUCTION

EuSpRIG was founded in March 1999 as a collaboration between spreadsheet researchers at the University of Greenwich, the University of Wales Institute Cardiff and HM Customs & Excise. Its mission was to bring together academics, professional bodies and industry practitioners throughout Europe to address the ever-increasing problem of spreadsheet integrity.

EuSpRIG held its first formal meeting in the offices of Her Majesty’s Customs and Excise as it was then known, in central London in January 2000. The group constitution was drawn up and the first officers appointed. The meeting decided to organise and hold a conference on Spreadsheet Risks at Greenwich University, London that July. There was some concern that anyone other than the organisers would turn up. The conference was, however, a great success with 50 delegates, some excellent papers and presentations, followed by a very enjoyable conference dinner.


EuSpRIG established a website in the year 2000, shortly after EuSpRIG’s formal establishment [Chadwick, 2003]. The EuSpRIG brand identity was revised and went online in October 2009 at the same time as a major update to the website (www.eusprig.org) which significantly extended the content. The input to this analysis is the complete web traffic data for calendar year 2010, with basic web site hits from 2004-2010 by month.

2. OBJECTIVES

There is very little work in the public domain outlining the characteristics and behaviour of spreadsheet users other than that which, by necessity, is part of spreadsheet error research [Panko, 2000] [Panko & Ordway, 2005] [Croll, 2005] [Caulkins et al, 2007]. An important exception is a large scale survey of MBA spreadsheet users [Baker et al, 2005] undertaken as part of the Spreadsheet Engineering Research Project (SERP). Their paper also briefly reviews spreadsheet user survey work over the past 20 or so years.

EuSpRIG objectives are clearly stated on their website:
EuSpRIG offers Students, Professors, Directors, Managers and Professionals in all disciplines the World’s only independent, authoritative and comprehensive web based information on the current state of the art in Spreadsheet Risk Management.

EuSpRIG is the largest source of information on practical methods for introducing into organisations processes and methods to inventory, test, correct, document, backup, archive, compare and control the legions of spreadsheets that support critical corporate infrastructure.

This paper seeks to extract pertinent information from spreadsheet users internet searches which have landed on the EuSpRIG website in order to inform future research about end-user computing in the EuSpRIG domain. Clearly, many related internet searches will not land on a EuSpRIG web page. We use Google page ranking statistics to qualitatively infer the likelihood that particular internet searches will land on a EuSpRIG web page.

The analysis of website logs is a common activity in the pursuit of commercial profit. We assume that investigation of the EuSpRIG website logs provides useful information about spreadsheet users thoughts and actions.

3 SITE HITS SUMMARY

The first analysis of EuSpRIG web site traffic was reported to the EuSpRIG executive in June 2006. It reported site hits (i.e. unique visitors) by month from January 2004 (Figure 1). Recorded site hits at the beginning of the period in January 2004 were 478 per month, which grew over a period of 30 months to 1949 hits in June 2006. A logarithmic regression model proved a good fit (Adj. $R^2$=80%) showing a compound growth of 60% per annum over the 30 month period. Site traffic came from diverse sources and there were no particular outliers which had to be accommodated in this early longitudinal analysis. The hit counter was changed in July 06 with no material changes observed.

As is the case with exponential models, they soon come to an end. The second analysis of monthly website statistics, which is reported to the EuSpRIG executive in this report,
occurred in January 2011. Exponential growth ceased in February 08, to be replaced after a short period of decline by slower probably linear growth and monthly website traffic of 2750 hits per month. There were three apparent exceptions to this pattern.

Figure 2

The first exception to the normal pattern of website traffic occurred in February 2007, when there was a notable one month decline in traffic for which no explanation is offered. The second exception followed an article by Dominic Connor in www.theregister.co.uk in February 2008 which generated about 3000 hits [Connor, 2008]. The timing of this article corresponded to the onset of the Global Financial Crisis. The third exception occurred in the early days of February 2011, two months after the bulk of this analysis had been done, when a link to the EuSpRIG website was posted on Twitter www.twitter.com generating about 1,200 additional hits.

In Figure 2 we have extended the log model to include the period up to February 08. Annual compound growth was slightly in excess of 60% for this extended period and the model offered an improved fit (Adj. R^2=86%). We also show the slower linear growth from January 09.

The total number of unique visitors to the EuSpRIG website 2000-2010, including those visiting before the web counter was established, is approximately 250,000. Visitor numbers grew from 32,915 in 2009 to 35,327 in 2010, an increase of 7.3%.

4 SITE HITS BY GEOGRAPHY

Over a period of 12 months, the majority of countries are represented by at least one visitor to the EuSpRIG website as recorded by Google Analytics. By December 2010, of 13,795 total page downloads, 99.35% were accounted for by the following 25 countries:
The United States, Canada, European Union, Australia and Japan account for the vast majority (93%) of page downloads. US dominance is almost overwhelming. European page downloads were 8.4%.

5 SEARCH PHRASES

5.1 SPECIFIC SEARCH TERMS

During the course of 2010, Google Analytics recorded that there were 12,520 unique search terms used which resulted in a visit, however brief, to the EuSpRIG website. The top 30 main search terms are given in Table 2 (note that minor spelling variations have been corrected & amalgamated for the more frequently occurring terms). If we regard the search terms of Table 2 as the “Specific” search terms, then these comprise 34.5% of the total. The “Generic” search terms, numbering approximately 6,500, comprise 65.5% of the total search terms.

In the final column of Table 2 we give the Google Page Rank of each search term. The Google Page Rank was obtained at the time of this analysis by searching again for each term in Column 1 of Table 2 and noting the Page Rank. So for example the Google Search term “EuSpRIG” gave www.eusprig.org as the first listed result. Likewise, the Google search term “importance of spreadsheet” gave www.eusprig.org as the 51st listed result. Clearly, the Google Page Rank will change over time and the manner and timing of such changes over a period may provide useful information into the state of art and the spreadsheet end-users mindset. There was little change in the page rankings of Table 2 in the approximate six month period between draft and final submission of this paper.
We believe it is highly likely that end users unfamiliar with EuSpRIG who see the EuSpRIG site as the first ranked site for their search term will click on the EuSpRIG site reference. Note that for 11 (36%) of the above 30 search terms, EuSpRIG is the top ranked site and for 18 search terms (60%), EuSpRIG appears on the first page of Google results.

Note that the word “Excel” appears only once in Table 2 and the words “Test” and “Document”, not at all. Note that the words “modelling” and “modeling” appear at almost the exact same frequency and are listed together.

It is interesting that site visits due to use of the groups unique or abbreviated name comprise only 12% of site visits. It is of further interest that the number of visitors reaching the EuSpRIG site by use of “Spreadsheet Error”, “Spreadsheet Risk” and related search terms comprise only about 6% of visitors. The EuSpRIG group has recently attempted to move its focus away from its initial interest in Spreadsheet Errors to a wider mandate suggested by the term “Spreadsheet Risk Management” (SRM). The evidence from end user internet searches suggests that end user interest in EuSpRIG is far more broadly based than even SRM. It is fascinating that three of the top 25 routes into the EuSpRIG website are from searches where the Google Page Rank is 300 or more.

It is encouraging to note that site visitors using “Spreadsheet Modelling”, “Spreadsheet Best Practice” and related terms comprise about 10% of the popular specific search terms.
5.2 GENERIC SEARCH TERMS

By inspection, the Generic search terms are fairly interesting. First of all, they are extremely numerous, there being approximately 6,500 of them in 2010 alone. Secondly, they are extremely diverse. It will not be possible to give a full analysis of the Generic Search terms here, and a more rigorous taxonomic approach could be adopted.

To give an idea of the diversity of terms which have caused searchers to arrive at the EuSpRIG web site we give in Table 3 a count of the frequency of occurrence of search terms.

Table 3

<table>
<thead>
<tr>
<th>FREQ</th>
<th>NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;20</td>
<td>37</td>
</tr>
<tr>
<td>&gt;10</td>
<td>37</td>
</tr>
<tr>
<td>&gt;5</td>
<td>72</td>
</tr>
<tr>
<td>4</td>
<td>71</td>
</tr>
<tr>
<td>3</td>
<td>124</td>
</tr>
<tr>
<td>2</td>
<td>519</td>
</tr>
<tr>
<td>1</td>
<td>5773</td>
</tr>
</tbody>
</table>

Thus there are 37 search terms which occur with a frequency of more than 20 times. All of these are documented in Table 2. Astonishingly, nearly six thousand visits to the EuSpRIG website in 2010 were by people who invented their own unique search term. Given the limited space available here, we document in Figure 3, search terms which occurred with a frequency of 4 or more, classified by a generic type which we manually assigned.

Figure 3 – Classification of Some Search Terms

Clustering of search terms was certainly possible and included thematic variations and incorrect spellings (of which there were remarkably few).
Frequently occurring patterns within the generic search phrases include questions such as “How do I”, “What is”, “Why”, “Who”, “What to”, together with a wide variation of every theme already reported and more. There are 700 single search terms that begin with the word “Spreadsheet”. Table 4 gives the frequencies of occurrence of the questioning search terms.

<table>
<thead>
<tr>
<th>Questioning Search Term</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>How</td>
<td>464</td>
</tr>
<tr>
<td>What</td>
<td>349</td>
</tr>
<tr>
<td>Why</td>
<td>86</td>
</tr>
<tr>
<td>When</td>
<td>71</td>
</tr>
<tr>
<td>Who</td>
<td>27</td>
</tr>
<tr>
<td>Which</td>
<td>26</td>
</tr>
<tr>
<td>Where</td>
<td>25</td>
</tr>
<tr>
<td>Total Questions</td>
<td>1048</td>
</tr>
<tr>
<td>Total Search Terms</td>
<td>12520</td>
</tr>
</tbody>
</table>

5.3 GENERIC SEARCH PHRASE CURiosITIES

There are a number of curious search terms which have lead end-users to the EuSpRIG website. One curiosity is where a user has typed a complete exam question into Google and then visited the EuSpRIG website. A good example is:

“3. discuss an example of how you have used a spreadsheet to help you make a decision. discuss the spreadsheet logic the data inputs and calculated values that you used to make a decision.”.

Another example is:

“in the plan section you are required to plan the layout of your spreadsheet. give your spreadsheet a title and using the information below start to think about what information is required. the layout of the spreadsheet should be clear and easy to read. remember to leave space for your bar graph. below is a list of the candidates who sat the exams and their results.”

Other curious examples are of the use of people or company names. These are substantially less frequent than one might have expected.

5.4 THEMATIC SEARCH TERMS

We used a simple spreadsheet to extract the frequency of occurrence of keywords relating to important themes within spreadsheet research from the 12,520 EuSpRIG 2010 search terms. The full software development lifecycle as it might be applied to spreadsheets has previously been established [Grossman & Ozluk, 2004].

5.4.1 Spreadsheet Testing

Search terms relating to testing spreadsheets [Pryor, 2004] [Panko, 2006] occurred 132 times – about 1% of the total. Table 5 gives the frequency of occurrence of the more frequently occurring terms relating to testing.
Table 5- EuSpRIG 2010 Search Terms – “TEST”

<table>
<thead>
<tr>
<th>Search Term</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Plan Spreadsheet</td>
<td>14</td>
</tr>
<tr>
<td>Spreadsheet Test Plan</td>
<td>7</td>
</tr>
<tr>
<td>Best Spreadsheet Test</td>
<td>3</td>
</tr>
<tr>
<td>Cross Foot Balance Test</td>
<td>3</td>
</tr>
<tr>
<td>Best practices spreadsheet testing</td>
<td>2</td>
</tr>
<tr>
<td>How to test a spreadsheet</td>
<td>2</td>
</tr>
<tr>
<td>Out of balance financial software testing</td>
<td>2</td>
</tr>
<tr>
<td>Spreadsheet practice test</td>
<td>2</td>
</tr>
<tr>
<td>Test plan for a spreadsheet</td>
<td>2</td>
</tr>
<tr>
<td>Test your spreadsheet</td>
<td>2</td>
</tr>
</tbody>
</table>
| The other 93 search terms relating to testing spreadsheets each occurred once. There were 47 occurrences of the use of the word “cross”, as in cross testing, cross foot totals etc, a simple to implement but effective method of checking spreadsheets [O’Beirne, 2009].

5.4.2 Spreadsheet Documentation

There were just 48 occurrences of the word “Document” [Pryor, 2006] [Payette, 2006] in the 12,520 EuSpRIG 2010 search terms (0.38%). All 48 search terms were different. A general lack of interest in documenting spreadsheets has been previously noted.

5.4.3 Spreadsheet Best Practice

It is pleasing to note that there were 756 occurrences (6%) of a search term that included the phrase “Best Practice” in the EuSpRIG 2010 search terms. 288 of these were unique. Table 6 lists the most frequently occurring “Best Practice” search terms.

Table 6 – EuSpRIG 2010 Search Terms “BEST PRACTICE”

<table>
<thead>
<tr>
<th>Search Term</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spreadsheet modelling best practice</td>
<td>153</td>
</tr>
<tr>
<td>Spreadsheet best practice</td>
<td>61</td>
</tr>
<tr>
<td>Spreadsheet best practices</td>
<td>31</td>
</tr>
<tr>
<td>Modeling best practices</td>
<td>27</td>
</tr>
<tr>
<td>Spreadsheet design best practices</td>
<td>25</td>
</tr>
<tr>
<td>Best practice</td>
<td>24</td>
</tr>
<tr>
<td>Spreadsheet modeling best practice</td>
<td>15</td>
</tr>
<tr>
<td>End user computing best practices</td>
<td>13</td>
</tr>
<tr>
<td>Spreadsheet modeling best practices</td>
<td>12</td>
</tr>
<tr>
<td>Best practices</td>
<td>11</td>
</tr>
<tr>
<td>Best practice pdf</td>
<td>10</td>
</tr>
<tr>
<td>Best practice modelling</td>
<td>9</td>
</tr>
<tr>
<td>Modelling best practice</td>
<td>9</td>
</tr>
<tr>
<td>Best practice spreadsheets</td>
<td>6</td>
</tr>
<tr>
<td>Spreadsheet modeling for best practice</td>
<td>6</td>
</tr>
<tr>
<td>Spreadsheet modelling best practice ibm</td>
<td>6</td>
</tr>
<tr>
<td>Spreadsheets best practices</td>
<td>6</td>
</tr>
<tr>
<td>Best practices modeling</td>
<td>5</td>
</tr>
<tr>
<td>Best practice modeling</td>
<td>4</td>
</tr>
<tr>
<td>Best practice spreadsheet modelling standards</td>
<td>4</td>
</tr>
<tr>
<td>Spreadsheet modelling best practices</td>
<td>4</td>
</tr>
</tbody>
</table>

Several of the above search terms are directed towards a popular Best Practice guide [Read & Batson, 1999] which is available on the EuSpRIG website.
5.4.4 Spreadsheet Engineering

There were 51 search terms out of 12,520 (0.4%) that contained the word “Engineer”. Only some of them related to Spreadsheet Engineering [Grossman, 2002]. All 51 terms were different.

6 SUMMARY AND CONCLUSION

We have briefly introduced the European Spreadsheet Risks Interest Group (EuSpRIG) outlining its establishment and development over eleven years 2000-2010. We have analysed summary and detailed web data for the same period by quantity, geography and content. The content analysis was performed by looking at specific search terms, generic search terms, curiosities and important themes relating to the activities of EuSpRIG.

In 2010 there were about 35,000 unique visitors to the EuSpRIG website, a total which had been growing exponentially at 60% per annum in previous years, but is now growing more slowly at 7.3% per annum. This may reflect an initial interest followed by a maturation phase. It is curious, and perhaps merely a coincidence, that the period of decline in between the two differing growth cycles corresponds to the period of the recent global financial collapse [Croll, 2009]

Analysis of the 12,520 unique search terms which resulted in visitors accessing the EuSpRIG website in 2010 provides an interesting insight into the thoughts and interests of spreadsheet users worldwide.

Only a small proportion of these users – around 10% - search for important themes within the software engineering paradigm, namely testing, documentation and best practice. It is likely, but not certain (given EuSpRIG’s generally high Google page ranking for these search terms), that these visitors to the EuSpRIG website are almost the only End-Users interested in these issues.

Based upon this analysis of the EuSpRIG 2010 web site statistics, it would appear to be the case that in 2010 only a few thousand people worldwide had a qualified interest in issues related to the integrity and quality of spreadsheets, best practices related to their development and the minimisation of risks related thereto. Given the ubiquity, importance and criticality of spreadsheets [Croll, 2005] within contemporary society, this is a continuing cause for concern which EuSpRIG quite rightly addresses.

7 ACKNOWLEDGEMENTS

The author thanks Patrick O’Beirne for his establishment of the original EuSpRIG web site, his colleagues on the EuSpRIG committee for their valuable contributions to the re-branding & re-writing of the EuSpRIG website and Tineke Warren of KINET e-solutions for its faultless reimplementation & maintenance. The author thanks the anonymous referees for their constructive comments.

REFERENCES


Requirements for Automated Assessment of Spreadsheet Maintainability

José Pedro Correia, Miguel A. Ferreira
Software Improvement Group,
Rembrandt Tower, 14th floor
Amstelplein 1, 1096 HA Amsterdam, The Netherlands
Email: j.p.correia@sig.eu, m.ferreira@sig.eu

ABSTRACT

The use of spreadsheets is widespread. Be it in business, finance, engineering or other areas, spreadsheets are created for their flexibility and ease to quickly model a problem. Very often they evolve from simple prototypes to implementations of crucial business logic.

Spreadsheets that play a crucial role in an organization will naturally have a long lifespan and will be maintained and evolved by several people. Therefore, it is important not only to look at their reliability, i.e., how well is the intended functionality implemented, but also at their maintainability, i.e., how easy it is to diagnose a spreadsheet for deficiencies and modify it without degrading its quality.

In this position paper we argue for the need to create a model to estimate the maintainability of a spreadsheet based on (automated) measurement. We propose to do so by applying a structured methodology that has already shown its value in the estimation of maintainability of software products. We also argue for the creation of a curated, community-contributed repository of spreadsheets.

1 INTRODUCTION

Although there is a considerable amount of work in the literature regarding the occurrence of errors in spreadsheets, another important aspect has been so far mostly neglected, namely maintainability. Most surveys conducted to determine the degree of usage of spreadsheets in large companies agree on one thing: their presence is almost ubiquitous. Furthermore, they very often play crucial roles in the organizations where they are used [Croll, 2009].

Much like software systems, their life span can be of many years and they can be developed and maintained by several different people. This makes it crucial to view spreadsheets from a different perspective, namely to study how easy they are to maintain, i.e., how easy it is for the person responsible for a spreadsheet to:

- understand it, diagnose it for deficiencies or determine where change is required;
- actually perform a required change;
- avoid unintended effects of a local change in the rest of the spreadsheet;
- validate the modified spreadsheet.

There is latent risk in a spreadsheet that is difficult to maintain, since errors will be more difficult to identify and/or address, and errors are extremely common to occur [Panko, 2008]. Organizations can therefore greatly benefit from insight into the maintainability of their spreadsheets.
sheets, since it makes them aware of the latent risks and thus empowers them to take action to address them.

An assessment of the maintainability of a spreadsheet can be performed by an expert (or several) who would inspect the spreadsheet and give his opinion on how maintainable he thinks it is. Another option, however, is to estimate the maintainability based on automated measurement. The advantages of such an approach are clear:

1. **Objectivity**: A metric-based assessment is not influenced by the particular opinion of one or more experts, but based solely on facts: the measurement results;
2. **Repeatability**: Since the assessment is purely fact based, it can be more easily repeated or reproduced, as long as the measurements are deterministic and can be redone;
3. **Cost-effectiveness**: It is expected that an automated approach delivers a faster assessment than an expert. Also, an expert's time is more expensive than the time of a computer;
4. **Scalability**: Given the time and cost gain by using an automated approach, assessing a large set of spreadsheets is much less of a problem.

The Software Improvement Group (SIG) has created such an automated approach for evaluating the maintainability of software products (see [Heitlager, 2007] and [Baggen, 2010]). The TÜVit (a German certification party affiliated with the TÜV NORD Group) has accredited the SIG as an evaluation laboratory for the Trusted Product Maintainability certification scheme. The two key elements of the approach are a layered quality model that relates measurement results to high-level notions of quality, based on the ISO/IEC 9126 standard [ISO, 2001] and a continuously growing repository of evaluation results [Correia, 2008]. This benchmark repository allows us to study metrics’ behavior in real-world software products and periodically recalibrate the quality model. Periodic calibration ensures that the quality model remains a reflection of the state of the art in software engineering.

We believe this methodology, which has shown its value both in practice as in some experimental studies (see [Luijten, 2010] and [Bijlsma, 2011]), can and should be applied to the evaluation of spreadsheet quality, in terms of maintainability.

The remainder of the paper is structured as follows. Section 2 presents some background on how the SIG’s methodology works for software products. Section 3 defines the requirements for automated assessment of spreadsheet maintainability. Section 4 briefly describes the EUSES spreadsheet corpus. Section 5 describes the challenges in developing a maintainability model and curating a spreadsheet corpus to support it. Finally, Section 6 concludes the paper with some final remarks.

This paper aims at creating awareness for the issue of spreadsheet maintainability and to clearly lay out the requirements for a methodology that could support its automated assessment.

2 BACKGROUND

As mentioned in the previous section, two elements form the backbone of our approach: a quality model based on (automated) measurement and a repository of real-world instances of the object of study.

The SIG Quality Model is essentially a layered aggregation method to transform low-level, quantitative measurements into high-level ratings. A description can be found in the original...
publication [Heitlager, 2007]. The model has undergone some recent modifications since then, which can be found summarized in subsequent publications (e.g., [Luijten, 2010], [Nugroho, 2011]).

In general terms, the application of the model involves the following steps:

1. Static analysis is performed on the source code to collect measurement data about the software system. Metrics are collected for the low-level system elements such as lines, units (e.g., methods or functions), and modules (e.g., files or classes). In the case of spreadsheets, low-level system elements can be cells, formulas, rows, and sheets;

2. Metrics for building blocks are mapped onto ratings for properties at the level of the entire software system, such as volume, duplication, unit complexity, etc. These ratings take values in the interval between 0.5 and 5.5, which can be rounded to an entire number of stars between one and five. (This constitutes a unitless ordinal scale that facilitates communication and comparison of quality results at the level of entire software systems.) The mapping functions for some properties are straightforward translations of system-level metrics to ratings. The remaining mapping functions make use of risk profiles as intermediate device. Each profile is a partition of the volume of the source code in each of four risk categories: low, moderate, high, and very high risk;

3. The ratings calculated per property are then aggregated to ratings for each of the four sub-characteristics of maintainability as defined in the ISO/IEC 9126, which are then finally aggregated to a rating for the maintainability of the whole software system. Which properties are deemed to influence which sub-characteristics is represented in a table with dependencies (for more on this mapping see [Correia, 2009]).

The mapping from metrics to ratings, described in point 2, is configured by a set of thresholds. These define the risk categories, as well as the mapping from either system-level metrics or risk profiles to ratings. The specific values for these thresholds have been calibrated based on statistical study of a large, representative sample of software systems [Alves, 2010], the SIG benchmark repository. This is crucial to the methodology, since it ensures that the model embodies realistic standards of quality. In order to make sure the model remains a reflection of the state of the art in software development, the repository is continuously updated with evaluation results, and re-calibration is performed periodically, as illustrated in Figure 1.

![Figure 1 – Calibration cycle (adapted from [Baggen, 2010]).](image-url)
4 WHAT IS NEEDED?

In order to adapt such an approach for spreadsheets, one can use a very similar set up, only tailoring it to account for the particular differences between these and traditional software systems. Nevertheless, this should be done carefully. We identify the following activities that need to be performed:

1. The goals of measurement and how we can achieve them through metrics need to be clearly defined. Many metrics for spreadsheets have already been defined in the literature (see [Bregar, 2004], [Hodnigg, 2008]), but in order to construct a model one needs to have present not just how to measure, but what (and why) to measure and how to interpret the results. We intend to apply the Goal Question Metric (GQM) approach [Basili, 1994] in order to achieve this. The GQM method provides a structured way of defining goals (in our case the ultimate goal is the assessment of spreadsheet maintainability), these goals are then broken down into different perspectives captured by questions and, finally, appropriate metrics are selected to provide data to answer the questions;

2. After a set of metrics is chosen, these should be investigated in a large, representative sample of spreadsheets. Namely, one needs to understand what the typical values of a given metric are, which extreme values can be found and whether they represent problematic situations or not. We intend to gather a large sample of spreadsheets, apply all the selected metrics consistently and perform statistical analysis, much in the fashion of [Alves, 2010].

The constitution of a representative repository of spreadsheets is very important for the implementation of the whole approach.

4 THE EUSES SPREADSHEET CORPUS

Since compiling a large and representative sample of spreadsheets can be quite challenging, we first decided to investigate what is available in the field of spreadsheet research. Apart from small sets used by different authors for their studies, we found only one large enough corpus, namely the EUSES Spreadsheet Corpus [Fisher, 2005].

This corpus is composed of 4498 spreadsheets and was set up by gathering spreadsheets from several different sources. Some curation was performed to exclude duplicated and unusable spreadsheets. The great majority (4401) was collected by querying the Google search engine¹. The authors also calculated several metrics per spreadsheet, mostly counts of cell types for different categories of cells.

From this large corpus, we were interested only in spreadsheets that matched the following criteria:

- The spreadsheet should contain computations. Excluding spreadsheets without formula cells reduces the set to 1977 data points;
- The spreadsheet should contain referenced input cells. Excluding spreadsheet without those further reduces the set to 1609 data points.

¹ http://www.google.com
We performed some preliminary analysis on this subset and determined that at least 365 spreadsheets (approximately 23%) have more than 25 unique formulas (copy-equivalent), which is an indication that there is a fare amount of non-trivial data points.

5 THE CHALLENGE

Even though the EUSES corpus is usable for a large-scale exploration of spreadsheet metrics, it has some disadvantages for its intended use in the context that we propose, namely:

- The corpus is most likely not representative of spreadsheets used in a professional context. The great majority comes from an internet search, where keywords such as “homework” or “grades” were used. The results of those queries are most likely spreadsheets used for personal management of information. Furthermore, most spreadsheets that implement some business value in an organization will not likely be publicly available on the internet;
- The corpus does not contain enough reliable meta-information to determine the actual representativeness of it, nor to allow for further analysis where grouping on different dimensions could provide some additional insight. It would be interesting, for example, to determine differences between spreadsheets used in different areas, developed by IT professionals versus end users, and so forth;
- There is no information on the legal permissions to analyze most spreadsheets in the corpus. The fact that they were publicly available through an internet search does not constitute legal permission of use;
- The age of the spreadsheets in the corpus is not clear. Furthermore, some 130 (out of the subset we considered) were found to be in a “BIFF5” format (format mainly used prior to Excel 97), which is an indication that they are quite old;
- The corpus is static: it was collected once and has not been updated since. We assume that spreadsheet development practices evolve and become better over time, this is one of the reasons why for software we re-calibrate the model periodically.

For all those reasons, we believe that there is a need for a spreadsheet repository compiled and managed in a similar manner to the SIG benchmark repository or, for example, the PROMISE\(^2\) data set for machine learning research. We consider the following to be important characteristics of such a repository:

- **Curated:** The repository should contain spreadsheets that have been curated on arrival. This is important not only to verify the usability of the spreadsheet (much like it was done in the EUSES corpus), but also to determine their relevance and to make sure the proper meta-information is also included and appropriately handled;
- **Dynamic:** The repository should continuously grow, in order to reflect the state of the art in spreadsheet development. This means, of course, that it needs to be versioned to enable the proper replication of studies;
- **Open:** The repository should be easily available to the community and everybody should be allowed to contribute. If authors or maintainers contribute with their own spreadsheets, they can also provide extra meta-information to enrich the repository with. Some contributors might however have security concerns in disclosing their spreadsheets or other information, thus part of the repository could be restricted and subject to non-disclosure agreements for those who would like to use it;
- **Useful:** In order to motivate contribution from researchers, as well as practitioners, the repository must prove useful to them. For researchers, the use would be the availabil-

\(^2\) http://promisedata.org/
ity of curated data for their studies. For any contributor, the repository could provide some general feedback on the quality of its contribution compared to the existing data. The whole repository could also be used to compile periodical, anonymous, “state of the art” reports. These are merely some examples.

6 CONCLUSION

In this paper we have presented our view on how an automated assessment of spreadsheet maintainability could work, what are the requirements for it and what is still missing. Furthermore, we challenge the spreadsheet developer and user communities to contribute to a curated spreadsheet repository that can support a spreadsheet maintainability model (among potentially other approaches to spreadsheet quality). We are prepared to lead the effort in creating such a repository, but we find the involvement of the community fundamental for the success of such an endeavor. The initiative already has the collaboration of the SSaaPP project3.

The approach we propose in this paper has already been successfully applied to assess the maintainability of software products. It is our conviction that with a carefully conducted study of spreadsheet metrics it is possible to select which are the most appropriate for building a similar maintainability model for spreadsheets.

We are open to other partnerships to support us in this effort, feedback on the whole project, but especially to contributions in terms of data points, so we leave the reader with a request: may we have your spreadsheet?

REFERENCES


3 SpreadSheets as a Programming Paradigm (http://ssaapp.di.uminho.pt)


Workbook Structure Analysis – “Coping with the Imperfect”

Bill Bekenn and Ray Hooper
Fairway Associates Ltd. PO Box 846, Ipswich IP5 3TT UK
info@fairwayassociates.co.uk

ABSTRACT

This Paper summarises the operation of software developed for the analysis of workbook structure. This comprises: the identification of layout in terms of filled areas formed into “Stripes”, the identification of all the Formula Blocks/Cells and the identification of Data Blocks/Cells referenced by those formulas. This development forms part of our FormulaDataSleuth® toolset. It is essential for the initial “Watching” of an existing workbook and enables the workbook to be subsequently managed and protected from damage.

1 INTRODUCTION

A study [Lawson et al, 2007] has shown that inexperienced spreadsheet developers have markedly less understanding of the benefits of practices such as integrating spreadsheet modules, keeping data and formulas separate and testing than experienced developers. This can lead to critical spreadsheets and workbooks being vulnerable to error because of poor workbook structure. This is a particular problem in small to medium sized organisations where resources to support spreadsheet development are limited. We have developed a software algorithm to enable the structure of a Workbook to be analysed and subsequently improved.

There is a school of thought that there is no such thing as “spreadsheet good practice” [Colver, 2004], arguing that there are many different ways of designing a workbook, each of which has advantages and disadvantages. In contrast, our experience of spreadsheet design and use shows that there is “good practice” at least for spreadsheet layout. This “good practice” is often adopted intuitively by experienced spreadsheet practitioners. This is distinct from overall workbook design (e.g. having Inputs, Outputs and Workings in separate sheets). Layout good practice can be based on assessing whether modifications (e.g. inserting rows) are likely to cause damage and tracking whether Blocks and Single Cells are accurately referenced by formulas.

Algorithms have been developed that perform Workbook Structure Analysis and find all the Formula Blocks/Cells and Data Blocks/Cells. To achieve this all the Formula Blocks are examined and the ones that share a common width or height are identified. The relationships between the Formula Blocks/Cells can thus be recorded. The Data Blocks/Cells are then found from the Formula References. Manual inspection by an expert practitioner could achieve the same end. These algorithms have evolved from a “Find Formulas” procedure that originally found formulas only and displayed the structure as borders around the filled Formula Blocks. This feature is not unique and is included in products such as OAK V4 [OPERIS, 2009] and SpACE [AuditWare, 2006].

Why perform the complex analysis needed to reveal the workbook layout and structure beyond identifying filled Blocks? At least one recent study [Aurigemma, 2010] has
shown that automated analysis performs less well at finding errors in spreadsheets than
detailed study by a person. Our Workbook Structure Analysis does not find all the errors
and instead concentrates on what can be done automatically to reveal the layout. It then
checks whether references in formulas accurately follow the layout/structure. Once the
underlying structure is revealed then areas of a workbook that appear untidy, e.g. with
Blocks that conflict with the overall structure, can be targeted for manual detailed
analysis. Our experience of using the algorithm, on real spreadsheets from our client base,
is that revealing the structure greatly speeds up the process of understanding and
analysing a workbook. This addresses some of the problems encountered when needing to
understand legacy spreadsheets [Hodnigg, 2008]. As well as visually revealing the
structure, potentially incorrect cell references are indicated as warnings or errors for the
user to inspect. The algorithm also gives an early indication of the likely scope of the
problems as a percentage “score”, and will show when the workbook is past redemption
and needs to be re-worked rather than checked [Murphy, 2007].

The original version of the “Find Formulas” procedure did not record spreadsheet
structure which could only be determined by the user inspecting the borders.
Nevertheless, familiar patterns reflecting both good practice and poor layouts could be
observed. Experience gained from manually analysing/checking spreadsheets has now
enabled the enhancement of our FormulaDataSleuth® toolset [Bekenn, 2008]. The latest
version of the “Find Formulas and Data” procedure records the spreadsheet structure and
flags layout anomalies. The correct identification of spreadsheet structure enables the
accurate detection of all Data Blocks/Cells. This includes identifying, if present, Data
Blocks which may be “unknown” to the user/developer but are nevertheless referred to by
Formula Blocks. These are denoted “Unintentional Data Blocks”.

2 STRUCTURE, “STRIPES”, AND “BLOCKS”

Spreadsheet structure may be defined in terms of “Stripes” and “Blocks” which are
defined in figure 1 and the following sub-sections. These concepts are examined in more
detail in [Bekenn, 2009].

![Figure 1: Spreadsheet Structure](image)

2.1 “Blocks” and “Single Cells”

An area of data or filled formulas is referred to as a “Block” [O’Beirne, 2005]. Blocks are
considered distinct from “Single Cells” which contain isolated items of data or one-off
formulas. There are three types of Block:-

Proceedings of EuSpRIG 2011 Conference
Copyright © 2011 EuSpRIG (www.eusprig.org) and the Author(s)
1. Single Row,
2. Single Column,
3. Two Dimensional (Multiple Row and Column).

2.2 “Stripes”

A horizontal set of Blocks spanning the same rows, make up a “Horizontal Stripe”. A vertical set of Blocks spanning the same columns, make up a “Vertical Stripe”. Stripes are more important if they contain multiple rows/columns than if they contain just single rows/columns.

2.3 “Idealised Layout”

An “Idealised Layout” is a layout composed only of Stripes. These stripes can have whole rows or whole columns inserted at any point, without introducing the risk of an error. This is indicated in figure 1 with the Vertical Stripes and the top Horizontal Stripe. Stripes are not allowed to overlap and must occupy their own set of rows or columns. A Vertical Stripe can intersect a Horizontal Stripe and thus a Two Dimensional Block can be part of both. Where possible, references within formulas should be to cells on the same horizontal row or the same vertical column.

It is possible to conceive a Column or Row Insertion Test (“CRIT”) to test the resilience of a spreadsheet to modification. Any contiguous assembly of blocks which cannot have whole rows/columns inserted without causing errors is considered as failing CRIT. An example of this situation is shown in figure 1 with the block at the bottom right. An “Idealised Layout” will always pass a CRIT.

3 HOW THE STRUCTURE ANALYSIS WORKS

Figure 2: Program Components

Figure 2 shows the main program components required to execute the Workbook Structure Analysis. The Find Formulas and Data algorithm calls three functions to (i) find
the Formula Blocks, (ii) find the Data Blocks and (iii) perform the Stripe Analysis. Any isolated formulas and data in Single Cells are also found. The functions interact with the two data sets containing the “Parsed Formula Descriptions” and the “Map of Stripes”. Using these components a workbook structural analysis is performed in broadly the following order:-

1. Formula Blocks are found and logged in the “Map of Stripes” using the Find Formula Areas Function.
2. The Formula Blocks are parsed and recorded in the “Parsed Formula Descriptions” also using the Find Formula Areas Function.
3. Using the “Parsed Formula Descriptions” data set the Data Blocks are found and logged in the “Map of Stripes”.
4. The Stripe Analysis function is invoked after 1, 2 and 3, to progressively refine the layout picture of the workbook contained in the “Map of Stripes”.

The algorithm works like an expert practitioner to first record all major Blocks and their interrelation and then move on to more detailed checking. The detailed checking is capable of identifying Blocks that do not fit well with the overall structure, i.e. Blocks that do not conform to an “Idealised Layout”.

Section 4 will describe in more detail the three functions and their interaction with the data sets. Section 5 will then explain in more detail how the algorithm uses the three functions described in section 4.

4 THE FIND AND STRIPE ANALYSIS FUNCTIONS

Three forms of information are recorded as the three functions are invoked:–

1. The location (cell address) of each Formula or Data Block/Cell, stored together with the “Parsed Formula Descriptions”.
2. The locations (cell addresses) referenced within each formula, stored within the “Parsed Formula Descriptions”.
3. The locations (row/column addresses) of the stripes detected by the functions, stored within the “Map of Stripes”.

4.1 The “Find Formula Areas” Function

This function finds the filled Formula Blocks. A Formula Block would be expected to contain a contiguous area of filled formulas that could be generated from a single cell instance (seed) within the area by “Fill” or Copy and Paste. This is a relatively simple matter for the function to achieve with a contiguous area as it only needs to find the extent of the “Fill”. A more complex situation is depicted in figures 3 & 4, where the extent of the “Fill” can be seen but the area is not contiguous, i.e. there are “Exceptions” within the area in the form of different formulas, data cells and blanks. These are shown in grey cross-hatch in figures 3 & 4.

There are two options which can be specified when this function is called, Simple Find (figure 3, any Exceptions will bound Blocks) and Complex Find (figure 4, defining the maximum extent of the Formula Blocks while skipping over any Exceptions).

As formula Bocks and Cells are found they are recorded in the “Map of Stripes” so as to be ignored if they are found by subsequent passes of the algorithm.
4.1.1 Simple Find (Exceptions bound Blocks)

The approximate sequence of operations, used for establishing the initial layout picture, is as follows:-

1. Find a new, un-recorded, formula (scanning from top right of a sheet to bottom left across columns and then down rows).
2. Assuming the new formula is in the top left cell of a Block, scan across the top row and down the left column, looking for the first cell containing something different from the top left cell. These cells, shown in grey, are used to define the potential extent of a simple filled Block (figure 3, Extent 1).

The dotted outline in figure 3 indicates Extent 1 for the simple formula Block containing F1. Initially, this extends to two cells across and three down. There is the possibility that other rows and/or columns may be filled to a greater extent.

3. Rows to the right of Extent 1 and columns below it are checked to see if some rows/columns have a greater extent for the fill. If so, the size of the filled Block is increased to match the greatest horizontal and vertical extent finally defining the filled Block.

In figure 3 the top left filled Block found by the sequence of operations above, shown by the grey dashed outline “Extent of Formula Block” in figure 3, is logged in the “Map of Stripes”. Other filled Blocks will also be found and logged.
4.1.2 Complex Find (Exceptions skipped)

The approximate sequence of operations, used for finding Formula Blocks, is as follows:-

1. Find a new, un-recorded, formula (scanning from top right of a sheet to bottom left across columns and then down rows).
2. Assuming the new formula is in the top left cell of a Block, scan across the top row and down the left column looking for the end of a “Region” which is either delimited by a blank cell, a cell containing text, an existing predominant Stripe boundary or a formula that has already been found.
3. Scan back across the top row and up the left column from the end of the Region to find the furthest extent of formulas matching the one found initially in 1. These cells, shown in grey, are used to define the potential extent of a complex filled Block (figure 4, Extent 2).
4. Continue the backward scan, checking the top row and left column of Extent 2 to see how many cells contain either formulas that differ from the one found initially or numeric data.
5. If the number of different cells (separately across and down) is large compared with the number of identical ones then it is assumed that the filled Block is smaller than Extent 2 in 3. In this case the scan is repeated forwards, looking for the first different cells, across and down, which are used to define the extent of a smaller filled Block.

The chained outline in figure 4 indicates Extent 2 for the complex formula Block containing F1. Initially, this extends to five cells across and three down. There is the possibility that other rows and/or columns may be filled to a greater extent. There is also no certainty that all the cells within the Block have the same formula.

6. Rows to the right of Extent 2 and columns below it are checked to see if some rows/columns have a greater extent for the fill. If so, the size of the filled Block is increased to match the greatest horizontal and vertical extent finally defining the filled Block.
7. Operations similar to 1 through 6 are repeated within the extent of the defined filled Block. These look for filled Blocks and Single Cells containing a different formula and for cells overwritten with data. All of these are marked as “Exceptions” and are potential errors.
The filled Block found by the sequence of operations above, shown by the grey dashed outline “Full Extent of Formula Block” in figure 4, is logged in the “Map of Stripes”.

### 4.2 The “Find Data Areas” Function

Each formula in a workbook is analysed (or parsed) and recorded in the Parsed Formula Description data set (figure 2). This is done for each Formula Block as they are found by the Find Formula Areas function (section 4.1 and 4.1.2). The locations of the top left and bottom right cells of each range referenced by the formula are also recorded. A filled reference, where the row and/or column are relative, is taken as referring to a range of the same size as the filled Formula Block.

Data Blocks are found in stages using the different types of reference in the order of precedence below:-

1. Data referenced by relative filled formula References,
2. Data referenced by Range References but not overlapping with 1,
3. Data referenced by Single Cell References but not included in 1 or 2,
4. Data referenced by Range References but falling outside or in between Blocks/Cells that have been identified in 1, 2 or 3.

---

<table>
<thead>
<tr>
<th>Data Block 1 (found from Formula Block 1)</th>
<th>Formula Block 1 referencing Data Block 1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Unintentional Data Block 4</strong></td>
<td><strong>Unintentional Data Block 5</strong></td>
</tr>
<tr>
<td>Data Block 2 (found from Formula Block 2)</td>
<td>Formula Block 2 referencing Data Block 2</td>
</tr>
<tr>
<td>Total 1 of Data Blocks 1 &amp; 2</td>
<td>Total 2 of Formula Blocks 1 &amp; 2</td>
</tr>
<tr>
<td>Data Block 3 (found from Formula Block 3)</td>
<td></td>
</tr>
<tr>
<td>Total 3 of Data Block 3</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 5: Finding Data Blocks**

Figure 5 shows a common sheet layout where Data Blocks 1 & 2, separated by a blank row, are summed together into Total 1. Similarly Formula Blocks 1 & 2 are summed together into Total 2. The blank cells referenced are shown in grey cross-hatch and designated as “Unintentional Data Blocks”. Formula Blocks 1 & 2 reference Data Blocks 1 & 2 respectively.

In figure 5 there are two types of Formula Block. Firstly there are the SUM() functions in Totals 1, 2 and 3. These will have range references. Secondly there are filled references in Formula Blocks 1 and 2. Which of these Formula Block types should be dealt with first?
It turns out that, from a structure analysis point of view, the order of precedence for the different types of reference is important. If range references were considered first then the Unintentional Data Block 4 would be combined with Data Blocks 1 & 2. This would form a single combined Data Block, which is not a correct analysis of the structure. Moreover, it conflicts with the Horizontal Stripes defined by Formula Blocks 1 & 2. On the other hand, if the filled references in Formula Blocks 1 & 2 are considered first, then the Data Blocks 1 & 2 are correctly identified. The Unintentional Data Blocks 4 & 5, found from the range references in Totals 1 & 2, are treated as separate. Data Block 3 is also found when considering the range reference in Total 3, as there is no filled reference to consider in this case. The “rule” is therefore to consider filled references before range references.

Each data block found is logged in the “Map of Stripes”. This map is processed by the Stripe Analysis function (section 4.3), which identifies the ones define the sheet structure.

### 4.3 The “Stripe Analysis” Function

The Stripe Analysis function processes the Map of Stripes (see figure 2) to increasingly make sense of what the Find Formulas and Find Data functions discover from each pass of the overall algorithm. The map initially contains a mixture of (i) stripes which define the sheet layout picture, (ii) stripes that exist due to anomalies in fill and (iii) stripes caused by Exceptions within the Formula Blocks found. The number of filled rows and columns are totalled for each of the Vertical and Horizontal Stripes recorded in the map. These counts are then used to determine an order of predominance for each Stripe with the highest counts coming first.

Where a workbook conforms strictly to the criteria for an Idealised Layout (section 2.3) all the Stripes logged in the “Map of Stripes” could be used to define the structure of each sheet. However, the Idealised Layout criteria are not often met. It is necessary, therefore, to identify predominant Stripes within the map so that any layout anomalies do not interfere with the analysis of the sheet structure.

In general, therefore, the “Map of Stripes” is accessed starting with those stripes with the highest predominance. These are found first and are used to define the structure of each sheet. Any other stripes overlapping with the ones with highest predominance are noted as not conforming to the Idealised Layout criteria. However, any single column or row stripe that overlaps with a predominant Vertical or Horizontal Stripe will conform to an Idealised Layout. This is because single column or row Stripes will not cause CRIT failures.

### 5 THE “FIND FORMULAS AND DATA” ALGORITHM

The “Find Formulas and Data” algorithm is a multi pass procedure. It progressively builds up a record of the predominant Formula Blocks that obey Idealised Layout criteria (i.e. form “Stripes”). It then identifies other Formula Blocks and Cells that do not obey the Idealised Layout criteria. After each of these passes the Data Blocks are found using the different types of reference as described in section 4.2.

The concept of an Idealised Layout is important as the Blocks within a spreadsheet that conform to the rules for “Stripes” (section 2.2) take precedence over Blocks that do not. There are therefore two sets of passes with the first set tackling only Blocks that obey the Idealised Layout criteria and the second set tackling the rest.
Figure 6 shows a simple spreadsheet layout where Totals 1 & 2 are filled to 5 columns wide to include the row Totals 3 & 4. Formula Blocks 1, 2 & 3 are referencing Data Blocks 1, 2 & 3 respectively using relative references. Whilst Totals 1 & 2 are single filled Blocks, the right hand column of the Totals rows (shown with diagonal hatching) have a logically different function from the other totals and should be treated separately in terms of layout.

5.1 Primary Passes (Idealised)

The first passes of the algorithm find those Blocks which obey the Idealised Layout Criteria, i.e. they would, on their own, meet CRIT.

5.1.1 Finding the Vertical and Horizontal “Stripes”

Figure 6 shows how a predominant Vertical Stripe is identified. This is achieved by invoking the “Find Formula Areas (Simple Find)” function (section 4.1 & 0) without skipping over “Exceptions”. This function builds a temporary version of the “Map of Stripes” which contains all the different fill widths and heights encountered. The “Stripe Analysis” function (section 4.3) is then invoked to establish which stripes are predominant and contribute towards the structure of each sheet.

Considering figure 6 in more detail the two grey diagonal hatched cells on the right are column totals and are part of the filled Blocks of Totals 1 and 2. However, they need to be considered as separate because the row total formulas above them are different. The two filled Totals rows (1 & 2) could indicate a 5 column Stripe, but there are five rows (in total) in Formula Blocks 1 & 2 filled to 4 columns wide. This means that the 4 column Stripe is predominant in terms of the sheet structure. Another predominant 1 column Stripe (containing five formula rows (in total)) is also found formed by the Totals column, but this is of less importance as there is no horizontal fill.

The predominant Stripes form the layout structure of the sheet and are thus logged as conforming to an Idealised Layout. Formula Block 3 contains 3 filled rows and its 3
column Stripe overlaps with the previously defined predominant Stripe. This 3 column Stripe is thus logged as not conforming to an Idealised Layout.

At this early stage in the algorithm the stripe on the left of figure 6 has not been found as only the Formula Blocks have been examined.

5.1.2 Finding the Main Formulas (obeying Idealised Layout Criteria)

Now the predominant Stripes have been found the “Find Formula Areas (Complex Find)” function (section 4.1 & 4.1.2) is invoked finding filled Blocks, but skipping over “Exceptions” due to any corrupted formulas within filled areas. Only Blocks of filled formulas that match Stripes that conform to an Idealised Layout are processed at this stage.

In figure 6 Formula Blocks 1 & 2 are found together with the first 4 columns of Totals 1 & 2. Four Formula Blocks in the final totals column are found, including Totals 3 & 4 and the last cell of Totals 1 & 2 (diagonally shaded in grey).

5.1.3 Finding the Referenced Data (obeying Idealised Layout Criteria)

Having identified some formula Blocks that conform to an Idealised Layout the Find Data Areas function (section 4.2) is invoked looking at all references in these formulas. References that point to other Formula Blocks already found (e.g. the Totals column) are skipped but those pointing to cells which do not contain formulas are used to identify Data Blocks/Cells.

In figure 6 Data Blocks 1 & 2 are found from the references in Formula Blocks 1 & 2. The Stripe occupied by Data Blocks 1 & 2 is also identified and logged.

5.2 Secondary Passes (Non-Idealised)

Subsequent passes of the algorithm find those Blocks which do not obey the Idealised Layout Criteria i.e. they would not meet CRIT.

5.2.1 Re-Finding the “Stripes”

At this stage a second invocation of the “Find Formula Areas (Simple Find)” function (section 4.1 & 0) ensures that any smaller Blocks, within the larger blocks already found, are removed from the “Map of Stripes”. The “Stripe Analysis” function (section 4.3) is then invoked, for a second time, to re-establish which stripes are predominant and refine the layout picture of each sheet. This will also assess the predominance of Stripes logged from Data Blocks which have already been found (e.g. the left hand predominant Stripe in figure 6).

5.2.2 Finding the Remaining Formulas (not obeying Idealised Layout Criteria)

Now the current set of predominant Stripes have been found the full “Find Formula Areas (Complex Find)” function (section 4.1 & 4.1.2) is invoked, for a second time, finding any remaining filled Blocks. All Blocks of filled formulas within all Stripes are processed at this stage, whether or not the stripes conform to an Idealised Layout.
In figure 6 Formula Block 3 is found as this is the only one left over because it did not conform to an Idealised Layout.

5.2.3 Finding the Remaining Referenced Data (not obeying Idealised Layout Criteria)

Having identified all the formula Blocks, including those that do not conform to an Idealised Layout, the Find Data Areas function (section 4.2) is invoked again, looking at all references in all formulas. References that point to other Formula and Data Blocks already found (e.g. the Formula Blocks 1 & 2) are skipped but those pointing to cells not found so far, and which do not contain formulas, identify additional Data Blocks/Cells.

In figure 6 Data Blocks 3 is found from the reference in Formula Block 3.

5.2.4 Data “Mop-Up” of any other Referenced Data

The final stage of the algorithm is to find any other data that is referenced but has been missed by all the stages so far. This will normally be data that is range referenced but which is outside or in between filled Formula Blocks, or Data Blocks that are referenced by filled Formula Blocks.

There are no data blocks of this type in figure 6 but figure 5 in section 4.2 shows the Unintentional Data Blocks 3 & 4 which would be found at this stage. These unintentional Blocks, in between other Blocks which are summed together, are important as any temporary data or sum formulas entered in these cells will affect the final totals.

6 CONCLUSION

The algorithms documented here have been implemented within our FormulaDataSleuth® toolset and have successfully analysed a variety of different types of workbook from varying sources. The multi-pass nature of the algorithms allows the predominant structures to be identified first. This helps to identify Formula and Data Blocks that may be problematic in terms of the spreadsheet layout, particularly when modifications are made.

The main purpose of the Workbook Structure Analysis is to allow FormulaDataSleuth® to manage an existing workbook and monitor it for damage. However, setting up the monitoring of a workbook requires the detail of the workbook structure to be defined and all the Formula and Data Blocks/Cells have to be identified. The procedure for doing this analysis has the spin-off of providing a comprehensive check on the structure of the sheets and how formula references access other related Blocks. It is possible to envisage the functionality described here as a separate entity from the error checking capabilities of our FormulaDataSleuth® toolset.

7 REFERENCES


O’Beirne, P (2005), Spreadsheet Check and Control: 47 key practices to detect and prevent errors, Systems Publishing ISBN 190540400X
Spreadsheet on Cloud – Framework for Learning and Health Management System

1K.S. Preeti, 2Vijit Singh, 3 Sushant Bhatia, 4Ekansh Preet Singh, 5Manu Sheel Gupta
WZ-106/73 Rajouri Garden Extension
New Delhi – 110027, India

1 Independent Developer, Email Add: kspreeti@nsitonline.in
2,3,4 Software for Education, Entertainment and Training Activities, Email Add: manu@seeta.in, vijit@seeta.in, sushant@seeta.in, ekansh@seeta.in

ABSTRACT

Cloud Computing has caused a paradigm shift in the world of computing. Several use case scenarios have been floating around the programming world in relation to this. Applications such as Spreadsheets have the capability to use the Cloud framework to create complex web based applications. In our effort to do the same, we have proposed a Spreadsheet on the cloud as the framework for building new web applications, which will be useful in various scenarios, specifically a School administration system and governance scenarios, such as Health and Administration. This paper is a manifestation of this work, and contains some use cases and architectures which can be used to realize these scenarios in the most efficient manner.

Keywords- Cloud Computing, Python, Spreadsheet, Web Application, Education, Administration, Health.

I. INTRODUCTION

With industry analysts such as Gartner placing cloud computing amongst the top 10 strategic technologies for 2011, it is not a surprise that the world is now looking at this as the next buzz word. [1] In this context, as enterprises and organizations across the world gear up to embrace the technology, it is up to the creators of software to localize Cloud Computing and customize it in a manner suitable to the situation in which it is being deployed. It has also become imperative to import the existing systems into the corresponding cloud versions and utilize them to expand our presence in the cloud.

One such essential platform is that of the Spreadsheet. In the business world, the enterprises and education system, the Spreadsheet has become an indispensable tool of productivity and organization. The Spreadsheet is not just an effective information system, but it is also a strong and reliable framework for building applications. While the movement of Spreadsheet towards the cloud has started taking place in various forms such as Google Docs and Editgrid, what remains to be done is using the Spreadsheet framework over the Cloud to create innovative services which utilize the mathematical and programming capabilities of the activity and at the same time leverage upon the collaborated environment of the cloud. Not only this, these services must also have the capability to be customized for the typical use-cases. Through this paper, we examine various such scenarios.
Our work with One Laptop Per Child (OLPC Inc.) [2] led us to develop a novel Spreadsheet activity, SocialCalc for desktops. As an experiment, in order to bring Computer Supported Collaborative Learning into the education system, we imported the Activity into the Cloud. Once this was achieved, we examined various scenarios, apart from education where Collaboration over the Spreadsheet framework could be utilized to create fresh innovative software. Our work in this sphere brought us to the Indian context and hence, to the idea of using Collaborative Spreadsheets for developing customizable web applications such as a School system. In the course of the paper, we have examined this concept and also explained briefly how the Collaboration over Spreadsheet activity was achieved. Also, we have investigated a new architecture for deploying these applications successfully over the cloud and allow multiple users to collaborate and work.

II. INTRODUCTION TO SOCIALCALC SPREADSHEET ACTIVITY

One Laptop Per Child (OLPC) is an organization dedicated to create educational opportunities for the world's poorest children by providing each child with a rugged, low-cost, low-power, connected laptop (XO) with content and software designed for collaborative, joyful, self-empowered learning. As community engineers associated with this unique proposition, we have constantly evolved our programming skills to align ourselves with the mission statement and develop software for educational purposes.

SocialCalc is a spreadsheet activity developed for functioning in the Sugar environment [3], OLPC’s software paradigm. Initially coded by Dan Bricklin, Founder and President of Software Garden Inc. for Socialtext, Inc. [4], the OLPC part was started by Manusheel Gupta, Managing Director of SEETA with K.S. Preeti, community friend of SEETA and alumnus from Netaji Subhas Institute of Technology and Luke Closs from Socialtext Inc. under the guidance of Walter Bender, Oversight Board Member at Sugar Labs.

The main idea of the Spreadsheet activity is to include features that would enable children to make easy use of the typical features of Spreadsheet activities such as organization, graphing and simple calculations in their respective languages. The main features of this spreadsheet activity are:

- Tabulation
- Organization
- Graphing and Calculation
- Localization in different languages
- Multi-user editing over the mesh network
- Ability to read and edit single sheet Excel 1997-2003 (.xls), Lotus (.wk4) and other popular spreadsheet files
- Optimization in saving of sheet data.
- Collaboration over the Cloud
- Chat integration

Over time, SocialCalc has grown to become an innovative platform over which we have experimented on several accounts of collaborative learning. The basic framework of the application is as follows:

- Application – Spreadsheet activity called SocialCalc, written in JavaScript
- Platform – Python, integrated through XOCOM Library
- Infrastructure – XO Laptop and School server
During the course of this development, we also created a library, XOCOM for running DHTML activities in the Sugar development environment [5]. The basic utility of the package is in its ability to help integrate JavaScript codes with Python codes, hence, ensuring a flexible and robust communication between both. We have utilized this library for all functionalities, including Collaboration.

III. TAKING SPREADSHEET TO THE CLOUD

In order to take our Spreadsheet Activity to the cloud, we devised an architecture which enables the application to reside on the school server, common to all the XO laptops. All the systems integrated to the cloud access the application on the browser, while the server handles operations such as saving, etc. Though SocialCalc was ready to be used by individual browsers on their XOs, additional infrastructure was needed in order to support collaboration. Changes were introduced in the Python as well as JavaScript parts with XOCOM acting as the base, to create the infrastructure.

To put into effect this use-case, we first accomplished the same on an established Cloud server, that is, the Google App Engine. The Python code was used for the server side scripting, while the JavaScript code running on the Browser acted as the main activity. This application was named SocialCalcNet.

IV. FEATURES OF SPREADSHEET ON THE CLOUD

A. Login - User login is required to keep record of the sheets that a user creates. SocialCalcNet provides the user an option of logging in either using his Google account or creating his own account with the application. Whenever a user logs in, a new session is created which identifies each user.

B. Account Details - This displays the account details of the user and also provides the options for further actions – new sheet, load a previous sheet, edit account details, etc.

C. Edit Account - Users can edit their accounts created with the application. They can change username or password. However, Google account details cannot be changed.
D. **Socialcalc Sheet** - Users can create new sheets and work on them. A tab by the name Options was added to the application. This tab provides options to the user such as saving the current sheet, opening a previous sheet or logging out of the current account.

E. **Editing a Sheet** - Users can also open previously saved sheets and continue their work on them. The name of the sheet to be opened has to be written in the textbox corresponding to the name.

F. **Interoperability** - Interoperability in SocialCalc with any file format means that files of that format can be read by the application. SocialCalcNet can currently open files with .wk4 and .xls (Excel sheets) extension. Users can upload these files from their system and then view them as SocialCalc sheet. The uploaded files are transferred to the server. The extension of the file uploaded, is retrieved from the file name. SocialCalc stores the sheet data in the form of a string. The data in the wk4 and XLS files is however not in the same format. Therefore to open those files in SocialCalc their contents is first converted to the SocialCalc string.

G. **Collaboration** - Collaboration is the feature wherein various users can work together on the same sheet, thus increasing work efficiency. The feature of collaboration in SocialCalcNet is still in the early development stages. The collaborated sheets are organized by defining the author of the sheet and the group name. These two are used together to identify a sheet uniquely. For secure collaboration, users are asked for a sheet-id that is specified by the sheet author at the time of creation.
SocialCalcNet uses JSON (JavaScript Object Notation) to make these RPC’s [13]. While making a RPC all the arguments at the client side are packed into a string and then sent to the server. The server unpacks these arguments and calls the appropriate routine to handle these requests. The results are again packed in the form of a string and then send to the client.

There are mainly two functions that the application currently calls via RPC –

(i) Sending the changes to the server. Whenever a change in the current sheet is observed it is sent to the server for other users to collect. This change is sent in the form of a command string which is then used by other users to introduce these changes to their sheets as well. For this data transfer a record of all the active users working on a sheet is stored along with the changes sent by each of them.

(ii) Collecting the changes from the server and displaying them on the sheet. For this the application currently uses synchronous checking. The client synchronously keeps on checking for any data submitted at the server by making RPC at fixed intervals. If any new data is found, it is collected and then displayed in the sheet.

V. ARCHITECTURE TO USE SPREADSHEET FRAMEWORK ON THE CLOUD

Once the basic collaboration over the cloud through the Spreadsheet was achieved, we devised architecture for using this Spreadsheet as a chassis for building several other Activities which require the support of a Spreadsheet. The diagram below shows this architecture in brief:

![Proposed Architecture](image_url)

In the architecture that has been proposed, the components which have been implemented in the previous sections have been marked in red. Now, in order to exploit this system for more robust and hard to implement use cases, we have introduced two new components:
(i) **RPC Connection Port:** RPC is a powerful technique for constructing distributed, client-server based applications. It is based on extending the notion of conventional or local procedure calling, so that the called procedure need not exist in the same address space as the calling procedure. The procedures may or may not be on the same system, but the RPC connection allows inter-process communication, without programmers actually coding details for the implementation. In our architecture, we introduced the RPC Connection in order to bring into picture parallelism that would be required by complex systems using this architecture. While each instance of the application would be requiring computation bandwidth and server storage space, it is not possible to do this on a single machine. In such a scenario, it is useful to implement RPC, such that the Server which acts as the RPC Connection port, handles all the computation, while the individual machines logging into the systems simply run the application remotely.

(ii) Also, in figure point systems, while running the data stream on the cloud, there is a threshold limit to the streaming rates. More often than not, this is far less than recommended speeds. In order to increase the speed of performance of a Cloud application, a local cache or mirror of the data and computation is maintained on this RPC Connection server. This comes into picture only when required, so that during the running of the application, only the important computation data is transferred instead of the entire data involved in the process.

(iii) **Auto-Pilot Engine:** Several data in the application requires pre-processing and analytics. Instead of relying on the local machine to do this computation, we use RPC Connection port, in tandem with an auto-pilot engine. This engine, performs group analysis of data, performs various analytics and operations on the data and predictions based on the grouping. This engine also performs the task of synchronizing the received data with the database and eventually, presenting it to the end-user in the right format.

![Figure 5: Spreadsheet on the Cloud](image)

The most important use of these components is in their support to the existing Cloud Spreadsheet. While in the cloud version of the activity, several tasks such as collaboration and computation can be performed on the local machine itself, enhancing it in order to expand it to more complex systems would require support from back-end servers. This is where the RPC connection port and the auto-pilot engine contribute. These components ease the pressure from local machines by taking part in the computation process and transferring the same to the local machine through remote calls. The advantage of this lies in the increased efficiency and speed of the entire process on cloud.
VI. WEB APPLICATIONS BASED ON SPREADSHEET

Once the Spreadsheet is running on the cloud, a combination of CSS and the coding of the Spreadsheet can be utilised in several scenarios to build complex web applications. The basic features such an exercise will utilise are:

- Computational prowess of Spreadsheet
- Design abilities of CSS
- Collaboration power of Spreadsheet
- Inter operability & cloud presence of Spreadsheet

This unique combination helps build customized spreadsheet based websites, which can be utilized to present complex data in simple formats, and at the same time, also be used in the context of collaboration.

One such use case which we have implemented is that of the School Administration system.

VII. BUILDING THE SCHOOL ADMINISTRATION SYSTEM

The Administration system of an educational institution is a complicated system, which requires collaboration on various fronts. It requires students to be able to view their course work, as well as access Administration related modules such as Performance analysis, attendance management and work submission. Similarly, for the teachers, such a system should be an easy access to the Students’ works, course material and administration modules such as Attendance management & performance analysis.

The School Administration system which we have built is based on these aspects. The basic framework is that of the Collaborative Spreadsheet SocialCalc.

The same framework is used to fetch the CSS outline for a design outlay spread over this Spreadsheet.

Such a system enables us to make complete use of the Spreadsheet’s exceptional abilities of computation & collaboration, at the same time, allows us to bring the design element associated with the website.

The various features included for students in the implementation include:

Learning Mandate – This section includes all lessons uploaded by the Teacher for the group. It also includes reminders on pending home-works and submissions. The mandate consists of theoretical chapters as well as specific tests and templates set by the teacher for these chapters.
Fig. 7  Framework for Teacher

Fig. 8  Framework for Student
1. **Attendance Management System** – This section includes the ability to fetch attendance for desired dates & submit leave applications.

2. **Calendar Management System** – This section is updated in real time with the latest happenings in school and the date-sheets regarding exams, results and events.

3. **Performance Management System** – This section consists of the history of records & grades obtained by the student. This is also refreshed in real time as and when the teacher uploads the Marks.

4. **Collaboration** – This feature is present throughout the website, wherein the student can simultaneously learn and collaborate with peers and teachers.

Similarly, such a system has also been designed for the Teachers and Administrators. This system also consists of:

1. **Learning Mandate** - Where the teacher can upload the chapters & tests and assign them to group of students or classes. This section can also be utilised to set the Revision tests along with dates, which are then displayed as pending tasks in the students’ landing pages. The Teacher uses this section to check the works submitted by students and mark them accordingly.

2. **Classroom Management System** – In this section, the teacher has the liberty to share her open spreadsheet with the class and utilise the collaboration tool to teach while working. She can also use this to observe while the students practise. She can also chat with any students who may be facing any problems during the class.

3. **Attendance Management System** - The Teachers can mark the attendances and submit the attendance to the Administrators for final evaluation.

4. **Calendar Management System** – This section is updated in real time with the latest happenings in school and the date-sheets regarding exams, results and events. The Teacher can utilise this section to send reminders to students about pending tasks, and this gets updated in the students’ landing pages.

5. **Performance Management System** – The Teacher uses this section to evaluate the students and submit their Report Cards class wise. This is updated in real time in the Student’s landing pages.

Hence, we can see that the basic Spreadsheet framework has been utilised to produce a web application for the complete school management. In a similar manner, several other applications can be built, which utilise the Spreadsheet on cloud to solve complex problems such as computation & fetching of data.

**VIII. SCENARIOS FOR SPREADSHEET ON THE CLOUD**

While Education and Enterprise are common use case scenarios for using a Spreadsheet activity on the cloud, it is more important to utilise the basic chassis for more complex situations. With the architecture defined above, composite applications, involving multiple users, huge databases and complex computations can be easily implemented. In the course of next section, we discuss two such useful cases, especially tailored for the Indian atmosphere:
(i) Health on Cloud – Implementing a Health record portal for Health professionals as well as patients, to interact and store data on the cloud.

(ii) Administration on Cloud – Implementing an Administration portal for Administrators as well as Consumers to interact and collaborate.

IX. HEALTHCARE ON CLOUD

The power of on-the-go computing applied to healthcare systems has long since become a technology advancement to keep the eyes on. Frost & Sullivan predicts that with healthcare will become a major trend in the coming years [14]. Healthcare providers are increasingly looking at automating processes at lower cost and higher gains; cloud computing can act as an ideal platform in the healthcare IT space. Google Health and Microsoft Health Vault are some examples of how the world is tuning itself to this new phenomenon.

With the architecture we discussed, we believe that the Spreadsheet as a framework can prove to be an extremely reliable background for building a good portal for Healthcare over the Cloud. In essence, this refers to making real time availability of patient and professional’s data, seamless access of data between various hospitals and healthcare communities, ability for patients to upload new data and collaborate with online doctors for referrals, ability for professionals to consult other doctors through collaboration over the cloud by sharing patient data and a lot more. This not only reduces the cost of data storage for Hospitals, but also makes the interaction between professionals and patients seamless. Also, this helps in the standardization of patient data across the globe.

In the Indian context, with rural healthcare becoming a pressing issue with the day, it has become imperative to find solutions which will help organize this extremely unorganized sector. With the Cloud infrastructure in place and Information & Communication Technologies gradually creeping in to rural India, this could be the perfect solution bringing the rural population into the realms of technology. The issue of healthcare professionals not being able to devote time to this sector can also be reduced to a large extent, with collaboration and interactivity between professionals being made possible over the Cloud. Some of the features which we propose to be build into our Healthcare portal, based on the Spreadsheet framework are:

— Cloud interface for storing Healthcare data on the Cloud
— Separate interfaces for Doctors and Patients/Consumers
— A web interface for allowing Consumers to upload data
— A web interface for allowing Doctors to upload data
— Templates with pre-defined fields which can be directly loaded
— All data to be saved in Spreadsheet formats
— Ability to save profiles of patients in folders
— Ability to share files and folders with other Doctors only
— Ability to comment on a diagnosis which has been shared by another doctor
— Ability for Consumers to share their profiles/reports with Doctors they chose
— Collaboration to allow different Doctors to work on same patient simultaneously
— Collaboration to allow Consumers to view online Doctors closest to their location and allow them to chat and take advice
**USE CASE:** Consider a scenario in a rural village in India. With ICT in place, the local health clinic embarks on a drive and after a rigorous exercise organizes all the data related to all residents into Spreadsheet format, with the help of pre-defined templates. This data is stored on the Cloud, the only infrastructure deployed at the site being a basic computer. This data is now shared with all hospitals in the nearest vicinity. Whenever there is a situation when the low level healthcare professional based in the local clinic is unable to make a diagnosis, he simply sends a message to all these connected hospitals and shares the details of the patient. The doctors can now collaborate over the Cloud, chat, give individual diagnosis, which other doctors can comply or overrule and the same is now interpreted to the Local Doctor. In case of emergencies, the nearest Hospitals with available doctors can even send resources.

Consider another scenario, in a city, where ICT has penetrated well. The patient, being well aware of his conditions, has already uploaded the data on the portal. Now, requiring a consultation from a good doctor, he can first contact available doctors through the portal, share his data, await their initial diagnosis and if need be, arrange for an appointment.

In a further scenario, the mobile version of the application being GPS enabled can help in emergency situations, by raising an alarm about a patient and sharing his data immediately with the Hospital in the nearest vicinity and demanding resources.

Hence, with these use cases, we can see that a simple architecture, based a spreadsheet activity can be a step forward to solving the problem of inaccessible healthcare solutions.

**X. ADMINISTRATION ON CLOUD**

In his Senate confirmation hearing in May 2009, Aneesh Chopra, USA’s first chief technology officer (CTO), stated that cloud computing holds a number of advantages for the government. [15] These include “reduced cost, increased storage, higher levels of automation, increased flexibility, and higher levels of employee mobility.” More than anywhere else, India requires adopting these technologies with an immediate urgency in order to cleanse a system continually marred by allegations of corruption and inefficiency.

One such area we identified is the Public Distribution System or the PDS, which is a national program that distributes food grains at subsidized process to the poor in the country. The grains are procured from the Food Corporation of India (FCI) and distributed at the Fair Price Shops spread across the country.[16] Considering the numbers such as 14 million tonnes of grains, and 16 states amongst which these were distributed, it is a mammoth exercise which leads to immense operational problems. [17] For instance, out of the stated grains, only 5 million tonnes of grains actually reached the stipulated population. The rest was unaccounted for. While the Government is taking steps through Biometric systems and over hauling of the mechanism of distribution, it is an estimated huge spend on infrastructure.

With the architecture we defined in the precious sections, we propose a low cost system of administration on the Cloud. Some of the features are:

— Position SocialCalc as a “Collaborative Resource Management Tool” that ensures the proper distribution of grain and keeps real time track of:
  * the Amount of grains credited to each FCS
  * the amount of grain being distributed to consumers
  * the amount of grain that is unused
Collaboration will allow hierarchal supervision of the entire process, reducing dependence on middlemen. Features include:

- Collaboration on same sheet to allow Real Time Tracking
- Chat Enabled Collaboration to allow interaction between Administration

Multiple User entry points to allow all Administrators to Log in from different areas and access the Resource Management Tool

Consumer Portal can allow users to enter their Ration Card number, view past transactions, get updates about next ration availability and amount of grains to be issued in their name

In case a portal for consumers is not accessible, the consumer issues can be sorted by the FCS Officer by simply entering the Ration Card number and checking details.

Administrator of local area can make information available to all Users regarding the Date of new Grain arrival, etc.

Customers can also chat with Administrators to issue complaints or discuss grievances

Issues and Problem identification is easier due to transparency. Any unnatural dealings can be caught by the system and officers can be warned.

**USE CASE:** Consider a rural scenario in India. When the FCI releases the grain, the various Fair Price Shops (FPS) receive the grains [18]. The FPS manager then logs into the system and maintains a log of the amount of grain received and the number of customers he will be catering to. The district administration has access to records of all the FPS logs. Now, the district administrator can create schedules and upload them on the portal, informing both the FPS managers as well as customers about the dates during which the grains will be available. On the stipulated days, the FPS start distributing the grains and the Administrator keeps real time track of the logs.

A certain customer having missed the date, can either log onto the portal himself or through the FPS Officers, to check his previous transactions and intimates the manager about this. The manager also checks past transactions and immediately sanctions a new date for this customer to collect the grains he missed.

In this manner, the administration can be made transparent, accountable and customer friendly. The same system can be replicated for various other Governmental procedures.

**XI. FUTURE OF CUSTOMIZED WEB APPLICATIONS ON THE CLOUD**

Technology in any form is ineffectual, unless transformed to suit the needs of the situation. Cloud computing, in various forms, can be customized to suit the Indian environment, and this paper was our small effort in achieving this. Through the paper, we discussed several scenarios, ranging from Education to Healthcare to Administration on the cloud. However, this is just the beginning. With a sound architecture and framework in place, the spreadsheet on cloud can be used a tool in several other scenarios. As we embark on this exciting journey, we hope to learn more and contribute more in development of web based applications.
REFERENCES


ABSTRACT

A huge amount of data is everyday managed in large organizations in many critical business sectors with the support of spreadsheet applications. The process of elaborating spreadsheet data is often performed in a distributed, collaborative way, where many actors enter data belonging to their local business domain to contribute to a global business view. The manual fusion of such data may lead to errors in copy-paste operations, loss of alignment and coherency due to multiple spreadsheet copies in circulation, as well as loss of data due to broken cross-spreadsheet links. In this paper we describe a methodology, based on a Spreadsheet Composition Platform, which greatly reduces these risks. The proposed platform seamlessly integrates the distributed spreadsheet elaboration, supports the commonly known spreadsheet tools for data processing and helps organizations to adopt a more controlled and secure environment for data fusion.

1 RISKS IN DISTRIBUTED SPREADSHEET COLLABORATIVE DATA PROCESSING

Companies and organizations in many critical business areas base many of their most critical data processing tasks on spreadsheet applications [1][2]. The reasons why this happens have been investigated and researched by the European Spreadsheet Risks Interest Group, as well as many of the potential risks associated with the use of spreadsheets in these processes [3][4].

Some of these risks are due to bad practices or lack of control in the development phase of the spreadsheets by the End Users [5], but many risks are also hidden in the run-time processing of data contained in distributed spreadsheets [6]. In many large organizations spreadsheet-based data processing often takes place in a distributed, collaborative way, where many actors produce or consume spreadsheet-based data and contribute to the setup of a real spreadsheet-based “distributed workflow”.

The propagation and the update of data in this workflow is commonly performed either using the tools provided by spreadsheet applications (like cross-spreadsheet links) or using manual techniques, like copy and paste operations over spreadsheets. Access to spreadsheet data is also not controlled, allowing different methods like concurrent access by different actors to a shared location or circulation of spreadsheets by email. The combinations of these practices may lead to errors and/or inefficiencies, the most common of which can be summarized as follows:

- inconsistent cross-spreadsheet links due to uncontrolled delete/rename/replace operations of spreadsheet files potentially leads to data loss;
proliferation of many copies of data due to spreadsheet circulation by email or other means leads to data replication, lack of control on spreadsheet versions and potential usage of not up-to-date data;
uncontrolled concurrent access to a shared spreadsheet for manual data update leads both to high latency time for update (only one actor at time is allowed to modify data) and to uncontrolled changes in the spreadsheet data;
manual copy/paste operations for propagating data from one spreadsheet to another leads to mistyping errors and violations of cell formats.

In order to provide a more controlled approach to distributed spreadsheet processing, we investigated the application of Data and Services Composition Platform [7] in several scenarios where spreadsheet-based data processing plays a major role. This investigation led both to the definition of an approach for applying data composition to the spreadsheet case and to the development of a prototype platform supporting this methodology. We will discuss why such approach and platform can help to reduce the risks described above and improve distributed spreadsheet processing.

The presentation is organized as follows. Section 2 introduces the proposal through an example. Section 3 describes the platform features and architecture, and provides implementation details. Section 4 discusses other approaches to the problem and presents a comparison with such approaches. Section 5 provides some concluding remarks.

2 INTRODUCTION THROUGH A REAL CASE EXAMPLE

2.1 A Real Case Scenario

A car selling enterprise is organized in a hierarchical structure: a certain number of Car Dealer entities (CDs) located in the same area are managed by an Area Sales Manager entity (ASM). The ASM needs to collect sales information from the CDs in order to evaluate them through a performance index. This performance index is the percentage of completion of a sales target assigned to a particular CD by the ASM: every CD must know only his own target whereas the ASM wants to publish a comparison of CDs based on the performance index (see Figure 1).

We suppose that:

- a Virtual Private Network exists among CDs and ASM, we call it Enterprise Domain Network;
- all the CDs keep track of their sales report on their local spreadsheets;
- the ASM calculates performance indexes using his own complex local spreadsheet;
- the ASM publishes the comparison to the entire enterprise through a new spreadsheet.

Every CD can view the performance index based comparison in any moment and it should be updated to the last car sold by any CD: when a CD insert a new sold car in its local spreadsheet, its performance index must change immediately, even though the ASM is not working on its spreadsheet.
This scenario identifies some problems the cars selling enterprise must face:

- the CDs and the ASM are widely distributed and potentially large in number, therefore there is a strong need for an automatic way of syncing information among them: the CDs should export only the information needed by the ASM in an easy way, avoiding to transfer their entire spreadsheet or communicate changes manually (via email and/or shared storage resources);
- the information exchanged between each CD and the ASM must not be viewed by the other CDs, only the final comparison is visible to everyone, therefore a permission manager system is needed;
- the three levels in Figure 1 form a “distributed spreadsheet chain”, since spreadsheets belonging to different levels and actors are linked through shared data, which must be kept always updated;
- every element in the distributed spreadsheet chain should not be aware of belonging to that chain, but it should work as a stand-alone unit relying on a mechanism automatically keeping the chain alive.

### 2.2 Requirements for a Spreadsheet Composition Approach

From a generalization of such a scenario we derive the following requirements for a distributed spreadsheet composition approach and for the supporting ICT tools/platforms:

- support of End-User Computing: being widely used and well known by end users, spreadsheet-based tools can be used as enablers for the creation of Distributed Spreadsheet Compositions, so that no programming skills are required for the composition to take place;
- support of reuse of already existing data: enterprise users often use spreadsheets to store information and/or data extracted from other company’s software systems (e.g., ERP system). Thus it is important to provide a tool that allows to combine data belonging to different spreadsheets easily and friendly;
- compliance with the distributed and hierarchical structure of enterprises: the platform is supposed to manage the case in which the spreadsheets belonging to a composition are stored in different machines (e.g., different units of the same company may be geographically distributed). Moreover, the Spreadsheet Composition platform must be aware of the different roles of the employees inside the company in order to apply the correct access rights according to the hierarchical structure of the company;
• support of automatic updates and “Always-on” distributed Composite Spreadsheets: the maintenance of complex relationships over distributed spreadsheet composition is necessary to preserve data consistency. In particular the system is supposed to synchronize the linked spreadsheets automatically even when one or more components of the distributed spreadsheet are offline.

3 NOVELTY OF THE APPROACH: PLATFORM FEATURES AND BENEFITS

3.1 Approach and Platform Features

The proposed approach to distributed spreadsheet data composition is supported by an IT platform named DISCOM (Distributed Spreadsheet COMposition) based on a client-server architecture: one or more clients interact with the server platform across a network, which can be either an Intranet, a VPN or the Internet.

• The client part is a Plug-in module integrated in the client’s local spreadsheet application. The module interacts with the composition platform by means of Web Services and allows the user to configure itself and make exportations and importations of spreadsheet cells by means of a graphical interface.

• The DISCOM platform exposes interfaces that support the execution of spreadsheet data exportation and importation, manage user accounts and synchronize data across dependent.

Figure 2 illustrates the system operation. Spreadsheet A exports a cell range (dotted area) and Spreadsheet B imports the same range. In the same way, Spreadsheet B can export a range (dashed area) that is imported by Spreadsheet C and so forth. In the above example, Spreadsheet A is an Exporter, Spreadsheet C is an Importer, and Spreadsheet B is both an importer and an exporter.

![Figure 2: Spreadsheet composition operation.](image)

Data propagation is implemented through the following mechanisms:

• the exporter contribution update is periodically performed by each exporter Plug-in towards the platform. The Plug-in checks the spreadsheet for exported range modification and refreshes the range image on the platform. In this way, the platform always contains the latest update of the exported ranges. If the user computer is offline, the Plug-in caches the modifications, and exports them to the platform as soon as possible;
• the importer spreadsheet update is periodically performed by the importer Plug-in. The Plug-in periodically polls the platform for new contributions or updates. As soon as a new contribution or update is detected the Plug-in imports the contribution and inserts it in the spreadsheet. In this way, the spreadsheet always contains the latest update and updates are propagated in real time.

If a user spreadsheet is both importer and exporter, and at least an exported cell is a function of an imported cell (via direct inclusion or formulas) we call it an Intermediate spreadsheet (in our Figure 2 example, Spreadsheet B is an intermediate spreadsheet via direct inclusion of the dashed range in the dotted range). Intermediate spreadsheets allow building spreadsheet chains.

As we assume that a user personal computer can be switched-off in every moment, a scenario that includes one or more intermediate spreadsheets presents a crucial issue: if an intermediate spreadsheet computer is offline (e.g., switched off or not connected to the network), the data updates cannot propagate through the chain. In order to overcome this issue, the system should support a feature, namely the data propagation function, which works as follows:

• when a client Plug-in realizes that the spreadsheet is intermediate it uploads the entire spreadsheet on the platform. Every time an intermediate spreadsheet is updated the Plug-in performs a new upload;
• when an intermediate spreadsheet is offline, the platform runs a local spreadsheet engine in order to recalculate the exported ranges based on fresh import ranges.

In the example of Figure 2, Spreadsheet B is an intermediate spreadsheet and is therefore uploaded on the platform in order to assure data updates from A to C.

The visibility of data exported to the platform can be controlled, allowing data to be exported to all users or part of them. In order to provide more flexible functions to control data lifecycle and visibility, the concept of Space has been introduced. A Space is a collection of platform users, one of them being the Space Creator, while the others are either data exporters or data importers.
The Space is a dynamic entity: a Space can exist on the platform to manage a spreadsheet based collaboration carried on by different actors in a limited time period (e.g., “2011 First Quarter Balance Assessment”). A space can also be created to manage spreadsheet based collaborations for a limited set of actors belonging to a well defined unit in an organization.

3.2 Platform architecture and implementation

The platform architecture is depicted in Figure 4: the two main platform components are the Plug-in Client and the DISCOM Platform. We adopted an architecture based on a central composition platform in order to provide a central repository that performs automatic synchronization even when one or more contributors are offline.

![Figure 4: Platform architecture.](image)

**Client Plug-In**

The Plug-in Client is written in C# using Visual Studio Tool for Office [8], developed for 2003 and 2007 Microsoft Excel versions. It provides a graphical interface that allows users to control the import/export operations; the interface maintains the same look and feel as the other Microsoft Excel functionalities, with respect to different versions of UI layout (command bar or ribbon interface). The main design objective was to integrate it as seamlessly as possible in Excel applications, letting users concentrate on normal Excel commands without worrying about other tools.

The Plug-in interacts with the DISCOM Platform through Web Services technology [9], as Web Services are widely developed and supported in enterprise environments. In particular we adopt standard SOAP messaging. The cell range to be exported is converted to a custom XML document and wrapped in a SOAP request which is sent to the DISCOM Platform. The DISCOM Platform module stores the XML translation of the exported cells in the database. Additionally, the Plug-in stores metadata locally on the client machine:

- exportations and importations information are kept in the spreadsheet, inside file custom properties;
- user identification, server address and other plug-in configuration information are kept in user’s local space on the filesystem.

In order to perform the data propagation function, if the Plug-in realizes that the spreadsheet contains an automatic exportation dependent on an automatic importation, it
uploads the entire spreadsheet on the platform. Otherwise, only cell ranges are exchanged.

**Server Side Platform**

The server side of the platform is written in Java, using Spring framework [10] as the engine for managing Web Services and reacting to HTTP and SOAP requests. The main interface exposed to the Client Plug-In is a Web Service which provides a set of functions that allow clients to authenticate themselves and perform CRUD (Create, Read, Update, Delete) operations on exported and imported data. It serves client SOAP requests by invoking Spring beans interacting with a MySQL database [11] in a transactional environment.

The data propagation function is realized by a Spring bean which perform this sequence of operations:

1) detection of the intermediate uploaded file to be re-evaluated;
2) insertion of updated values in its automatic importations;
3) recalculation of cell formulas;
4) update of automatic exportations (those modified by the recalculation).

Every step is performed using Apache POI library [12], a set of objects able to open and manipulate Microsoft Excel file directly within Java code.

For administration purposes a web application is provided by the platform. Through this component an administrator can manage user accounts and control the platform correctly working.

The platform can be deployed in two different ways:

- as an enterprise service, creating a “ad-hoc” server inside an enterprise domain to give the enterprise full control over the system and limit spreadsheet data circulation within the enterprise domain;
- as a cloud service hosted on a IaaS (Infrastructure-as-a-Service) provider like Amazon Machine Image – AMI on the Amazon EC2 Infrastructure [13], to enable an easier and quicker setup for spreadsheet compositions which don’t need strong control and data flowing-over-the-internet limitations.

### 3.3 The platform at work

In this section we explain how the DISCOM platform can solve the issues presented above using the car selling enterprise scenario presented in section 2.1 as a reference scenario.

Each Car Dealer (CD) and the Area Sales Manager (ASM) are requested to install the plug-in in their Microsoft Excel local applications and should have access to the platform installed in the enterprise domain (or to the platform located in the Cloud).

All the CDs agree with ASM to structure the spreadsheet information using a common format. As an example we assume that the format is 4 columns containing car model, number of sales, average price and total income. Figure 5 depicts the export operation: every CD, through the plug-in interface, assigns its data a name and a description and makes such data available only to the ASM.
Figure 5: Cell range exportation. With reference to the car selling enterprise scenario presented in section 2.1, car dealer John Smith is exporting Oct. 2010 sales figures to ASM Carl Black.

The ASM can import the CD contributions using the plug-in interface and choosing among the accessible data (see Figure 6). Then it sets the rules (spreadsheet functions) to calculate the performance indexes starting from the imported data, without worrying about the actual data availability. All the CD exportations and the ASM importations are performed automatically, so the plug-in and the platform work in background to keep the data constantly updated to the last changes made by the CDs. To publish the final CDs comparison based upon the calculated performance indexes, the ASM exports it the same way as the CDs do, with the difference that in this case it makes the data available to the entire Space, which represents its area and includes itself and the CDs.

Figure 6: Cell range importation. With reference to the car selling enterprise scenario presented in section 2.1, ASM Carl Black is importing sales figures from three car dealers.

Since the ASM local spreadsheet is an intermediate spreadsheet (see section 3.1), the platform requires ASM to upload an entire copy of it in order to keep the distributed spreadsheet chain (from CDs exported data to final comparison) always updated. Therefore any CD who wants to view the comparison always views the latest performance indexes even if the ASM is not working on its spreadsheet: the platform has its own copy, knows its dependencies (importations from CDs) and can recalculate it in any moment, publishing the updated final comparison in place of the ASM.
4 COMPARISON WITH OTHER APPROACHES AND TECHNOLOGIES

4.1 Other approaches and technologies for Distributed Spreadsheet Composition

Several approaches and solutions are already available for Distributed Spreadsheet Composition: in particular, we will focus on Google Docs [14], on Microsoft SharePoint [15], and on the Data Mashup approaches.

Google Docs is a collaborative tool allowing the sharing of document files among users [14]. It is implemented following the Software-as-a-Service, SaaS, Cloud Computing model according to which users can manipulate spreadsheet files by accessing a Web Application instead of installing software like Microsoft Excel, Open Office, etc. on their own pc. With Google Spreadsheets, users can create, edit, and share spreadsheets without taking care about how and where these files are located because they are stored “in-the-cloud”. Moreover, this tool allows users to work simultaneously on the same spreadsheet (i.e., colored cursors will be assigned to the users where each color represents a different user working on the file). Finally, Google Spreadsheets presents the “Cross-workbook references” feature which enables the interconnection among spreadsheets to create the chain of spreadsheets described in Section 3.

One of the drawbacks related to the usage of this tool is that users are supposed to learn how to use a new tool that, even if it is very similar to the one they are already able to use (e.g., MS Excel), it is slightly different (on the contrary the system described in section 3 represents just a new MS Excel feature to the user who can continue using the same tool). Since Google Spreadsheets provides fewer features compared to MS Excel, there might be some compatibility issues between existing MS Excel-created files and the Google Spreadsheets counterpart. A similar tool is proposed in Microsoft Office Web Apps [16] that extends the basic functionalities of Microsoft Office suite products (like the spreadsheet software Excel) to the cloud. In particular, the interaction model is similar to the one provided by Google Docs, allowing multiple users to edit an online spreadsheet using the browser.

Microsoft SharePoint is a suite of enterprise collaboration products, developed by means of the integration of SaaS technologies and client software, providing a large set of
functionalities such as website visual composition, enterprise social networking, document sharing, content management, advanced search support and business intelligence applications. In particular, Microsoft SharePoint offers a *Excel Service*, that allows users to publish an Excel spreadsheet on a SharePoint server, thus allowing the sharing of the spreadsheet among user in a workgroup. The spreadsheet can be read and modified using Microsoft Excel software, using a browser and a web application as in Microsoft Office Apps service, or modified by means of Web Service calls.

Spreadsheet hosted in Google or Microsoft services can be accessed also through a set of APIs [17], thus allowing the creation and the maintenance of spreadsheets in a programmatic way (i.e., a programmer can develop an application that retrieves and/or updates cells’ values without any intervention by human beings). This feature can be used to exploit the system as the technology standing behind the server-side platform introduced in section 3 because it might cover the role of the component storing the exportation/importation information as well as keeping the chains updated thanks to the “Cross-workbook references” feature. Although this approach is very similar to the one proposed in this paper, it shows the following drawbacks:

- it lacks flexibility because the functionalities provided by the server-side component are limited to those provided by the platform, whereas the server-side component described in section 3 has been developed from the scratch thus it can be easily extended;
- it provides a rigid scheme for security management (see[17]Error! Reference source not found.) which represents one of the most important features of the system described in section 3.

Another approach proposed in the literature for the composition of spreadsheets is that of Data Mashups [7] (i.e., applications combining information retrieved from mixed sources and possibly provided in mixed formats). The distinctive feature of products like Yahoo!Pipes [18], Apatar [19], etc. is their focus on the management of different data formats (e.g., spreadsheets, databases, XML files, CSV files, plain text, etc.) and not on the creation/maintenance of distributed spreadsheets continuously kept up to date. Some of the tools supporting Data Mashups might provide a spreadsheet-based user interface in order to exploit the well known paradigm based on grids and cells which is widely used by people all around the world (see [6] for a survey of the available solutions). Although Data Mashup platforms might be used to solve (some of) the issues described in section 1, they have been designed for other purposes (e.g., merging data provided by means of different formats) thus they are not optimized to manage the composition of spreadsheets. In addition Data Mashup tools often require the installation of a new software on the user’s pc thus there is - again - the problem related to the fact that a user is supposed to learn how to use a new tool. Finally, the most of the available Mashup tools don’t provide capabilities supporting the security thus data stored in spreadsheets might be stolen or changed by malicious behaviors.

### 4.2 Benefits and comparison with other approaches

When compared with the traditional practices for distributed spreadsheet composition, where spreadsheet data composition takes place with almost manual techniques like cross-spreadsheet cell references, manual copy-paste operations, email spreadsheet circulation or single shared spreadsheet location, the proposed approach introduces a different way to compose spreadsheets, removing some or all of the limits and risks which affect the traditional practices.

In particular:
• there is no need for storing spreadsheets in a shared location: after data have been exported to the composition platform, these are available for importation for every user of the platform which is allowed to do it;

• data importations are performed with the assistance of the specific platform Plug-in component, reducing the risks for incorrect copy/paste operations;

• the platform eliminates the need for spreadsheet circulation by email or other means, since data are stored in a centralized server and kept up to date by the platform: whenever a user updates the data in a spreadsheet area which has been “exported” to the platform, these are pushed to the platform server in an automatic way;

• only a single (centralized) copy of the data is maintained, thus eliminating the risk of multiple data sources;

• moreover, the data are updated by the platform even when one of the actor participating to the data workflow is not available (e.g. he/she does not have Excel running on her desktop).

When compared with other approaches to spreadsheet composition, the proposed platform exhibits the interesting following features:

• the platform supports spreadsheet based data composition and aggregation in a hierarchical and distributed way, which is compliant with the usual collaboration practices for distributed spreadsheet management in a complex organization or enterprise;

• the target of the composition is itself a spreadsheet, within which data are kept up to date in real time thanks to the tools provided by the platform, in a transparent way for the end user;

• data updates take place even whenever one of the actor participating in the distributed spreadsheet elaboration process is not available or is not running the spreadsheet application from his desktop;

• no specific software programming skills are required for adopting the approach, since data composition takes place through the usage of the well known tools and methods provided by the traditional spreadsheet editing tools: the proposed approach encourages “end user computing”, while providing at the same time a controlled environment when it can take place;

• the proposed platform easily integrates with the usual tools for spreadsheet management and at the same time extends them for supporting collaborative and distributed processes in spreadsheet elaboration in a non invasive way within an organization or industry.

5 CONCLUSIONS AND FUTURE DEVELOPMENTS

The DISCOM Platform is an ongoing project: platform and methodology on-the-field trials and experimentations are currently going on in order to evaluate the platform features and usability of the approach. At the moment, an experimental prototype has been installed and is currently used at the Regional Branch of the ICT Department of the Italian Ministry of Justice (CISIA), in order to assess and enhance the platform features and to evaluate the usability of the approach. Several improvements are already under investigation, among which it’s worth mentioning:

Data Modeling and Validation: thanks to the fact the spreadsheet data are internally represented as XML documents on the platform server, it will be possible to exploit XML Schema Validation [20] in order to provide modeling tools for server side compliance enforcement every time data are exported from the spreadsheet to the platform. This feature will help to greatly reduce format and value errors in spreadsheet cells.
**Data versioning**: as a future work the platform will be equipped with a versioning mechanism, in order to maintain data exportation changes history. Users will be allowed to browse different versions during the importation process.

**ECM integration**: since we are aware that many complex organization runs Enterprise Content Management systems, future research direction will investigate how the proposed approach can fit into the current way ECM are used within the enterprises, and also how the proposed platform will contribute to the ECM features or will benefit from a strong server side integration with them.

More information on the DISCOM platform is available at the address [http://www.m3s.it/discom](http://www.m3s.it/discom).

6 REFERENCES


[10] [http://www.springsource.org/](http://www.springsource.org/) 12.00 pm 3/30/11

[11] [http://www.mysql.it/](http://www.mysql.it/) 12.00 pm 3/30/11

[12] [http://poi.apache.org/](http://poi.apache.org/) 12.00 pm 3/30/11


[14] [https://docs.google.com/](https://docs.google.com/) 12.00 pm 3/30/11


[17] [http://code.google.com/more/](http://code.google.com/more/) 12.00 pm 3/30/11

[18] [http://pipes.yahoo.com/](http://pipes.yahoo.com/) 12.00 pm 3/30/11

[19] [http://wapater.com/](http://wapater.com/) 12.00 pm 3/30/11

[20] [http://www.w3.org/XML/Schema](http://www.w3.org/XML/Schema) 12.00 pm 3/30/11
Breviz: Visualizing Spreadsheets using Dataflow Diagrams

Felienne Hermans, Martin Pinzger, Arie van Deursen
Delft University of Technology, Mekelweg 4, Delft, The Netherlands
{f.f.j.hermans,m.pinzger,arie.vandeursen}@tudelft.nl

ABSTRACT

Spreadsheets are used extensively in industry, often for business critical purposes. In previous work we have analyzed the information needs of spreadsheet professionals and addressed their need for support with the transition of a spreadsheet to a colleague with the generation of data flow diagrams. In this paper we describe the application of these data flow diagrams for the purpose of understanding a spreadsheet with three example cases. We furthermore suggest an additional application of the data flow diagrams: the assessment of the quality of the spreadsheet’s design.

1. INTRODUCTION

Spreadsheets are used widely in industry, for all kinds of tasks, like financial modeling, reporting and planning [Rittweger, 2010]. A study from the year 2005 shows about 23 million American workers use spreadsheets, which amounts to about 30% of the workforce [Scaffidi, 2005].

Many of the spreadsheet used are of great importance to companies, Hall [Hall, 1996] interviewed 106 spreadsheet developers and found that only 7% of the spreadsheets were of low importance and that as much as 39% were of high importance. In a recent study we found similar results [Hermans, 2011].

In that study we furthermore investigated the most prevalent problems spreadsheet users have in their daily work with spreadsheets, by interviewing 27 spreadsheet users working at Robeco, a Dutch investment bank. The results showed that problems arose when spreadsheets are transferred from one employee to another. As spreadsheets have an average lifetime of more than five years, and individual spreadsheets are used by 13 different employees, this happens quite frequently. We identified three different transfer scenarios: from one employee to another, from an employee to IT, and from an employee to an auditor [Hermans, 2011].

In the case of a transfer, the receiving party often has to spend hours browsing through
the spreadsheet to understand its structure and purpose, since only one third of the spreadsheets contains documentation. The transferring party often feels the spreadsheet is not very complicated, and does not spend enough time to explain it. We have addressed this problem by creating a data flow diagram visualization that can be used during a transfer scenario to support the explanation of the spreadsheet.

When the tool was finished, it was installed at Robeco, so employees could use it when needed. When we analyzed the use of Breviz at Robeco, we found that it was not only useful in the transfer scenarios, but also for individual comprehension: when a spreadsheet user analyzes a spreadsheet by themselves. This scenario occurs when it is not possible to ask the creator of the spreadsheet for advice, for instance when he left the company, or is on a holiday.

In this paper we briefly describe our data flow diagram generation approach, and explain how it can be useful for the individual comprehension of a spreadsheet, with three examples, based on our experiences in practice. We also discuss additional uses of our approach, for the identification of errors in the spreadsheet.

The remainder of this paper is structured as follows: Section 2 briefly explains the algorithm for the extraction of data flow diagram from a spreadsheet. Section 3 describes the three different views we support on those diagrams. In Section 4 the implementation of our approach into a tool—Breviz—is described. Section 5 subsequently explains the applicability of our approach with three practical examples. Section 6 discusses new applications of Breviz. Concluding remarks finally can be found in Section 7.
2. DATA FLOW DIAGRAMS CREATION

This section briefly explains how we extract a data flow diagram from a spreadsheet. For a more detailed description we refer to previous work [Hermans, 2011]. Dataflow diagrams---or similar techniques for representing the flow within systems, such as flowcharts---have been present in literature since the seventies [Gane, 1977].

The extraction of a data flow diagram from a spreadsheet is done in six steps, as illustrated by Figure 1.

The first two steps originate from our earlier work extracting class diagrams from spreadsheets [Hermans, 2010]. The first step determines the cell type of all cells in the spreadsheet, which can be Data, Formula, Label or Empty. The second step identifies data blocks within a spreadsheet. Data blocks are rectangles of non-empty cells in a worksheet, separated from other data blocks by empty cells.

In the third step, labels describing Data and Formula cells are computed. This is done by inspecting the borders of the data block the cell lies in. The fourth step generates an initial dataflow diagram by creating entities for cells of type Data and Formula and creating arrows between them, corresponding to formula dependencies. Subsequently, in the fifth step, the labels of cells that were computed in step 3 are attached to the corresponding entities in the diagram. The final step adds the levels to the dataflow diagram. A level is introduced for each worksheet within the spreadsheet and for each data block within every worksheet. Again we refer to our previous paper for the specifics of the data flow diagram generation.

---

Figure 1 Overview of the data flow diagram extraction algorithm
3. DATA FLOW VIEWS

We support three different views on to the data flow diagram, to help users in navigating them. Examples of these views will be provided in Section 5.

The first view we support is the global view. This view shows all worksheets within the spreadsheet and the relations between them. An arrow from worksheet A to worksheet B indicates a formula in worksheet B refers to a cell in worksheet A. Multiple arrows are grouped into one, so the thicker the arrow is, the more formulas reference cells of another worksheet. The second view is the worksheet view, which shows all data blocks within a worksheet, as well as the dependencies between them. The view is obtained from the global view by expanding a level representing a worksheet. In this way, details of a worksheet are revealed while keeping the overall picture of the spreadsheet. Finally there is the formula view, where the relation between formulas and the cells they depend on is shown. To obtain this view the user opens a data block-node in the worksheet view, showing all calculations in the data block.

4. BREVIZ

Our initial prototype GyroSAT created the data flow diagram by generating a DGML file. The DGML (Directed Graph Markup Language) file format is an XML schema for hierarchical directed graphs that can be viewed with the graph browser that is part of Microsoft Visual Studio 2010 Ultimate. It is intended to visualize software architectures and dependency graphs for systems.

Because we did not have to implement a graph viewer ourselves, the use of DGML enabled us to create a prototype quickly. It however led to problems when we wanted to use the data flow diagram generation in an industrial setting, since not all spreadsheet users have Visual Studio installed, and were thus unable to work with the graphs we generated.

We decided we needed to create a standalone version of GyroSAT, so spreadsheet users could install it on their machine, to visualize the spreadsheets they work with. Therefore we adapted our implementation, and started to use the YFiles Graph Library for Windows Presentation Foundation for the graph visualization part. Breviz was born. We chose YFiles (available from yworks.com) since it supports hierarchical graphs out of the box, and provides great support for customizable user interaction.

Figure 2 shows a screen shot of the global view of a data flow diagram in Breviz.
5. EXAMPLES

While originally created to support spreadsheet users in transfer scenarios, we have seen that Breviz is also useful for scenarios in which a spreadsheet user analyses the spreadsheet by itself. During an industrial case study at Dutch investment banker Robeco, we have found that the data flow diagrams really support users in understanding a spreadsheet. As one of the participants there stated “this diagram (the global view was meant) shows me the idea behind the spreadsheet.” Unfortunately we cannot describe the details of spreadsheets from the industrial case here, since they are confidential.

We therefore illustrate the usefulness of our approach by describing three cases that are loosely based on our experience in practice. Each of the three cases describes the application of one of the three views: global view, worksheet view and detailed view. The spreadsheets can be found on our website [http://swerl.tudelft.nl/bin/view/FelienneHermans/Publications](http://swerl.tudelft.nl/bin/view/FelienneHermans/Publications)
5.1 Global view: Exam

This spreadsheet is used by a university professor to calculate the grades for a course. It consists of six worksheets. From the global view, shown in Figure 3, some aspects of the spreadsheet immediately catch the eye. For instance, one of the worksheets ‘lab-osiris’ is not connected to the other sheets, this sheet contains the data from Osiris, the university’s grading system. This information can be crucial when working with the spreadsheet, since someone working with the spreadsheet might mistakenly think that all scores are updated when updating the information form Osiris. To determine this without Breviz would require the user to select all cells for all worksheets one by one and checking their dependents.

Furthermore the 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.

Besides helping to identify unexpected links between worksheets, the global view can serve as documentation for the spreadsheet. In the case study at Robeco we have seen that users paste an image of the global view in the spreadsheet to document it.
5.2 Worksheet view: Income statement

The second example is a worksheet that describes an income statement for a company. The worksheet is divided into a number of data blocks, but it is not immediately clear how they are related to each other.

Understanding the relation between the data blocks however contributes to understanding the worksheet as a whole. Figure 4 shows the worksheet view for this worksheet. From this diagram a spreadsheet user can immediately see how the data blocks are related. The diagram shows the ordering of the data blocks in the calculation. For instance, ‘net sales’ and ‘cost of sales’ are at the same level in the calculation, while in the worksheet one is located above the other, possibly confusing the spreadsheet user.

Since Breviz gathers names for the data blocks, the user can also see---to a certain extent---whether the relations between make sense.

![Figure 4 Worksheet view for the Income example](image)

5.3 Formula view: Financial Performance

In the third example the financial performance of a company is calculated. This calculation is quite complicated, and to get an overview of what is calculated, spreadsheet users usually click all the cells to view the formulas. Breviz’s formula view is meant to support the user in understanding the calculation easier. Figure 5 shows the formula view for this example.
Without the need to inspect all formulas, spreadsheet users can see what exactly is calculated in the formulas. Breviz users were very satisfied with the help of the formula view. Many of them stated that it saves them a huge amount of time when analyzing a spreadsheet.

6. DISCUSSION

Section 5 gives an overview of the applicability of our approach for understanding and documenting spreadsheets. In this section we describe an additional application of Breviz we envision, but that has not been evaluated thoroughly in practice, namely the use of Breviz diagrams to detect anomalies in a spreadsheet.

As shown in Section 5.1 sometimes the data flow diagram gives rise to questions about the structure of the spreadsheet. This especially is the case in the global view, since we could see that level as the architecture of the spreadsheet, so it is logical to also analyze the quality of that architecture, as is common with software architecture.

Here we list a number of those spreadsheet structure smells of which we suspect that they might indicate errors. As stated before, this idea still needs more attention and a subsequent empirical evaluation.

- A loop between two worksheets: Data is going back and forth between two worksheets. This might indicate that the spreadsheet is not structured in a logical way, increasing the change of errors.

- A single arrow ‘against the stream’: All worksheets are connected is a single direction, but there is one link in the opposite direction. The one opposite dependency could be an error, or a special case that needs additional attention

- Worksheets that are disconnected from the data flow diagram. Users can mistakenly think the data in the disconnected sheet is taken into account in calculations in the other worksheets.

- Two worksheets that are very heavily coupled: the structure could possibly be improved by merging the worksheets. On a side note: merging heavily connected worksheets often has a positive impact on the performance of the spreadsheet, since worksheets with many connections tend to make the spreadsheet slow.
The further investigation of the effect of these spreadsheet smells is an interesting avenue for future research.

7. CONCLUSION

The goal of this paper is to explain how the technique of extracting a data flow diagram from a spreadsheet, which was created to support spreadsheet users in transfer scenarios, can be useful to help individual spreadsheet users to understand a spreadsheet quicker and better.

The current research gives rises to several directions for future work. Firstly the application of data flow diagrams for individual users could be refined. It might be the case that individual users have different information needs, so it would be logical to perform a new round of interviews in which we gather those information needs.

Furthermore the use of data flow diagrams to assess the quality of spreadsheets and spreadsheet design deserves more attention.

REFERENCES


DRIVERS OF
THE COST OF SPREADSHEET AUDIT

David Colver
Director, Operis Group PLC
Dcolver@operis.com

ABSTRACT

A review of 75 formal audit assignments shows that the effort taken to identify defects in financial models taken from the domain of limited recourse (project) finance is uncorrelated with common measures of the physical characteristics of the spreadsheets concerned.

1 THIS PAPER

Croll [1] has described a “a spreadsheet model audit process typical of that presently used in the City [ie the financial services sector] of London”. He asserted that “Discoverable model characteristics such as formula length, ratio of original to repeated cells, numbers of cell precedents and dependents and the locality and non-locality of cell linkages can be used to infer information about the relative ease or difficulty and time to review a given model.”

It makes intuitive sense that a ten-cell trivium should be quicker to check than a multi-megabyte monster. At a finer-grained level, however, this paper offers evidence that widely used metrics are in fact poor guides to the time taken to review a given model.

2 BACKGROUND

Project finance is an informal term for making loans on terms which include agreement that opportunities for recourse in the event of default will be limited to the assets being financed. Developed initially in the natural resources industry, project finance has in the last decade been applied increasingly to the funding of infrastructure projects, such as airports, roads, bridges, power stations, prisons, hospitals and schools. Of the projects identified by InfraNews, a news service specialising in the subject, as reaching financial close around the world in 2009, a role in over half was performed by Operis, a specialist in project finance based in London.

Decisions whether to provide more general corporate finance are typically made on the basis of financial statements. Those essentially record what has happened in the past. What is distinctive about project finance is the centrality in the credit evaluation process of projections of financial performance in the future. The projections are derived from financial models.

Operis develops these financial models, both as a stand-alone service and as part of a wider remit as financial adviser, and provides training in the development of models of this kind. It is best known for conducting formal audits of models that others have prepared. It also sells software (OAK, the Operis Analysis Kit) relevant to these activities.
Operis is therefore exposed to the issues surrounding the auditing of spreadsheets used to price, structure and illustrate substantial transactions on a significant scale. The objective of a full formal audit is to become sufficiently confident that material defects have been removed from a spreadsheet to deliver a letter, addressed to the prospective lenders, that they can rely on the figures presented to them. The liability cover provided if such an opinion later turns out to be ill-founded can run to the tens of millions of pounds.

As part of its continuous quest for improved productivity in this activity, Operis records the effort it takes to complete these reviews in a time recording system, custom designed to track not only to which projects hours are devoted, but how they are accounted for by the different tasks on the workplan. Tasks that, over a number of assignments, consistently take longer than expected from standard measures are candidates for investigation, which might lead to improved training, partial automation of the task or process reengineering aimed at doing away with the step altogether.

3 DATA

Operis has analysed the data recorded on its timesheets for the audits it has recently conducted. Operis has then correlated the data with the value of the transaction, as a proxy for deal size, and various measures of the size of spreadsheet.

- How large the document is in megabytes
- How many worksheets it contains*
- How many unique formulae are involved*
- The maximum and average complexities of the formulae, measured by the number of operators and functions they contain.*

The starred items were extracted using Operis’s software product, OAK v4.

The transaction values are exact for projects that were structured in Euros. They are approximate for projects involving other currencies, as they were converted to Euros at the rates prevailing at the time of writing this paper rather than the rate that actually applied when the project reached financial close; but it still gives some measure of the scale of the asset being modeled.

Also included in the database is the number of different versions, or iterations, of the spreadsheet that were reviewed. The significance of this detail is explained in section 6, Interpretation.

The database used for the analysis covers 75 assignments, concerning spreadsheets responsible for the structuring of over €54bn of financing\(^1\). Attention has been confined to audits completed in recent months, so that they are reasonably consistent in terms of audit process as it evolves in light of experience over time.

4 ADJUSTMENTS

It is necessary to exclude some samples from the database for a variety of reasons.

- Some models have been audited already, and are submitted for re-examination because they have been adjusted to reflect some change in the deal. This can happen several times.

\(^1\) €54bn is the simple total of the financings represented in each spreadsheet. A small number of transactions, including the €14bn largest, are represented by two or more models in the sample.
times in the life cycle of a transaction. The second and any subsequent inspections are
artificially short as they can reuse much of the earlier work.

• Sometimes a standard template model is applied to a series of similar transactions. Again, the economics of the later audits are flattered by the ability to reuse earlier work.

• Some assignments are for a smaller scope of work than a full formal audit. Operis terms such exercises High Level Reviews.

• Some assignments ended prematurely because the initiative to which they were directed collapsed or was cancelled.

It also turns out that a fair proportion of the spreadsheets were provided in the format used by Excel 2007. This is a recent development, Excel 2007 having achieved widespread adoption only slowly. However, on the evidence of this sample, the migration is now well under way. With Excel 2007 Microsoft introduced a new file format that uses compression to store spreadsheets more compactly, a manoeuvre that makes sense given that Excel 2007 permits spreadsheets to be much larger. Since Excel 2003 and 2007 file sizes are not directly comparable, all the Excel 2003 workbooks have been re-saved in the Excel 2007 xlsx format, and it is that measure that is used as an indication of file size.

5 ANALYSIS

The relationship between the items listed above as logged for each assignment is set out below, as measured by R-squared.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>0.02</td>
<td>0.00</td>
<td>0.00</td>
<td>0.01</td>
<td>0.03</td>
<td>0.02</td>
<td>A</td>
</tr>
<tr>
<td>1.00</td>
<td>0.00</td>
<td>0.07</td>
<td>0.01</td>
<td>0.03</td>
<td>0.01</td>
<td>B</td>
<td>Document size (Mb)</td>
</tr>
<tr>
<td>1.00</td>
<td>0.04</td>
<td>0.17</td>
<td>0.02</td>
<td>0.01</td>
<td>C</td>
<td>Unique formulae</td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td>0.27</td>
<td>0.01</td>
<td>0.08</td>
<td>D</td>
<td>Average complexity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td>0.01</td>
<td>0.04</td>
<td>E</td>
<td>Maximum complexity</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td>F</td>
<td>0.27</td>
<td>0.27</td>
<td>Number of Iterations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td>G</td>
<td>0.00</td>
<td></td>
<td>Hours logged</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

All R-squareds are below 0.2 except for the ones linking
• average and maximum formula complexity: this may be dismissed as trivial as the maximum drives the average;
• number of iterations to hours logged in the audit process. Even that R-squared is only 0.27.
A more sophisticated analysis uses a multiple regression, exploring the relationship between items A-F with item G, Hours logged. This allows for the possibility that the number of hours logged in auditing a spreadsheet is driven by some of the spreadsheet characteristics in combination. For reasons of commercial confidentiality, the number of hours consumed by each audit is represented in a normalised form, as a percentage of the average hours of all 75 assignments in the database.

<table>
<thead>
<tr>
<th></th>
<th>Coefficients</th>
<th>Standard Error</th>
<th>t Stat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>-12.78%</td>
<td>25.98%</td>
<td>-0.49</td>
</tr>
<tr>
<td>A</td>
<td>0.00%</td>
<td>0.00%</td>
<td>-0.37</td>
</tr>
<tr>
<td>B</td>
<td>-0.13%</td>
<td>1.00%</td>
<td>-0.13</td>
</tr>
<tr>
<td>C</td>
<td>0.00%</td>
<td>0.00%</td>
<td>-1.20</td>
</tr>
<tr>
<td>D</td>
<td>18.42%</td>
<td>9.98%</td>
<td>1.85</td>
</tr>
<tr>
<td>E</td>
<td>0.35%</td>
<td>0.51%</td>
<td>0.69</td>
</tr>
<tr>
<td>F</td>
<td>6.54%</td>
<td>1.48%</td>
<td>4.41</td>
</tr>
</tbody>
</table>

This suggests that average formula complexity is on the verge of joining the number of iterations as a significant contributor to the time taken to complete an audit. To the extent this is true, we can infer that an increase of 1 in the number of terms and operators in the average formula increases the time to audit the spreadsheet by about 18%.

6 INTERPRETATION

A typical project finance model connects project cash flows, which describe the revenues and costs that arise from building and operating an asset, with financing cash flows, which concern how the asset construction is paid for. The modelling of the financing cash flows may well be more extensive in larger transactions, largely because it may be necessary to draw on a greater number of sources of finance to get a sizeable project financed.

The complexity of the project cash flows, by contrast, is influenced by the nature rather than the size of the project. Some have many streams of revenue or costs, built up in intricate ways, and in others the revenues and costs can be derived very simply. It is therefore no surprise that the correlations between the value of the transaction and the various measures of spreadsheet size are positive, but not especially strong.

Less obviously, the correlation between the number of formulae and the effort required to audit the model is very low.

Operis’s first rationalisation of this result is that auditing a financial model does not simply involve checking a spreadsheet. The process followed by Operis, once the formalities of engagement are completed, involves a process of reviewing successive iterations of the model as they are refined by the client, in a process detailed below.
### Iterative Review Process for a Full Formal Audit

1. The client sends a spreadsheet model, along with relevant documentation such as loan agreements or subcontracts.
2. Operis reviews the model, and delivers a report itemising anomalies that it has found.
3. The client fixes the model and/or gets its lawyers to align the contractual paperwork with the model; in due course, it delivers a second version of model and the documents.
4. Operis reviews the second version and updates its report.

   *High level reviews, excluded from the database surveyed in this study because they have a more economical scope of work, stop at this point. Only the original version of the spreadsheet and maybe one revision are examined, and no formal opinion letter is delivered.*

5. The client and its lawyers fix any remaining issues and deliver a third version of the spreadsheet and documents.
6. Operis reviews the third version and updates its report.

   and so on (for an average of 8.1 versions in the assignments studied) until:

7. Operis undertakes final quality control processes, and reviews sensitivity analysis specified by the banks and prepared by the client.
8. Operis attends financial close, the legal ritual at which all contracts are signed, including the credit (loan) agreement.
9. Operis delivers a letter setting out its formal opinion about the fitness for its intended purpose of the model.

This process is aimed at ensuring a separation between the teams that develop and review the spreadsheets, so that the auditor is at no point auditing his own work.

Some clients react well to the reports and address the issues raised in them quickly and effectively. Others are less skilled, and need several attempts before they get the spreadsheets right. As a result, the number of iterations of the model increases, and with it the amount of hand-holding necessary. It is this activity that is the primary driver of the cost of the review exercise. (For clarity, it is the hours expended by Operis, not the client, that are the focus of this analysis.)

At any point in this process, the deal can change as it is the subject of continuing negotiations. That too adds to the number of versions of the model that need examining, just considered in isolation. But isolation is probably inappropriate, as the phenomenon likely interacts with the skill of the model developer. Skilled modelers may be expected to handle change in the high pressure circumstances of getting a transaction concluded with more serenity, and fewer adverse consequences for the integrity of the spreadsheet, than individuals who are at the edge of their competence.

Operis’s second rationalisation of the low correlation between the spreadsheet metrics presented and the effort to audit is that there are diverse ways to audit a spreadsheet. One way is to check the formulae one by one. Panko has blessed this approach repeatedly in...
his addresses to past EuSpRIG conferences, inferring from similarities between spreadsheeting and traditional software development in the nature and incidence of types of error that what has emerged as good practice in traditional software development is relevant to spreadsheets also. However, double checking Panko’s seminal paper, What We Know About Spreadsheet Errors [2], shows that he gives equal weight to code inspection and to data testing, of which formula reperformance is arguably a form.

“Although we still have far too little knowledge of spreadsheet errors to come up with a definitive list of ways to reduce errors, the similarity of spreadsheet errors to programming errors suggests that, in general, we will have to begin adopting (yet adapting) many traditional programming disciplines to spreadsheeting”.

“In programming, we have seen from literally thousands of studies that programs will have errors in about 5% of their lines when the developer believes that he or she is finished (Panko, 2005a). A very rigorous testing stage after the development stage is needed to reduce error rates by about 80% (Panko, 2005a). Whether this is done by data testing, line-by-line code inspection, or both, testing is an onerous task and is difficult to do properly”.

Exhaustive formula checking is described by Croll in the paper already mentioned, and certainly used by some large accounting firms. Alternative approaches are to reconstruct the model, by:

• building an entirely new one from scratch and reconciling the outputs, as one Big Four accounting firm prefers to do;
• keying the assumptions into its own trusted, standard model, as a smaller accounting firm claims to do;
• reperforming or reconciling independently the revenue, the costs, the taxes, the debt, the equity, and the financial ratios, as Operis does.

The time taken to reconstruct the distinct modules of a spreadsheet in a systematic and much-practiced way is unrelated to the number of formulae that the spreadsheet being tested happened to use, which may be highly variable for stylistic reasons.

Reperformance has a number of practical advantages over formula checking. Insertion of a single row near the top of a worksheet has the potential to alter every formula on that worksheet. Auditors who follow the formula checking approach are therefore often led to impose limits on what can be changed in the spreadsheet. Some insist that nothing is altered in the final week before the deal is signed. Others allow changes, but only at the bottom of each worksheet: no rows are to be inserted or deleted. Reperformance and reconciliation can be done in a way that is tolerant of the repeated examination of successive, potentially changing, versions of a model described in the audit process. Given that the number of iterations inspected for each model in the sample averaged 8.1, tolerance of change has real practical value to the customer.

Checking individual formulae amounts to determining whether the route followed to derive a number is the right one by following again that same route, this time very carefully. Reperformance and reconciliation amount to pursuing a route that is intentionally different from the first one and seeing if it takes us to the same place. Operis instinctively has more comfort with the two-route approach, likening it to a surveyor’s use of triangulation or an accountant’s insistence on double entry.
The most important advantage of audit methods involving reperformance and reconstruction is that it is hard to maintain focus during audit by formula inspection. Panko acknowledges this issue in the sentence following the extract above:

“In code inspection, for instance, we know that the inspection must be done by teams rather than individuals and that there must be sharp limits for module size and for how many lines can be inspected per hour”

To state the point more starkly, formula inspection is boring. It is difficult to retain staff who are willing to continue with it for any extended interval. This means that the average experience levels of those doing the work is likely to be lower in firms whose spreadsheet review methodologies involve that approach. A number of firms are explicit in making it a temporary rite of passage on the way to a more interesting job. Operis is doubtful that formula inspection is the One Right Way to perform spreadsheet verification on any scale, but even if it is, this retention/experience point more than offsets it in practice.

Operis’s third rationalisation of the low correlation between the spreadsheet metrics presented and the effort to audit is that the hours logged are given equal weight in this analysis regardless of who contributed them. What can take a new recruit many days to complete can be done by a manager with years of experience in a few minutes. As a rule the different steps in an audit are allocated to suitable levels of experience and seniority drawn from the pool of consultants. The vagaries of scheduling can mean that some tasks get done by individuals who are arguably over-qualified for the task, in order to meet transaction deadlines.

A fourth rationalisation is that the recipients of an opinion letter vary in how clean they want that opinion to be. Some require essentially every last defect to be driven out of models before they will advance loans on the basis of them. Others prioritise getting a transaction concluded, and are more readily satisfied that the projections provided are close enough. To the extent shortfalls remain in the model, compared with the ideal, these organisations are content for them to be listed as qualifications to the reported opinion. There will tend to be more iterations of a model before the first group is ready to close a deal than are demanded by the second. One manifestation of the general risk aversion since 2007 has been the migration of institutions from the second category to the first. That aside, the tolerance for risk varies fairly randomly from institution to institution, and even among individuals responsible for transactions.

7 CONSEQUENCES

When asked to quote to review a financial model, Operis used to base its price on the various measures of spreadsheet size. Operis has in OAK, the Operis Analysis Kit, a product that has been engineered to perform, rapidly, this measuring function among many others. Competing firms use a similar approach to pricing financial model audits. However, Operis has now ceased to use these metrics other than in exceptional conditions, in light of the good data it has amassed showing that they don’t capture the real drivers of spreadsheet review costs.

When developing, rather than auditing, a spreadsheet Operis follows a methodology that is distinctive. At Eusprig’s 2010 conference, Tom Grossman compared the Operis approach with two others [3].

These approaches share common aims, and agree on much more than they disagree over. One area of divergence, though, is over the practice of staging intermediate results, that
is, marshalling at the top of a block of calculation the items needed by that calculation. The benefit is that the formulae that do the most meaningful calculations can be seen to refer to cells that are very near at hand, making them quick and easy to check. The cost is that the spreadsheet is much taken up by simple formulae devoted to restating and marshalling locally to a calculation data that has been derived elsewhere in the model. To give some idea of the scale of this effect, Operis has rewritten some competitor models, which do use intermediate result staging, in its own style, which does not advocate this redundancy, and found that the result is typically as much as three times more compact.

Some followers of the staging methodologies have found themselves penalised when they come to procuring audit for the resulting models. The audit firms approached have been open that it is due to the distended formula counts. It follows from the analysis presented here that that handicap originates from an understanding of the cost of model audit that wants for sophistication, and should not be viewed as a defect of the modeling methodology (even if the methodology is one that competes for attention with Operis’s own).

8 NO PRECEDENT

The conclusions in this paper address a highly streamlined process for reviewing on a substantial scale spreadsheets that are confined to a narrow domain. They do not necessarily extend completely to a wider range of spreadsheets of arbitrary purpose. The dispersion seen in this study could be even wider in a randomly drawn sample from arbitrary disciplines, as would be the case in a financial institution that sought to embark on a firm-wide programme of spreadsheet remediation.

Audit costs are here shown to be driven to a degree by the process of iterating spreadsheet models with the developer or owner until the process of aligning them with the deal documentation is judged complete. It is possible to imagine an internal spreadsheet audit function within a financial institution for which this iteration between model developer and auditor has no parallel or relevance. In the absence of the iteration, the spreadsheet review effort could be more correlated with common metrics of spreadsheet size. More often, however, it would seem likely that a discovery of defects in spreadsheets would be followed up by a process of addressing the issue, which will likely involve human interactions of some kind that are roughly analogous to the formal iteration described.

REFERENCES


This volume contains the proceedings of the eleventh European conference on

‘Spreadsheet Governance – Policy and Practice’,

EuSpRIG 2011 conference held in July 2011 at

The objective of this conference is to promote discussion and co-operation amongst those concerned with authorising, auditing or developing spreadsheet models and by so doing, improve the reliability and integrity of information portrayed in spreadsheet models.

The papers cover a broad spectrum of practical experience and research.

Published By

| ISBN         | 978-0-9566256-9-4 |