Merging two sorted lists omitting common records

1. Place both lists in ascending order.
2. Select the first item from the first list and compare with the first item of the second list.
3. If both items are the same, remove both and move to the next items in both lists.
4. If the first item of the first list is greater than the first item of the second list, move to the next item in the first list.
5. If the first item of the second list is greater than the first item of the first list, move to the next item in the second list.

Steps:
1. Trial and error (rule based)
2. Information based
3. Algorithmic based
4. Complex algorithms and debugging based (CAD)
5. Concept based (don’t handle non-comparable data)
Proceedings of the EuSpRIG 2016 Conference

“Spreadsheet Risk Management”

First published 2016

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

Held at the Institute of Chartered Accountants in England and Wales (ICAEW), Moorgate, London.

July 7th 2016

Copyright: European Spreadsheet Risks Interest Group

www.eusprig.org


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.
EuSpRIG 2016 Conference

“Spreadsheet Risk Management”

EUSPRIG 2016 ORGANISING COMMITTEE

Patrick O’Beirne (Chair), Systems Modelling, Co. Wexford, Ireland
David Colver (Treasurer), Operis, London, UK
Grenville J. Croll, Bury St Edmunds, UK
Simon Thorne, Cardiff Metropolitan University, UK
Angela Collins, (Secretary) Manchester, UK

EUSPRIG 2016 PROGRAMME COMMITTEE

Simon Thorne (Chair), Cardiff Metropolitan University, UK
Grenville J. Croll, Spreadsheet Engineering, Bury St Edmunds, UK
Patrick O’Beirne, Systems Modelling, Co. Wexford, Ireland
EUSPRIG 2016 CONFERENCE PROGRAMME

Thursday 7th July 2016, ICAEW, Moorgate, London
The Science of Spreadsheet Risk Management

08:30 – 09:30 Arrival, Registration & Coffee

09:30 – 09:35 Chairman’s Welcome

09:35 – 09:45 Sponsor’s Welcome / Keynote

09:45 – 10:30 Session 1: Teaching methods are erroneous: approaches which lead to erroneous end-user computing. Mária Csernoch, Piroska Biró

10:30 – 11:15 Session 2: Characteristics of Spreadsheets Developed with the SSMI Methodology. Paul Mireault

11:15 – 11:45 Coffee Break

11:45 – 12:30 Session 3: A Programmatic Approach to the development of Solutions in Excel. Peter Bartholomew

12:30 – 14:00 Lunch

14:00 – 14:45 Session 4: A Conceptual Model for Measuring the Complexity of Spreadsheets. Thomas Reschenhofer, Bernhard Waltl, Klym Shumaiev, Florian Matthes

14:45 – 15:30 Session 5: A Pilot Study Exploring Spreadsheet Risk in Scientific Research Ghada AlTarawneh, Simon Thorne

15:30 – 16:00 Tea break

16:00 – 16:45 Session 6: The use of the Power Query / Get & Transform tools in Excel Simon Hurst

16:45 – 17:00 Q&A

17:00 -- Conference Close

AGM to discuss future direction of EuSpRIG.
# EUSPRIG 2016 PROCEEDINGS

## TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Author(s)</th>
<th>Paper</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mária Csernoch</td>
<td>Teaching methods are erroneous: approaches which lead to erroneous end-user computing.</td>
<td>1</td>
</tr>
<tr>
<td>Piroska Biró</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paul Mireault</td>
<td>Characteristics of Spreadsheets Developed with the SSMI Methodology.</td>
<td>15</td>
</tr>
<tr>
<td>Peter Bartholomew</td>
<td>A Programmatic Approach to the development of Solutions in Excel.</td>
<td>25</td>
</tr>
<tr>
<td>Thomas Reschenhofer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bernhard Waltl</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Klym Shumaiev</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Florian Matthes</td>
<td>A Conceptual Model for Measuring the Complexity of Spreadsheets.</td>
<td>37</td>
</tr>
<tr>
<td>Ghada AlTarawneh</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Simon Thorne</td>
<td>A Pilot Study Exploring Spreadsheet Risk in Scientific Research</td>
<td>49</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
PREFACE

Dear Colleagues,

You are very welcome to the Seventeenth Annual Conference of the European Spreadsheet Risks Interest Group.

This year our conference was preceded by SEMS in Vienna, the workshops on Software Engineering Methods in Spreadsheets, where academic researchers further explore the possibilities of applying successful software methods to spreadsheets. Our conference is very much on the practical details of enforcing structure on the untamed spreadsheet.

Currently we have about 4,000 unique visitors to our website per month, the most popular page being the horror stories; and an active Yahoo Group of 900 members.

It is my pleasure to once again acknowledge the keen work of our conference and programme organiser Simon Thorne from Cardiff Met. The committee also depends upon the wise counsel and active support of David Colver, Grenville Croll, and Angela Collins (secretary). They 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.

The programme of speakers includes contributions by researchers from the UK, Hungary, Canada and Germany.

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

Patrick O'Beirne, Chair 2016

http://www.eusprig.org

http://tech.groups.yahoo.com/group/eusprig
Teaching methods are erroneous: approaches which lead to erroneous end-user computing

Mária Csernoch, Piroska Biró
4028 Kassai út. 26. Debrecen, Hungary
csernoch.maria@inf.unideb.hu, biro.piroska@inf.unideb.hu

ABSTRACT

If spreadsheets are not erroneous then who, or what, is? Research has found that end-users are. If end-users are erroneous then why they are? Research has found that responsibility lies with human beings’ fast and slow thinking modes and the inappropriate way they use them. If we are aware of this peculiarity of human thinking, then why do we not teach students how to train their brains? This is the main problem, this is the weakest link in the process; teaching. We have to make teachers realize that end-users are erroneous because of the erroneous teaching approaches to end-user computing. The proportion of fast and slow thinking modes is not constant, and teachers are mistaken when they apply the same proportion in both the teaching and end-user roles. Teachers should believe in the incremental nature of science and have high self-efficacy to make students understand and appreciate science. This is not currently the case in ICT and CS, and it is high time fundamental changes were introduced.

1 INTRODUCTION

Research focusing on spreadsheet analysis has come to the conclusion that almost without exception spreadsheet documents, – more than 90% of them – have various types of errors, and that these errors – along with the extremely high level of human and computer resources required to administer the documents (EuSpRIG, Panko, 2008; Powel et al., 2008; Thorne, 2010) – cause serious financial losses (Van Deursen & Van Dijk, 2012). Speaking generally, it has been accepted that spreadsheets are erroneous. However, Panko (2013) claimed that it is not spreadsheets which are erroneous but rather the end-users who create the documents. He explains that one of the reasons for making mistakes is the overuse of attention mode (ATM) thinking compared to automatic mode (AUM) thinking (Panko, 2013, 2015; Maynes, 2015; Kahnemann, 2011). This finding is closely related to Kelemen’s, who claims that there is unreliability in metacognitive accuracy, while both memory and confidence are usually consistent between tasks (Kelemen et al., 2000).

Compared to Panko, our research group took several further steps by analyzing the different metacognitive computer problem solving approaches, the problem solving approaches of end-users, the mathability level of software tools, cognitive load theory, the teaching methods applied in end-user teaching and training, the textbooks and coursebooks, teacher education, as well as several informatics and computer science curricula. Considering all these different approaches and their connection to spreadsheets, we have found that one of the main reasons spreadsheet users make mistakes is that teaching methods and materials are erroneous. Consequently, until we transform end-user-teaching approaches, nothing will change. In the present paper we focus on the educational aspect of the TEAM (Tools Education Audit Management) Approach (Chadwick, 2002). We argue that we have both the theoretical background and the teaching tools needed to introduce concept- and algorithmic-based spreadsheet management as an effective tool in end-user computing.
2 PROBLEMS TO DEAL WITH

We are faced with a high number of problems in end-user-teaching. (1) As mentioned in the Introduction, Panko (2013), based on his research on cognitive science, claimed that most spreadsheet errors are due to ATM thinking. Consequently, we have to develop end-users’ AUM thinking to reduce spreadsheet errors. (2) Panko & Port (2013) have also claimed that “[end-user computing] … seems to be invisible to the central corporate IT group, to general corporate management, and to information systems (IS) researchers.” (Panko & Port, 2013; Burnett, 2009). (3) “The public image of computer science does not reflect its true nature. The general public and especially high school students identify computer science with a computer driving license. They think that studying computer science is not a challenge, and that anybody can learn it. Computer science is not considered a scientific discipline but a collection of computer skills.” (Hromkovic, 2009). These misleading opinions are openly expressed by Gove (2012) and Bell & Newton (2013). “…children bored out of their minds being taught how to use Word and Excel by bored teachers…” (Gove, 2012). “…a collection of low-level routine knowledge such as how to format pages in a word processor, or how to make tables in HTML.” (Bell & Newton, 2013).

The following three problems are straightforward consequences of problem (3) mentioned above. (4) Teaching materials – textbooks, coursebooks, recently published e-materials, etc. – do not support the development of computational thinking, which Wing claimed was the newly emerged basic skill of the digital era. “Computational thinking is a fundamental skill for everyone, not just for computer scientists. To reading, writing, and arithmetic, we should add computational thinking to every child’s analytical ability.” (Wing, 2006). These teaching materials are mostly out of context, beyond this they focus on the details of the tools, are written in cookbook style (Appendix 4) – giving sequences of clicks as instructions (Angeli, 2013) or, in reference style, – replicate reference materials. (5) Teachers, almost unconditionally, accept these teaching materials, and the approaches and methods which they suggest. (6) Teacher education is not prepared for the challenges of the digital era, neither in the case of teachers of informatics and computer sciences, nor non-professionals (Csernoeh, 2015; European Schoolnet, 2011, 2013, 2015).

3 TEACHING MATERIALS

By analyzing numerous spreadsheet teaching materials we have found that these books, on-line courses, and the teachers who follow them are one of the main reasons for failure. We focused on general purpose informatics coursebooks with a section on spreadsheets, as well as on books specializing in spreadsheets (Csernoeh et al., 2014).

The analyses of these books revealed that they do not fulfill the requirements of the general concepts and basic rules of informatics textbooks detailed in the paper by Freiermuth et al. (2008), but rather follow the “…the misleading concepts of computer science education that were broadcasted in many countries as the consequences of the emphasis created by the fast development of information technologies.”

In general coursebooks the spreadsheet sections/chapters focus on formatting details: how to color cells and borders, how to change fonts and font styles, etc., creating diagrams, and providing different lists of functions. What we miss in these books is real world problems and problem solving. The tasks, if there are any, are only fabricated examples based on non-existing or fake tables (Appendix 3), focusing on the details of the language, mostly listing functions and their arguments (Csernoeh et al., 2014; Appendix
5: students’ and books’ list of functions, S and B, respectively). The only exception found with real-world problems is Gross et al.’s book (2014), however they also present a fictional company and detail general informatics. None of the books corrects the errors contained in the wizards, helps, and/or references, as, for example, is shown by the arguments of the MATCH() and the IF() functions. The MATCH() function (MATCH function, nd) does not accept any array as an argument, and only a one-dimensional array (vector) serves as the lookup array. In the reference of the IF() function (IF function, nd) the ‘logical test’, ‘logical expression’, ‘condition’ is named as the first argument, but untrained end-users do not understand these expressions, while they are familiar with the notion of ‘yes/no question’. In a similar way, end-users do not understand the match-type argument of the MATCH() function, since the reference is based on the different selection algorithms (MATCH function, nd); however, they understand the concepts ‘descending’, ‘ascending’, and ‘no order’, which are necessary to select the correct match-type.

The spreadsheet books we analyzed turned out to be ill-named spreadsheet books. Instead of teaching spreadsheets they teach general ICT skills. In addition, the same introductory chapters on managing text, presentations, and spreadsheets, etc. can be found in all the birotical – office applications – coursebooks (Appendices 1, 2, and 4).

In these spreadsheet books a great range of basic knowledge about informatics is detailed at great length, just like in general coursebooks (Appendix 4. We have to mention here, that copying is four-step process, which last two are merged in the example). Beyond these, the newest features are emphasized and only extremely short sections deal with functions. The only exceptions were the books by Walkenbach (2002, 2010) and Advanced Excel Essentials (Goldmeier, 2014). Walkenbach works with formulas, functions, and even with array formulas. He mentions that one of the advantages of array formulas over copying formulas is that they reduce the vulnerability of spreadsheets. Goldmeier’s book is much less conceptualized, seeming to feature ideas which pop up randomly, without a clear understanding of the concepts of problem solving and arrays.

Considering the contents analyzed, we have come to the conclusion that these books represent a paradox. First of all, their content and style have not changed over the last two decades. This means that even the newest books published in the digital era, in which the greatest number of generations of end-users are using computers, explain how to start a program, and how to open and save files (Appendices 1 and 2). This information is either completely unnecessary or presupposes that the readers’ digital competence is at an extremely low level. The paradox is here: if end-users are able to handle files, these contents should not be in spreadsheet books; if end-users do not know how to handle files, they should be taught, but this is not the task of spreadsheet textbooks. Anyway we claim that these chapters should be omitted from spreadsheet textbooks. In a similar way, formatting and typographic details, along with knowledge about styles, do not constitute spreadsheet knowledge. These elements should also be part of end-users’ digital competence. In general, these books mainly focus on the interface and mix basic ICT knowledge with the most recent features of the programs, and miss the essence of spreadsheets.

On the other hand, at the very beginning of these books readers are overwhelmed with subjects for which they are not prepared. Is it worth explaining all the different data types for beginners in one group? No, it is just waste of time and energy. Instead, we can open files with obvious examples, students can analyze them and recognize the different data types – and not all of them at the same time. Or we can open a file with mismatched data types and students will see it even more clearly. Explaining references for beginners? They will not understand them. No macros in introductory courses, please!
What is not included in these books? There is no real problem solving in coursebooks, not even in online tutorials (e.g. for a sample of formulas on an empty table see Appendix 3), where adding tables with authentic content would not be a problem. These books are weak copies of references. The books do not mention how to design content (Angeli, 2013), how to solve problems, or how to build algorithms (Hubwieser, 2004; Csernoch, 2014a, 2014b, 2015; Csernoch & Biró, 2015a, 2015b, 2015c, Biró & Csernoch, 2015a, 2015b). The concept of function, introduced in maths classes, and the idea that spreadsheets and spreadsheet functions are closely related to it is not mentioned at all.

We can conclude, in general, that these books are extremely contradictory; consequently, they cannot be used, either in classroom teaching, or in autonomous learning. These methods have led to risky spreadsheet documents containing formula errors routed in copying, using constants in formulas, ill-used references, incorrect selection of functions, incorrect argument list, etc. In general, the documents lack of design and concept, which unplugged phases are rarely taught either in schools or in additional materials for lifelong learners (Angeli, 213; Raffensperger, 2001; Thorne, 2005; Thorne, 2010). Further consequences of the ‘classical’ teaching approach and materials are the time, human and computer resources used up (Van Deursen & Van Dijk, 2012), and overconfidence (Kruger & Dunning, 1999; Thorne, 2005). Since schema construction is not preferred in these methods, applying them is extremely demanding – considering cognitive load –, with all its consequences (Thorne, 2005).

One further problem has to be mentioned here. Analyzing textbooks and publishing the results would improve their quality. However, we have experienced that some authors consider themselves so highly qualified that they refuse to accept these analyses (Csernoch 2014b; Koreczné, 2014).

Panko and Port claimed that CS should take end-user computing seriously – problem (2). We go two steps further back: (1) we claim that CS should take teaching and teacher education seriously, and (2) teaching and teaching education should also take end-user computing seriously. However, not in the way suggested by Gove (2012), by banishing end-user computing from education. If we did so, we would increase the number of self-trained end-users, who would accept the ‘user-friendly’ approaches of profit oriented software companies, spending lots of time learning the interfaces. We argue that teaching concept-based problem solving and schemata construction requires professionals.

4 ATM VS. AUM THINKING

4.1 Mathability levels of problem solving approaches

Panko (2013) claimed that AUM thinking would reduce the number of errors in end-user computing, while Gove (2012) and Bell & Newton (2013) asserted that routine activities kill the algorithmic aspect of these tools and activities. This contradiction tells us that teachers have to find the right proportion of ATM and AUM thinking. We have found that a typology of computer problem solving approaches (Csernoch & Biró, 2015a) and the mathability levels of software tools (Biró & Csernoch, 2015a, 2015b) would provide guidelines.

There is a close connection between the higher mathability level approaches and the types of thinking required, from Level 5 to 3. Level 5 is the concept based approach, which requires the most ATM thinking. At this level a real world problem is presented and, following Pólya’s problem solving guide (1954) (see also Thorne, 2005 and Gross et al., 2014), we can reach a satisfactory result. At Level 4 the original world problem is...
somewhat simplified and the problem solving process starts from the building of the algorithm(s). In any other aspects there are no differences between Levels 5 and 4.

Figure 1. Computer problem solving approaches matched with the mathability level of problem solving and software tools

Methods at Level 2 focus on the details of the language and the environment, while at Level 1 the unplanned surface browsing leads to some kind of output. Neither of these levels is considered a problem solving approach, but rather a planned or unplanned sequence of clicks. Operating at Level 2 and/or 1 would lead to misconceptions; Sewell & Thede (2011) clearly stated that “spreadsheet languages are terse - hard to document and hard to read, hard to debug, and suitable for short subroutines or macros”.

Level 3 plays a crucial role in the problem solving process, since it connects the deep and surface approach methods. The major characteristic of this level is the application of the users’ own schemata – algorithms –, which is the platform where the proportion of ATM and AUM thinking can be controlled. At Level 5 and 4 ATM thinking is dominant; however, building schemata (Merrienboer & Sweller, 2005; Chi et al., 1982) at these levels would lessen the strain and the burden of ATM thinking. The schemata construction with high mathability level approaches would lead to AUM thinking, and consequently to fewer erroneous end-user activities. However, schemata construction requires teaching methods which hardly exist in end-user computing, or in spreadsheet development. This approach is well accepted in teaching ‘serious’ programming, but not in other computer related activities. We can also find effective schemata construction methods used in teaching maths to young children (Kemp, 1971; Morgan et al., 2014).

“Routine practice is the strongest educational practice that teachers can use in their classroom to promote achievement gains,” From these practices in teaching programming and maths for beginners, we can adapt these methods to educate end-users.

4.2 SPREGO: from ATM to AUM thinking

We claim that spreadsheet environments are as good as ‘serious’ programming languages, both for high mathability problem solving and for building schemata. Based on previous (Booth, 1992; Warren, 2004; Sestoň, 2011) and parallel research results (Hubwieser, 2004; Schneider, 2005) we have introduced Sprog – Spreadsheet Lego – and developed a complete methodology for introducing and teaching spreadsheets with this approach. Sprog is a Level 5 mathability approach, focusing on real world problems in various contents, which adapts the problem solving method of Pólya (1954), accepted in other sciences and also in programming, detailed in Thorne’s paper (2005). The other feature of Sprog is schemata construction. Sprog introduces only a limited number of general
purpose functions – a dozen for beginners – (Csernoch & Balogh, 2010; Csernoch, 2014a; Csernoch & Biró, 2015b, 2015c). Based on these functions, not only is ATM thinking and real world problem solving supported, but routine algorithms are developed, and based on them, meta-schemata are constructed. With the schemata construction ability of Sprego we can transfer knowledge from Level 4 to 3 on the mathability scale.

The limited number of Sprego functions is in accordance with findings in programming and ‘classical’ spreadsheets. Hromkovic claimed (2008) that “One can learn programming by starting with five instructions only and working totally with about fifteen instructions that are sufficient for programming any complex behavior of the [Logo] turtle. Our philosophy is to follow the history of programming, and so to derive all complex instructions as programs consisting of a very small set of basic instructions.” Considering spreadsheet environments, Walkenbach (2010) found that “People in average do not use more than a dozen functions.” With Sprego we are within the limit of 12–15 functions and have adapted the methods which have proved effective and efficient in teaching programming. We have also found that the guidelines for Logo programming would match our requirements, since, similar to Logo, the idea of Sprego is “…not to completely replace a programming course in a high-level language”, but to introduce programming and algorithms and offer a tool for end-user computing. “Spreadsheets are code.” (McKee; 2015) and we have to support this fact with our teaching approaches.

Conrad Wolfram, in his speech at TEDGlobal 2010 (Technology, Entertainment and Design), emphasized that in maths classes we have to “Stop Teaching Calculating, Start Teaching Math—Fundamentally Reforming the Math Curriculum”. This statement is in complete accordance with the idea that “the stronger the belief in the importance of computation and correct answers the lower the mathematical content knowledge.” (Francis et al., 2015). We claim that the same is true for end-user computing: Stop Teaching Software Usage, Start Teaching Computer Problem Solving—Fundamentally Reforming the Informatics Curricula.

Similar to Conrad Wolfram’s ideas, Sprego is a completely new approach to teaching spreadsheet management and introductory programming in already existing environments. Sprego does not start with the introduction of the interface, the different settings of spreadsheet interfaces, saving and opening documents, or entering data. This is not spreadsheet knowledge, but general ICT skills, digital literacy, or digital competence, which are brought into Sprego classes and practiced thoroughly, but do not constitute learning objectives.

4.3 SPREGO: tool for functional data modelling

What Sprego stands for is in complete accordance with Hubwieser’s (2004) and Schneider’s (2004, 2005) theories: “Some reader may wonder why functional data modeling opens the mandatory subject informatics in the 8th grade, since until now the so called classical way was favored, i.e. the teaching of some “hard” programming skills, namely imperative-like control structures. Moreover, one will be reminded through the attribute “functional” to the paradigm of functional programming. … one has to emphasize that functional modeling is pure sequential modeling technique. Only the causal structure and the functional data flow of a context can be represented. On the other hand, a new empirical study on the learning process of students at university level has shown that students have lowest problems with the functional modeling technique but greater problems with imperative one. So it is obvious to start yet at school with functional data modeling.”
Schneider (2005) did not know about the similar results which Booth obtained in 1992. However, it is heartwarming that they came to the same conclusion, unaware of each other's work. Two researchers from different surroundings, using different measuring methods achieved similar results, which supports the claim that functional programming is perfect for beginners. Our team went one step further and provided a methodology based on this theoretical background.

4.4 Teachers: ATM and AUM thinking

The bad news is that the proportion of ATM and AUM thinking is not constant. The proportion which would work well in end-user computing would not be appropriate in end-user teaching. Using an application and teaching it require different skills, different approaches, and different thinking modes. Teachers should be open-minded and they should be able to recognize when it is time for change. The teaching methods and the coursebooks clearly demonstrate that teachers apply AUM thinking even if it obstructs their introduction of novel teaching approaches. This problem of teachers' beliefs and their effectiveness in the teaching process is well presented in Chen et al. (2015), and the relationship between teachers' beliefs and effectiveness and students' results are presented in Figure 2 (published in the paper of Chen et al., 2015).

Based on our testing project (TAaAS, Biró & Csernoch, 2013b, 2014b; Csernoch et al., 2015; Biró et al., 2015a, 2015b) we have found that most students are prepared only for tests which rely heavily on surface-knowledge (Hromkovic, 2009), and they fail when language- and interface-independent problems are presented (Csernoch et al., 2015). Students' progress in informatics does not reach the required level of effectiveness. Beyond this, we have to keep in mind that we are not necessarily preparing students for tests, but for performing in real life, which is even more demanding than classroom and testing environments. Consequently, although the beliefs held by teachers specialized in informatics require further testing and analysis, it is clear that they have a negative impact on end-user computing.

Figure 2. The meaning system model of Chen et al. (2015). Two different beliefs in the nature of science (fixed and incremental natures, FN and IN, respectively) and two self-efficacy (high and low teaching self-efficacy, HSE and LSE, respectively) are distinguished.
The performances of the four groups of students (Figure 2, right cells) are closely related to the mathability level of computer problem solving approaches (Figure 1): Level 3—FN+HSE, Level 2 and 1—FN+LSE, Level 5—IN+HSE, Level 4—IN+LSE, from top to bottom, respectively.

What we most miss from the teaching of end-user computing is the appreciation of this science. Panko and Port (2013) found that end-user computing is not taken seriously, “seems to be invisible…” and “It is time to stop ignoring end-user computing in general and spreadsheets in particular.” We claim that the main reason for this misconception is that education is not prepared for end-user teaching. Most of our teachers do not use the algorithmic approach to end-user computing, and their teaching materials are not high mathability tools. Teachers fall for the software companies’ misleading ‘user-friendly’ slogans and approaches, focus on technical details, develop low mathability level materials or unconditionally accept them.

5 CONCLUSION

Is there any reason for being optimistic or should we give up? Are end-users second hand participants in the digital word, as Asimov predicted, when he wrote that “Paul knew mysterious things about what be called electronics and theoretical mathematics and programming. Especially programming. Nicole didn’t even try to understand when Paul bubbled over about it.” (Asimov, 1982)? Should we also accept that “…Excel is broken. And I strongly suspect it can’t be fixed. Yet it’s ubiquitous and business critical. We need to reinvent the wheel and change all four whilst the car is driving down the motorway — and I don’t know how to do that…” (McKee, 2015).

We cannot give up! We have to find ways for teachers to educate for effective end-user computing, especially spreadsheet management. The good news is that we already have the theoretical background (Booth, 1992; Hubwieser, 2004, Warren, 2004; Merribenboer, & Sweller, 2005) and methods (Csénoch, 2014a; Csénoch & Biró, 2015b, 2015c) which would allow us to increase the level of end-user computing, and end-users’ computational thinking. The effectiveness of Sprego (Csénoch, 2014a; Csénoch & Biró, 2015b, 2015c) has been testing since the academic year of 2011/2012. The preliminary results (Biró & Csénoch, 2014) clearly demonstrate that with Sprego we can change the students’ approach to spreadsheet problem solving and solutions. These results suggest that we are able to solve the problem of “changing all four whilst the car is driving down the motorway “. On the other hand, it is already clear that schemata are built with Sprego, which is necessary for the reliable decisions of fast thinking mode.

One might ask, why spreadsheets? The answer lies in their special characteristics. On the one hand, spreadsheet management is end-user activity, while on the other hand it is a form of programming. By accepting this two-fold approach in the teaching-learning process, we would raise end-users’ skills and end-user computing to a higher level.

6 BIBLIOGRAPHY


MATCH function. Retrieved February 9, 2016 from https://support.office.com/en-gb/article/MATCH-function-e8df1d45-c762-47d6-bf89-533f4a37673a


Wing, J. M. (2006), Computational Thinking. March 2006/Vol. 49, No. 3 Communications of the ACM.
Appendices

Appendix 1.

Inhalt

1. Allgemeine Grundlagen
   1.1 Daten speichern, schreiben, öffnen, drucken ........................................ 15
   1.2 Die Benutzерfelder ............................................................................. 26
   1.3 Datenerstellung ................................................................................ 48
   1.4 Übersicht ....................................................................................... 53
   1.5 Vorsicht ......................................................................................... 65

2. Grundlagen von Excel ........................................................................... 66
   2.1 Texte und Zahlen einfügen .............................................................. 76
   2.2 Zahlen einfügen ............................................................................. 80
   2.3 Zahlen einfügen ............................................................................. 87
   2.4 Roten-einfügen .............................................................................. 89
   2.5 Bezeichnungen und einfügen .......................................................... 97
   2.6 Individuelle Formen einer Zelle einfügen ........................................... 103
   2.7 Muster und Muster einfügen ............................................................. 109
   2.8 Übersicht ....................................................................................... 124
   2.9 Vorsicht ......................................................................................... 137

3. Tastatur und Tastaturkombinationen ...................................................... 139
   3.1 Tastatur und Tastaturkombination .................................................... 152
   3.2 Tastatur und Tastaturkombination .................................................... 162
   3.3 Tastatur und Tastaturkombination .................................................... 171
   3.4 Tastatur und Tastaturkombination .................................................... 182
   3.5 Tastatur und Tastaturkombination .................................................... 192
   3.6 Übersicht ....................................................................................... 200
   3.7 Übersicht ....................................................................................... 204
   3.8 Übersicht ....................................................................................... 209
   3.9 Übersicht ....................................................................................... 215

4. Zellen markieren ................................................................................... 219
   4.1 Zellen markieren ............................................................................ 224
   4.2 Zellen markieren ............................................................................ 227
   4.3 Zellen markieren ............................................................................ 231
   4.4 Zellen markieren ............................................................................ 234
   4.5 Zellen markieren ............................................................................ 237
   4.6 Zellen markieren ............................................................................ 239

5. Excel und Excel-Tabellen einfügen ....................................................... 241
   5.1 Formeln und Funktionen einfügen .................................................... 248
   5.2 Formeln und Funktionen einfügen .................................................... 249
   5.3 Übersicht ....................................................................................... 254
   5.4 Übersicht ....................................................................................... 259

6. Der Arbeitsbereich einrichten ................................................................. 261
   6.1 Text und Zahlen einfügen .............................................................. 265
   6.2 Tastatur und Tastaturkombination .................................................... 269
   6.3 Tastatur und Tastaturkombination .................................................... 270
   6.4 Tastatur und Tastaturkombination .................................................... 272
   6.5 Tastatur und Tastaturkombination .................................................... 274
   6.6 Tastatur und Tastaturkombination .................................................... 276
   6.7 Tastatur und Tastaturkombination .................................................... 277

7. Diagramme ........................................................................................... 279
   7.1 Diagramme einfügen ..................................................................... 294
   7.2 Diagramme einfügen ..................................................................... 302
   7.3 Übersicht ....................................................................................... 328
   7.4 Übersicht ....................................................................................... 332
   7.5 Übersicht ....................................................................................... 334
   7.6 Übersicht ....................................................................................... 337

8. Excel und Excel-Tabellen einfügen ....................................................... 345
   8.1 Formeln und Funktionen einfügen .................................................... 353
   8.2 Formeln und Funktionen einfügen .................................................... 357
   8.3 Übersicht ....................................................................................... 367
   8.4 Übersicht ....................................................................................... 373

9. Excel und Excel-Tabellen einfügen ....................................................... 379
   9.1 Übersicht ....................................................................................... 387
   9.2 Übersicht ....................................................................................... 393
   9.3 Übersicht ....................................................................................... 395
   9.4 Übersicht ....................................................................................... 397
   9.5 Übersicht ....................................................................................... 399
   9.6 Übersicht ....................................................................................... 401

10. Excel und Excel-Tabellen einfügen ...................................................... 403
    10.1 Übersicht ..................................................................................... 407
    10.2 Übersicht ..................................................................................... 410

Appendix 2.

Getting Started
  - Spreadsheet editor
  - Quick Access Toolbar
  - Quick Access Toolbar

Customizing Excel
  - Customize Quick Access Toolbar
  - Customize Quick Access Toolbar

Working with a Workbook
  - Create a Workbook
  - Save a Workbook
  - Open a Workbook
  - Save/Save As

Manipulating Data
  - Select Data
  - Copy and Paste
  - Cut and Paste
  - Undo and Redo
  - AutoFill

Modifying a Worksheet
  - Insert Cells, Rows and Columns
  - Delete Cells, Rows and Columns
  - Find and Replace
  - Go To Comment

Performing Calculations
  - Excel Formulas
  - Insert Function
  - Insert Function
  - Insert Function
  - Insert Function
  - Insert Function

Macros
  - Recording a Macro
  - Running a Macro

Sort and Filter
  - Basic Sorts
  - Examine Sorts
  - Filter

Graphics
  - Adding a Picture
  - Adding a Row of Text
  - Adding a Picture

Charts
  - Create a Chart
  - Modify a Chart
  - Chart Type

Formatting a Worksheet
  - Convert Text to Column
  - Modify Cells
  - Format Cells Dialog Box
  - Add Borders and Gridlines
  - Change Column Width and Row Height
  - Hide or Unhide Rows and Columns
  - Merge Cells

Manipulating a Worksheet
  - Select and Sort Columns
  - Insert Sheet Tab
  - Change Sheet Tab

Page Properties and Printing
  - Page Properties Tab
  - Page Setup Tab
  - Page Layout Tab
  - Page Orientation
  - Page Breaks

Customize the Layout
  - Split a Worksheet
  - Freeze Panes and Unfreeze Panes
  - Freeze and Unfreeze Panes

Appendix 2.
Appendix 3.

Excel Formulas

A formula is a set of mathematical instructions that can be used in Excel to perform calculations. Formulas are entered in the formula bar with an = sign.

References:
- The cell or range of cells that you want to use in your calculation
- Operators: +, -, *, /, etc.
- Functions: Predefined formulas in Excel

To create a basic formula in Excel:
1. Select the cell for the formula
2. Type the equal sign (=) and the formula
3. Click Enter

Appendix 4.

Copy formulas into non-numeric cells

1. Select the cell or cells being copied.
2. Press the F4 function key to add the dollar sign ($) to make the cell reference absolute or mixed.
3. Change the cell references to absolute references.

There are many elements to set Excel formulas.

Appendix 5.
Characteristics of Spreadsheets Developed with the SSMI Methodology

Paul Mireault
Founder, SSMI International
Honorary Professor, HEC Montréal
Paul.Mireault@SSMI.International

ABSTRACT

The SSMI methodology was developed using concepts from Computer Science, Software Engineering and Information Systems and has been taught to undergraduate and MBA students and in Executive training seminars. In this paper, we describe the major characteristics of the spreadsheets developed using the methodology and show how they contribute to reduce many error causing factors.

1 INTRODUCTION

Developing a spreadsheet is a complex task, often performed by people with little or no training. These may be accountants, analysts or department directors who have a well-defined job to do, which is not being a spreadsheet specialist. The free-form nature of spreadsheets lead to all sorts of designs.

The SSMI methodology was developed to help developers structure their spreadsheet in a way that makes them easy to understand and to maintain.

2 REVIEW OF THE SSMI METHODOLOGY

The Structured Spreadsheet Modelling and Implementation (SSMI) methodology is described in (Mireault, 2016). It uses the following concepts that are commonly used in domains related to systems development.

1. Conceptual model. In Information Systems, the conceptual model is used to describe what a system does, or should do, without considering the technology that will be used to implement it. It uses a vocabulary familiar to the user.

2. Names. In Computer Science all computer languages use symbolic names to indicate what variables represent. The only restriction, in Excel, is that names cannot contain spaces and some other special characters. When Excel creates names from cell labels, it replaces spaces and special characters with the underscore “_” character.

3. Modules. In Computer Science, modules are self-contained portions of code that have a specific list of inputs and produce a specific output. Modules are easier to understand and to debug.

4. 3-tier architecture. In Software Engineering, the 3-tier architecture consists of separating an implementation in elements that handle different operations. The usual tiers are the Interface, the Application and the Services. With this separation of major tasks, one can change database systems by modifying only the Services tier and leaving the other tiers untouched.

In the SSMI methodology, the conceptual model consists of the Formula Diagram which is a representation of the problem’s variables and how they are related, and the Formula List which specifies the nature of these relationships as formulas. Figure 1, taken from (Mireault, 2016), illustrates a Formula Diagram and Table 1 its Formula List. In this example, the user wants to examine scenarios where he modifies the product’s price and...
see the corresponding regional Profit and Total Profit. He uses a demand function to estimate the total Demand for a given Price. The dash-bordered rectangle indicates the portion of the model that depends on the Region dimension: each region has its own delivery cost and the Distribution parameter represents its portion of the overall Demand.

Rectangles represent values that will be entered by the user during the normal use of the spreadsheet and triangles represent constants that are not usually changed. Circles and ovals represent variables that are calculated with a formula, with ovals representing variables whose value are of interest to the user. Rectangles and ovals will be part of the Interface tier.

![Formula Diagram](image)

**Figure 1 - Formula Diagram**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Price</td>
<td>Input</td>
<td>(To be set by user)</td>
</tr>
<tr>
<td>Profit</td>
<td>Output, repeating</td>
<td>Revenue – Total Cost</td>
</tr>
<tr>
<td>DemParA</td>
<td>Parameter</td>
<td>376,000</td>
</tr>
<tr>
<td>DemParB</td>
<td>Parameter</td>
<td>1.009</td>
</tr>
<tr>
<td>Fixed Cost</td>
<td>Parameter</td>
<td>$2,500,000</td>
</tr>
<tr>
<td>Manufacturing Cost</td>
<td>Parameter</td>
<td>$120</td>
</tr>
<tr>
<td>Distribution</td>
<td>Parameter, repeating</td>
<td>48%, 23%, 29%</td>
</tr>
<tr>
<td>Delivery Cost</td>
<td>Parameter, repeating</td>
<td>$50, $80, $60</td>
</tr>
<tr>
<td>Total Demand</td>
<td>Calculated</td>
<td>DemParA * DemParB^Price</td>
</tr>
<tr>
<td>Regional Demand</td>
<td>Calculated, repeating</td>
<td>Total Demand * Distribution</td>
</tr>
<tr>
<td>Variable</td>
<td>Type</td>
<td>Definition</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------------------</td>
<td>-----------------------------------------------------</td>
</tr>
<tr>
<td>Total Cost</td>
<td>Calculated, repeating</td>
<td>Regional Fixed Cost + Variable Cost</td>
</tr>
<tr>
<td>Regional Fixed Cost</td>
<td>Calculated, repeating</td>
<td>Fixed Cost * Distribution</td>
</tr>
<tr>
<td>Variable Cost</td>
<td>Calculated, repeating</td>
<td>Regional Demand * Unit Cost</td>
</tr>
<tr>
<td>Unit Cost</td>
<td>Calculated, repeating</td>
<td>Manufacturing Cost * Unit Cost</td>
</tr>
<tr>
<td>Revenue</td>
<td>Calculated, repeating</td>
<td>Regional Demand * Price</td>
</tr>
<tr>
<td>Total Profit</td>
<td>Output</td>
<td>SUM(Profit)</td>
</tr>
</tbody>
</table>

Table 1 - Formula List

Names are implemented directly using Excel's *Create Name from Selection* button (or menu item), as shown in Figure 2. Names created in the structured implementation refer to either single cells, entire row or entire columns.

![Creating Names using the row labels](image)

Figure 2 - Creating Names using the row labels

In the structured implementation, each calculated variable is defined at the bottom of its definition block, as illustrated in Figure 3.

![Definition blocks of variables Regional Demand and Revenue](image)

Figure 3 - Definition blocks of variables Regional Demand and Revenue

The top part of a definition block consists only in reference formulas whose purpose is to make a local copy the values used in the definition formula, as shown in Figure 4.

![Formula view showing the structure of a definition block](image)

Figure 4 - Formula view showing the structure of a definition block
The SSMI methodology can be taught in 8 to 10 hours (Mireault, 2016).

In the following section, we will describe the major characteristics of the spreadsheets developed by following the SSMI methodology.

3 SSMI CHARACTERISTICS

3.1 Use of worksheets

The SSMI methodology uses worksheets to implement the 3-tier architecture. Worksheets are dedicated to a single tier:

- The Application tier: the Model worksheets contain only the definition blocks of the model's variables. There are never any input cells in the model worksheets. Also, the model worksheets are not designed for the worksheet user: the layout elements are present only to help the worksheet developer of the worksheet auditor. The definition of a variable is indicated by a bold-italic font. There can be more than one Model worksheet to suit the developer's needs.

- The Services tier: the Parameters worksheets are the sheets that collect all the inputs. The inputs can come from the Interface sheet or from sheets that import raw data from external sources.

- The Interface tier: the Interface sheets are the sheets that are actually used by the spreadsheet's users. This is where they enter specific values and see the results for the scenarios that interest them. The input values are referenced in the Parameters worksheet and the output values are referenced from the Models worksheets.

Using dedicated worksheets is also recommended by many researchers, such as (Read & Batson, 1999).

3.2 Use of Names

Using names indiscriminately has been identified as a source of error. (McKeever & McDaid, 2010) devised an experiment whose results suggest that names has a negative effect on debugging performance. (Kruck & Sheetz, 2001) suggest that naming cells contribute to making formulas easier to understand Nonetheless, we consider that the way we use names is very different than the unstructured way that was used in those studies.

The structured implementation of the SSMI methodology uses names only in the reference formulas that are part of the upper portion of the definition blocks. The definition formula itself does not use names. This way, we take advantage of Excel's color coding to help us validate the definition formula: as shown in Figure 5, each element of the formula in cell B6 has a colour code and each element of the top portion of the definition block also has a colour code. If one element did not have a colour code, then we would investigate further.
3.3 Far and local references

Far references in formulas, called *coupling by* (Kruck & Sheetz, 2001), has been identified as a source of errors. (Raffensperger, 2003) says that a formula that references a cell that is not immediately visible and understood is harder to understand.

A definition formula uses local references to values that are situated in the top part of its definition block. This way, far references are not specified by point-and-click; they are specified by the name of the variable. An error could occur when the developer types the wrong name, but since we display the variable's values to the right of its label the developer should notice the error. Figure 4 shows that it is easy to verify that the proper reference is made in the top portion of the definition blocks.

3.4 Transitive references

A transitive reference is a reference to a location where a variable is used, not where it is defined.

The left side of Figure 6 illustrates a situation where the developer refers to cell B10 in cell B14 instead of referring to the definition of *Number of Items Delivered* (cell B3). Later, he modifies the spreadsheet to take into account the fact that all the items delivered incur a delivery cost but only the items effectively sold contribute to the total sales. He does the correct modification in rows 10 to 12, but he does not realize that the transitive reference in cell B14 now produces an error.

While a transitive reference does not produce an error when the spreadsheet is initially built, it can produce a logical error when the spreadsheet is modified later. Spreadsheet developers are tempted to use transitive references because it is faster to point to a cell that is close (usually just above) than to navigate to the location where the variable is actually defined (which can be far).

Transitive references have been identified as a cause of errors by many authors. The errors can appear during the spreadsheet maintenance when new variables are created to take new nuances into account.
Figure 6 - Example of a transitive reference

Using the SSMI methodology, the developer simply cannot create a transitive reference since the references in the top portion of a calculation block use names that always refer to the definition of the variable. This illustrated in Figure 7: the left side shows the formula view of the left side of Figure 6 developed without the SSMI methodology and the right side shows the formula view of the spreadsheet developed with the SSMI methodology.

Figure 7 - The SSMI methodology cannot produce a transitive reference

Figure 8 illustrates the formula view of the right side of Figure 6.
3.5 Formula Complexity

The complexity of the formula in a cell is often cited as a source of errors. According to (Hermans, et al., 2012) simpler formulas have a low complexity score.

The SSMI methodology's most important rule is to never mix operators or functions in a formula. The developer is encouraged to create variables as needed. The formula

\[ \text{Total Cost} = \text{Fixed Cost} + \text{Quantity} \times \text{Unit Cost} \]

uses two different operators: the addition and the multiplication. It is thus replaced by the two following formulas:

\[ \text{Variable Cost} = \text{Quantity} \times \text{Unit Cost} \]
\[ \text{Total Cost} = \text{Fixed Cost} + \text{Variable Cost} \]

When implemented, this rule has the advantage of creating blocks that are easy to validate by eye, as shown in Figure 9.
3.6 Formula copying

Copying formulas has been cited by many authors as a source of errors. The errors can be due to improper relative or absolute references or to partial copying.

Relative and absolute references are an artefact introduced by spreadsheet program companies, like Microsoft, Lotus and Apple, to understand the spreadsheet developer’s intentions when he copies a formula.

In an SSMI spreadsheet, there is never any need to use absolute or mixed references in a formula. As mentioned above, the definition formula uses only the cells that are immediately above it, in the top portion of the block, and uses the standard relative references. The cells in the top part of the block always use names that are interpreted correctly. A name that refers to a single cell is interpreted as an absolute reference, and a name that refers to a row or a column is interpreted as a mixed reference. This behaviour is illustrated in Figure 10, showing the normal and the formula view of the same worksheet with the Trace Precedents arrows.

![Figure 9 - Formula complexity example](image)

![Figure 10 - Behaviour of Names. Normal view (top) and Formula view (bottom)](image)
A partial copy happens when the developer does not copy the cell’s formula all the way to the end of the row or column. The result is that some cells in the row, or column, have one formula and the others have another.

In a repeating model worksheet, the model is implemented in a single column of its worksheet. Even if a few formulas have been modified, it is easier to copy the whole column instead of copying the modified formulas one at the time. Reducing the number of times the developer needs to copy formulas should reduce the risk of making copy errors.

3.7 Auditing

When we audit a spreadsheet we are in fact verifying two things: the model and its implementation.

Understanding the model

The first step consists of examining each variable of the Formula Diagram and determining if its formula is correct. During this step, we also have to determine if we have forgotten or ignored other variables.

Verifying the spreadsheet

Once we are satisfied that the model is correct, we now need to verify if its implementation is also correct. Verifying an SSMI spreadsheet is pretty straightforward. There are two cases to consider: the non-repeating model and the repeating model.

In the case of the non-repeating model we need only display the formula view in Excel and examine each definition block to see if it conforms to the Formula Diagram and the Formula List.

In the case of the repeating model we do the same examination on the leftmost column of the repeating model worksheet. Then, to make sure that no formula has been altered in the other columns or that there hasn’t been a partial copy, we can proceed as follows.

1. Work on a copy of the file you are auditing.
2. In the repeating model worksheet, copy everything: Ctrl+A, Ctrl+C.
3. Create an empty worksheet and paste the values only. Select everything and create a conditional format highlighting the cells that have a different value than the corresponding cell in the repeating model worksheet.
4. Return to the repeating model worksheet and copy the first model column to all the others on the right.
5. Examine the worksheet with the pasted values to see if any cell is highlighted.

4 CONCLUSION

(Panko, 2015) says that “Given that [our] experience of errors in unreliable, intuitions about how to reduce errors should not be taken seriously unless they are rigorously tested.” The SSMI methodology has not been tested yet. But neither have the other standards proposed by different organizations: FAST (Fast Standard Organization, 2015), SMART (Corality, 2015) and SSRB (Spreadsheet Standards Review Board, 2012). According to (Grossman & Özlıük, 2010), these standards “do not attempt to address “writing spreadsheets” in general” but are specialized for financial modelling.

The SSMI methodology has been developed to be a general spreadsheet development methodology that can be used in any domain, like accounting, actuary, economics,
engineering, human resources, insurance, logistics, management, marketing and, of course, finance.

Further research, like (McKeever & McDaid, 2011), should be done to evaluate its performance with regards to errors and ease of use.

5 REFERENCES


Read, N. & Batson, J., 1999. Spreadsheet Modelling Best Practice. s.l., IBM.

A Structured Approach to the development of Solutions in Excel

Peter Bartholomew
MDAO Technologies Ltd
peter.bartholomew@btconnect.com

ABSTRACT

Spreadsheets offer a supremely successful democratisation platform, placing the manipulation and presentation of numbers within the grasp of users that have little or no mathematical expertise or IT experience. What appears to be almost completely lacking within a ‘normal’ solution built using Excel default settings is the deployment of any structure that extends beyond a single-cell formula. The structural elements that allow conventional code to scale without escalating errors appear to be absent. This paper considers the use of controversial or lesser-used techniques to create a coherent solution strategy in which the problem is solved by a sequence of formulas resembling the steps of a programmed language.

1 INTRODUCTION

Spreadsheet technology, here the use of Excel is considered in particular, has achieved remarkable success in making ‘numbers’ available to business and industry with little or no reliance upon the user having any interest or knowledge in the mathematical theories or IT structures that make analysis possible in other domains. Indeed Raffensperger [2001, 2003] almost celebrates this lack of competence and concentrates upon cell-by-cell construction as a desirable characteristic, unlike Hellman [2005] who argued that the cell/matrix formula concept is so deeply flawed that a separation of conceptual model from the presentation structure of the spreadsheet is long overdue. Raffensperger rejects the “metaphor” of the spreadsheet as a program and draws upon text documents, mathematics and graphic art for style guidance. In doing so, he chooses to ignore why modularity is so highly prized for code development or why the more abstract constructions of linear algebra are important in mathematics or even why technical reports tend to be structured using a hierarchy of numbered sections. Excel is superbly successful for the solution of ad-hoc problems but the structural elements that allow conventional code to scale without escalating errors appears to be sadly lacking in the spreadsheet solutions I have seen or even from the good-practice advice available within the Excel community.

The approach outlined in this paper started as an experiment conducted in response to a statement deprecating the use of Names in Excel and claiming their use to be limited to the “simplest” of applications¹. The objective of the work is to examine whether that claim is true or whether it is possible to adopt the converse strategy and eliminate the use of direct cell referencing in its entirety. It turned out the use of Names for all referencing leads to the adoption of other techniques which, although standard features of Excel, tend not to receive extensive mainstream use. Overall the approach is more consistent with the programmatic approach of Bewig [2005] than it is to that of Raffensperger or the FAST Standard [2015].

The techniques adopted include the widespread use of array formulas, with the formula being decomposed into single-cell formulas through the use of relative referencing only where the use of array formulas proves impractical. Another feature of the approach is to

¹ Names are better (or only) suited to simple spreadsheets with limited complexity, where reading a simple natural language formula such as “=Price * Quantity” is a real possibility
use Named formulas (these do not refer to ranges) to reduce or eliminate the need to work with deeply nested formulas. This is shown to generate linear sequences of statements that, when documented, are far more reminiscent of a procedural language than of a typical worksheet formula but at the same time the immediacy of the spreadsheet solution and the opportunity for graphic presentation and user interaction are retained. It is only when one starts digging to see the formulas that the difference even becomes apparent; something that may affect the auditor but should not be obvious to the user.

The change of mind-set embodied in the new approach also leads to changes within the formatting and presentation in order to reduce the emphasis on the location of data referenced within the worksheet and, instead, to stress the significance of the data as properties or components of a container data object. Because of this, there is little or no benefit in showing the normal alphabetic form for sheet column headers. The location of an object within the worksheet is simply not relevant to the solution so its position on the worksheet may be determined solely by aesthetic considerations and the need to achieve a clear visual presentation of content to the user.

Another feature of the approach is that, provided formulas are only entered as the properties (ideally the array property) of Named Ranges, the documentation of Names provides a complete description of the workbook. By ‘complete’ it is meant here that it is possible to rebuild the workbook in its entirety by using the documentation as steering data to direct VBA utility code. Only the input data and additional features such as charts or shapes and ad-hoc annotation would need to be provided by the developer.

2 DISCUSSION OF THE TECHNIQUES

2.1 Applicability

The approaches advocated and described in this paper relate to the building of spreadsheet-based numerical models and are far less applicable to the analysis of corporate data. In a model, the data may well have been created with the specific goal of animating the model and the data lifecycle tend to be one of archiving one set of data within a version of the model and then loading a fresh dataset into the model, although this might require some degree of modification re-sizing arrays to accommodate it.

This is a world away from the massive corporate databases that accumulate over time. Such data may accumulate in use but otherwise needs protection against unauthorised change if it is to retain its integrity. Eventually any such data-processing activity will crumble under the weight of accumulated data but the point may be deferred by rigorous optimisation for speed.

The author has experience of building models for simulating ship motion in the North Atlantic, calculating the strength and stiffness of carbon composite laminates, simulating the output from statistical analysis, animating the user interface for games such as 2048 and, more recently, financial modelling within a comparative study. The common feature of these models is that they are light on data but require multiple processing steps.

2.2 Example: Model with similar line items

The first example was a shared exercise conducted under Levi Bailey’s leadership within a LinkedIn discussion. The solution shown in Figure 1 drew heavily upon techniques developed for problems far removed from financial modelling and was intended to provide an option within a spectrum of solutions, the principal ones being drawn from the finance modelling community. This solution has a number of highly distinctive features, which will be further discussed as topics within this section.
An immediate visual distinction is that the worksheet is presented without gridlines or sheet headings since the location or even the relative alignment of data is considered immaterial both to the business problem and to its solution. In the main, the visual appearance is produced by applying modified versions of the Microsoft built-in styles. An exception is the presence of somewhat lurid magenta cells (here circled), each one representing an array formula.

The region over which these identifiable array formulas are applied typically corresponds to a Named Range. The uniformity of a formula throughout the time series is generally considered good-practice but here it is actually enforced by the use of array formula and, as a result, consistency checks are not required; also there are very few formulas to verify as being correct. Moreover, a simple looking formula such as “= Price * Quantity” can be applied to calculate results across large regions within a workbook.

The other conspicuous visual feature is the vertical band of highlighted cells. This is created by conditional formatting used to show an entire column range that creates a temporal slice through the model allowing any aggregation of data that is used to characterise the entire model to be restricted to the selected time periods. It would be possible to achieve the same result using a system of flags and SUMIFS but that would lack the elegance of the range intersection solution that exploits the overall structure of the problem rather than developing multi-step formula solutions ‘bottom-up’.

2.3 Named Ranges

Whereas a programming language will use variables to allow the developer to reference values held in computer memory, a spreadsheet uses the location of cells presented within a rectilinear grid. In each case this represents a level of indirection; the developer has no need to know the actual location at which the value is held in the computer memory.

In the case of the code variable, the name chosen in any but the most primitive machine code is likely to provide some description of the intended content. Spreadsheets are different; the cell reference such as A1, simply aims to identify the content by its location within an apparent grid. Any meaning is normally to be inferred from annotation held in adjacent cells.
The use of defined names to reference the content comes as close to defining a variable as the spreadsheet software will allow. Strictly, the name is a formula that references the required cell or range and, in doing so, conceals the direct reference from the user. The name does not store the result; that is the province of the cells, which hold the calculated value as a property. The Name can, nevertheless, provide a meaningful and well-structured description of the cell’s intended content. Normally, a Name will refer to a range as an absolute reference. In that instance, an inverse relationship will also exist whereby the Range object will have the Name as one of its properties.

There is, however, nothing that can be achieved using defined Names that could not also be achieved with direct referencing; it is a simple mechanical process to substitute each Name by its direct reference. All one loses by avoiding the use of Names is structure and meaning. Formulas will no longer be expressed in terms of the business rules they implement but will instead merely reference values by location.

Whereas direct cell referencing provides the means to address any of the 16 billion or so cells on the worksheet, the use of Names limits the address space to the previously declared ranges, in doing so reducing risk. Used in this manner, Names provide the functional equivalent of the VBA

```vba
Option Explicit
Dim myArray(1 To 100, 1 To 2) As Variant
```

which is generally considered good practice within the developer community.

In fact though, there is significant resistance to the use of defined names in the Excel community with a number of well-known publications in which the authors deprecate their use. When single-cell naming practices were studied by McKeever & McDaid, using names like “HMV2009Profits” it was found that they tend to have a detrimental effect on both the ease of creating a spreadsheet solution and upon the detection of errors, although the latter conclusion was originally drawn using inexperienced Excel users. In the opinion of the present author, the idea of having names in one-to-one correspondence with used cells renders them practically valueless; the number of names will be so large that they have little more significance than the cell references they replace and the simplest of tasks such as summing a contiguous range would become a nightmare.

Names come into their own in that they provide structure to the solution process by identifying large blocks of numbers that together represent features of the problem to be addressed at a more abstract conceptual level than may be achieved by referencing individual cells. The number of Names that need to be understood to develop a solution can be orders of magnitude fewer than the count of used cells, so should be far more memorable. The use of names also permits the more complex methods of defining ranges to be considered in order to capture the structure of the problem being addressed. For example, in the similar line item model shown in Section 2.2, the entire model timeline fell within a band of complete columns defined to be “model” that refers to

```
= $F:$X
```

The range representing a single period would be found by using MATCH to give an index value “selectedPeriod”. In this case, the range “inPeriod” comprising the single column containing data for that period would be given by

```
= INDEX( model, 0, selectedPeriod )
```

If one were to have the revenue per period for a list of products identified by the range “revenue”, the intersection
would be a fully dynamic range containing all the sales values for the chosen period and

\[ = \text{SUM(revenue inPeriod)} \]

would be the total revenue for the period across all line-items. The same result might
equally be achieved using direct cell referencing but it lacks the simplicity and readability
of the named version.

A key aspect of the use of Names is that they define the Objects, simultaneously
capturing meaning as well as defining their location and extent, whereas labels merely
annotate with the intention of informing the reader as to the intended significance of
adjacent data but they lack any direct linkage to it. The data is only identified though the
observance of stated or assumed presentation standards which is a vastly weaker concept.
A Name also has the advantage of being computer intelligible; it is part of the Names
collection it can be accessed by computer program and the content can be read and
processed programmatically, as will be shown in Section 4.

In the opinion of the author, Names are grossly underused but there is hope; at least one
company (a firm of actuaries) has implemented an automated system that rejects any
model that has even one cell address reference as part of its quality control process.

2.4 Array formulas

It is often claimed that array formulas are “powerful” and they certainly carry the stigma
of being “advanced”. To my mind that is an erroneous perception on both counts. The
output of any array formula is defined in terms of the cell-by-cell calculations it repre-
sents. Provided one is willing to use helper cells to store intermediate results, it is always
possible to replace an array formula by the cell-by-cell formulas that define it. Array
formulas are not powerful, they only appear that way because they can hold intermediate
results in memory, so making it seem that a final aggregation appears as if by magic. If
one wishes to examine the intermediate calculation, the values generated may be output
as an array formula anywhere on any worksheet without regard to relative positioning.

What array formulas really achieve, is an order of magnitude reduction in the number of
independent formulas that are used to build a solution and, related to that, they provide a
heavily constrained and restricted solution process. At first sight, it might appear that this
loss of flexibility is a major handicap for anyone trying to build a solution from array
formulas. In practice, it is usually the case that valid solutions conform to the constraints
of array formulas whilst errors violate the constraints and require the use of single-cell
formulas and relative references. By way of example, in Section 2.2 all 228 cells of the
product revenue range are given by the single formula

\[ = \text{volume} \times \text{product.price} \]

Even more challenging, some line items are specified as having prices that escalate over
time whilst others are constant. Exception handling is required both for the initialisation
at the first period and to eliminate the escalation where it is not required. Whereas single-
cell strategies allow the formula to be varied from cell to cell, the use of an array formula
demands that the exception handling is built into the formula. Thus the ‘isEscalated?’
flag is an array that applies item by item whilst ‘initialise?’ applies to the first period only.
The range ‘price’ is identical in size to ‘product.price’ but displaced one cell to the left.
The result is an accumulation that runs left to right across the entire range

\[
= \text{IF( isEscalated?, IF( initialise?, price.initial, \text{price initial} ) \times ( 1 + \text{price escalationPerPeriod} ), price.initial )} \]

Copyright © 2016, EuSpRIG European Spreadsheet Risks Interest Group (www.eusprig.org) & the Author(s)
2.5 Formatting and Graphics

My expectation of a model built upon Excel as a platform is that it should allow the client to explore a business scenario; it is not created in order that the user should be able to demonstrate their Excel skills; there may be no requirement even to expose the Excel development interface. In the World Cup predictor application, below, hyperlinked Group tabs replace the normal sheet tabs, the grid has no role in the location of cells and is not visible and the Excel ribbon is hidden.

The worksheet is designed to provide a graphic user interface and it is hoped that the user will have an intuitive understanding of its operation. This concept of ‘affordance’ was introduced by Norman, [1988] in the context of the design of everyday objects and applied to spreadsheets by Hellman [2005]. This design has achieved its objective if the user is able to move between Groups by clicking the appropriate tab and understands that they are to complete the scores either as a prediction or following the matches. The colour palette is based upon the national flag of the host nation Brazil but that is a purely aesthetic consideration.

The user is not expected to interact with the application in the terms of an Excel workbook, nor examine the logic of the tiebreak algorithm; they simply see the result as the Group winner and runner-up names and their associated flags and see the teams transferred to the sheet representing the knockout stage of the competition.

Microsoft offers two very different styles of default format. The most familiar is the grid of grey lines forming an array of white cells limited only by the worksheet boundaries. The more recent exemplar is the gaily-coloured horizontal stripes of the Table object with contrasting headings and filter dropdowns.

![Figure 2 Group stage sheet from the World Cup predictor workbook](Image)
It may be argued that both represent an element of over-formatting; after all while space is not formatted as cells in any other media form and the positive formatting used to outline empty cells merely distracts attention from the significant content appearing elsewhere within the worksheet within data objects. Raffensperger, however, argues that blank cells are not the equivalent of graphic blank space because there is no guarantee that other cells do not reference it and the cell may contain hidden content. While it is clear that the developer can use formatting to conceal information in this manner, it is far more productive to consider how well-chosen formatting may best be used to convey information.

I personally favour using the Excel “normal style” to invoke a background ‘colour wash’ that eradicates all indication of cells. In that way, the developer is almost forced to format any cells they wish to include as part of the model. Most users find this very disconcerting at first because they are conditioned to think in terms of the grid system to identify individual cells and expect to see clear links between the content and the sheet headers. Once one dispenses with direct cell referencing, however, the ability to locate individual cells by coordinates is simply not required; the content is, instead, identified by Name and index within large, structured data Tables and Ranges.

Since the proposed development practice requires all referencing to be done through the use of named ranges, the cell-formatting could be applied automatically by scanning through the names collection using VBA utilities to identify data cells or formula ranges and formatting them using the appropriate style. I tend to favour a white background for input cells but might also introduce a white fill for output since it is these categories of information are of importance to the user and should appear with the greatest clarity. A font difference could be introduced to distinguish the two especially if they appear in close proximity. The intermediate formula cells are formatted with a grey fill to emphasise their secondary role as well as to discourage attempts to change these cells which are normally protected when the workbook is in use.

Equally, the Excel Tables may be regarded as over-formatted to the point of being lurid. Whilst the tables themselves stand out, the content is not as easy to read against a coloured background as it is against white. The direction of the stripes may be relevant as a device for identifying the properties of a record within a list but is less appropriate for a horizontal timeline or a matrix. Similarly, the concept of a row filter is not universally applicable.

3 NAMED FORMULA AS AN AID TO ENCAPSULATION AND REUSE

Names provide clear-cut interfaces between one functional unit of spreadsheet coding and the various functions that reference the data. There is, however, nothing to stop the developer from completely reworking the calculation methodology, provided the meaning is maintained and the datatype is unaltered.

For example, some taxes like sales tax in the US or VAT in the EU vary by jurisdiction. It is quite possible that a spreadsheet solution originally intended to work within a single jurisdiction might eventually need to be generalised to cope with varying values. An absolute reference to a single cell

<table>
<thead>
<tr>
<th>Name</th>
<th>Refers to (absolute range)</th>
</tr>
</thead>
<tbody>
<tr>
<td>taxRate</td>
<td>= $J$16</td>
</tr>
</tbody>
</table>

could be replaced by some form of LOOKUP (one of a number of possible implementations) such as
taxRate = LOOKUP( state, state.codes, state.taxRate )
or = INDEX( state.taxRate, MATCH( state, state.codes, 0 ) )

to create an array of values, one for each line-item. Any array formula that references taxRate will adapt automatically. Interestingly, a named formula may also provide a simple method of regressing the workbook functionality to its legacy mode if that were still required. Just slightly more complicated is

Name Refers to (formula)
taxRate = IF( isLegacyCalculation?,
default.taxRate,
LOOKUP( state, state.codes, state.taxRate )
)

The named formula provides a single point of access to correct or enhance the functionality of the spreadsheet solution by encapsulation of the means of its calculation. Had every tax calculation formula within the workbook made direct reference to the cell containing the original tax rate, the process of updating the workbook would require the addition of a new helper range and would have incurred significantly greater risk.

A more drastic form of modular reuse is made possible by the embedding a copy of a worksheet taken from a template workbook to provide new functionality, as illustrated in the following section. The formulas embedded within the imported worksheet may be viewed in the same light as a call to a library in a procedural code or an additional method within OO programs.

3.1 Merging two sorted lists using an imported Excel module

This example illustrates the use of Names to allow the import of a ‘black box’ module from a separate workbook. The main window of Figure 3 shows the master sheet whilst the module is superposed on it. The key element of the strategy is that the module contains only Names scoped to the worksheet that is to be imported. The private properties ‘private.list.A’ and ‘private.list.B’ are initially linked to dummy data to allow the module to be unit tested independently. By redefining these Names to refer to the public variables held in the master document links are made that achieve the same objective as the Lets and Sets of the Object-Oriented paradigm. Similarly the equivalent of a Get to return the result is achieved by setting the workbook name ‘merged.list’ to refer to the module name ‘=mergeRoutine!value’ essentially mimicking the ‘value’ property of a Class.

The purpose in presenting this example is to demonstrate the way in which well-chosen Names can be used to encapsulate both methods and properties (strictly just range values and formulae). The objective of allowing the functionality of the master workbook to be modified and further developed is far more readily achievable if the existing solution may be reused with little change. The meaning and values of the Names can be changed without requiring changes to be made to the formulas that reference them.

Some degree of caution in handling Names is required though. In particular, it should be recognised that deleting a worksheet to replace it with a fresh version does not remove the Names defined as being local to the sheet [Grossman & Burd, 2015]. This can create difficulty, so it is recommended that the local Names are all deleted before one attempts to delete the worksheet.
What is vital is that every defined Name should be local to the imported worksheet, otherwise undesirable and confusing crosslinks between workbooks will result. Once the new workbook is in place, the local Names used for data input would be reassigned to refer to the appropriate global Names from the host workbook (previously they may have been used to refer to dummy data held within the worksheet so that the functionality could at least be demonstrated). If the calculation simply defines a sequence of array formulas performed in memory there is no problem with needing to access the worksheet holding the new module (it could even be imported as a hidden worksheet) but it is more likely that some scratch workspace would be required in the form of a worksheet range. Whereas named formula will be sized entirely by the referenced input data arrays, any formula held within a range can only expand until it reaches the bounds imposed by the size of range. The presence of cells containing #N/A at the end of the helper ranges should not be a problem either in terms of aesthetics or performance but some care is needed to ensure the developer is warned if the size allocated is insufficient.

Just as the input links are made by allowing the local name to refer to the global name, there must be a suitable mechanism for returning the data objects that result from the calculation. The recommended strategy is simply to document the local names used for the output. Assuming the sheet name is readily associated with the new functionality, the reference to the output using the full name including the sheet name should be acceptable. Other names which should not be touched could be hidden for their protection. Unfortunately Excel does not provide any means of protecting defined names so it may be useful to rebuild the names from documentation if there is any suspicion that any defined names may have been corrupted.
4 AUDITING AND AUTOMATION

4.1 Example: Standard loan repayment model

This example is based upon the FAST modelling guide [Sternberg] that describes how their methodology may be applied to a loan calculator with a choice of debt repayment profiles and includes the possibility of a period of grace in which the borrower is obliged neither to pay debt interest nor principal.

Whilst the present model follows the commentary very closely, basing the name selection upon the text description, it departs from the FAST standard in several respects.

As already described, the output of this workbook is intended to show the salient features of the loan repayment but with no allusion to the Excel grid or cell references. The user is simply not expected to inspect the technicalities of the calculation.

In fact the formula bar shows the Debt Balance as being calculated using the formula

\[
\{ = \text{IF( initialise.loan?,
   loan.amount,
   ←debt.balance + interest.expense - debt.service )} \}
\]

which may be reasonably meaningful to someone who has no knowledge of Excel. It may be more of a challenge for a reviewer that attempts to use conventional auditing techniques. Even the use of the symbols: “ ←” (back arrow) to indicate the previous period; “.” (full stop) to distinguish some property of the root name; and “?” to indicate a Boolean flag will initially appear alien.

4.2 Validating formulas without direct cell referencing

One of the criticisms made of the use of Names is that they insert an additional layer of indirection that makes it harder for the reviewers to navigate to the source of referenced data. That is undoubtedly true but the question is, “Does that matter?” In developing, verifying or auditing traditional workbooks we are conditioned to accept the need to branch about the workbook in the hope that the intent of the formula we are evaluating will be clearer once we see the cells that it references. The trouble is that content of
referenced cell may be as obscure as the original in providing meaning. In a well-ordered workbook it is likely that there will be some form of annotation or table heading that may give a clue to the meaning of the referenced cell but that is far from universally true and it is very easy to destroy the original order of lists in workbooks based upon direct single-cell references or to separate items from the intended annotation.

By comparison, validating a workbook based upon defined names can be a very unfamiliar process. There are plusses though; the number of Names should be orders of magnitude fewer than the number of used cells. Also, provided the cells containing a worksheet formula are always defined as a named range, it is possible to create a complete description of the workbook formulas using Name Manager or other, improved, versions of the tools. Used with care, the names pasted to the worksheet should provide a reasonably good view of the flow of calculation through a sequence of readable and semantically meaningful formulae.

Better still is to use a VBA utility to parse the formulas and use the resulting predecessor graph to order the output for presentation as a navigable diagram.

In Figure 5, the diagram is set with its focus upon the debt balance and shows the formula contained within the range. The connectors provide links to the ranges referenced within the formula. It may appear a little surprising that the ‘debt.balance’ should appear with no dependents (i.e. nothing is shown to the left as referencing it). That is simply because the balance references are all, in fact, to the previous period’s balance ‘←debt.balance’, as shown in Figure 6.

The other element of evaluating the spreadsheet solution is to check that the Named Ranges actually contain the data that one is led to expect from the name; the spectre of a name being misapplied is often raised as an issue, though I would argue the problem is the other way round. The Name defines the data structure; it may be incorrectly dimensioned but it cannot, by definition, be incorrectly placed. The question should be, “Given the Named Range, is the data appropriately entered?” This documentation tool addresses the problem by providing the ‘down arrow’ control, which takes one to the target workbook with the relevant range selected on the worksheet.
5 CONCLUSIONS

The point of this paper is not to suggest that developers should change the habits of a lifetime and adopt the strategy as outlined. It is more the intention to raise the awareness that the adoption of less common approaches within Excel can lead to a coherent solution strategy in which the problem is solved by a sequence of formulas resembling the steps of a programmed language. Individually the techniques can deliver benefit when building models and the developer ought to be aware of the possibilities even though they may choose to stay with default techniques that will be more familiar to their clients.

The author has developed spreadsheet solutions with array formula and named ranges for a number of years now but the experience within the community of the risks of such an approach is miniscule and the scarcity of Excel specialists capable of maintaining such workbooks is itself a risk that must be mitigated. It is certain that the end user is likely to have far less opportunity to ‘meddle’ but, there, improved workbook integrity must be balanced against possible client dissatisfaction.

6 REFERENCES

Bewig Philip L “How do you know your spreadsheet is right? Principles, Techniques and Practice of Spreadsheet Style”, EuSpRIG July, 2005


ABSTRACT

Spreadsheets are widely used in industry, even for critical business processes. This implies the need for proper risk assessment in spreadsheets to evaluate the reliability and validity of the spreadsheet’s outcome. As related research has shown, the risk of spreadsheet errors is strongly related to the spreadsheet’s complexity. Therefore, spreadsheet researchers proposed various metrics for quantifying different aspects of a spreadsheet in order to assess its complexity. However, until now there is no shared understanding of potential complexity drivers for spreadsheets. The present work addresses this research gap by proposing a conceptual model integrating all aspects which are identified by related literature as potential drivers to spreadsheet complexity. In this sense, this model forms the foundation for a structured definition of complexity metrics, and thus enhances the reproducibility of their results. At the same time, it forms the foundation for identifying further applicable complexity metrics from other scientific domains.

1 INTRODUCTION

Spreadsheets are the Swiss Army Knife for decision support in enterprises: They empower business users from different domains to manage, analyze, and visualize their domain-specific data for deriving – very often business-critical – decisions [Panko and Port, 2012; Reschenhofer and Matthes, 2015]. Spreadsheets are used in a variety of application areas, e.g., for financial reporting [Panko, 2006] and workload planning [Pemberton and Robson, 2000]. Due to a wide dissemination of spreadsheets across nearly all business domains [Scaffidi, Shaw and Myers, 2005; Bradley and McDaid, 2009], but their invisibility to the corporate IT departments, Panko and Port [Panko and Port, 2012] call spreadsheets the “dark matter” of IT. However, spreadsheets are not only widely used in a plethora of application areas, but also important and critical for companies [Chan and Storey, 1996; Gable, Yap and Eng, 1991; Hall, 1996]. Therefore, errors in spreadsheets can have significant negative impact [Caulkins, Morrison and Weidemann, 2007; Powell, Baker and Lawson, 2009]. This becomes even more critical since numerous studies [Panko, 2000; Powell, Baker and Lawson, 2008] have shown that spreadsheets are indeed very error-prone. Even worse, users tend to overlook the risk of errors in their spreadsheets since they are not able to assess this risk [Hall, 1996]. Therefore, they often blindly trust the respective outcomes and thus make potentially costly decisions.

Spreadsheets as a generic measure for complex calculation on data are heavily used in the financial industry, e.g., determination of risks based on predefined formulas, measurement of equity requirements using given regulation policies such as Basel II, respectively Basel III, and the Sarbanes-Oxley-Act (SOX). Obviously, spreadsheets...
have become very popular for banks and insurances [Hall, 1996; Janvrin and Morrison, 2000]. Supervising authorities responsible for auditing, such as the German BaFin (engl. Federal Financial Supervisory Authority), are also aware of the importance of spreadsheets in the financial industry [Bretz, 2012]. Therefore, supervisors have a particular focus on spreadsheets, their structure, and calculation. Based on their experience they also state that complex calculations in spreadsheets are vulnerable to errors and can become an additional source for risks [Bretz, 2009].

The lack of separation between data and logic and the usage of difficult formulas with many dependencies hardens the problem of creation, controlling and maintenance of spreadsheets [Bretz, 2009]. The analysis of spreadsheets regarding complexity based on a conceptual model is a step towards detailed investigation of the structure and semantics of spreadsheets as sources of risks. An important driver for the risk of errors in spreadsheets is the complexity of its design which mainly constituted by the formulas and dependencies between cells [Teo and Lee-Partridge, 2001; Bregar, 2004; Janvrin and Morrison, 2000]. In this sense, the complexity of a spreadsheet correlates with its understandability, and hence is an indicator for the probability of errors. Therefore, determining the complexity helps during the assessment of errors and risks [Bregar, 2004; Hermans, Pinzger and van Deursen, 2012b].

In many domains, metrics are already common to assess the complexity or understandability of artifacts. Analyzing linguistics properties has always been an objective to gain insights to a particular text or discourse or to the structure of written language in general [Graesser and McNamara, 2011]. Several approaches exist aiming at the determination of qualitative and quantitative textual properties [Köhler, 2005]. Linguists have always tried to measure understandability and readability as objective and measurable indices. Those have been used in the assessment of texts for education [Flesch, 1948], journalism, military, healthcare [DuBay, 2004], and recently also in the legal domain [Waltl and Matthes, 2014]. Thereby, they have developed and reused several metrics representing the properties of a text quantitatively. Assessing textual complexity can be achieved by a set of metrics, rather than by one single metric. Consequently, the resulting metrics developed by linguistics can be adapted and reused in other domains in order to investigate and quantify complexity of formulas and dependencies in spreadsheet formulas.

The objective of the present work is to propose a conceptual spreadsheet model forming the foundation for identifying applicable complexity metrics in a structured way, and thus enhances the reproducibility of metric results through a unified spreadsheet model. In this sense, this model captures all aspects of spreadsheets, which are potential drivers for its complexity. Furthermore, this spreadsheet model can serve as a starting point for the identification of metrics from other domains. Based on this model, we refine metrics from the domain of software engineering which were already adapted to spreadsheets, and identify metrics from linguistics, which are also applicable for assessing the complexity of spreadsheets. Finally, we want to answer the question how complex today’s spreadsheets are. This leads to the following research questions constituting the present work’s contribution:

- What is a spreadsheet model capturing potential complexity drivers for spreadsheets, and which enables the formal definition of complexity metrics?
- How can the metrics from software engineering and linguistics be defined based on the proposed conceptual model?
- According to those indicators, how complex are today’s spreadsheets, and how do those metrics correlate to each other?
In order to answer those research questions, the remainder of this paper is organized as follows: Section 2 summarizes related work in the field of spreadsheet research, in particular about risk assessment in spreadsheet. Thereafter, Section 3 describes the applied research methodology. In Section 4 we propose a model for the definition of spreadsheet complexity metrics and thus the answer for the first research question. Based on this model, Section 5 answers the second research question by describing a set of metrics from the domains of software engineering and linguistics, which then are applied to two spreadsheet corpora as shown in Section 6.

2 RELATED WORK

Due to the vulnerability of spreadsheets to errors and the potentially high negative impact of those, the assessment of spreadsheets regarding complexity has been subject of recent research. Hermans et al. [Hermans et al., 2012b] correlate the risk and complexity of spreadsheets with the understandability of formulas. They derived a set of metrics for measuring formula understandability by conducting interviews with spreadsheet experts. Those metrics capture both the complexity of formulas and their placement within the spreadsheet. Furthermore, they evaluated their work by correlating the metrics to the perceived understandability by spreadsheet experts. Although Hermans et al. come up with a list of important factors to understand a spreadsheet, they do not provide a formal model based on these findings, which in turn is the main contribution of the present paper. Additionally, while the proposed metrics only assess the understandability at the level of formulas, the present work’s spreadsheet model allows the definition of metrics at higher levels, e.g., on the level spreadsheets which can capture the interrelations between formulas and worksheets.

In another publication, Hermans et al. [Hermans, Pinzger and van Deursen, 2014] investigate code smells within spreadsheet formulas. A “code smell” is a concept from the software engineering domain, describing symptoms which are likely to produce errors in execution and maintenance. By adapting the concept of code smells to the spreadsheet domain, it is possible to generate risk maps visualizing code smells and thus to indicate the exposure to risk at certain locations within a spreadsheet. Again, the objective of the present paper is to provide a formal foundation for the definition of those code smell metrics.

While the present paper seeks for a conceptual model for measuring spreadsheet complexity and assessing spreadsheet risk, Bregar [Bregar, 2004] defines complexity metric based on a mathematical model. He proposes a set of metrics, which are mostly adapted from the domain of software engineering, and provides a mathematical definition for most of them. However, while this type of formal definition enables a reliable reproducibility, an underlying conceptual model for spreadsheet complexity – as proposed by the present work – explicitly outlines the aspects of spreadsheets, which are considered to be drivers for the complexity and risk. Hence, contrary to the mathematical model by Bregar [Bregar, 2004], the conceptual model facilitates the identification and derivation of additional metrics from other domains, e.g., linguistics (as described in Section 5 of this paper).

Similarly, Hodnigg and Mittermeir [Hodnigg and Mittermeir, 2008] formally define complexity metrics based on a mathematical and graph-based notation, whereas the graph describes the dependencies between cells determined by cell references within formulas. The purpose of those metrics is to visualize the conceptual model of the spreadsheet in order to facilitate a spreadsheet’s maintenance. They also distinguish
between two types of metrics, namely general indicators capturing data and structure-related aspects of the spreadsheet, and formula complexity metrics which basically relate to the metrics proposed by Hermans et al. [Hermans et al., 2012b] and Bregar [Bregar, 2004]. And again, while they provide formal definitions of metrics for measuring the complexity of spreadsheets, the purpose of the present paper’s conceptual spreadsheet model is to explicitly capture the complexity drivers of spreadsheets. Hence this model can serve as a foundation for the works by Hodnigg and Mittermeir [Hodnigg and Mittermeir, 2008] as well as Bregar [Bregar, 2004].

Cunha et al. [Cunha, Fernandes, Peixoto and Saraiva, 2012] present a quality model for spreadsheets based on a common software engineering standard. Thereby they define metrics for quality aspects including functionality, maintainability, and usability. While many of those metrics are covered by the works of Hodnigg and Mittermeir [Hodnigg and Mittermeir, 2008] as well as Bregar [Bregar, 2004], the usability-related metrics by Cunha et al. also capture visual attributes (e.g., color-coding for different cell types) as aspects affecting the understandability and thus the perceived complexity of spreadsheets. Therefore, this work provides additional input for the derivation of the conceptual model as described in Section 4.

Seila and Banks [Seila and Banks, 1990] propose simulation as an alternative approach for assessing the risk of errors in spreadsheets. Thereby they simulate an uncertainty in input variables, and thus observe the dynamics of the investigated spreadsheet in particular. The simulation reveals how vulnerable the spreadsheet’s output in case of erroneous inputs. However, the conceptual model as described later on in the present work does not capture the dynamics of spreadsheets, but – similar to Hermans et al. [Hermans et al., 2012b], Bregar [Bregar, 2004], Hodnigg and Mittermeir [Hodnigg and Mittermeir, 2008], and Cunha et al. [Cunha et al., 2012] – focuses on the structural aspects of spreadsheets.

Shubbak and Thorne [2016] present a software tool for identifying and assessing the risk of a given spreadsheet by taking into account the spreadsheet’s nature, importance, use, and complexity. While they already use some basic complexity measures, their work can profit directly from the model as proposed by the paper at hand, since it enables a structured and model-based approach to complexity metrics.

3 RESEARCH METHOD

For deriving the conceptual model for spreadsheet complexity as described later on in Section 4, we applied the design-science research method as defined by Hevner et al. [Hevner, March, Park and Ram, 2004], and as depicted in Figure 1.

The environmental part of the Information Systems Research Framework [Hevner et al., 2004] was already introduced in Section 1 by motivating the need for risk assessment in spreadsheets and a structured approach to the same. Furthermore, we design the conceptual model for spreadsheet complexity on the foundations of the existing knowledge base (see Section 2) which already describe different indicators and aspects of complexity in spreadsheets. In this sense, the model as presented in this work integrates the findings of the related papers and proposes a comprehensive conceptual model capturing all aspects, which were identified as potential drivers for complexity in spreadsheets. The conceptual model for spreadsheet complexity forms the artifact as defined by the Information Systems Research Framework. Its design is affected by both the relevance of the environmental part and the rigor through the knowledge base [Hevner et al., 2004]. We use an analytical and descriptive evaluation...
method to justify and argument for the conceptual model’s utility and correctness. For this, we apply the already proposed spreadsheet complexity metrics on the one hand, and apply them to two common spreadsheet corpora as described in Section 6.

Figure 1: Overview over this work’s research methodology based on the Information Systems Research Framework by Hevner et al. [Hevner et al., 2004].

In addition to the design-science research method, we also follow a quantitative descriptive content analysis method [Neuendorf, 2002] – in particular in the evaluation phase of the Information Systems Research Framework. Thereby, the quantitative metrics as described in Section 5 indicating formula complexity were derived from the scientific discipline of linguistics. Consequently, we methodologically transfer established insights from this discipline to determine metrics representing complexity of spreadsheets based on the proposed conceptual model for measuring spreadsheet complexity.

4 A CONCEPTUAL MODEL FOR SPREADSHEET COMPLEXITY

Based on the design-science research method as well as the related work about measuring and assessing the complexity of spreadsheets, we propose the conceptual model in Figure 2, capturing and integrating all aspects of spreadsheets, which were already identified as potential drivers for complexity.

The concept of worksheets not only forms a basic part of a spreadsheet’s structure, but is also an important aspect in the context spreadsheet complexity. For example,
Hermans et al. [Hermans, Pinzger and van Deursen, 2012a] use the worksheet information for determining interdependencies between them. A high coupling between worksheets indicates a code smell and thus implies an increased complexity.

A worksheet consists of cells that are organized in a two-dimensional grid. As shown by Rajalingham et al. [Rajalingham, Chadwick, Knight and Edwards, 2000], the placement of formulas and location of cells in general has a huge impact on the perceived complexity and thus understandability of spreadsheets. Therefore, we include the coordinate of the cell within a worksheet as an attribute of the Cell class. As suggested by Cunha et al.[Cunha et al., 2012], a cell’s visual properties, e.g., its color, contribute to a spreadsheets understandability and perceived complexity, wherefore we add the class VisualProperty to the model. Due to a lack of evidence from related literature we do not specify this concept in detail, e.g., by defining which kind of visual properties are affecting the understandability of a spreadsheet (apart from the already mentioned color of cells [Cunha et al., 2012]).

Hodnigg et al. [Hodnigg and Mittermeir, 2008] differentiate between different kind of cells, namely empty cells, label cells, and value cells. Empty cells do not even contain any data or text. Label cells are not referenced by formulas and only serve as documentation and description of value cells, which in turn contain the actual data of the spreadsheet. Value cells can be further classified into cells containing formulas and cells serving as input fields for the spreadsheet data. Thereby, the InputValueCell class captures the type of its data, which is also an aspect potentially affecting the spreadsheet’s complexity [Cunha et al., 2012]. In this context, the ValueType enumeration containing possible value types strongly depends on the actual spreadsheet software and has to be adapted accordingly when applying the model to a certain spreadsheet tool. The class FormulaCell represents cells which compute their value based on an expression – the formula. Hermans et al. [Hermans et al., 2012b], Bregar [Bregar, 2004], and Hodnigg et al. [Hodnigg and Mittermeir, 2008] identify the level of nesting of formulas – with respect to the formula’s representation as an abstract syntax tree (AST) – as a complexity driver for spreadsheets. This AST is realized in the model in Figure 2 by the Expression class and its reflexive association. Thereby the formula refers to exactly one expression, which forms the root of a potentially nested tree structure of different kinds of expressions.

The number of conditionals (or decision count) within a formula is also considered to be a measure for its complexity [Bregar, 2004]. The occurrence of functions like lookup and offset adds further complexity to the formula [Hodnigg and Mittermeir, 2008]. We define the Function class as a concrete expression to capture those different kinds of functions in spreadsheet formulas. Furthermore, we differentiate between Operator (e.g., plus and minus for the arithmetic addition and subtraction), Constant of different types, and Parenthesis expressions in order to capture the diversity and nestedness of formulas. Again, the set of operator types and the set of value types strongly depends on the actual spreadsheet software.

As suggested by Hermans et al. [Hermans et al., 2012b], Bregar [Bregar, 2004], and Hodnigg et al. [Hodnigg and Mittermeir, 2008], the dependencies between formulas are one of the main drivers for spreadsheet complexity. Therefore, the conceptual model as proposed in this work captures those dependencies by the Reference and Range classes. The former one describes references to single cells, while the latter one represents a reference to a one- or two-dimensional cell block. According to Hermans et al. [Hermans et al., 2012b], range references have an even higher impact to a spreadsheet’s complexity than references to a single cell. Furthermore, in another
work [Hermans et al., 2012a] they suggest that cell references by name have to be differentiated from references by grid coordinates. For this reason, we add the Boolean attribute \texttt{byName} to the \texttt{Reference} class.

The integrated conceptual model in Figure 2 describes a network of cells, whereas most of its nodes contain an abstract syntax tree representing a formula. This model not only integrates knowledge about aspects for spreadsheet complexity based on existing complexity metrics, but also serves as a starting point for identifying new ways and metrics for quantifying complexity in related domains.

5 APPLICATION OF THE MODEL TO DEFINE COMPLEXITY METRICS

Measures of complexity are well known in various scientific disciplines to get a theoretical and empirical insight into a system and its behavior. However, our research focuses on two disciplines that investigate different artifacts, namely software engineering and linguistics. Both deal – at least to some extent – with the analysis of man-made objects, which is software on the one hand, and language, i.e. text, on the other hand. The following two sections summarize the integration of the two disciplines regarding their understanding and measurement of complexity and show how these fit to our conceptual model of spreadsheet complexity.

5.1 Metrics in Software Engineering

Over the years, software engineering (SE) has become a mature scientific discipline dealing with a variety of challenges, i.e. planning, development, maintenance, etc. Metrics in SE are used by industry and research to understand and improve both software products and software development processes. Various metrics exist covering different aspects of SE processes and artifacts. However, validation of such metrics is difficult to generalize and mostly done ad-hoc [Meneely, Smith and Williams, 2013]. Therefore, we refer to established SE metrics, which were already adopted to the area of spreadsheets by Hermans et al. [Hermans et al., 2012b], Bregar [Bregar, 2004], Hodnigg et al. [Hodnigg and Mittermeir, 2008], and Cunha et al. [Cunha et al., 2012]. We show that existing and well-studied metrics can be redefined through our model (see Table 1). We exemplarily select 16 representative metrics covering all aspects of our model, except VisualProperty class, since there is no respective concrete metric definition in related literature.

5.2 Metrics in Linguistics

As mentioned in the introduction, linguists have eagerly defined metrics and indicators representing linguistics features on various levels. Graesser and McNamara et al. have defined over 100 different indices for evaluation of text and discourse [Graesser, McNamara, Louwerse and Cai, 2004; McNamara, Graesser, McCarthy and Cai, 2014]. Thereby they classified their metrics regarding the complexity and difficulty to which they contribute. This classification covers categories such as “Descriptive”, “Readability”, “Referential Cohesion”, “Lexical Diversity”, “Syntactic Complexity”, etc. [McNamara et al., 2014]. This classification also contributes to a common understanding of a multilevel framework for discourse comprehension [Graesser and McNamara, 2011]. The levels cover the surface code, textbase, situation model, rhetorical structure, and pragmatic communication level. This structure refers to the rationale, that distinct linguistic properties influence the text on a different level, such as different entities in spreadsheets contribute to its complexity differently. However, the two top-most levels, namely rhetorical structure and
pragmatic communication level cannot easily be transferred to the domain of spreadsheet complexity. Surface code, textbase, and situation model on contrary cover technical and structural properties of text and can be reused for analysis of spreadsheet formulas based on a constructive model. Based on the metrics provided by Grasser and McNamara et al. [McNamara et al., 2014] we selected a subset for the analysis and adapted those to the domain of spreadsheet formulas (see Table 2).

Table 1. A selection of complexity metrics from the software engineering domain.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average AST depth per formula</td>
<td>The average depth of the abstract syntax tree which is formed by the Expressions of a FormulaCell.</td>
</tr>
<tr>
<td>Max AST depth per formula</td>
<td></td>
</tr>
<tr>
<td>Number of formula cells</td>
<td>The size of the spreadsheet which is determined by the number/ratio of different kind of Cells, e.g., ValueCell and FormulaCell.</td>
</tr>
<tr>
<td>Ratio of formula cells to non-empty cells</td>
<td></td>
</tr>
<tr>
<td>Number of input cells</td>
<td></td>
</tr>
<tr>
<td>Ratio of input cells to non-empty cells</td>
<td></td>
</tr>
<tr>
<td>Ratio of formula cells to input cells</td>
<td></td>
</tr>
<tr>
<td>Number of distinct formulas</td>
<td>Number of distinct FormulaCells which have an equally structured AST which only differs with respect to its References.</td>
</tr>
<tr>
<td>Average fan-out per formula</td>
<td>Number of incoming and outgoing references of a FormulaCells.</td>
</tr>
<tr>
<td>Max fan-out per formula</td>
<td></td>
</tr>
<tr>
<td>Average fan-in per formula</td>
<td></td>
</tr>
<tr>
<td>Max fan-in per formula</td>
<td></td>
</tr>
<tr>
<td>Average number of conditionals per formula</td>
<td>Number of Boolean expressions within one FormulaCell, e.g., functions with the names IF, COUNTIF, SUMIF, etc.</td>
</tr>
<tr>
<td>Max number of conditionals per formula</td>
<td></td>
</tr>
<tr>
<td>Average spreading factor per formula</td>
<td>Maximal Euclidian distance between References in a Formula. In this sense, rowIndex and columnIndex of the Cell and index of the Worksheet represent the x, y, z coordinates of a three-dimensional space.</td>
</tr>
<tr>
<td>Max spreading factor per formula</td>
<td></td>
</tr>
</tbody>
</table>

Table 2. Selected metrics from the linguistics domain.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average number of functions per formula</td>
<td>Capture the lexical diversity of a FormulaCell by counting the number of used Functions.</td>
</tr>
<tr>
<td>Max number of functions per formula</td>
<td></td>
</tr>
<tr>
<td>Average number of distinct functions per formula</td>
<td>Capture the lexical diversity of a FormulaCell by counting the number of used Functions, without counting duplicates.</td>
</tr>
<tr>
<td>Max number of distinct functions per formula</td>
<td></td>
</tr>
<tr>
<td>Average number of elements per formula</td>
<td>Capture the connective incidence of a FormulaCells by counting the number of expressions within its AST.</td>
</tr>
<tr>
<td>Max number of elements per formula</td>
<td></td>
</tr>
</tbody>
</table>

The selection of six linguistic metrics enables the quantification of properties of spreadsheet formulas. Additionally, the number of references in texts are well-known linguistic metrics. However, since they are already covered through Software Engineering metrics, they are omitted in this table. Our selection represents the most fundamental quantifications that can be used straightforward.
6 APPLICATION OF METRICS

Based on the conceptual model for spreadsheet complexity as described in Section 4, we applied the metrics from Section 5 to the EUSES and Enron spreadsheet corpora. The application of the metrics enables the comparison of them with respect to different aspects of spreadsheet complexity on the one hand, and the determination of correlations between potential drivers to complexity on the other hand. The EUSES spreadsheet corpus [Fisher and Rothermel, 2005] contains over 4,000 spreadsheets which were mainly crawled with search engines. The Enron corpus [Hermans and Murphy-Hill, 2014] consists of over 15,000 industrial spreadsheets gathered from the Enron Corporation. Hermans et al. [Hermans and Murphy-Hill, 2014] already state the spreadsheets of the Enron corpus are significantly more smelly and thus they are considered to be more complex than those from the EUSES spreadsheet corpus. This claim can be partially supported by the statistics in Table 3, in particular by the metrics from the domain of SE. For example, the values for the fan-in and fan-out metrics capturing the incoming and outgoing references of formula cells are much higher for the spreadsheets of the Enron corpus than for those of the EUSES corpus. Interestingly enough, though, there is no significant difference between spreadsheets of those two corpora from a linguistics perspective.

Table 3. Average complexity values for the EUSES and Enron spreadsheet corpora

<table>
<thead>
<tr>
<th></th>
<th>EUSES</th>
<th>Enron</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ratio of spreadsheets with formulas</td>
<td>43 %</td>
<td>58 %</td>
</tr>
<tr>
<td><strong>Software Engineering Metrics</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average AST depth per formula</td>
<td>1.92</td>
<td>1.51</td>
</tr>
<tr>
<td>Max AST depth per formula</td>
<td>4.63</td>
<td>4.47</td>
</tr>
<tr>
<td>Number of formula cells</td>
<td>350</td>
<td>2107.53</td>
</tr>
<tr>
<td>Ratio of formula cells to non-empty cells</td>
<td>0.23</td>
<td>0.30</td>
</tr>
<tr>
<td>Number of input cells</td>
<td>4931.90</td>
<td>11170.50</td>
</tr>
<tr>
<td>Ratio of input cells to non-empty cells</td>
<td>1.55</td>
<td>5.38</td>
</tr>
<tr>
<td>Ratio of formula cells to input cells</td>
<td>3.63</td>
<td>2.54</td>
</tr>
<tr>
<td>Number of distinct formulas</td>
<td>3.13</td>
<td>10.50</td>
</tr>
<tr>
<td>Average fan-out per formula</td>
<td>167.94</td>
<td>473.27</td>
</tr>
<tr>
<td>Max fan-out per formula</td>
<td>476.79</td>
<td>4709.88</td>
</tr>
<tr>
<td>Average fan-in per formula</td>
<td>0.93</td>
<td>7.70</td>
</tr>
<tr>
<td>Max fan-in per formula</td>
<td>9.20</td>
<td>50.53</td>
</tr>
<tr>
<td>Average number of conditionals per formula</td>
<td>0.09</td>
<td>0.07</td>
</tr>
<tr>
<td>Max number of conditionals per formula</td>
<td>0.27</td>
<td>0.33</td>
</tr>
<tr>
<td>Average spreading factor per formula</td>
<td>148.13</td>
<td>374.80</td>
</tr>
<tr>
<td>Max spreading factor per formula</td>
<td>350.94</td>
<td>1522.60</td>
</tr>
<tr>
<td><strong>Linguistics Metrics</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average number of functions per formula</td>
<td>0.53</td>
<td>0.48</td>
</tr>
<tr>
<td>Max number of functions per formula</td>
<td>1.00</td>
<td>1.17</td>
</tr>
<tr>
<td>Average number of distinct functions per formula</td>
<td>0.51</td>
<td>0.47</td>
</tr>
<tr>
<td>Max number of distinct functions per formula</td>
<td>0.88</td>
<td>0.93</td>
</tr>
<tr>
<td>Average number of elements per formula</td>
<td>5.43</td>
<td>4.25</td>
</tr>
<tr>
<td>Max number of elements per formula</td>
<td>14.64</td>
<td>15.71</td>
</tr>
</tbody>
</table>

As shown by the histogram in Figure 3, the ratio of formula cells metric indicates that most spreadsheets have a low complexity. However, at the same time there is also a considerable amount of spreadsheets having high values for this metric wherefore they are considered to be very complex. This distribution of the complexity is even more significant for the average fan-out metric in Figure 4.
Computing the correlation coefficient for each pair of metrics based on the almost 20,000 spreadsheets of both corpora reveals that only those metrics correlate to each other which capture the same concepts of the conceptual model in Figure 2 (e.g., number of input cells and average fan-out). This is interesting in the sense that the aspects captured by the conceptual model for spreadsheet complexity are mostly independent from each other, and a high complexity with respect to a certain aspect does not imply a high complexity with respect to another one.

7 CONCLUSION

This paper presents an approach that fits neatly into the research dealing with the analysis of spreadsheets complexity. It proposes a conceptual and integrated model of spreadsheets allowing a detailed investigation of today’s spreadsheets, which is the answer to the first research question from Section 1. Beyond the qualitative structuring of attributes and properties of spreadsheet formulas, the paper argues for a quantification of those. Thereby, it transfers well-studied and established results from two different but related domains dealing with the determination of complexity and understandability of man-made artifacts, namely software engineering and linguistics. In both disciplines, metrics to evaluate complexity are common and have been repeatedly reported as useful. Our paper maps those metrics to our model in a way that relevant aspects of spreadsheet complexity are covered by at least one metric. Thereby, we answer the second research question raised in the introduction. We evaluate these metrics by their application to a large set of existing spreadsheets, namely the EUSES and the Enron corpus. The empirical evaluation shows that many spreadsheets used in industry do not even contain formulas. However, if they do, the distribution of the ratio of formula cells metric indicates that most have a supposedly low complexity, whereas there are also numerous spreadsheets with an increased complexity. The average fan-out distribution shows a similar distribution: While most spreadsheets have only a little average fan out (<5), there is also a considerable amount of spreadsheets with a very high value for this metric (>30).

Ongoing research may integrate temporal aspects into the conceptual model for spreadsheet complexity, e.g., the temporal evolution of complexity metrics, but also the impact of changes of a spreadsheet to its complexity. Thereby, the model would be able to not only capture static and structural aspects of spreadsheets, but also the respective dynamics. Apart from this, the proposed model does not capture macros which usually have a high impact on the understandability and complexity of spreadsheets. This would be an additional aspect which should be added to a holistic complexity model of spreadsheets. Furthermore, the proposed conceptual model can serve as a foundation for deriving new complexity measures through adoption from other scientific domains. For example, due to the network structure of formulas as defined by the proposed conceptual model, sophisticated graph algorithms could be applied in order to reveal structural aspects. Similarly, methods from social network

Figure 3. A histogram showing the number of spreadsheets by the ratio of formula cells.

Figure 4. A histogram showing the number of spreadsheets by the average fan-out per formula.
analysis could be adapted to the field of spreadsheets. Another example for metrics which can be adapted to the domain of spreadsheets are measures describing the diversity of elements within a certain context by applying the Shannon entropy, e.g., the entropy of functions or operators within a function. As these examples suggest, our conceptual model for spreadsheet formulas can serve as a theoretical foundation and starting point for upcoming metrics, measurements, and analysis in general.

8 REFERENCES


Abstract

This paper discusses the risks and potential impacts of spreadsheet errors in scientific research data in a Neuroscience research centre in the UK.

Spreadsheets usage in neuroscience, or indeed any medical discipline, is a largely unreported area of spreadsheet research. This paper presents a case study exploring the possible risks and impacts of spreadsheet errors in the neuroscience research centre at the University of Newcastle. Data was collected using an online questionnaire with 17 participants and two detailed semi-structured interviews.

The analysis highlights that errors in research data may lead to severe impacts such as misleading science and damaged personal and organisational reputations. In addition, many risks factors arise from using spreadsheets such as inadequate design and a lack of training.

Spreadsheets are used widely in business and the impacts and risks in these fields have been studied and highlighted in detail. However, scientific research and spreadsheets have also a significant relationship that has not been clarifed. The paper also draws out the similarities in spreadsheet practice between the scientific and business communities.
1.0 Introduction

This paper discusses the risks and potential impacts of spreadsheet errors in scientific research data in a Neuroscience research centre in the UK.

Spreadsheets usage in neuroscience, or indeed any medical discipline, is a largely unreported area of spreadsheet research. Although little is published on this subject, it seems likely that the medical discipline will make extensive use of spreadsheets for a variety of clinical and non-clinical activities. This assumption is based on the observation that spreadsheet use is ubiquitous in almost all areas of business, government and education. To that end, this paper aims to answer the following questions:

1. To what extent are spreadsheets used by the Neuroscience Research Centre at the University of Newcastle for data processing and decision making activities?
2. How are spreadsheets planned, developed and maintained by the research centre?
3. What are the specific risks and potential sources of error arising from Neuroscience spreadsheet use?
4. What is the likely impact of spreadsheet risks and error on neuroscience research data?

Data was collected using an online questionnaire with 17 participants and followed up by two detailed semi-structured interviews. Given the small sample size, only limited generalisations to the wider neuroscience and medical communities can be made. However, this research will gather some specific, interesting data and will highlight important areas for further research.

1.1 Neuroscience

Neuroscience was defined by the U.S. congress as “the study of the nervous system, how it affects behaviour, and how it is affected by disease. The goal of neuroscience is to define and understand the continuum from molecular to cell to behaviour” (Congress, 1984).

Several biological and human cognitive developments have been discovered in neuroscience research. Furthermore, many medical and mental health issues have been treated by its findings. Neuroscience did not gain the appropriate attention as a new science until the last decade.

In fact, studying the brain and the nervous system sets the foundation for many other studies. Psychology for example is a wide and rich discipline and many recent neuroscience research studies and experiments on the brain have explained many irrational attitudes and behaviours through studying precursor neural circuit activity (Diamond and Amso, 2008). Moreover, the impact of neuroscience on the wider medical discipline has been significant. Neuroscience research has made important contributions to the understanding of many medical conditions and has highlighted new avenues of research in multiple medical disciplines (Conn, 2008).
1.2 Research Data confidentiality and Integrity

As with other areas of research, confidentiality and data protection are of the upmost importance in neuroscience. There are several sets of standards and recommendations to that end such as the guidelines published by The National Human Research Protections Advisory Committee (NHRPAC) (NHRPAC, 2002). These guidelines highlight the importance of research data and therefore, the risks associated with it. However, there are no explicit guidelines for spreadsheets.

Research data services at Wisconsin-Madison University (WISC, 2014) discuss spreadsheets risks, errors and research data. They also published a set of guidelines and recommendations when using spreadsheets in order to minimize these errors in research data. Although these guidelines offer some basic advice, they are far from detailed.

- Research data in the organization is very important and losing it could lead to many results such as:
  - Losing valuable experimental data or simulation results.
  - Drawing incorrect conclusions from data leading to negative research impact.
  - Damage to the reputation of the organization conducting the research.
  - Potential for regulatory or legal action from governmental or other institution.
  - Requirement for corrective actions or repairs which would take years of research.
  - Violation of University or organization mission, policy, or principles.

This lack of detailed advice is in contrast to the regulation and control of spreadsheet applications in other medical fields such as the pharmaceutical industry in the United States. The pharmaceutical industry has highly specific controls placed on the use of spreadsheet applications for data analysis, reporting and decision making under Title 21 of the Code of Federal Regulations Part 11 (21 CFR 11). Spreadsheets are explicitly mentioned in this legislation which demands companies provide evidence of: audits, validation, electronic signatures and documentation for any software artefact including spreadsheets. The legislation also dictates that electronic artefacts such as spreadsheets be stored in a secure server so that once the spreadsheet has been created and audited, it cannot be changed without authorisation.

1.3 Neuroscience data management strategies

There are many approaches available to manage and distribute research data. Spreadsheets are one of the most popular data manipulation and analysis tools used among researchers.

Lacroix and Critchlow (2003) discuss spreadsheets as one of the two popular data management strategies in research. According to this research, spreadsheets offer quick data browsing, simple mathematical operations and easy distribution to collaborators. However, they also highlighted many points as disadvantages, in particular the lack of data validation when entered which can increase the possibility of errors.

In their research (Anderson, et al., 2007), Anderson and his colleagues interviewed 286 researchers from different research fields including neuroscience. The majority of the researchers interviewed admitted that they rely on general-purpose applications such as spreadsheets to manage their data. The main reason for this reliance is the simplicity of interface, the range and power of data manipulation.
tools and its short learning curve. According to one of the interviewees “Yeah, the spreadsheet has been our main workhorse, unfortunately”.

Anderson et al. (2007) also note that several of the interviewees have encountered problems when using spreadsheets:

“Well, we have multiple spreadsheets - that’s one of the problems. We sort of have a master spreadsheet ... We try to minimize it as much as we can, but I think that’s a major problem.”

“However, that exceeds the capabilities of the spreadsheet. Spreadsheet really bogs down any time you get past say 20,000 individual cells with columns.”

“Well, it’s very cumbersome, I can’t print anything, I’d have to paste it together. I end up just doing a freeze frame so that I can scroll this way.”

Although Anderson et al (2007) do not mention the wealth of research on spreadsheet error, it is clear from the quotes on spreadsheet problems that users experience problems with concurrency, computational power and usability.

Spreadsheets are widely used in research for tabulating, analysing and sharing data; “Recent research in multiple disciplines shows that the use of spreadsheets to store and structure numeric and text data is commonplace” (WISC, 2014). The main reason for this is probably the same as the reason that business rely on spreadsheets. It represents a fast way to test hypotheses, plot data, conduct pilot experiments and prototype ideas. Moreover, it is ‘easy’ to learn and affordable. There are alternatives to spreadsheets for medical statistical analysis such as SPSS, R and Matlab. Both R and Matlab resemble programming languages and require a detailed knowledge of the syntax and argument construction to be wielded effectively. SPSS requires less specialised knowledge and resembles a spreadsheet. However, SPSS costs significantly more per license than Microsoft Excel (SPSS is starts at $1170 USD per year as a subscription, Excel costs $331 for a full user owned copy). Between the usability, flexibility, perceived ‘easiness’ of spreadsheet software and license cost, it is no surprise that spreadsheet software is the de-facto choice for data analysis in the medical field. Indeed, it is these same reasons that spreadsheet software is used extensively in the business world.

1.4 Spreadsheets Risks

Spreadsheet software is amongst the most utilised commercial software in organisations world-wide. Users cite ‘ease of use’ and a wide range of functionality that could replace many complex information systems. The business world makes extensive use of spreadsheet software for data processing, analysis, decision science and data storing needs (SERP, 2006).

Spreadsheets are however, prone to multiple risks and since they are standalone files, they lack system-wide controls off the shelf. Almost any employee can create access, manipulate, and distribute spreadsheet data. Hence, almost any employee can make a small risky error while manually entering data or configuring formulas (Deloitie, 2009).
Spreadsheets errors are prevalent and can cause crises in organisations. Spreadsheets error rates and cell error rates are high and there are many real stories worldwide that show that these errors can cause serious problems in the business world (EuSpRIG, 2016). Through many years of research, it has become clear that spreadsheet use can carry multiple serious risks (Panko, 2008).

The risk of making a simple mistake appears to be high, field studies show error rates shows that up to 90% of spreadsheet models contain at least one error (Panko, 2008). Spreadsheets are created and used without proper documentation and organisations generally do not have strict criteria governing their use.

Loss of data is a particularly dangerous risk having all data on spreadsheets and not having a centralized and data recovery environment could lead to a crisis in any financial or nonfinancial system. Data availability environments should be created also to ensure business continuity in the event of data loss.

Unskilled users could be considered as a risk since business spreadsheets are designed by both IS and non-IS professionals. The important issue to consider here is that non-IS professionals are unlikely to be trained in information systems development methods, meaning that the process of creating a spreadsheet is far more ad-hoc and is unlikely to follow standards dictated by software engineering. Indeed research shows that almost all spreadsheet modellers have no formal training (SERP, 2006). Because of this, it is impossible to guarantee the adherence of standards to any one spreadsheet modeller or the validity of a particular spreadsheet model. Research shows that most errors do not arise from mistakes in programming the spreadsheet, rather they arise from the misapplication of programming logic (Panko, 2008). This makes the lack of user training in formal development methods even more critical since, once committed, logic errors are difficult to find and correct. Without the knowledge of how to test and debug the spreadsheet, the chance of a user noticing and correcting such a mistake is low.

1.5 The Pilot Study

This paper considers the research environment in the neuroscience department at the University of Newcastle. Within this department, there is diverse research being undertaken ranging from the basic biology of neurons to the abnormal activity associated with epilepsy, from music perception to mood disorders, from visual object recognition to retinal prostheses for the blind, from animal decision-making to anaesthesia to neurological disease. The department has various tools at its disposal including:

- Brain scanning (MRI)
- Cellular imaging and electrophysiology
- Computational modeling
- Molecular genetics
- Animal and human behavioral laboratories
- Psychophysics

The neuroscience department consists of 17 researchers comprised undergraduate, postgraduate and post-doctoral students. There are also a number of senior academics within the department. The
department produces world class research, publishing papers in leading neuroscience outlets such as the *Journal of Neuroscience, Brain and Language* and *Physics life reviews*.

1.6 Research Materials and Methods

A number of different research materials are employed to gather relevant information. Firstly an in-depth questionnaire across all members of the research centre was distributed. There are also two in-depth 45 minute interviews conducted with senior members of the department to supplement this information.

1.6.1 Questionnaire

The questionnaire was designed to gather information on several different dimensions of spreadsheet use in the Neuroscience lab. The questionnaire posed questions on the following themes: Participant demographics; The importance of research data; Spreadsheet use in the organisation; Participant knowledge and experience of spreadsheets; Spreadsheets and other statistics software; The spreadsheet lifecycle; Backup and security.

The questionnaire contained 35 questions and although the large majority were closed multiple choice questions there were also some open free typing questions for balance.

<table>
<thead>
<tr>
<th>Question area</th>
<th>Questions posed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demographics</td>
<td>Age, Sex, Education and Role in the organisation</td>
</tr>
<tr>
<td>The Importance of research data</td>
<td>Frequency of spreadsheet use; size of spreadsheets in the department; number of users per spreadsheet and motivations for spreadsheet use.</td>
</tr>
<tr>
<td>Spreadsheet knowledge and experience</td>
<td>Methods of learning spreadsheets; self-assessed proficiency and willingness to train</td>
</tr>
<tr>
<td>Spreadsheets and other statistics software</td>
<td>How useful are spreadsheets for data analysis; other potential statistics software and personal advantages of spreadsheet software</td>
</tr>
<tr>
<td>The spreadsheet lifecycle</td>
<td>Approaches to design, separation of input, calculation and output; use of guidelines in development; approaches to testing; documentation</td>
</tr>
<tr>
<td>Spreadsheet backup and security</td>
<td>Organisational backup strategies; cell protection; password protection;</td>
</tr>
</tbody>
</table>

Table 1 Questionnaire areas and questions

1.6.2 Interviews

Two semi-structured interviews were conducted with two post-doc researchers in the laboratory. Post-doc researchers were chosen since they were likely to have a more mature, fuller understanding of conducting research, analysis and the workings of the lab. In addition, the post-doctoral students are those who are still working in the lab, whereas the senior members of the academic staff assume a more supervisory role for the undergraduate and postgraduate students. Both interviews lasted around 45 minutes, a range of questions were posed to each interviewee, see table 2.
The interviews were designed to get the participants to extrapolate on the answers provided in the questionnaire and to explore the detail and subtlety of the relationship between spreadsheets and scientific research.

<table>
<thead>
<tr>
<th>Questions relating to conducting research</th>
<th>How many experiments do you conduct in the year?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>How often do you deal with research data?</td>
</tr>
<tr>
<td></td>
<td>What type of data do you usually deal with in your research?</td>
</tr>
<tr>
<td></td>
<td>How many people work on the same experiment and share work together?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Questions relating to spreadsheet use</th>
<th>Do you use spreadsheets daily?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Why do you use spreadsheets?</td>
</tr>
<tr>
<td></td>
<td>What is the percentage of time using spreadsheets compared to other software?</td>
</tr>
<tr>
<td></td>
<td>Do you back up your spreadsheets? How often? What method?</td>
</tr>
<tr>
<td></td>
<td>Who can access your spreadsheet? Could he/she edit?</td>
</tr>
<tr>
<td></td>
<td>Are there any guidelines for using spreadsheets?</td>
</tr>
<tr>
<td></td>
<td>Documentation?</td>
</tr>
<tr>
<td></td>
<td>How do you protect spreadsheets data?</td>
</tr>
<tr>
<td></td>
<td>What level of Knowledge do you have on spreadsheets?</td>
</tr>
<tr>
<td></td>
<td>Is there any second level of checking? What is the method (code inspection…etc.)?</td>
</tr>
<tr>
<td></td>
<td>Are there any preparation steps or guideline to create spreadsheets?</td>
</tr>
<tr>
<td></td>
<td>If data is lost what are the effects (in terms of effort, reputation, legislation and finance)?</td>
</tr>
<tr>
<td></td>
<td>If the data has errors, what are the arising issues? (in terms of effort, reputation, legislation and finance)?</td>
</tr>
<tr>
<td></td>
<td>If data is lost is it hard to get it again?</td>
</tr>
<tr>
<td></td>
<td>What is the period of time needed to collect research data?</td>
</tr>
<tr>
<td></td>
<td>Is there any policy regarding dealing with spreadsheets?</td>
</tr>
<tr>
<td></td>
<td>How much does it cost to run an experiment? (Per day and Per experiment)</td>
</tr>
<tr>
<td></td>
<td>How is your research funded?</td>
</tr>
<tr>
<td></td>
<td>Lack of training and overconfidence - are these a problem from your point of view?</td>
</tr>
<tr>
<td></td>
<td>Is there a risk of losing funds?</td>
</tr>
<tr>
<td></td>
<td>Is there a risk of delaying a research results?</td>
</tr>
</tbody>
</table>

Table 2 Interview questions

1.6.3 Limitations

The number of respondents was relatively small with only 17 completing the questionnaire and two completing interviews. This obviously limits the generalisation that can be drawn from the research but it still provides an interesting initial view of the risks and difficulties of the use of spreadsheets in this clinical setting. There is very little research presently that considers the influence of electronic resources in clinical medicine and hence this paper presents a valid contribution to this field.
2.0 Results

The analysis identifies spreadsheet risks in the following themes: Training; Overconfidence; Spreadsheet design approaches; Documentation.

2.1 Training

The results of questions related to training shows that none of the respondents held any certified training in spreadsheets, see figure 1. More than two-thirds of the respondents learned spreadsheets through self-tutoring. This lack of training is typical finding of almost all surveys of spreadsheet training practice. (Taylor et al 1998, SERP 2006). Self-tutoring does not necessarily indicate a lack of understanding but it does likely suggest that those participants are not formally trained in approaches to planning software, managing data and objectively testing for errors. Spreadsheet modelling has been identified as a cognitively complex activity comparable to that of medical diagnosis (Kruck, 2003). However, consider that routine medical diagnosis is only possible after years of specialised training. Superficially, spreadsheets seem to be simple and straightforward tools but error rates show they carry significant risk. Without an understanding of approaches to planning, development and testing, it would seem more likely that mistakes committed would go unnoticed.

However, a lack of training is not considered an unusual situation as one of the participants suggested in the interview:

“\textit{In research, you just have to pick new skills as you go in job. So, peer learning and picking up things by reading online manual is something all researchers have to do}”.

![Figure 1 Spreadsheet training](image)

According to the ‘internal auditor’ (Larry, 2007) a lack of training is identified as the first source of risk when dealing with spreadsheets. Larry (2007) recommends that ‘untrained users’ should be educated to more than a basic level to reduce the risk of error.

2.2 Overconfidence

Ease of use and overconfidence are related phenomena. The questionnaire showed that although none of the respondents had a certified training, more than three quarters claimed that they are on or above the intermediate level, see figure 2. Overconfidence tends to increase specially in people who are
highly educated and doctoral research represents the highest level of qualification in the education system.

Formal training in spreadsheets is something the researchers do not look at as a necessity, as one of the interviewees replied. It might be correct that all sources of knowledge are available online, but there are a lot of hidden risks that could be eliminated by simple steps. For example, multi-level code inspection could eliminate errors significantly, but it is not something that is typically recommended in online instructional material.

Overconfidence is risk as discussed by Panko (2008) said: “Overconfidence is corrosive because it tends to blind people to the need for taking steps to reduce risks”. In his paper, Panko realized the severity of this issue and the fact that it exists more than we think which is reflected in figure 2, and is referred to in his previous research:

“... when Brown and Gould gave three spreadsheet development tasks to nine highly experienced spreadsheet developers, all made at least one error, and 63% of the spreadsheets overall had errors. Yet when asked about their confidence in the correctness of their spreadsheets, their median score was “very confident.”

Overconfidence affects both novice and expert users (Panko, 2008). It also appears that the self-efficacy of the individual modeller is the most important factor when considering how overconfidence can be observed and mitigated (Takaki, 2005).

2. Design and testing

Almost all of the researchers (94%) start designing their own spreadsheets by directly inputting data into the computer. This finding is typical of most spreadsheet user surveys (SERP 2006, Taylor et al. 1998) and highlights a particularly risky behaviour amongst most spreadsheet developers. Several spreadsheet design standards recommend sketching the design on a piece of paper first (Grossman 2002, Grossman and Ozluk, 2004). However, none of the respondents indicated they sketch the model on paper first, instead they go straight to the spreadsheet and enter formulae and data directly. It would also seem that such behaviour is consistent with overconfident modellers, as indicated in figure 2 – i.e. an overconfident modeller does not see the need to plan the model before they start coding.

When asked how they approach testing the spreadsheets they create, the largest majority (40%) indicated they use self-revision checks to ensure the model is free of errors, 13.3% indicated they did
not use checks, 20% said they submit spreadsheets for peer review and 26.7% said they use a calculator to check the accuracy of the calculations, see figure 3. This is an encouraging finding since most participants do some form of checking. The most promising of these are using a calculator and peer review. Peer review is a good idea from the perspective that if the work contains errors, it is more likely that the error will be spotted by a team than an individual. Research shows that spreadsheets audited by teams generally find more mistakes, in one study (Panko, 2008) individuals found and corrected 60% of an error seeded model, whereas groups of three found and corrected 80%. The use of a calculator to check the accuracy of the mathematics in the spreadsheet is a promising approach since it requires the tester to reconsider the dimensions of the model, indeed one would need to examine all assumptions to effectively check using this approach. If there had been a mistake in the logic or mechanics of the spreadsheet, cross checking the outcome with a calculator might expose those mistakes (Colver 2008, O’Beirne 2009, Ayalew et al. 2000).

2.4 Documentation

Documentation is an essential aspect of the spreadsheet life cycle. Ideally, every spreadsheet should have supporting documentation to describe the input, computation, output and data points contained in it.

Documentation assures better understanding of the spreadsheet (Pryor, 2004) which leads to better accuracy and fewer errors. In addition, it is particularly critical when there are multiple users using the same spreadsheet. The participants indicated around a third of the respondents share spreadsheets with other users. The questionnaire showed that on average, 2-5 researchers share the same spreadsheet.

The data also showed that nearly a third of the respondents do not document their spreadsheets at all. And nearly half of the respondents use the cell-comments feature to document. It is encouraging that comments feature at all since this can be viewed as a kind of documentation. However, cell comments can be thought of as the equivalent as annotated code in software engineering, whilst these comments are important and assist in making the spreadsheet more understandable, it does not offer the depth of analysis or guidance as a conceptual model might.

2.5 Security

This research considers the spreadsheet security from several different angles that align with the Spreadsheets Standards Review Board (SSRB) standards (SSRB, 2003). The first line of protection is
setting a strong password on the spreadsheet. The majority of the respondents (82%) didn’t use this feature to protect their spreadsheets. Moreover, the majority that used passwords had them written down somewhere in the office. Finally, cells that contained output data had not been protected from manipulation. Since more than third of the respondents share spreadsheets, the output cells should be protected to prevent accidental overwriting of formulae.

However, the IT infrastructure of the premises offsets the risk of theft via local secure storage. Each lab has its own share drive to store spreadsheets and data and only. Access is only granted to the lab researchers so the chance of theft is greatly reduced. This does not mitigate the risk of accidentally overwriting cells in the spreadsheet.

2.6 Backup solutions

The backup ‘rule of three’ states that for a file to be sufficiently backed up it should be kept in three separate locations using two different types of media with one offsite backup.

A lack of an adequate backup solution could mean permanently lost data, effort and time. In this research, more than 82% of the respondents seemed to be unaware of suitable backup procedures to protect their data. Some respondents kept a single backup of work on external hard disks. Others used the Universities local networked servers as their means of backup. Whilst the networked infrastructure of the university offers some security from lost files, it does not meet the conditions of the ‘rule of three’ and hence one should not consider these files adequately backed up.

During the interview stage of the study, interviewees were asked about the consequences of losing research data:

“Losing data could result in a lot of time and effort being put in to repeating the research if results were not also recorded elsewhere (which they usually are). If repeating experiments is necessary this may cost the organization to fund another set of experiments. This would also delay publishing the data which could lead to a delay in publishing (manuscripts) which would also result in a delay in any benefits to the general population the research may provide. For example the creation of new treatments”

“…damage to your personal reputation, huge amount of time wasted and have to do it again, would have to prove to tutor/supervisor that it won’t happen again somehow which is likely really hard in today's dog-eat-dog research society Organization. Ethical consequences of not keeping original results, would practically invalidate studies that might have been published, if other researchers/ethics committees wanted to look at original results we would be in legal trouble, staff members may be sacked, even criminally liable in some cases. Media coverage of bad science would reflect badly on the university…”

“lost effort collecting data, lost potential for important discoveries which could have had widespread implications”

Clearly the risks of losing research data are severe to the participants, the university, the research discipline and the wider public. The questionnaire data suggests that whilst some backup precautions are followed, there is more that could be done to secure the departments data.

2.7 Complexity, Frequency of use and number of users
Defining complexity in spreadsheets objectively is difficult since the scope and use of functions varies from one application to another. Therefore, data on the type of features used in spreadsheets was used to determine its complexity in addition to determining the percentage of cells with formulae, see figures 4 and 5. This is broadly in line with other studies of functionality use (Chan and Storey 1996, Ballinger et al. 2003, Thorne and Ball 2008).

From figures 4 and 5, it is obvious that the majority of the researchers use spreadsheets to plot data and do some calculations using equations. However, around half of the respondents indicated that less than 10% of their spreadsheet cells contain formulae. The majority of the participants indicated they use ‘simpler’ excel functions in their spreadsheets rather than make use of nested functions which are more complex to program. It would seem then that the complexity of the spreadsheets at the lab is at the lower end of the complexity spectrum. This picture is fairly typical.

2.9 Spreadsheet errors impacts

This section discusses some of the potential impacts arising from the use of spreadsheets in neuroscience research. The answers in this discussion arose mostly from the interview process and examine a specific set of risks for those working with scientific research data.

2.9.1 Financial loss

Although financial issues are well considered in business data, financial loss should be taken into consideration when dealing with research data.
The neuroscience lab at Newcastle University is funded through a mix of European Science Council funding, internal funding and

Establishing a science lab, such as the one study, with state of the art technology is a costly exercise. The lab also has significant ‘on-costs’, meaning that the running costs sometimes exceed the setup. Every experiment run in the lab therefore has its own cost which is deducted from a limited budget. According to the respondents, the average cost per neuroscience experiment is around £1000 per day. However, the cost varies according to the type of experiment, the tools used, the materials available and the number of staff involved. Since the costs are so high, any data generated from these experiments is valuable, and losing data would mean a significant financial expenditure to re-gather the data. Furthermore, the budgets are tight and there is no space for errors or time wasting.

If data was lost, additional funds to rerun the experiments might be sought as one of the respondents described:

“Lots of effort would be required to correct the data by, for example, rerunning that part of the experiment. This would also be likely to cost more money and might require additional legislation”

To conclude, financial loss is critical to the organisation, but it would even be catastrophic if financial loss as combined with other impacts.

2.9.2 Wasted effort and data loss

Wasted effort is one of the key impacts that almost all of the participants agreed on. Wasted effort includes the effort of the original experiment, the actual data collection, error checking and the effort to correct or re-run the experiment.

“Lots of effort would be required to correct the data by, for example, rerunning that part of the experiment”.

“It takes a long time to go back and check spreadsheets for errors”

“The biggest effects would be loss of effort and reputation”

“It would take a long time to possibly correct the errors, or if the results hadn't been backed up elsewhere, the research would have to be carried out again”

“The biggest impact from input errors would be the time consumption of checking for errors”

Clearly the participants recognise the risks of having to re-run the experiments and the significant monetary costs of repeating experimentation. A single error that materially affects the outcome of the experiment would cost a significant amount of money to correct. This is especially true when the data collection period lasts months.

In addition to the wasted time, wasted data is also crucial. More than two-thirds of the respondents of the questionnaire selected “Losing valuable experimental data or simulation results” as one of the impacts and risks of spreadsheets errors.
2.9.3 Lower Productivity

Each research group in the neuroscience institute has between 18 - 20 experiments per year to conduct. Therefore, having errors and wasting time could reduce this number which in turn would lead to a significant reduction in the productivity of this group in terms of research conducted, outputs produced (Academic papers) which in turn has a knock on effect on the credibility of the department and the University.

“It may reduce the amount of papers I produce in my PhD therefore reducing my attractiveness to a future employer”.

2.9.4 Causing unnecessary suffering to animals

One of the respondents pointed out that unnecessary animals suffering as one of the impacts of errors in research data. Newcastle neuroscience department uses animals in some of their experimentation. Therefore, losing data or even having errors in it not only causes loss of effort and money, but also wasted days of experiments on animals.

No researcher or scientist intends to harm or cause suffering to animals if it could be avoided. Therefore, there is a significant ethical impetus to handle experimental data correctly and carefully.

2.9.5 Publishing corrections in publications

In the worst case scenarios, data with errors could pass all internal checks unnoticed and get published. In this case, the honesty of the researcher plays a major role in not misleading the discipline, journal and readership. One of the interviewers acknowledged the impact of errors in published papers, and when he was asked about the steps he would do if he discovered any errors in his experiments: “You have to be very honest if you find any mistake and to make sure your correction reach everybody read your publications”.

However, the risk of an error reaching the publication stage is lower since the paper will go through at least one process of peer review when it is being considered for publication. A paper is also likely to be reviewed internally, especially if the paper is the result of PhD study or research grant further reducing the chance of the paper reaching publication with an error. Multiple authoring further reduces the likelihood of unnoticed mistakes since the work is likely to be read and considered by multiple experts.

“It is unlikely results with errors would be published into the scientific community since there are many stages where different people check and analyse the results. In summary: If there are errors more time and effort would be used”

2.9.6 Unreliable research

Although internal and external peer review processes are likely to highlight errors, there is a chance that manuscripts with errors could be published in a journal. This could mean that research is published with misleading conclusions. A good example of this is the Reinhardt and Rogoff paper
“Growth in a Time of Debt” which was published in the peer reviewed journal *American Economic Review* in 2010. This paper contained a statistical analysis that suggested that if a country’s external debt exceeded 90% of the country’s Gross Domestic Product (GDP), negative growth was the consequence. However, the statistical analysis contained a flaw which meant that the 90% figure was incorrect. In actual fact, the analysis was completely wrong, a corrected analysis issued by Herdon *et al*. 2013 showed that when debt exceeds 90% of a country’s GDP growth is still positive. In the time between the article being published and the discovery of the error, the paper was used by a number of governments in the US and Europe as a justification for austerity. The paper was even quoted directly by the UK chancellor George Osborne in his discussions on the economy. Hence decisions were made and policies formed on erroneous data.

Neuroscience is a complex field of study with only a few institutions globally dedicated to neuroscience research. It is conceivable that an error could reach publication and that the published manuscript could form the basis for real world decision making as in the Reinhardt and Rogoff case.

Unlike books, research publications in conferences and journals do not generally have post-publication review and feedback. This ensures fewer corrections to the journal, but it also means that if a mistake makes it through the peer review process that the work is unlikely to be questioned.

When asked about the chance of errors creeping into published manuscripts, the participants recognised the risk of unreliable research:

“Unfortunately, since the data are published there is no feedback and the data are considered as right”

“In general, making errors in the data involve having wrong conclusions and guiding the researcher to the wrong hypothesis”

### 2.9.7 Misleading science

Errors in research data have severe impacts on the science fields in general. Although the consequences of an error could include financial loss, some of the potential impacts could carry far greater ramifications. Unlike Reinhardt and Rogoff, some neuroscience researchers are dealing with real lives and the impact of an error could be catastrophic to the health of individuals following treatment plans informed by such research. Again, the participants recognise these issues in both the questionnaire data and follow up interviews.

“Drawing incorrect conclusions from data leading to negative research impact” was one of the multiple choice answers to the impacts and risks of errors in spreadsheets. 14 respondents out of 17 selected this choice in the questionnaire. Therefore, this issue is clearly recognised as a significant risk in scientific research.

One participant distinguished the impact of erroneous data based on the stage of its discovery:

“If the error is detected before publication, only the lab wastes time/effort/money. If the error is not detected until after publication, the lab/university's reputation will be affected. If the error is not detected at all, the entire field is misled”
2.9.8 Damage to Reputation

Respectable research institutions are expected to publish reliable and well explored research manuscripts. Errors that reach publication could therefore severely damage the reputation of the institution if the research was widely published as many of respondents described:

“It is damaging to reputation to retract any statements/findings”

“The organization would lose credibility because of poor research techniques, which would impact the university as a whole”

“The reputation of the organization would be negatively affected if it turned out that errors had arisen in the data and not been checked/realized.”

“It is demoralizing, bad for reputation of the researcher and the school, time consuming”

Other respondent highlighted a case that occurred in the medical research field recently by saying:

“if the errors are highlighted as it was seen for a recent Nature paper published by Japanese students, that the reputation of the lab considerably decrease and it may happen that some people lose their position”.

3.0 Conclusions

The following conclusions are divided into three sections: The risk profile of the participants; The risk profile of the spreadsheet artefacts and the impacts of risk. Finally a comparison is made between business and academic spreadsheet risks.

3.1 Risk profile of the participants

Based on the questionnaire and interview data, the profile of the participants of the study are typical of similar studies in business research. Almost all participants indicated that they had no formal training in spreadsheet software, which is typical of spreadsheet users irrespective of the discipline of the sample. The participants also displayed classic evidence of overconfidence in the answers to a number of questions. When asked to evaluate their own competence in spreadsheet development, the large majority of the participants (76%) chose to rank themselves as “intermediate or expert” despite not a single participant having received any formal or certified training. When asked about spreadsheet design, 94% of the participants indicated that when developing spreadsheets, they opt to directly enter data and formulae into the spreadsheet without planning the model first. When developing spreadsheet models the participants produced very little supporting documentation, although some participants used cell comments to document certain features. Very few participants (18%) chose to protect their spreadsheets with cell locking on the sheets. The most significant reason to lock the cells on a spreadsheet is to prevent yourself or others accidently overwriting portions of the spreadsheet, this practice particularly prudent if the spreadsheet is being shared with others.

3.2 Risk profile of spreadsheet artefacts

Based on the information gathered from the participants, the level of complexity assigned to the artefacts is intermediate. Figures 4 and 5 show a ‘simple’ use of spreadsheet functions and relatively small spreadsheets. However, when looking at research outputs of the department, for instance
(Gregson et al, 2014), it is clear that statistical analysis techniques are utilised to validate data. Presumably this analysis is done within Excel and therefore could redefine the complexity classification as high. However, this is anecdotal since none of participants indicated they perform these relatively complex calculations in the answers given. The questionnaire data also suggested that the number of new spreadsheets being created is relatively low, with participants indicating they tend to work on the same spreadsheet frequently.

3.3 Risk impacts in research

The specific risks arising from this use are numerous and will be listed sequentially in order of potential negative impact to the organisation.

The first potential risk is financial loss through lost data. The neuroscience lab has significant running costs, these costs are a combination of staffing, equipment and materials for experimentation. The average cost for running an experiment in the lab is around £1000 pound a day. Hence if spreadsheet data was lost through whatever means, it would cost a significant amount of money to re-run the experiments which would have to be accounted for as part of the Universities tight budget for running this lab.

Wasted effort and lost data are both significant risks to the researchers at the neuroscience lab. Any lost data would take significant time and effort to reproduce. In the lab environment, the amount of time available for the researchers to use the lab is limited by budget and demand, meaning that finding spare capacity and budget to re-gather data is difficult.

Having to repeat experimentation also means that the overall productivity of the lab and the researchers is impacted since fewer measurable research outputs (such as academic journal papers) can be published if experiments must be re-run.

Since some of the experimentation is done with live animals, any errors or lost data could mean that any suffering the animals experienced as part of the research was unnecessary. Obviously this is of the upmost ethical importance to the staff who would never intentionally cause suffering to animals. Indeed, if it was possible to do the experimentation without the animal subjects then that would be the de facto position however it remains the case that many medical breakthroughs have been made through the use of animal subjects.

Errors in data that reach publication could have a number of serious consequences for the institution. Firstly errors in data or interpretation of data would equate to published research that would be considered unreliable. It’s possible that others could then make decisions on the basis of erroneous research such as the Reinhard and Rogoff case, the consequences here are that the conclusions are misleading. This may in turn influence other researchers in to making poor decisions, such as choosing research questions based up erroneous conclusions.

3.4 Comparing academic and business risks of spreadsheets

The profile of spreadsheet modellers between the academic and business world is similar. This is evidenced through a lack of formal training, issues of overconfidence, a lack of preparation and documentation. These findings are typical of spreadsheet surveys conducted in the business environment too, hence the profile of spreadsheet modellers in academic and business worlds are similar.
One of the major differences between these worlds are the mechanisms that each uses to communicate information externally. In the business world, information contained in spreadsheets might be communicated through a variety of means to the general public but importantly, it is up to the business to ensure that the information being communicated is sound. In the academic community, most external communications comes through research outputs such as papers published in conference or journal proceedings. Typically, these research outputs are double blind peer reviewed before publication. This provides academic research with a significant safety net when considering spreadsheet errors, since if mistakes are made in publications or if falsified data is presented, it is likely that these issues will be spotted before being communicated to the community. Of course such processes aren’t infallible (Reardon & Cyranoski, 2014) but they must prevent a good proportion of mistakes from reaching the public eye. Peer review processes have long been advocated as one of the best means to reduce the likelihood of spreadsheet errors (Panko, 2008) – this pilot study really underlines that issue since the number of retractions in the academic world are very low. Hence it would seem that the peer review process is particularly effective at preventing spreadsheet errors (amongst other types of errors) from reaching publication.

3.5 Identified areas for future research

This study highlights two main opportunities for further interesting research, the first considers the internal peer review process and the second focuses on the ‘important’ spreadsheets in the organisation. To that end, the following research questions have been identified:

1. How are processes of internal verification and review implemented at the Newcastle neuroscience lab to reduce errors in spreadsheets, research and publication?

This research question will explore how the internal peer review process is used to reduce defects in spreadsheets, research and publications. A successful system of this type could provide a model approach that could be adopted by other organizations to reduce the likelihood of spreadsheet defects be communicated outside of the organization.

2. What is the character and make up of ‘important’ spreadsheets in the neuroscience lab and what specific risks do these spreadsheets carry?

This research question will offer a detailed examination of spreadsheets identified as ‘important’ to the lab. The spreadsheets will be examined in terms of programming structure, functions used, process of development and documentation. The spreadsheets will be analysed for vulnerabilities and ‘what-if’ risk scenarios will be explored with the modelers.

3.6 Conclusion

This pilot study has examined spreadsheet use at the neuroscience lab at the University of Newcastle. From the study we can conclude that spreadsheets play an important role in conducting, recording, analysing and communicating scientific research. The risks present in these activities relate to misanalysis, misrepresentation, data loss or corruption, wasted time, lower productivity and unnecessary suffering to animals. The potential consequences of such risks mainly relate to the publication of data or analysis that is erroneous or misleading. However, the neuroscience lab has a process of internal verification that explores the data, analysis and inference before they leave the
confines of the laboratory. A more detailed examination of the ‘important’ spreadsheets and the internal verification process is proposed since a process like this could be adapted to a business context to help reduce the number of errors in the business world.
References


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.

Front cover images courtesy of EuSpRIG Authors.